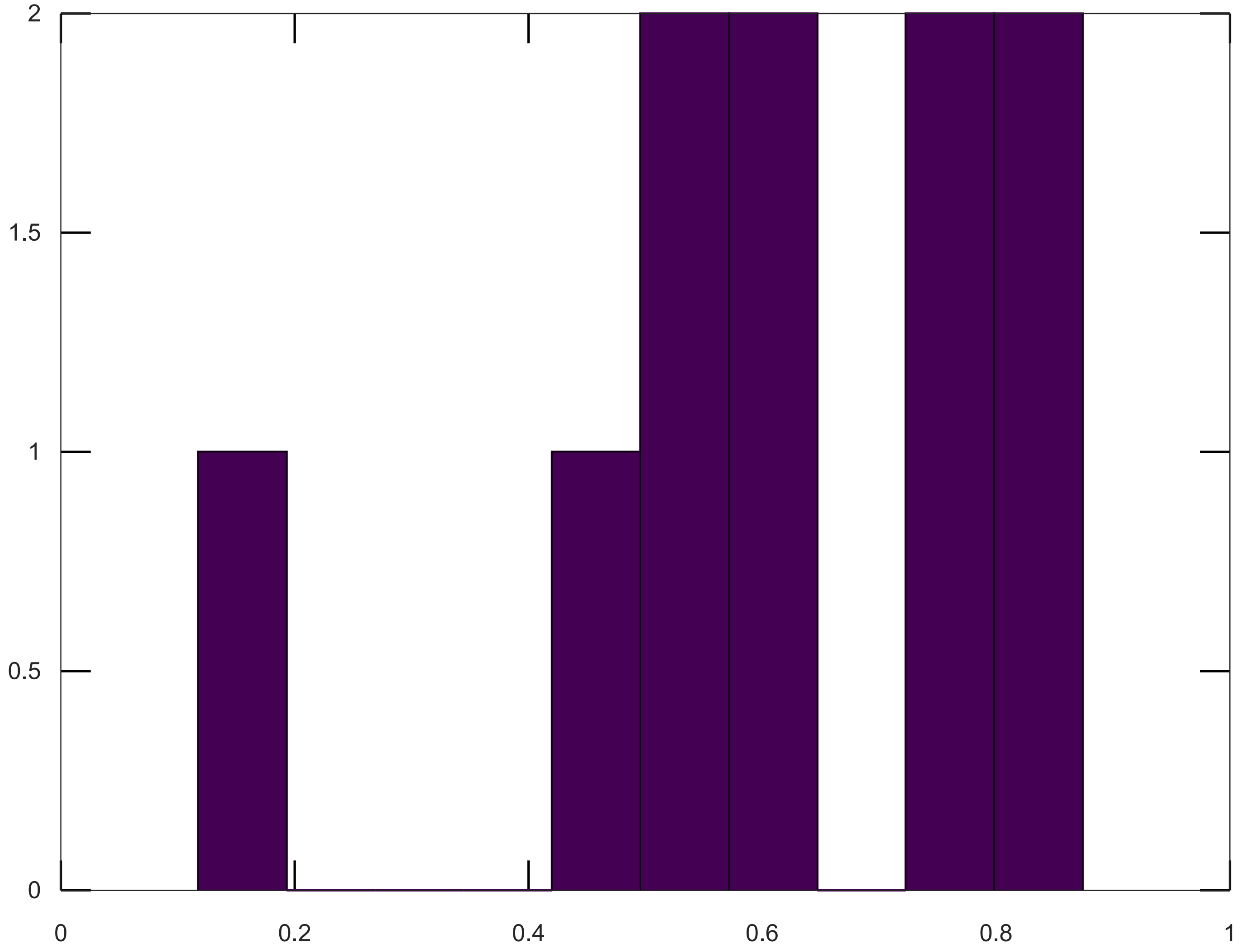
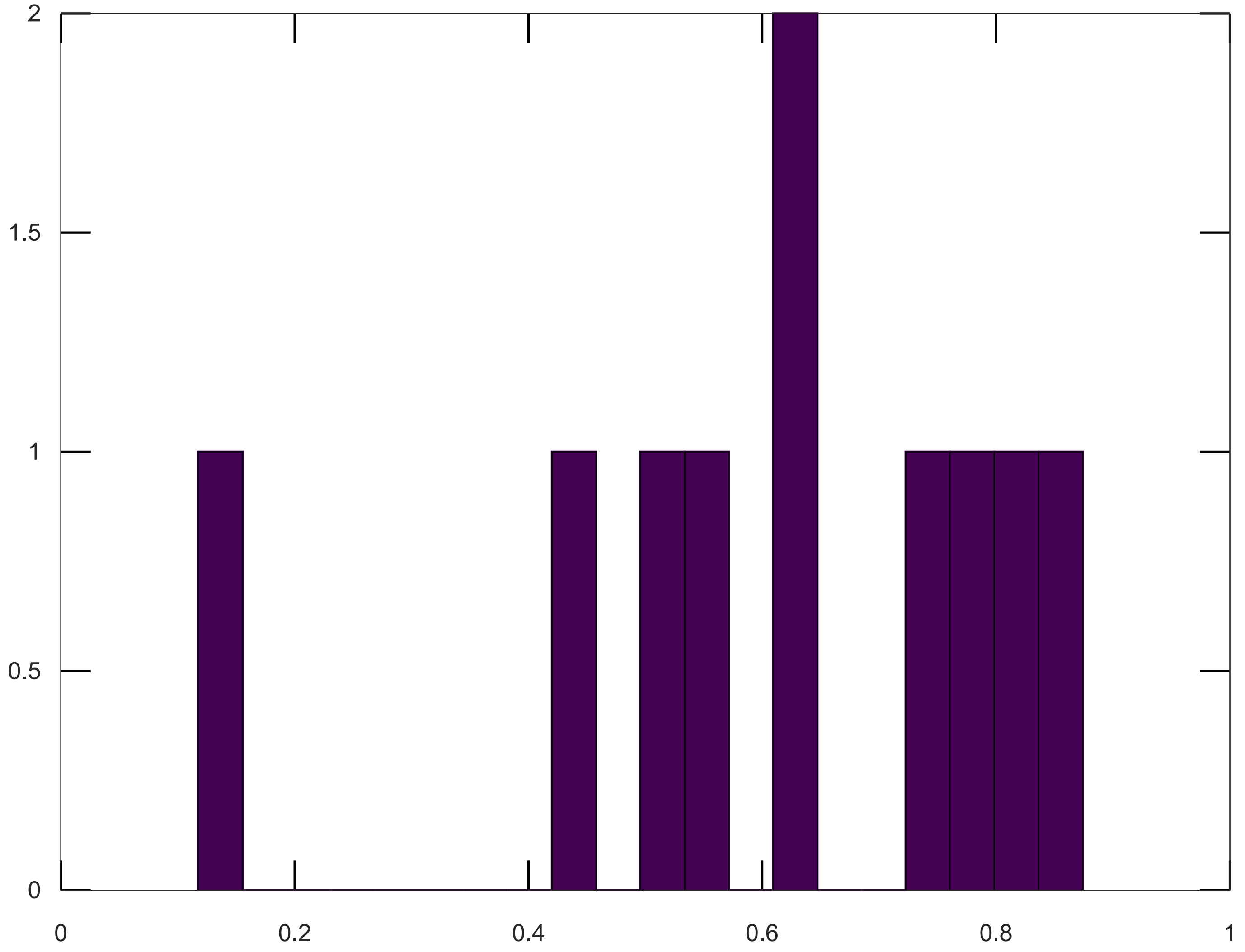
Octave Online

```
octave:2> disp('Octave Tutorial')
Octave Tutorial
octave:3> 5+6
ans =  11
octave:4> 5*6
ans =  30
octave:5> 8-7
ans =  1
octave:6> 30/6
ans =  5
octave:7> 2^32
ans =    4.2950e+09
octave:8> a = 2^32
a =    4.2950e+09
octave:9> disp(sprintf('%.10f',a))
4294967296.0000000000
octave:10> disp(sprintf('%.20f',a))
4294967296.00000000000000000000
octave:11> %%matrix and vectors
octave:11> mat =[1 2; 3 4; 6 7]
mat =

   1   2
   3   4
   6   7

octave:12> 1:0.1:3
ans =

 Columns 1 through 8:

    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000    1.6000    1.7000

 Columns 9 through 16:

    1.8000    1.9000    2.0000    2.1000    2.2000    2.3000    2.4000    2.5000

 Columns 17 through 21:

    2.6000    2.7000    2.8000    2.9000    3.0000

octave:13> 1:10
ans =

    1    2    3    4    5    6    7    8    9   10

octave:14> ones(2,3)
ans =

   1   1   1
   1   1   1

octave:15> zeros(1,5)
ans =

   0   0   0   0   0

octave:16> rand(4,5)
ans =

   0.684534   0.337185   0.054637   0.339106   0.292682
   0.059592   0.128698   0.771652   0.667704   0.558483
   0.492201   0.070819   0.134464   0.649446   0.596195
   0.123910   0.968228   0.639157   0.243870   0.546264

octave:17> w= rand(1,10)
w =

 Columns 1 through 8:

   0.87406   0.43764   0.50207   0.73418   0.79389   0.55489   0.64466   0.82749

 Columns 9 and 10:

   0.11753   0.61544

octave:18> hist(w)
```



```
octave:19> hist(w,20)
```



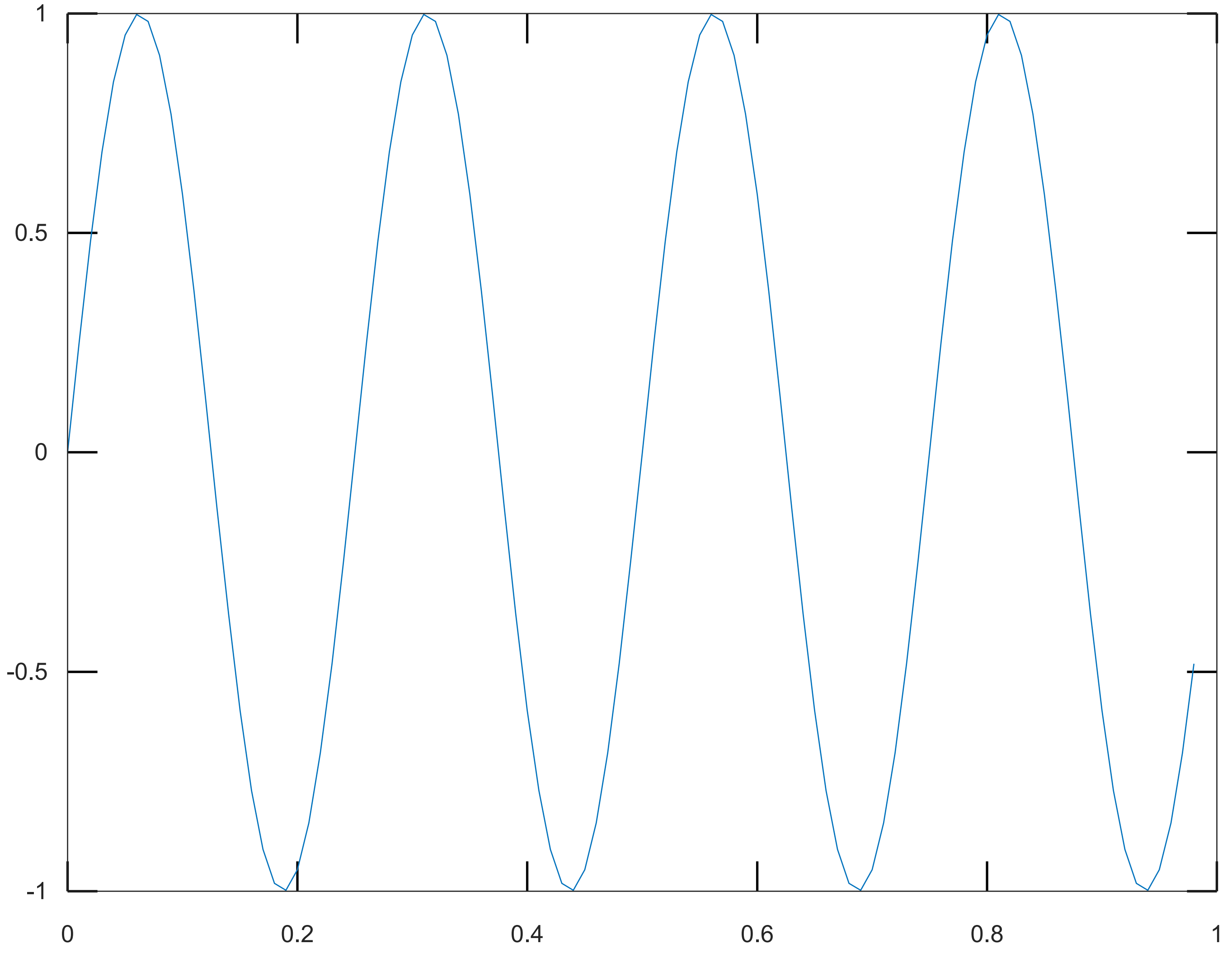
```
octave:20> eye(10)
ans =

Diagonal Matrix

   1   0   0   0   0   0   0   0   0   0
   0   1   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0
   0   0   0   1   0   0   0   0   0   0
   0   0   0   0   1   0   0   0   0   0
   0   0   0   0   0   1   0   0   0   0
   0   0   0   0   0   0   1   0   0   0
   0   0   0   0   0   0   0   1   0   0
   0   0   0   0   0   0   0   0   1   0
   0   0   0   0   0   0   0   0   0   1

octave:21> %% initialize variables
A = [1 2;3 4;5 6]
B = [11 12;13 14;15 16]
C = [1 1;2 2]
v = [1;2;3]
%% matrix operations
```

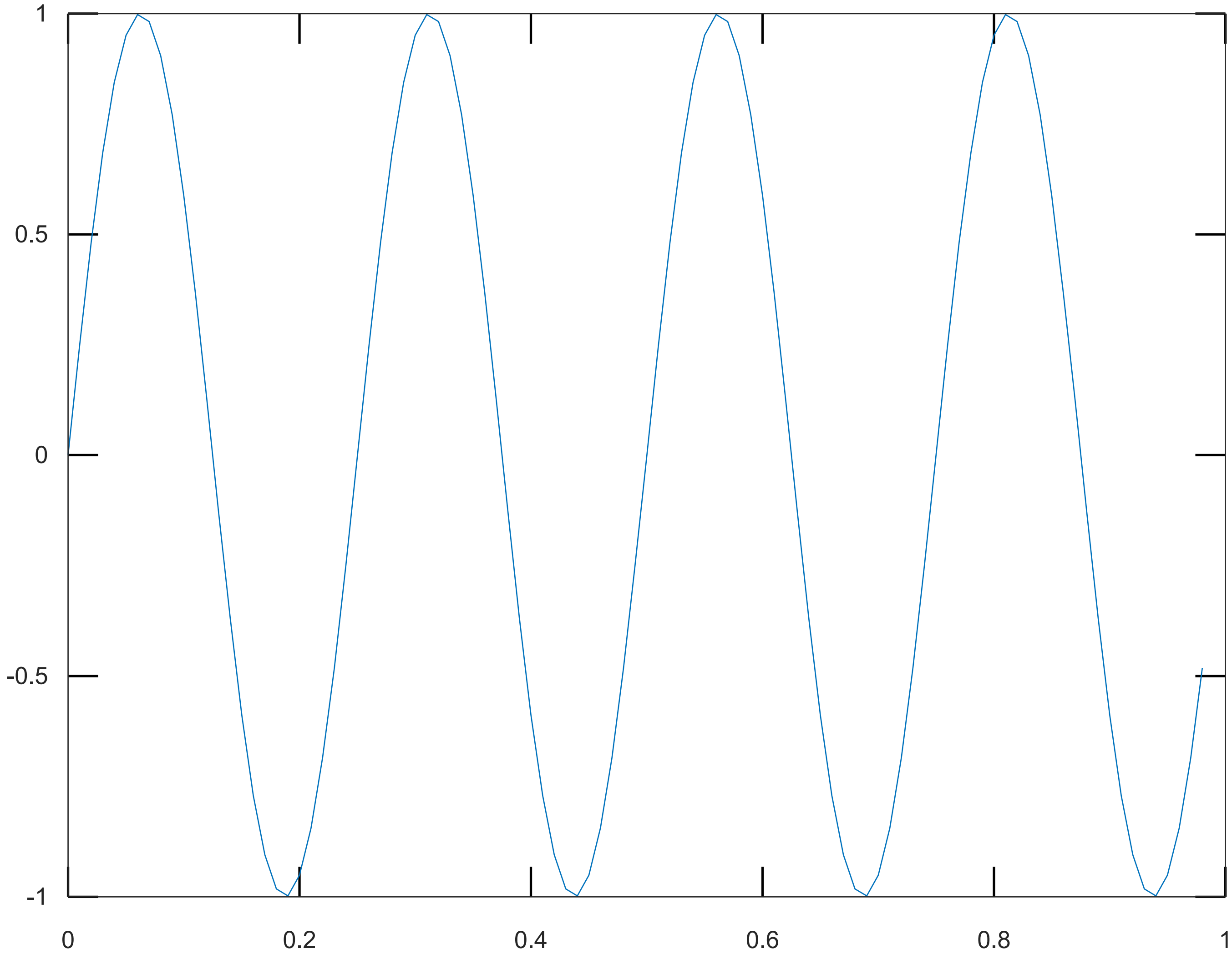
```
A * C % matrix multiplication
A .* B % element-wise multiplication
% A .* C or A * B gives error - wrong dimensions
A .^ 2 % element-wise square of each element in A
1./v % element-wise reciprocal
log(v) % functions like this operate element-wise on vecs or matrices
exp(v)
abs(v)
A =

   1   2
   3   4
   5   6

B =

   11   12
   13   14
   15   16

C =

   1   1
   2   2

v =

   1
   2
   3

ans =

    5    5
   11   11
   17   17

ans =

   11   24
   39   56
   75   96

ans =

    1    4
    9   16
   25   36

ans =

   1.00000
   0.50000
   0.33333
```

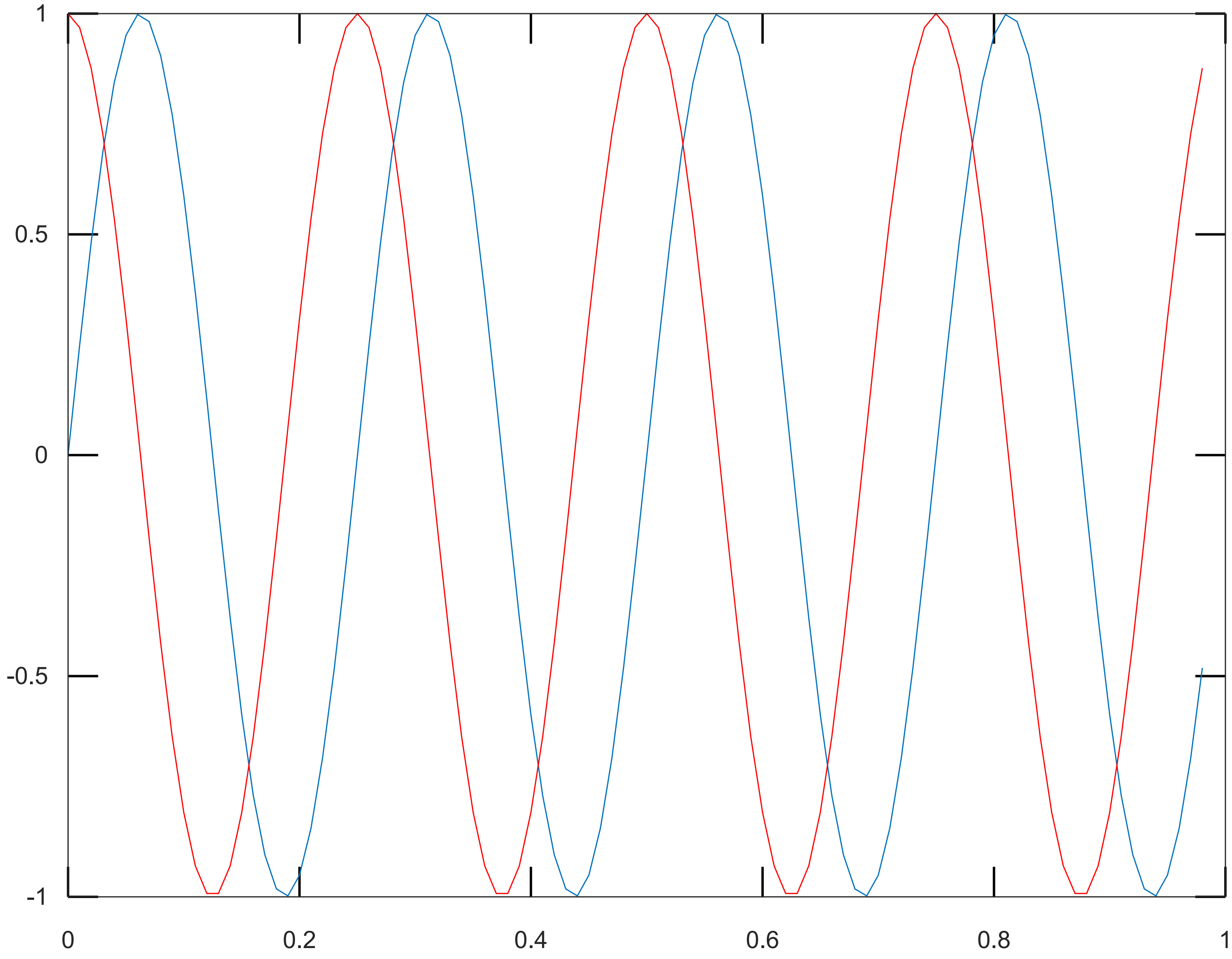
```
ans =

   0.00000
   0.69315
   1.09861

ans =

    2.7183
    7.3891
   20.0855

ans =

   1
   2
   3

octave:32> a = [1 15 2 0.5]
val = max(a)
[val,ind] = max(a) % val - maximum element of the vector a and index - index
value where maximum occur
val = max(A) % if A is matrix, returns max from each column
% compare values in a matrix & find
a < 3 % checks which values in a are less than 3
find(a < 3) % gives location of elements less than 3
A = magic(3) % generates a magic matrix - not much used in ML algorithms
[r,c] = find(A>=7) % row, column indices for values matching comparison
% sum, prod
sum(a)
prod(a)
floor(a) % or ceil(a)
max(rand(3),rand(3))
max(A,[],1) %- maximum along columns(defaults to columns - max(A,[]))
max(A,[],2) %- maximum along rows
A = magic(9)
sum(A,1)
sum(A,2)
sum(sum( A .* eye(9) ))
sum(sum( A .* flipud(eye(9)) ))
% Matrix inverse (pseudo-inverse)
pinv(A) % inv(A'*A)*A'
a =

    1.00000   15.00000    2.00000    0.50000

val =  15
val =  15
ind =  2
error: 'value' undefined near line 1 column 1
val =

   5   6
```

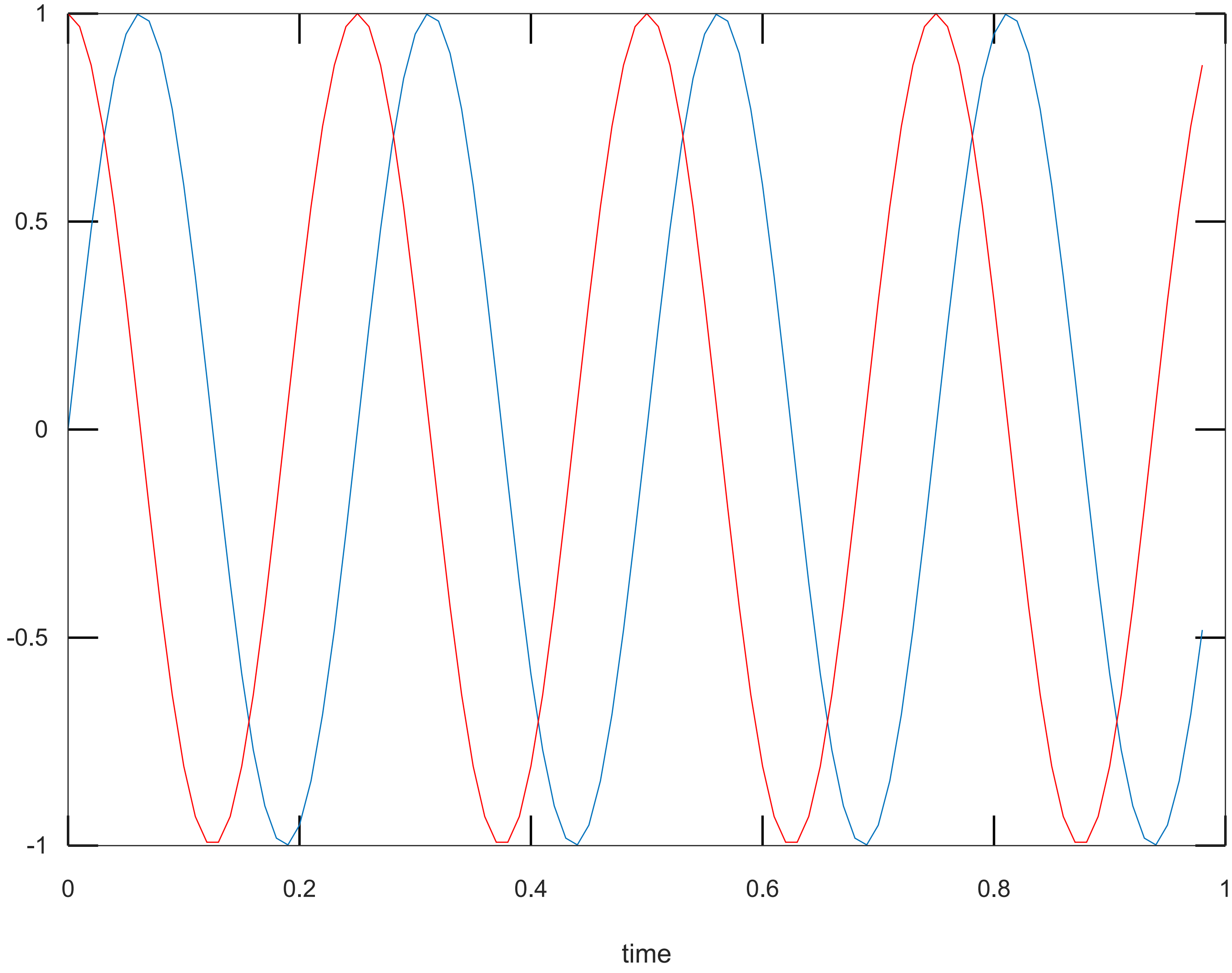
```
ans =

  1  0  1  1

ans =

   1   3   4

A =

   8   1   6
   3   5   7
   4   9   2

r =

   1
   3
   2

c =

   1
   2
   3

ans =  18.500
ans =  15
ans =

    1   15     2    0

ans =

   0.56716   0.25111   0.70272
   0.84502   0.79745   0.61642
   0.96680   0.28888   0.95318

ans =

   8   9   7

ans =

   8
   7
   9

A =

   47   58   69   80    1   12   23   34   45
   57   68   79    9   11   22   33   44   46
   67   78    8   10   21   32   43   54   56
   77    7   18   20   31   42   53   55   66
    6   17   19   30   41   52   63   65   76
   16   27   29   40   51   62   64   75    5
   26   28   39   50   61   72   74    4   15
   36   38   49   60   71   73    3   14   25
   37   48   59   70   81    2   13   24   35

ans =

   369   369   369   369   369   369   369   369   369

ans =

   369
   369
   369
   369
   369
   369
   369
   369
   369

ans =  369
ans =  369
ans =

 Columns 1 through 6:

   4.5353e-04  -1.2230e-03   1.6729e-03   1.2647e-02  -1.2062e-02   3.1805e-04
   3.0111e-04   3.0111e-04   1.2801e-02  -1.2199e-02   3.0111e-04  -1.0878e-03
  -1.0706e-03   1.4019e-02  -1.2045e-02   3.0091e-04   4.5374e-04   1.4870e-04
   1.2647e-02  -1.2045e-02   3.0132e-04   3.0300e-04  -1.0725e-03   1.6729e-03
  -1.0810e-02   3.0111e-04   3.0111e-04   3.0111e-04   3.0111e-04   3.0111e-04
   2.8418e-04   4.7047e-04   1.4870e-04  -1.0706e-03   1.6747e-03   2.9923e-04
   3.0300e-04   2.8230e-04   3.1805e-04   4.5353e-04   1.4849e-04   3.0132e-04
   3.0111e-04   3.0111e-04  -1.0878e-03   1.6900e-03   3.0111e-04   1.2801e-02
   3.0091e-04   3.0320e-04   2.9923e-04   2.8418e-04   1.2664e-02  -1.2045e-02

 Columns 7 through 9:

   3.0300e-04   2.9902e-04   3.0132e-04
   1.6900e-03   3.0111e-04   3.0111e-04
   2.8418e-04   3.1993e-04   2.9923e-04
   4.5353e-04   1.3176e-04   3.1805e-04
   3.0111e-04   3.0111e-04   1.1412e-02
   3.0091e-04   1.2647e-02  -1.2045e-02
   1.2647e-02  -1.3416e-02   1.6729e-03
  -1.2199e-02   3.0111e-04   3.0111e-04
  -1.0706e-03   1.8253e-03   1.4870e-04

octave:52> %% plotting
t = [0:0.01:0.98];
y1 = sin(2*pi*4*t);
```

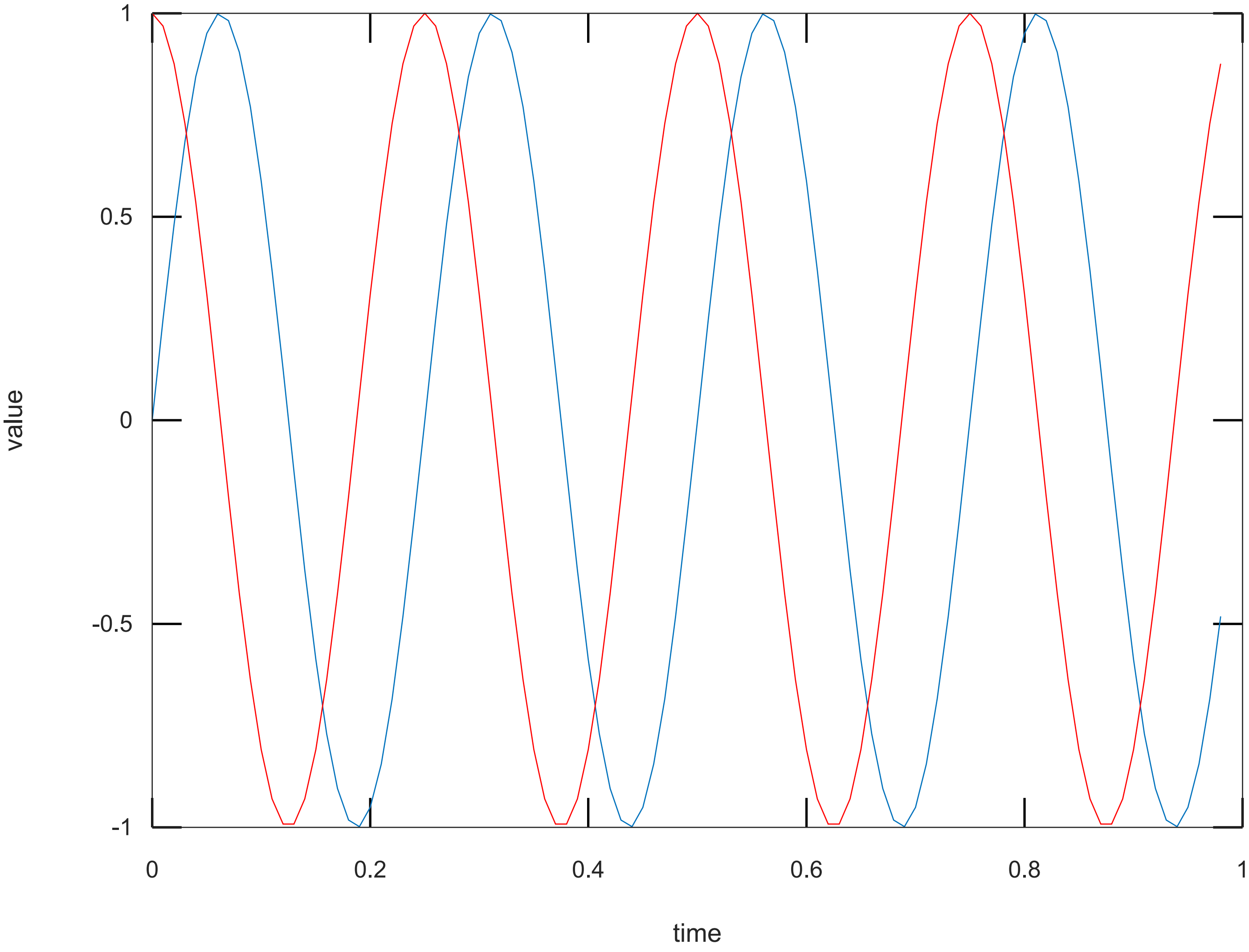
```
plot(t,y1);
y2 = cos(2*pi*4*t);
hold on; % "hold off" to turn off
plot(t,y2,'r');
xlabel('time');
ylabel('value');
legend('sin','cos');
title('my plot');
print -dpng 'myPlot.png'
close; % or, "close all" to close all figs
figure(1); plot(t, y1);
figure(2); plot(t, y2);
figure(2), clf; % can specify the figure number
subplot(1,2,1); % Divide plot into 1x2 grid, access 1st element
plot(t,y1);
subplot(1,2,2); % Divide plot into 1x2 grid, access 2nd element
plot(t,y2);
axis([0.5 1 -1 1]); % change axis scale
%% display a matrix (or image)
figure;
imagesc(magic(15)), colorbar, colormap gray;
% comma-chaining function calls.
a=1,b=2,c=3
a=1;b=2;c=3;
```

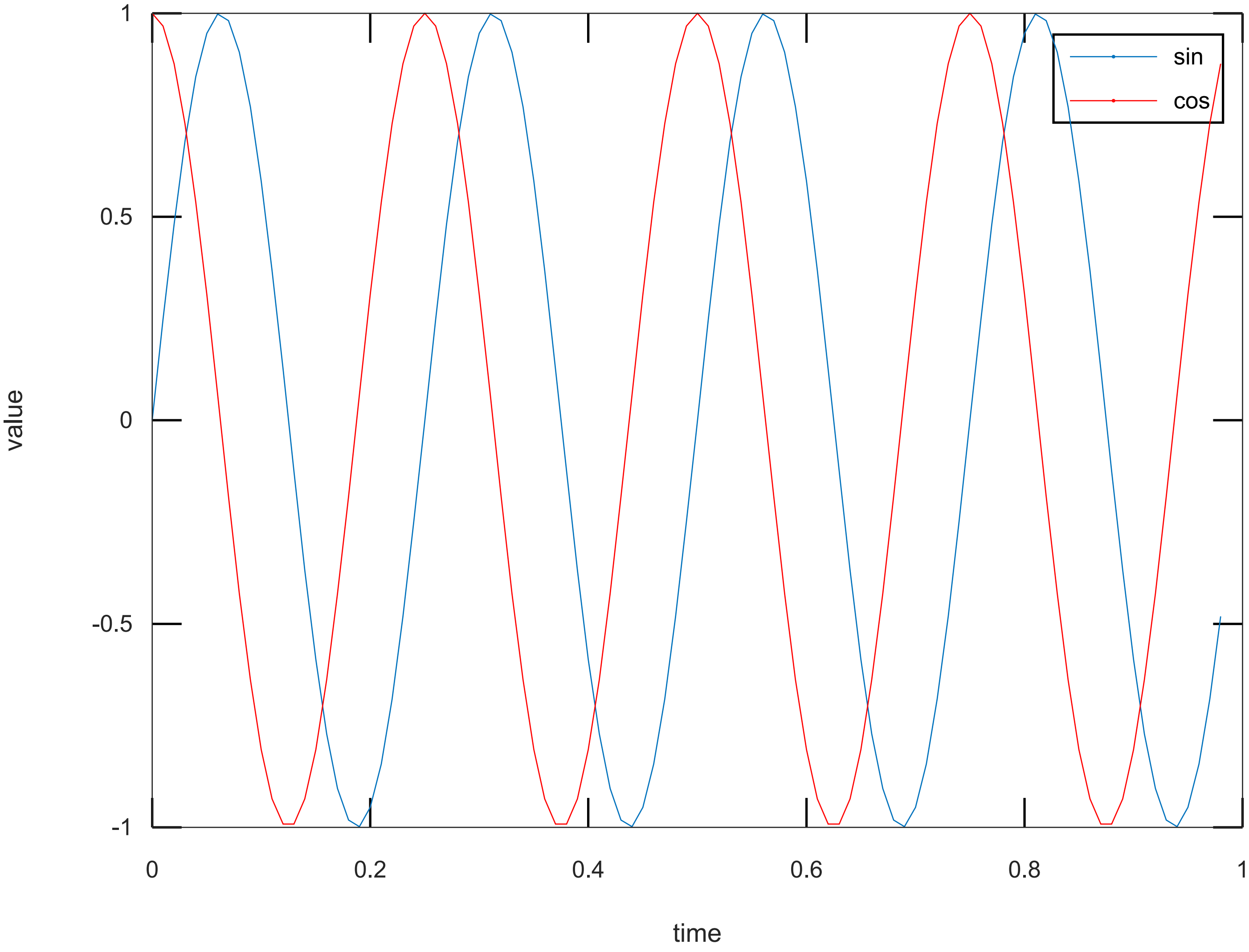

1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1

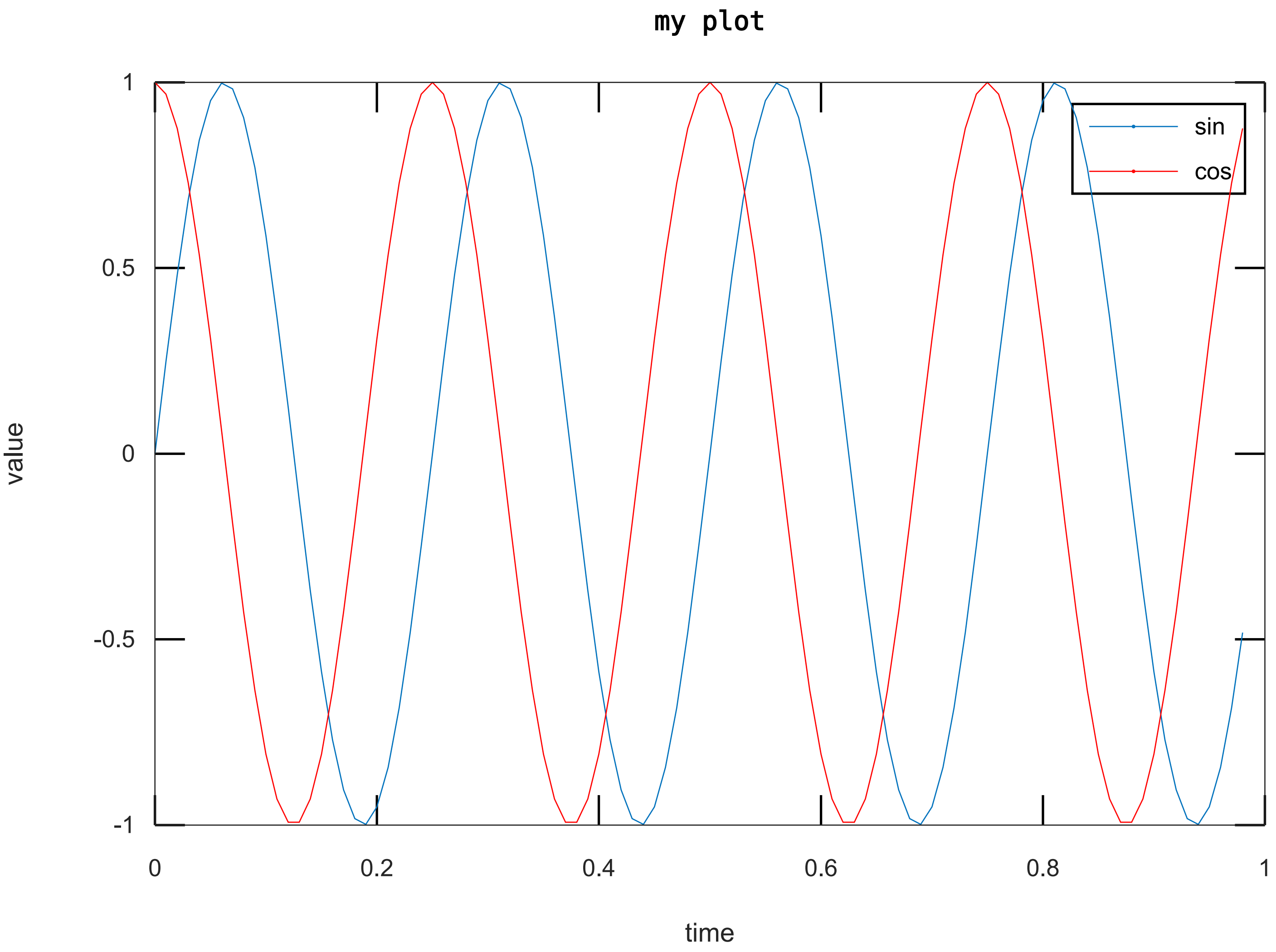

1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1


1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1

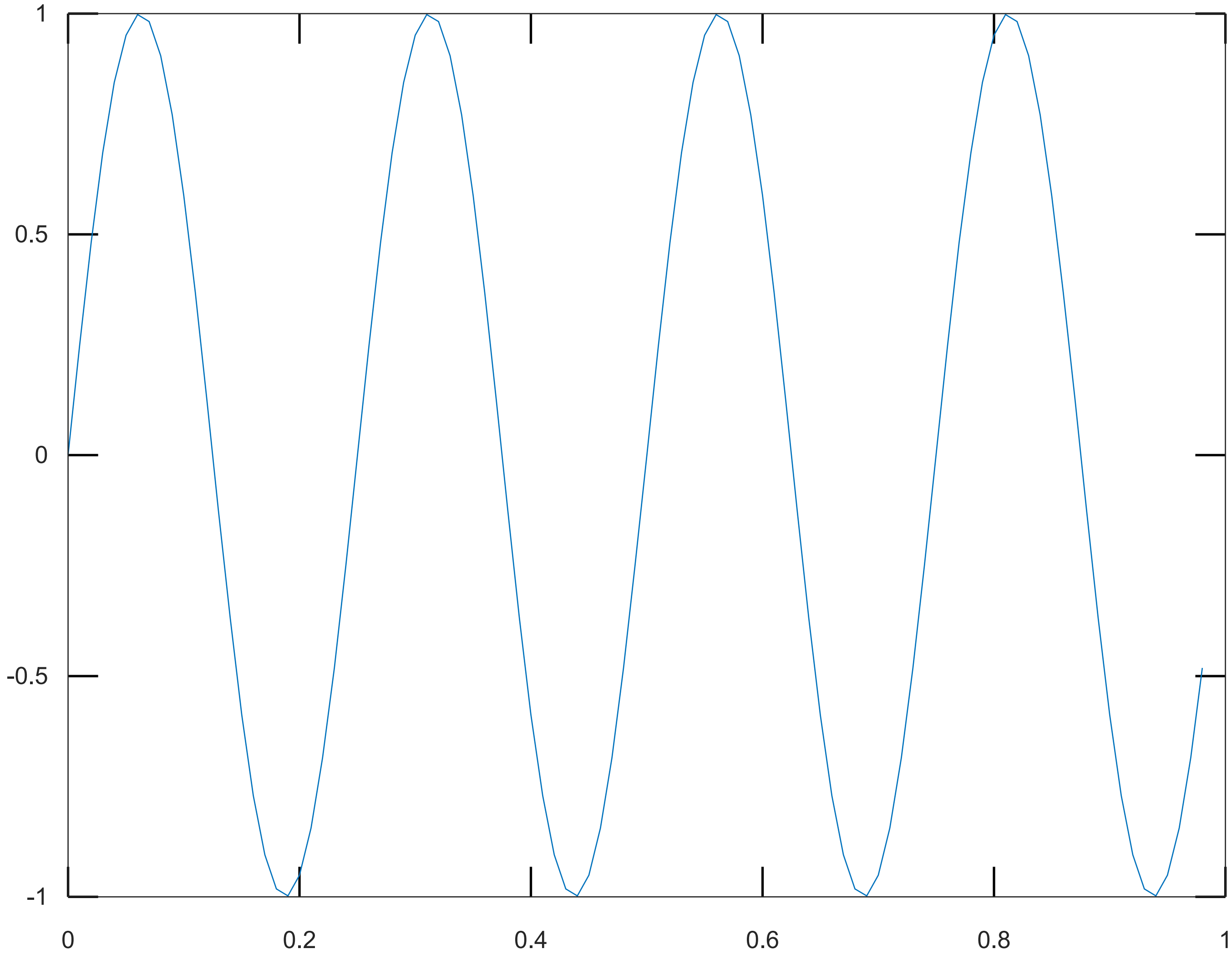

1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
time

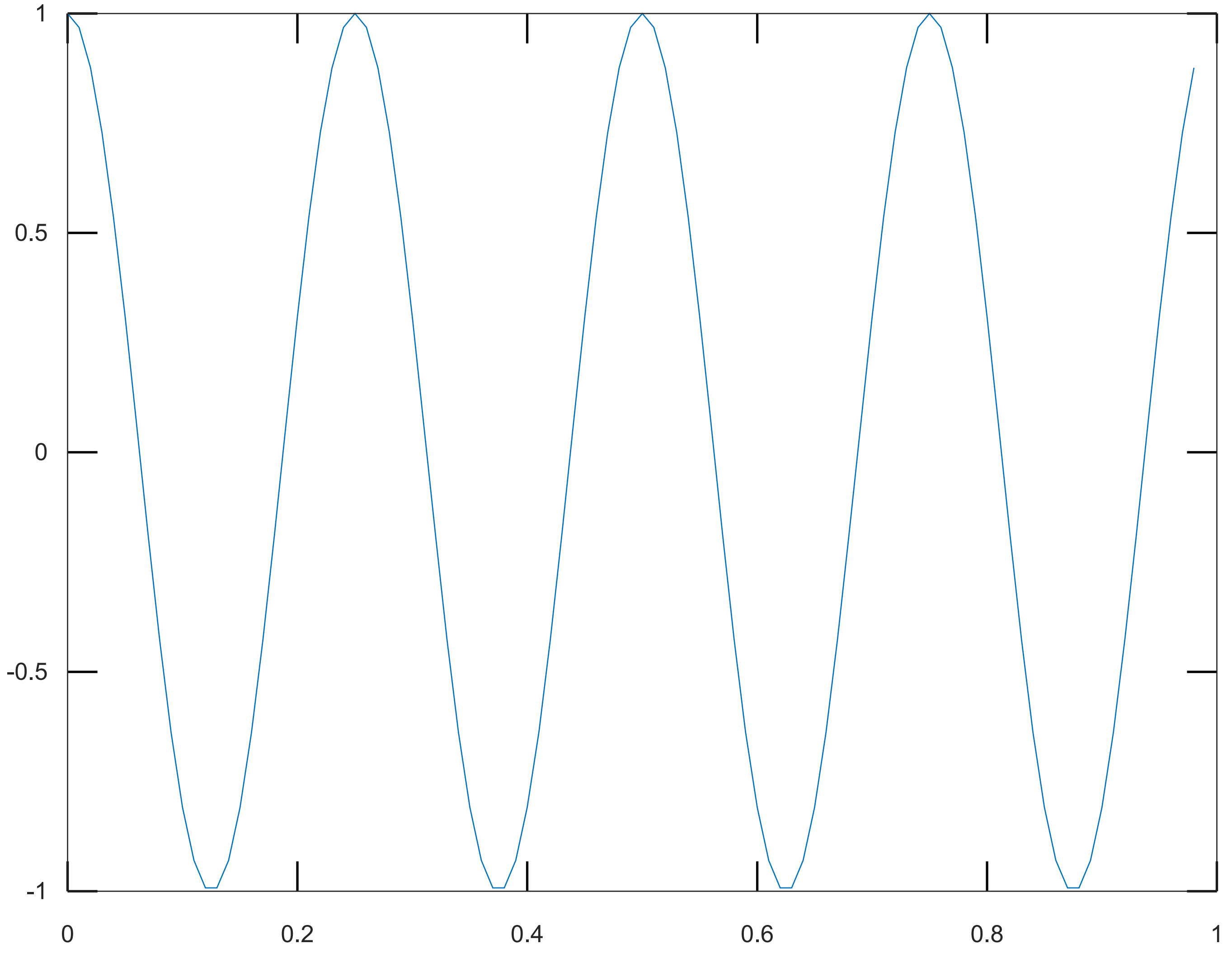

1
0.5
0
-0.5
-1
value
0
0.2
0.4
0.6
0.8
1
time

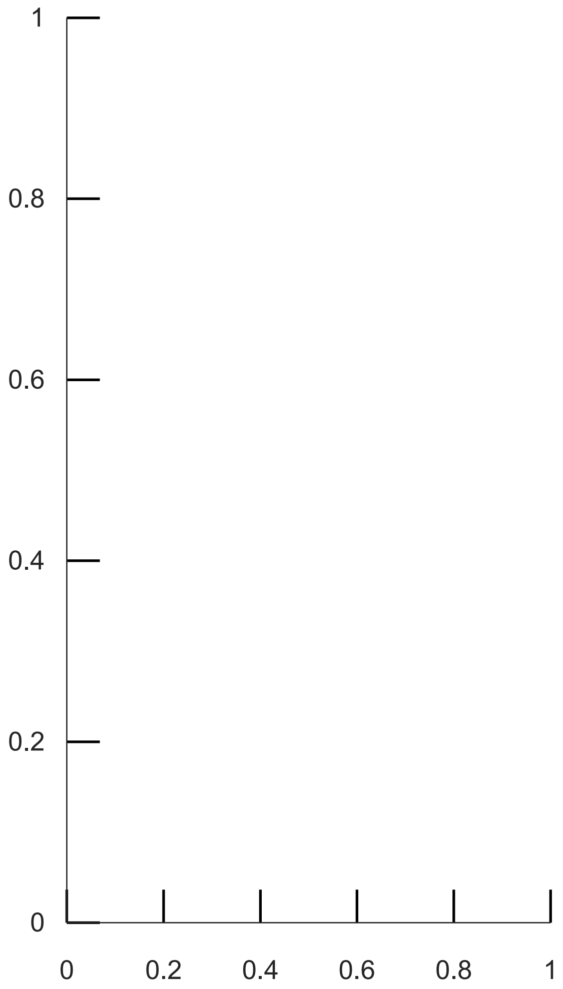

1
0.5
0
-0.5
-1
value
0
0.2
0.4
0.6
0.8
1
time
sin
cos

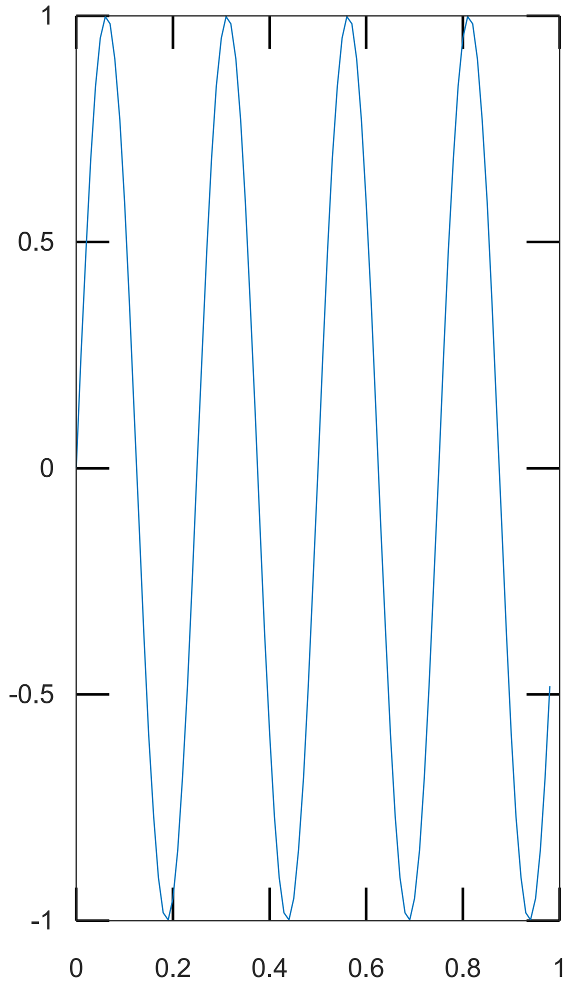


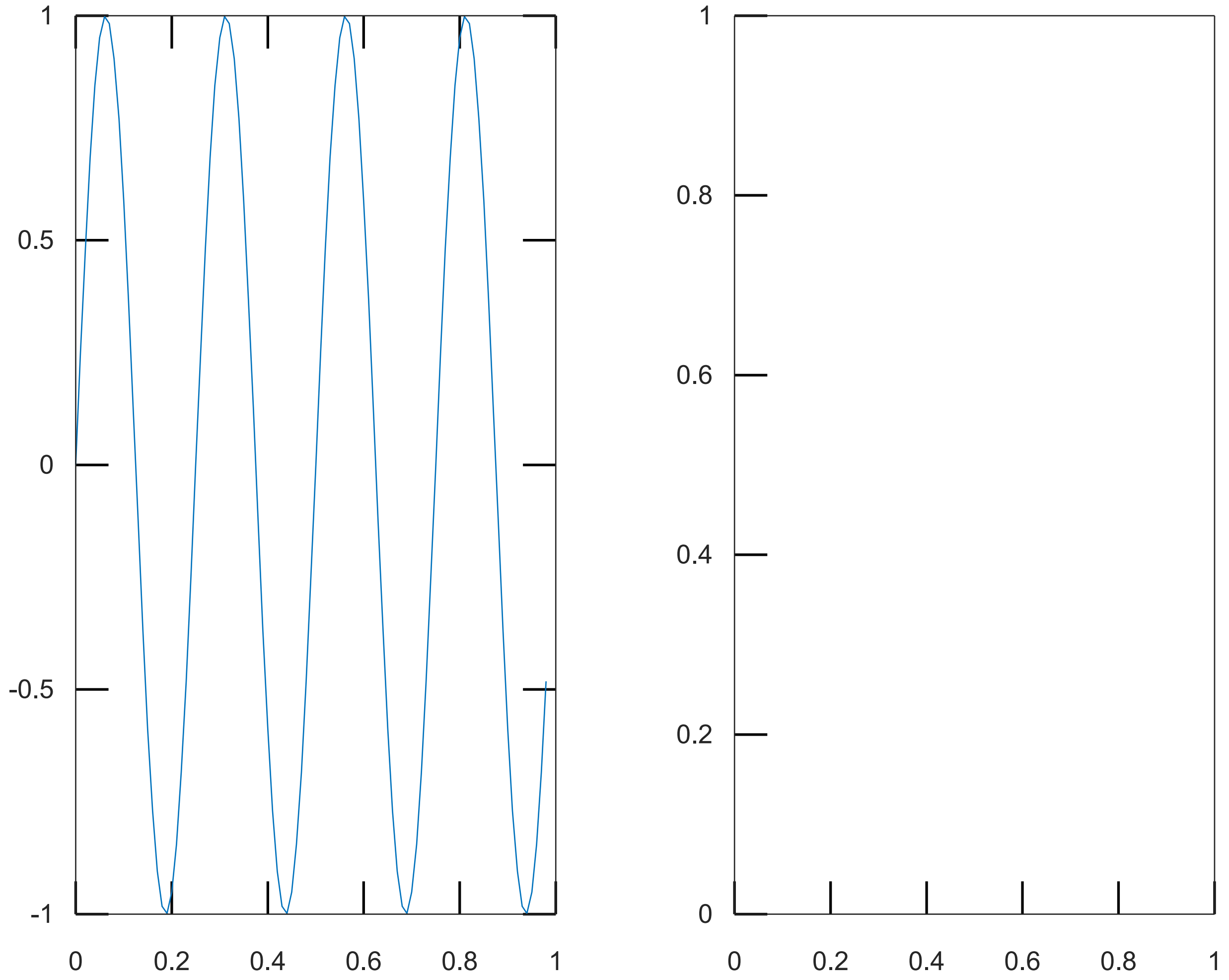
```
parse error:

  syntax error

>>> print -dpng 'myPlot.png'
          ^
```

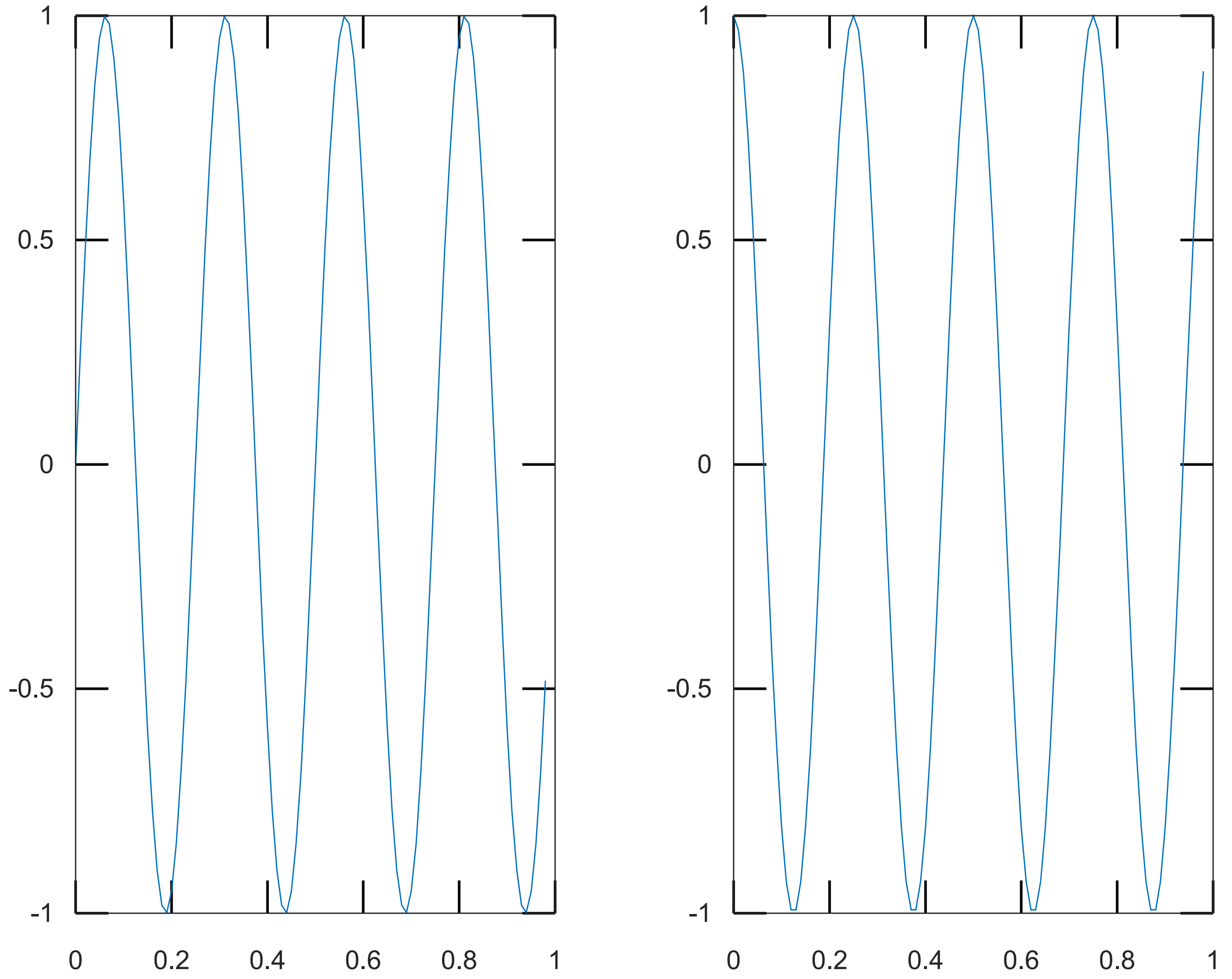

1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1


1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1

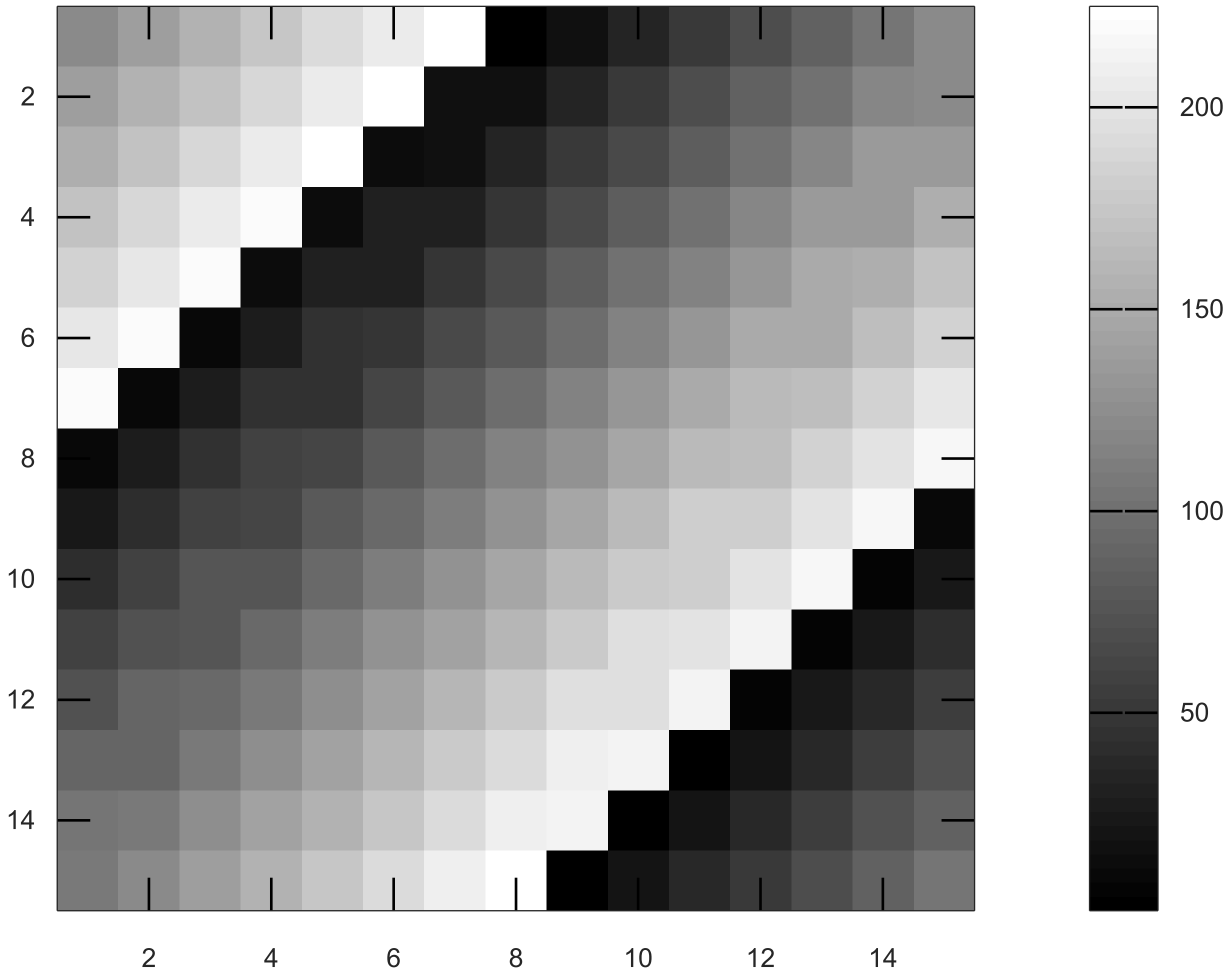

1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1


1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1



```
parse error:

  syntax error

>>> axis([0.5 1 -1 1]); % change axis scale
               ^
```



```
a =  1
b =  2
c =  3
```