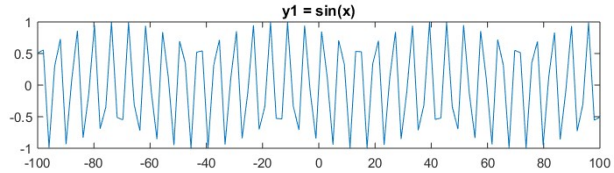
CmpE 362 Homework 1 : Getting Started with MATLAB

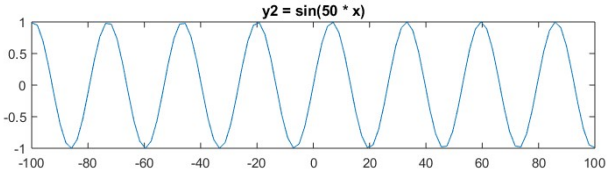
- I learnt how different sin functions will have different patterns and how different random number generation strategies will yield different distributions.
- I leart how to use some basic functions like linspace, randn, rand, subplot, plot, histogram, title, figure, fopen, fscanf, fclose, findpeaks, imread, rgb2gray, mean2, std2, ind2sub, disp, num2str. I learnt how to plot graphs and compare them. I learnt how to read different types of files and process their matrices.
- I sometimes faced difficulties whie using element wise arithmetic operators e.g. where to use vector and where to use element wise ones. I also had difficulties while finding the row and column of min and max values of a matrix.
- MATLAB has lots of builtin functions that makes programming easier. It is also very efficient for matrix computations. But there also some points that you need to be careful while using MATLAB such as you must not use i and j as loop variables and indexes start from one.

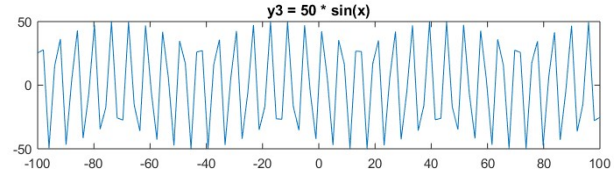
Berk Erol 2014400189

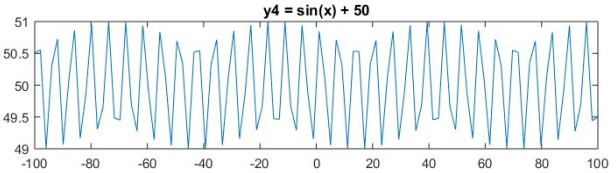
Problem 1

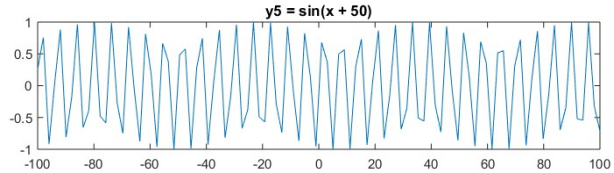
```
% Create a vector from -100 to 100
x = linspace(-100, 100);
% Create functions for plotting
y1 = sin(x);
y2 = sin(50 * x);
y3 = 50 * sin(x);
y4 = sin(x) + 50;
y5 = sin(x + 50);
y6 = 50 * sin(50 * x);
y7 = x .* sin(x);
y8 = sin(x) ./ x;
% Draw all plots into a 4x2 subplot with their titles
subplot(4, 2, 1);
plot(x, y1);
title('y1 = sin(x)');
subplot(4, 2, 2);
plot(x, y2);
title('y2 = sin(50 * x)');
subplot(4, 2, 3);
plot(x, y3);
title('y3 = 50 * sin(x)');
subplot(4, 2, 4);
plot(x, y4);
title('y4 = sin(x) + 50');
subplot(4, 2, 5);
plot(x, y5);
title('y5 = sin(x + 50)');
subplot(4, 2, 6);
plot(x, y6);
title('y6 = 50 * sin(50 * x)');
subplot(4, 2, 7);
plot(x, y7);
title('y7 = x .* sin(x)');
subplot(4, 2, 8);
plot(x, y8);
title('y8 = sin(x) ./ x');
figure;
```

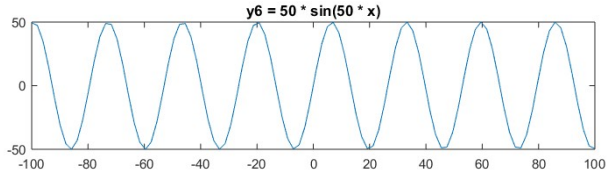

y1 = sin(x)
1
0.5
0
-0.5
-1
-100 -80 -60 -40 -20 0 20 40 60 80 100

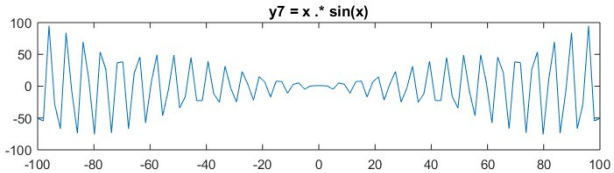

y2 = sin(50 * x)
1
0.5
0
-0.5
-1
-100 -80 -60 -40 -20 0 20 40 60 80 100

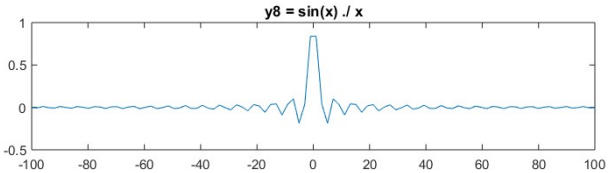

y3 = 50 * sin(x)
50
0
-50
-100 -80 -60 -40 -20 0 20 40 60 80 100


y4 = sin(x) + 50
51
50.5
50
49.5
49
-100 -80 -60 -40 -20 0 20 40 60 80 100


y5 = sin(x + 50)
1
0.5
0
-0.5
-1
-100 -80 -60 -40 -20 0 20 40 60 80 100

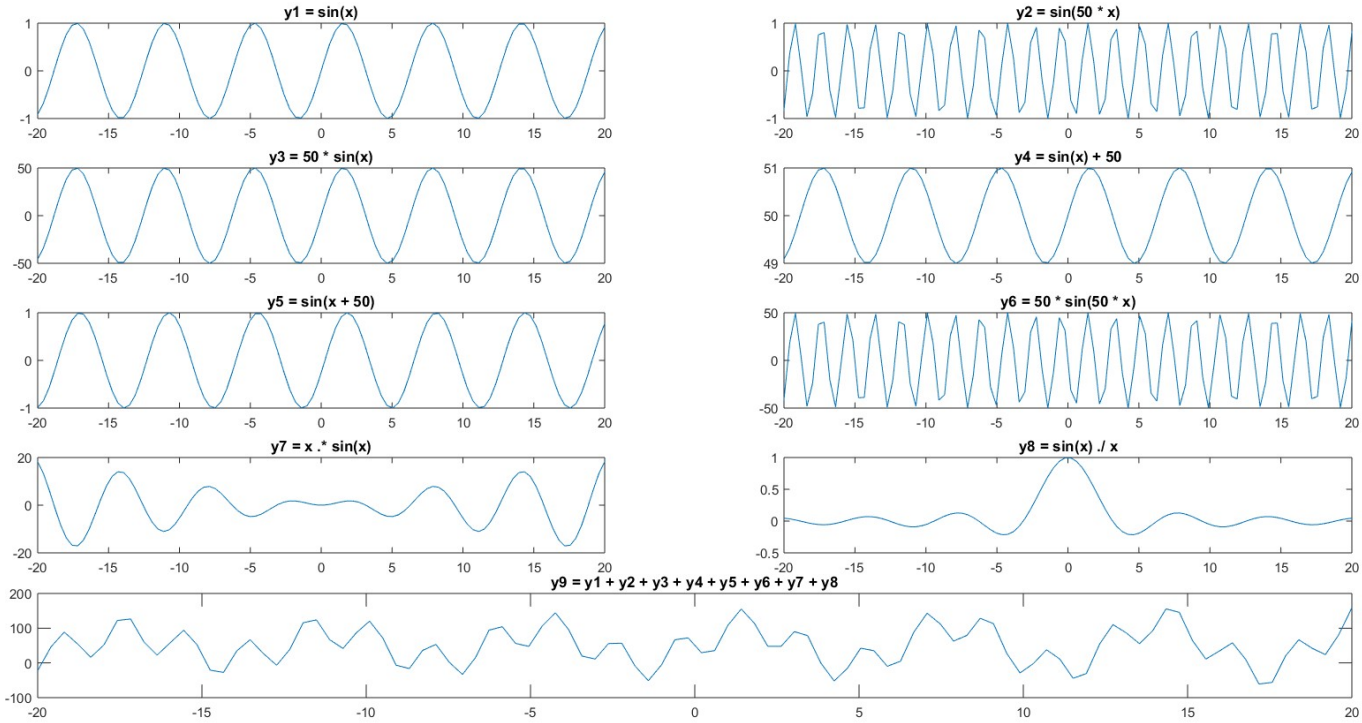

y6 = 50 * sin(50 * x)
50
0
-50
-100 -80 -60 -40 -20 0 20 40 60 80 100


y7 = x .* sin(x)
100
50
0
-50
-100
-100 -80 -60 -40 -20 0 20 40 60 80 100


y8 = sin(x) ./ x
1
0.5
0
-0.5
-100 -80 -60 -40 -20 0 20 40 60 80 100

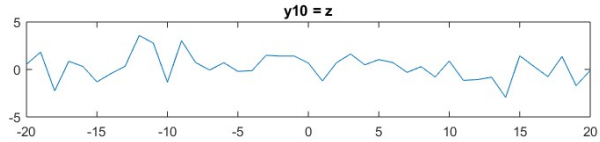
Problem 2

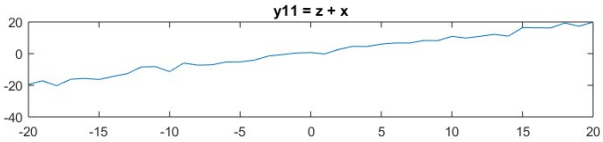
```
% Create a vector from -20 to 20
x = linspace(-20, 20);
% Create functions for plotting
y1 = sin(x);
y2 = sin(50 * x);
y3 = 50 * sin(x);
y4 = sin(x) + 50;
y5 = sin(x + 50);
y6 = 50 * sin(50 * x);
y7 = x .* sin(x);
y8 = sin(x) ./ x;
y9 = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8;
% Draw all plots into a 5x2 subplot with their titles
subplot(5, 2, 1);
plot(x, y1);
title('y1 = sin(x)');
subplot(5, 2, 2);
plot(x, y2);
title('y2 = sin(50 * x)');
subplot(5, 2, 3);
plot(x, y3);
title('y3 = 50 * sin(x)');
subplot(5, 2, 4);
plot(x, y4);
title('y4 = sin(x) + 50');
subplot(5, 2, 5);
plot(x, y5);
title('y5 = sin(x + 50)');
subplot(5, 2, 6);
plot(x, y6);
title('y6 = 50 * sin(50 * x)');
subplot(5, 2, 7);
plot(x, y7);
title('y7 = x .* sin(x)');
subplot(5, 2, 8);
plot(x, y8);
title('y8 = sin(x) ./ x');
subplot(5, 2, [9 10]);
plot(x, y9);
title('y9 = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8');
figure;
```

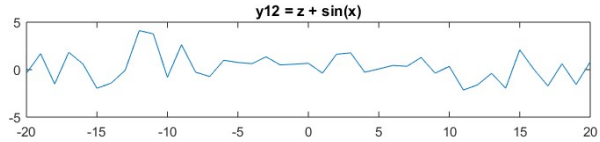

y1 = sin(x)
y2 = sin(50 * x)
y3 = 50 * sin(x)
y4 = sin(x) + 50
y5 = sin(x + 50)
y6 = 50 * sin(50 * x)
y7 = x .* sin(x)
y8 = sin(x) ./ x
y9 = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8

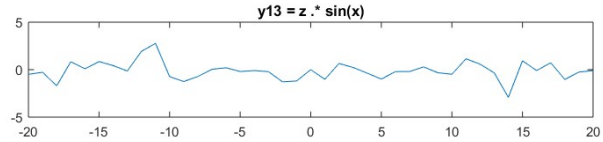
Problem 3

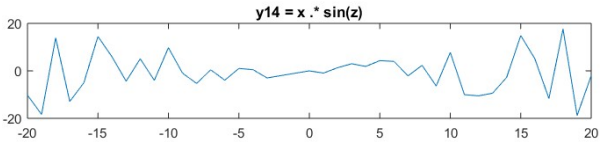
```
% Create a vector from -20 to 20 with 41 elements
x = linspace(-20, 20, 41);
% Generate 41 Gaussian distributed random numbers
z = randn(1, 41);
% Create functions for plotting
y10 = z;
y11 = z + x;
y12 = z + sin(x);
y13 = z .* sin(x);
y14 = x .* sin(z);
y15 = sin(x + z);
y16 = z .* sin(50 * x);
y17 = sin(x + 50 * z);
y18 = sin(x) ./ z;
y19 = y11 + y12 + y13 + y14 + y15 + y16 + y17 + y18;
% Draw all plots into a 5x2 subplot with their titles
subplot(5, 2, 1);
plot(x, y10);
title('y10 = z');
subplot(5, 2, 2);
plot(x, y11);
title('y11 = z + x');
subplot(5, 2, 3);
plot(x, y12);
title('y12 = z + sin(x)');
subplot(5, 2, 4);
plot(x, y13);
title('y13 = z .* sin(x)');
subplot(5, 2, 5);
plot(x, y14);
title('y14 = x .* sin(z)');
subplot(5, 2, 6);
plot(x, y15);
title('y15 = sin(x + z)');
subplot(5, 2, 7);
plot(x, y16);
title('y16 = z .* sin(50 * x)');
subplot(5, 2, 8);
plot(x, y17);
title('y17 = sin(x + 50 * z)');
subplot(5, 2, 9);
plot(x, y18);
title('y18 = sin(x) ./ z');
subplot(5, 2, 10);
plot(x, y19);
title('y19 = y11 + y12 + y13 + y14 + y15 + y16 + y17 + y18');
figure;
```

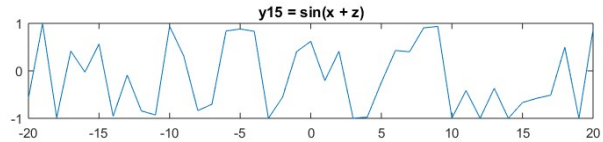

y10 = z
5
0
-5
-20 -15 -10 -5 0 5 10 15 20

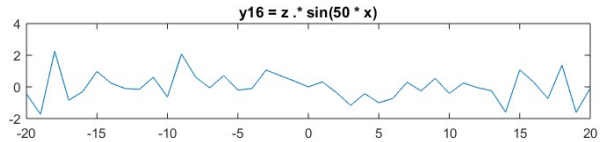

y11 = z + x
20
0
-20
-40
-20 -15 -10 -5 0 5 10 15 20

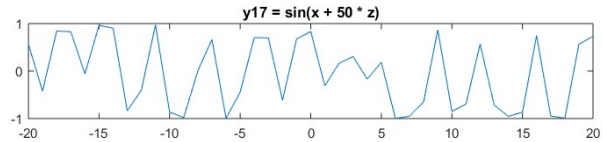

y12 = z + sin(x)
5
0
-5
-20 -15 -10 -5 0 5 10 15 20

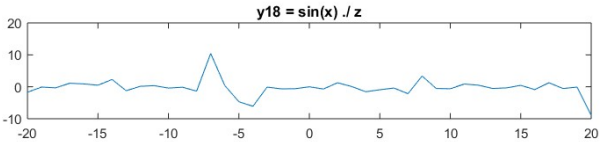

y13 = z .* sin(x)
5
0
-5
-20 -15 -10 -5 0 5 10 15 20

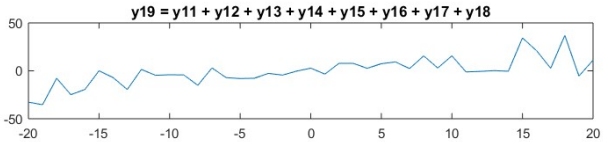

y14 = x .* sin(z)
20
0
-20
-20 -15 -10 -5 0 5 10 15 20


y15 = sin(x + z)
1
0
-1
-20 -15 -10 -5 0 5 10 15 20


y16 = z .* sin(50 * x)
4
2
0
-2
-20 -15 -10 -5 0 5 10 15 20

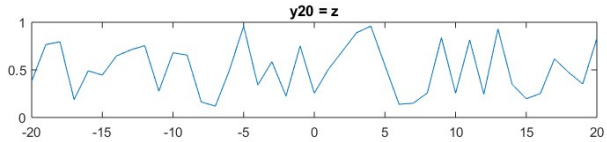

y17 = sin(x + 50 * z)
1
0
-1
-20 -15 -10 -5 0 5 10 15 20

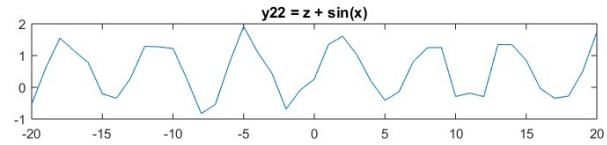

y18 = sin(x) ./ z
20
10
0
-10
-20 -15 -10 -5 0 5 10 15 20

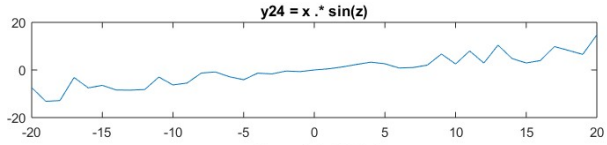

y19 = y11 + y12 + y13 + y14 + y15 + y16 + y17 + y18
50
0
-50
-20 -15 -10 -5 0 5 10 15 20

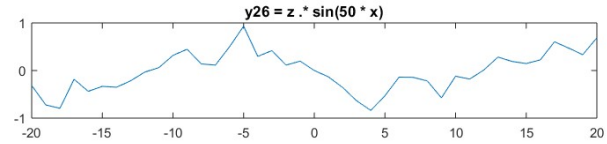
Problem 4

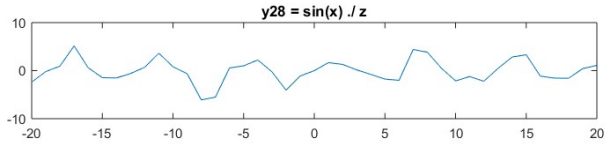
```
% Generate 41 uniformly distributed random numbers
z = rand(1, 41);
% Create functions for plotting
y20 = z;
y21 = z + x;
y22 = z + sin(x);
y23 = z .* sin(x);
y24 = x .* sin(z);
y25 = sin(x + z);
y26 = z .* sin(50 * x);
y27 = sin(x + 50 * z);
y28 = sin(x) ./ z;
y29 = y21 + y22 + y23 + y24 + y25 + y26 + y27 + y28;
% Draw all plots into a 5x2 subplot with their titles
subplot(5, 2, 1);
plot(x, y20);
title('y20 = z');
subplot(5, 2, 2);
plot(x, y21);
title('y21 = z + x');
subplot(5, 2, 3);
plot(x, y22);
title('y22 = z + sin(x)');
subplot(5, 2, 4);
plot(x, y23);
title('y23 = z .* sin(x)');
subplot(5, 2, 5);
plot(x, y24);
title('y24 = x .* sin(z)');
subplot(5, 2, 6);
plot(x, y25);
title('y25 = sin(x + z)');
subplot(5, 2, 7);
plot(x, y26);
title('y26 = z .* sin(50 * x)');
subplot(5, 2, 8);
plot(x, y27);
title('y27 = sin(x + 50 * z)');
subplot(5, 2, 9);
plot(x, y28);
title('y28 = sin(x) ./ z');
subplot(5, 2, 10);
plot(x, y29);
title('y29 = y21 + y22 + y23 + y24 + y25 + y26 + y27 + y28');
figure;
```

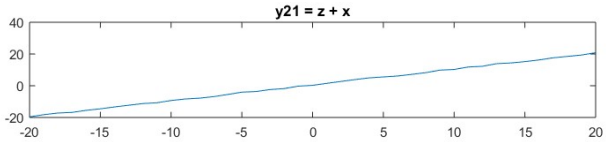

y20 = z

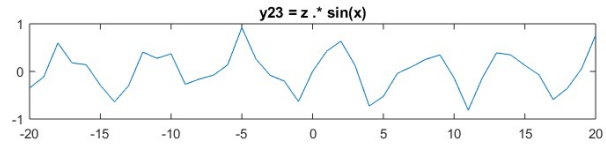

y22 = z + sin(x)

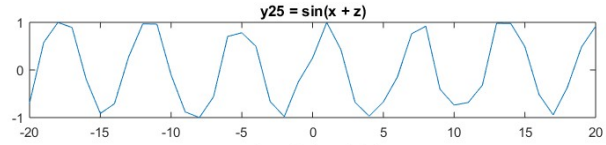

y24 = x .* sin(z)

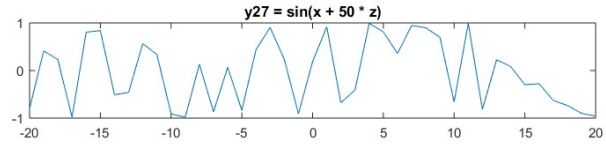

y26 = z .* sin(50 * x)

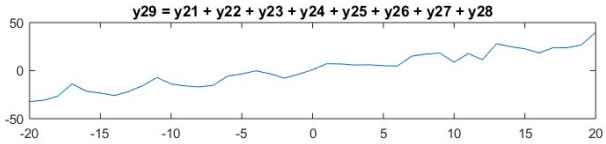

y28 = sin(x) ./ z


y21 = z + x


y23 = z .* sin(x)

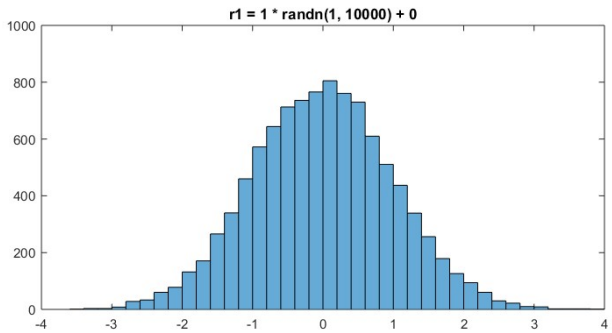

y25 = sin(x + z)

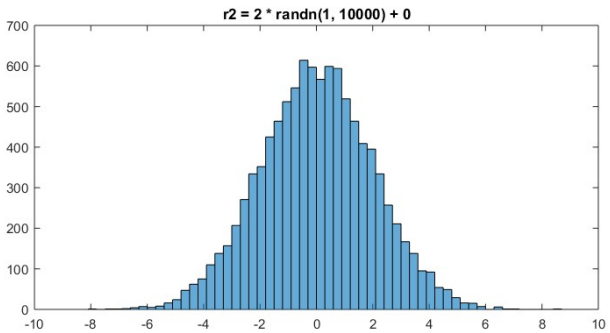

y27 = sin(x + 50 * z)

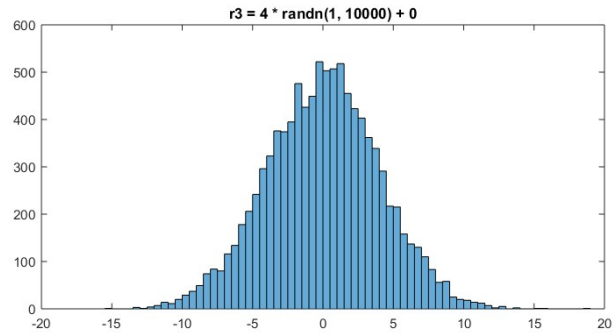

y29 = y21 + y22 + y23 + y24 + y25 + y26 + y27 + y28

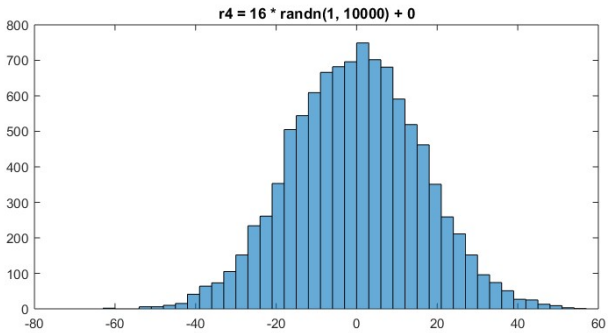
Problem 5

```
% Generate 10000 Gaussian distributed random numbers with mean 0, variance
1
r1 = 1 * randn(1, 10000) + 0;
% Generate 10000 Gaussian distributed random numbers with mean 0, variance
4
r2 = 2 * randn(1, 10000) + 0;
% Generate 10000 Gaussian distributed random numbers with mean 0, variance
16
r3 = 4 * randn(1, 10000) + 0;
% Generate 10000 Gaussian distributed random numbers with mean 0, variance
256
r4 = 16 * randn(1, 10000) + 0;
% Draw all histograms into a 2x2 subplot with their titles
subplot(2, 2, 1);
histogram(r1);
title('r1 = 1 * randn(1, 10000) + 0');
subplot(2, 2, 2);
histogram(r2);
title('r2 = 2 * randn(1, 10000) + 0');
subplot(2, 2, 3);
histogram(r3);
title('r3 = 4 * randn(1, 10000) + 0');
subplot(2, 2, 4);
histogram(r4);
title('r4 = 16 * randn(1, 10000) + 0');
figure;
```



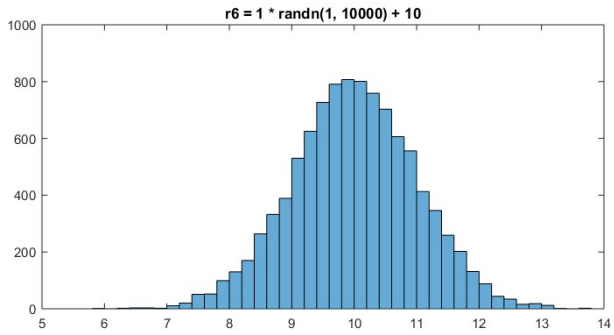


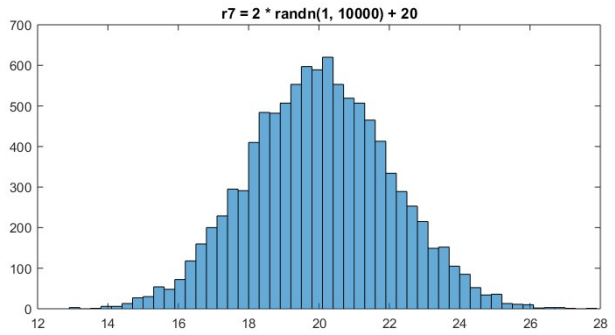


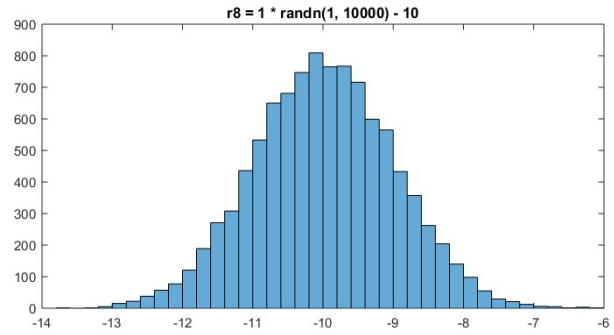


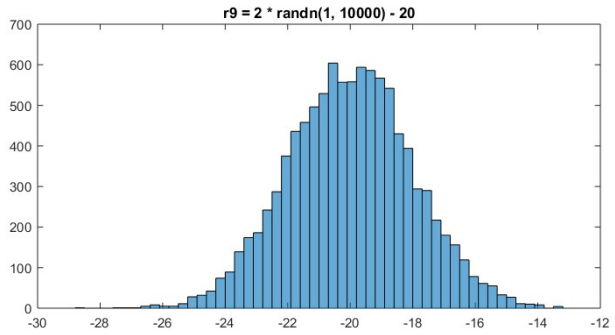
Problem 6

```
% Generate 10000 Gaussian distributed random numbers with mean 10, variance
1
r6 = 1 * randn(1, 10000) + 10;
% Generate 10000 Gaussian distributed random numbers with mean 20, variance
4
r7 = 2 * randn(1, 10000) + 20;
% Generate 10000 Gaussian distributed random numbers with mean -10,
variance 1
r8 = 1 * randn(1, 10000) - 10;
% Generate 10000 Gaussian distributed random numbers with mean -20,
variance 4
r9 = 2 * randn(1, 10000) - 20;
% Draw all histograms into a 2x2 subplot with their titles
subplot(2, 2, 1);
histogram(r6);
title('r6 = 1 * randn(1, 10000) + 10');
subplot(2, 2, 2);
histogram(r7);
title('r7 = 2 * randn(1, 10000) + 20');
subplot(2, 2, 3);
histogram(r8);
title('r8 = 1 * randn(1, 10000) - 10');
subplot(2, 2, 4);
histogram(r9);
title('r9 = 2 * randn(1, 10000) - 20');
figure;
```



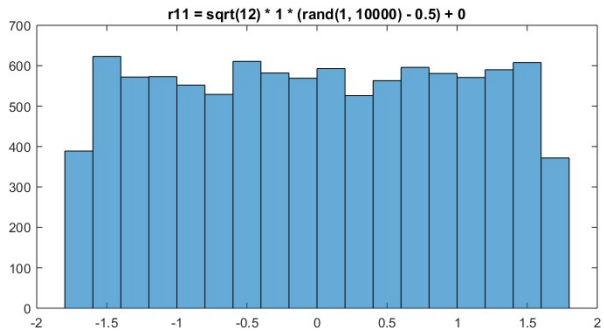


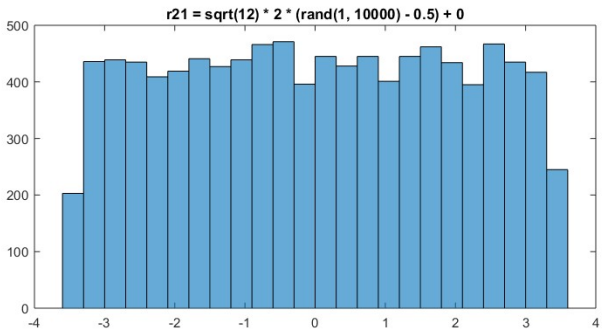


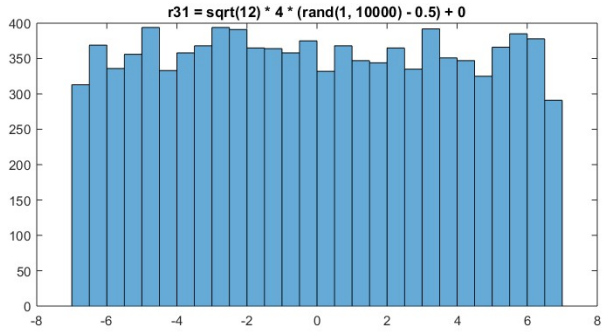


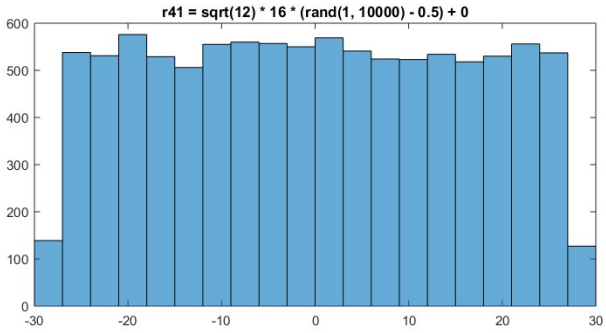
Problem 7

```
% Generate 10000 uniformly distributed random numbers with mean 0, variance
1
r11 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) + 0;
% Generate 10000 uniformly distributed random numbers with mean 0, variance
4
r21 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) + 0;
% Generate 10000 uniformly distributed random numbers with mean 0, variance
16
r31 = sqrt(12) * 4 * (rand(1, 10000) - 0.5) + 0;
% Generate 10000 uniformly distributed random numbers with mean 0, variance
256
r41 = sqrt(12) * 16 * (rand(1, 10000) - 0.5) + 0;
% Draw all histograms into a 2x2 subplot with their titles
subplot(2, 2, 1);
histogram(r11);
title('r11 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) + 0');
subplot(2, 2, 2);
histogram(r21);
title('r21 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) + 0');
subplot(2, 2, 3);
histogram(r31);
title('r31 = sqrt(12) * 4 * (rand(1, 10000) - 0.5) + 0');
subplot(2, 2, 4);
histogram(r41);
title('r41 = sqrt(12) * 16 * (rand(1, 10000) - 0.5) + 0');
figure;
```



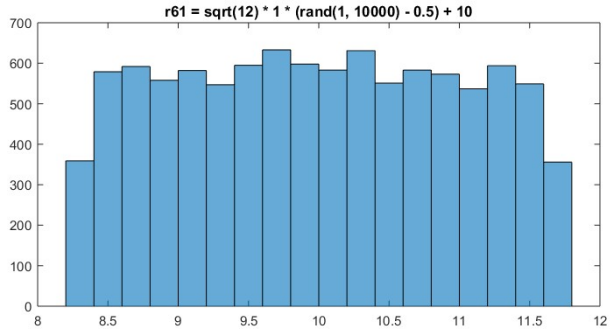


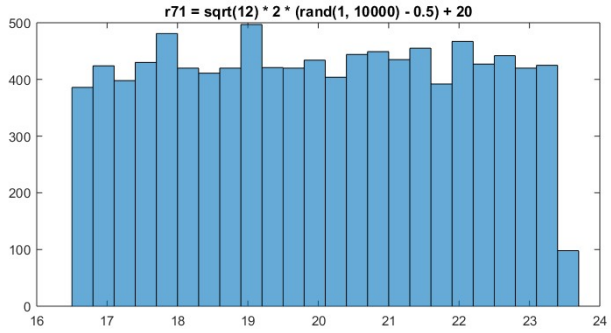


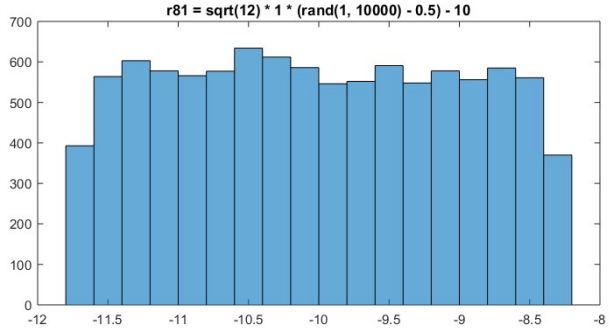


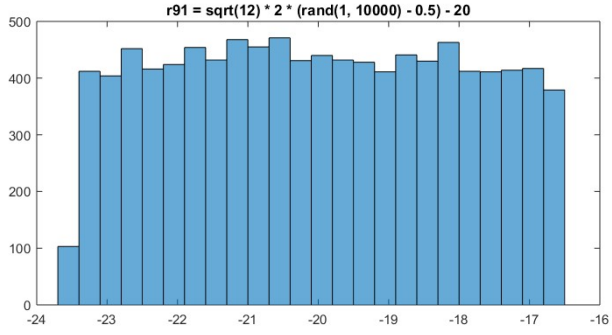
Problem 8

```
% Generate 10000 uniformly distributed random numbers with mean 10,
variance 1
r61 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) + 10;
% Generate 10000 uniformly distributed random numbers with mean 20,
variance 4
r71 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) + 20;
% Generate 10000 uniformly distributed random numbers with mean -10,
variance 1
r81 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) - 10;
% Generate 10000 uniformly distributed random numbers with mean -20,
variance 4
r91 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) - 20;
% Draw all histograms into a 2x2 subplot with their titles
subplot(2, 2, 1);
histogram(r61);
title('r61 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) + 10');
subplot(2, 2, 2);
histogram(r71);
title('r71 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) + 20');
subplot(2, 2, 3);
histogram(r81);
title('r81 = sqrt(12) * 1 * (rand(1, 10000) - 0.5) - 10');
subplot(2, 2, 4);
histogram(r91);
title('r91 = sqrt(12) * 2 * (rand(1, 10000) - 0.5) - 20');
```



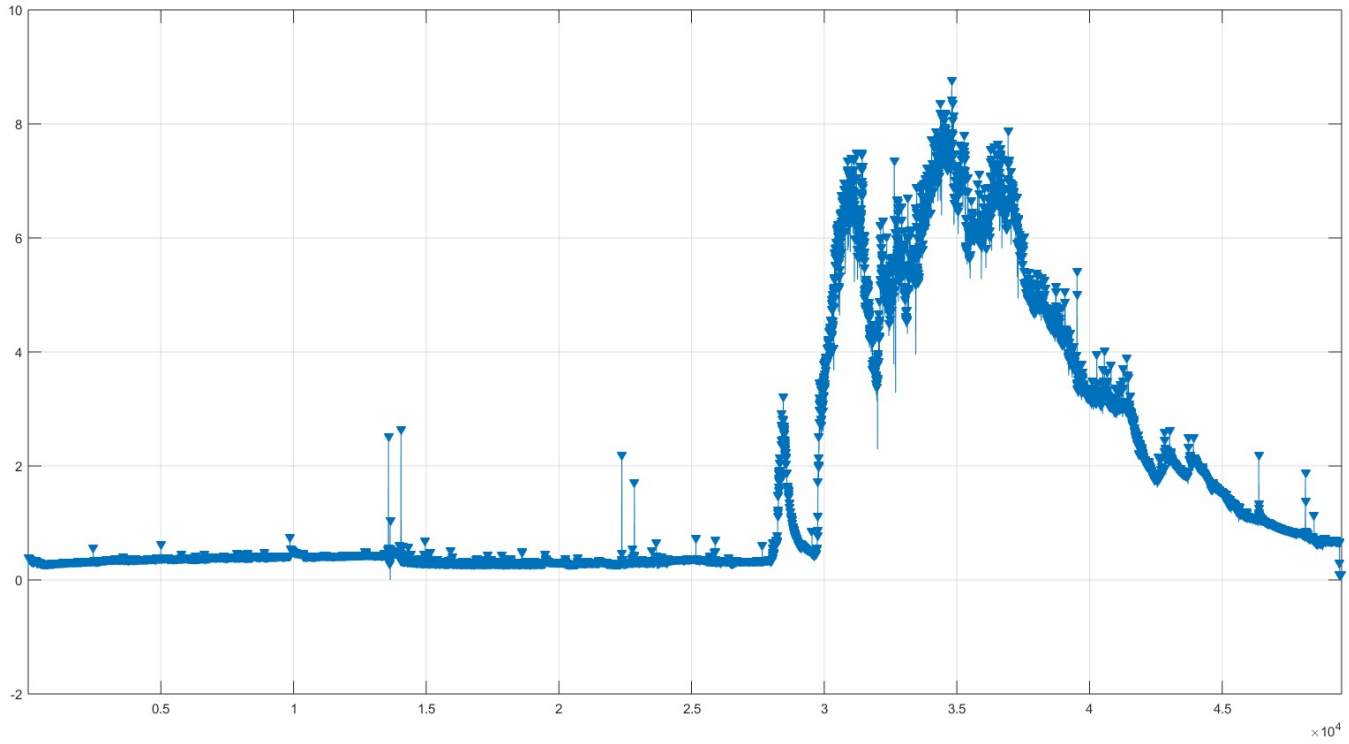






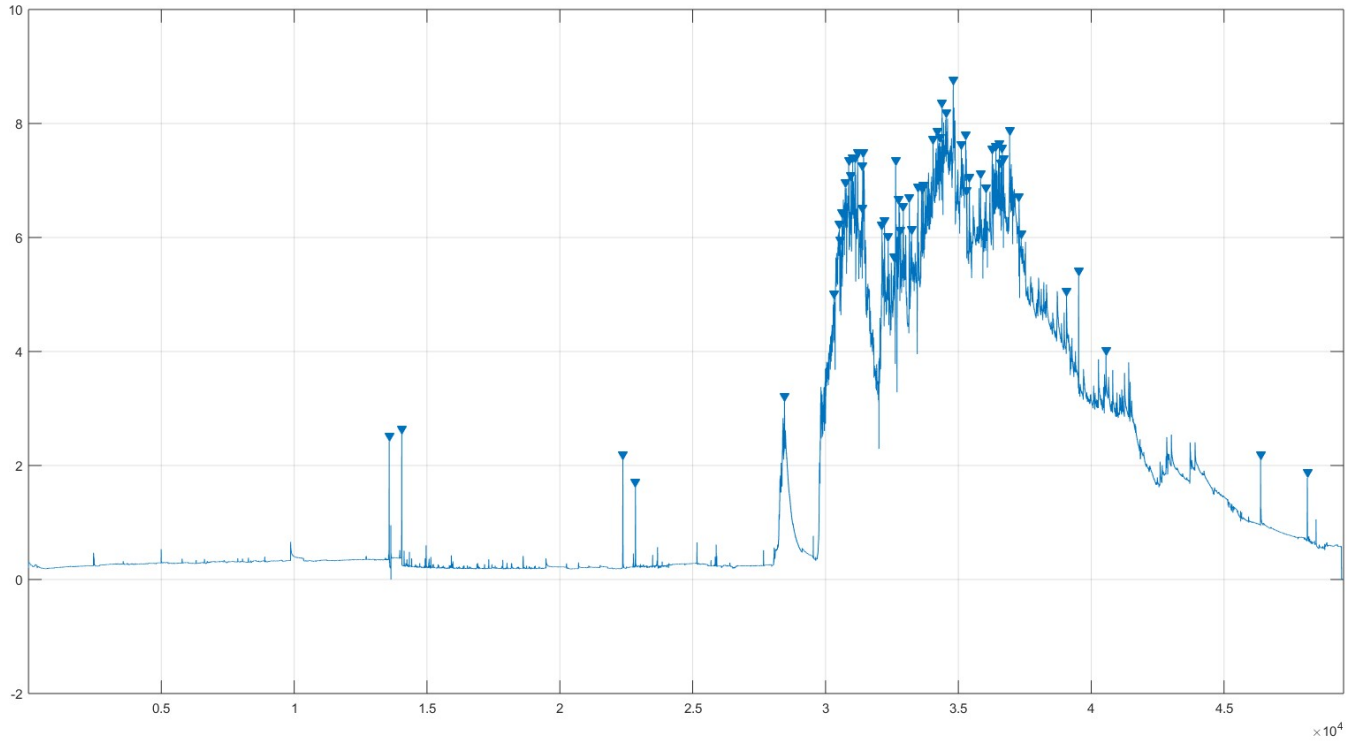
Problem 9

```
% Prepare file for reading
file = fopen('HW1_material/exampleSignal.csv', 'r');
% Read the file
v = fscanf(file, '%f');
% Close the file source
fclose(file);
% Discard the first three elements
v1 = v(4:end);
% Find peaks from data
findpeaks(v1);
```

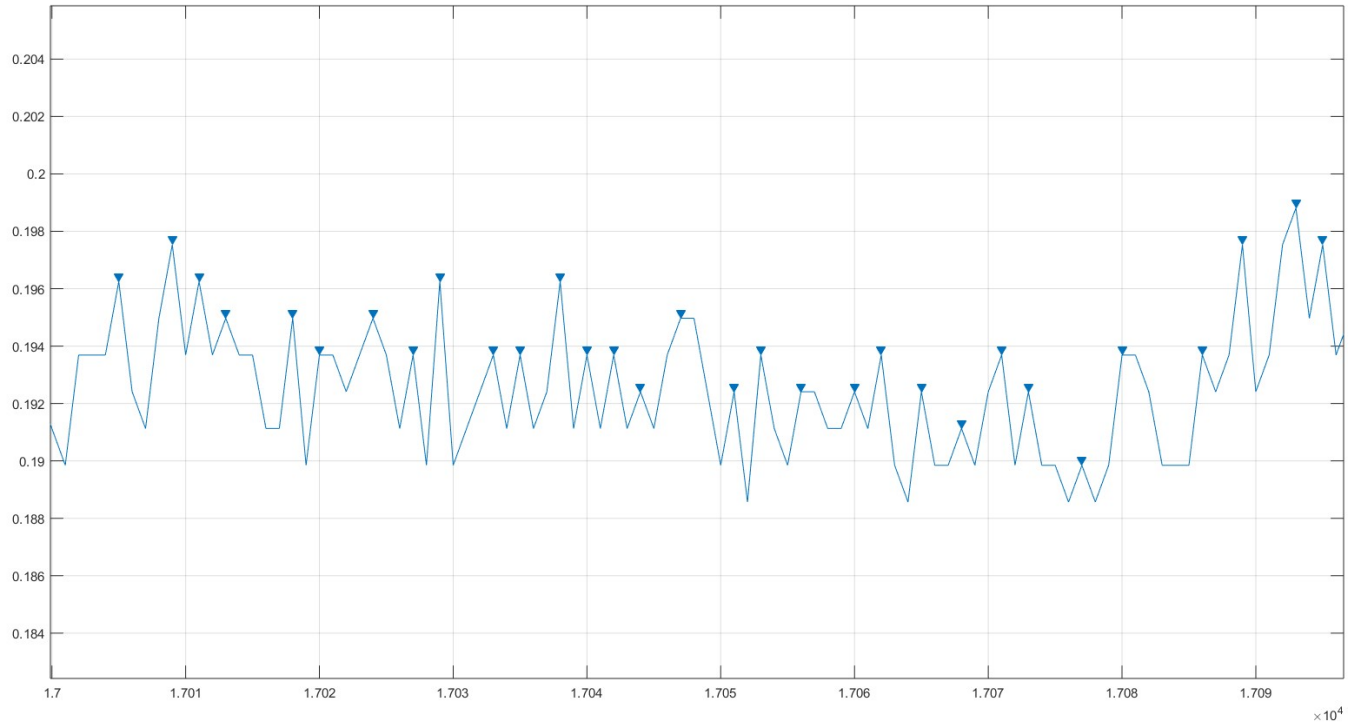


```
% Find peaks from data while clearing some of the noise
findpeaks(v1, 'MinPeakProminence', 1);
```

I set the MinPeakProminence property to 1 to clear some noise and have a more understandable image. It discards some peaks that have a vertical drop less than 1 on both sides.



Without the MinPeakProminence property, it will still not show some peaks. If after the peak the graph continues with the same value, it only highlights the first point in that continuous area (1.7048 and 1.708).



Problem 10

```
% Read the image file into a matrix
img = imread('HW1_material/lena.png');
% Convert the image to black&white
bw = rgb2gray(img);
% Find mean of matrix
m = mean2(bw);
% Find standard deviation of matrix
s = std2(bw);
% Find the maximum and its index in column representation
[vmax, imax] = max(bw(:));
% Find row and column of maximum
[rmax, cmax] = ind2sub(size(bw), imax);
% Find the minimum and its index in column representation
[vmin, imin] = min(bw(:));
% Find row and column of minimum
[rmin, cmin] = ind2sub(size(bw), imin);
% Print the results
disp(['Mean: ' num2str(m)]);
disp(['Standard Deviation: ' num2str(s)]);
disp(['Maximum value: ' num2str(vmax)]);
disp(['Index of maximum in column representation: ' num2str(imax)]);
disp(['Row of maximum: ' num2str(rmax)]);
disp(['Column of maximum: ' num2str(cmax)]);
disp(['Minimum value: ' num2str(vmin)]);
disp(['Index of minimum in column representation: ' num2str(imin)]);
disp(['Row of minimum: ' num2str(rmin)]);
disp(['Column of minimum: ' num2str(cmin)]);
```

```
Mean: 124.0425
Standard Deviation: 47.8557
Maximum value: 245
Index of maximum in column representation: 202514
Row of maximum: 274
Column of maximum: 396
Minimum value: 25
Index of minimum in column representation: 1608
Row of minimum: 72
Column of minimum: 4
```