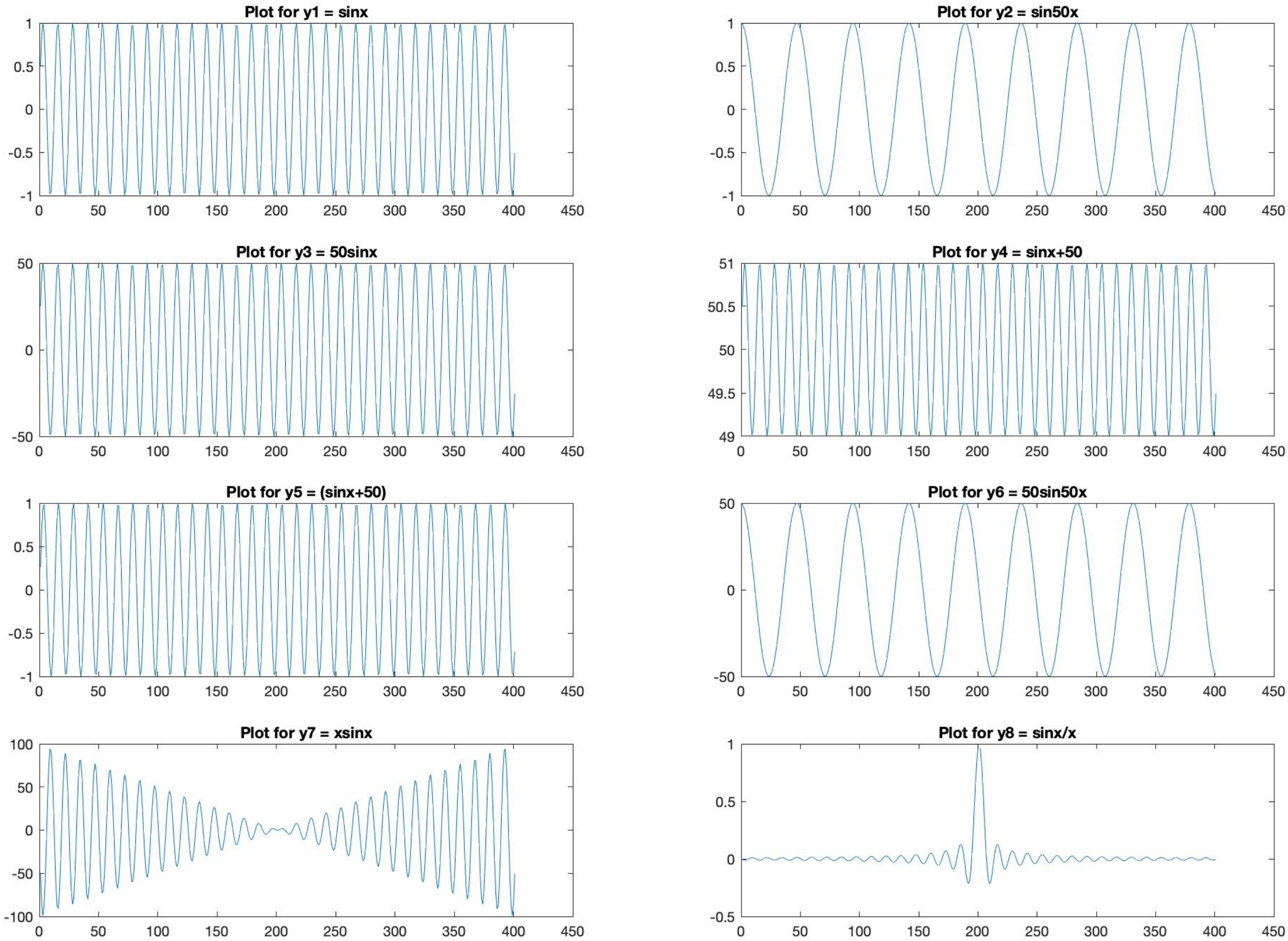
Hatice Melike Ecevit
Cmpe362
2016400138

Homework 1

Getting Started with Matlab

Problem 1:



If we think the wave equation y(t) = Asin(wt+ϕ);

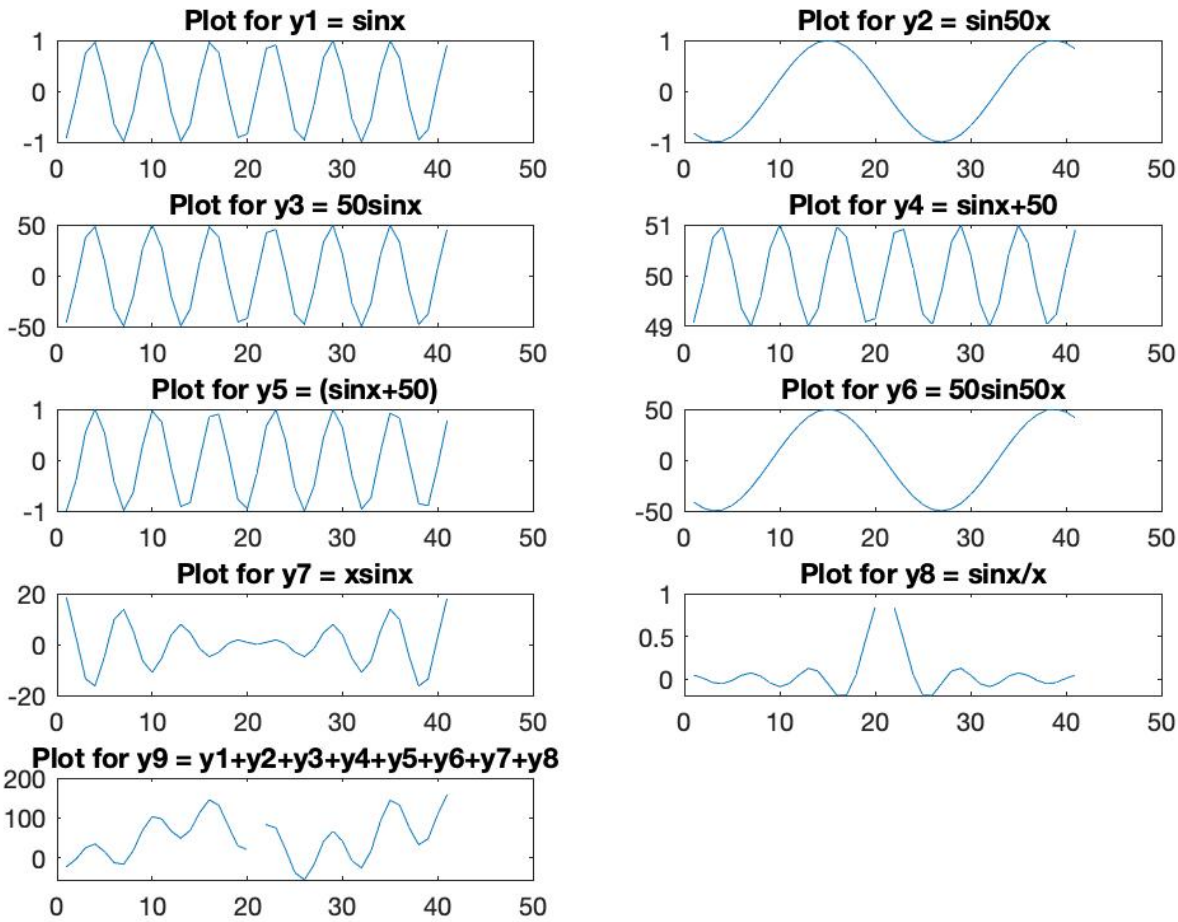
If we increase frequency, plot becomes more sparse as we see the difference between sin(x) and sin(50x). We can think these plots as waves and x as time axis. If we increase A, the plot becomes longer in y direction as we see the difference between sin(x) and 50sin(x). Also if we increase the ϕ, the plot shifts to the right. As we see the difference between sin(x) and sin(x + 50).

Code:

```
%problem 1
x = -100:0.5:100;
y1 = sin(x);
y2 = sin(50.*x);
y3 = 50*sin(x);
y4 = sin(x)+50;
y5 = sin(50.+x);
y6 = 50*sin(50.*x);
y7 = x.*sin(x);
y8 = sin(x)./x;

figure(1);
subplot(4,2,1);
plot(y1);
title('Plot for y1 = sinx');

subplot(4,2,2);
plot(y2);
title('Plot for y2 = sin50x');

subplot(4,2,3);
plot(y3);
title('Plot for y3 = 50sinx');

subplot(4,2,4);
plot(y4);
title('Plot for y4 = sinx+50');

subplot(4,2,5);
plot(y5);
title('Plot for y5 = (sinx+50)');

subplot(4,2,6);
```

```
plot(y6);
title('Plot for y6 = 50sin50x');

subplot(4,2,7);
plot(y7);
title('Plot for y7 = xsinx');

subplot(4,2,8);
plot(y8);
title('Plot for y8 = sinx/x’);
```

Problem 2:



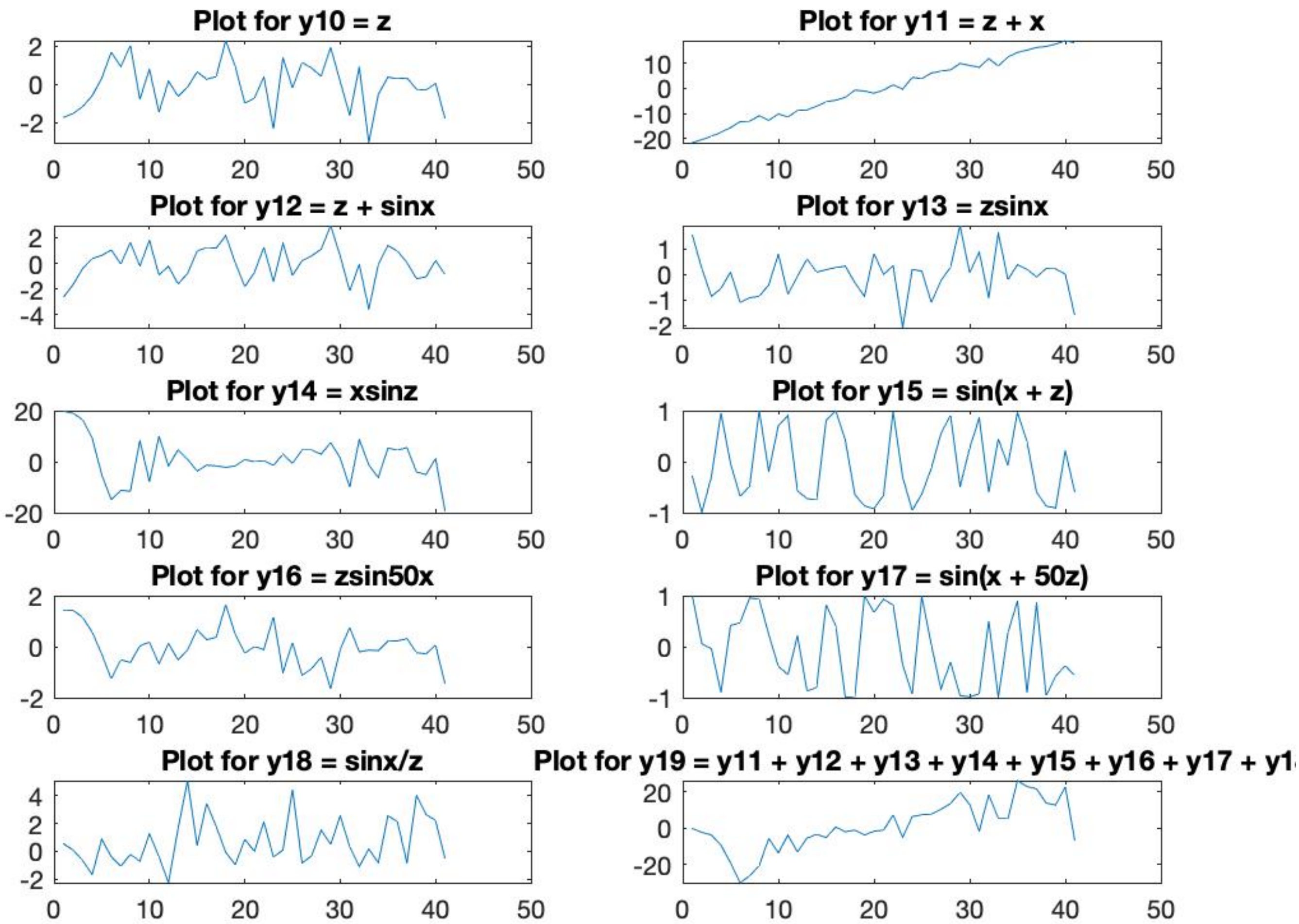
In this problem, we are looking the wave a bit more closer because our time axis is smaller. Also since we have less points, our plots does not look more smooth.

Code:

```
%Problem 2

x = -20:20;

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

figure(2);
subplot(5,2,1);
plot(y1);
title('Plot for y1 = sinx');

subplot(5,2,2);
plot(y2);
title('Plot for y2 = sin50x');

subplot(5,2,3);
plot(y3);
title('Plot for y3 = 50sinx');

subplot(5,2,4);
plot(y4);
title('Plot for y4 = sinx+50');

subplot(5,2,5);
```

```
plot(y5);
title('Plot for y5 = (sinx+50)');

subplot(5,2,6);
plot(y6);
title('Plot for y6 = 50sin50x');

subplot(5,2,7);
plot(y7);
title('Plot for y7 = xsinx');

subplot(5,2,8);
plot(y8);
title('Plot for y8 = sinx/x');

subplot(5,2,9);
plot(y9);
title('Plot for y9 = y1+y2+y3+y4+y5+y6+y7+y8');
```

Problem 3:



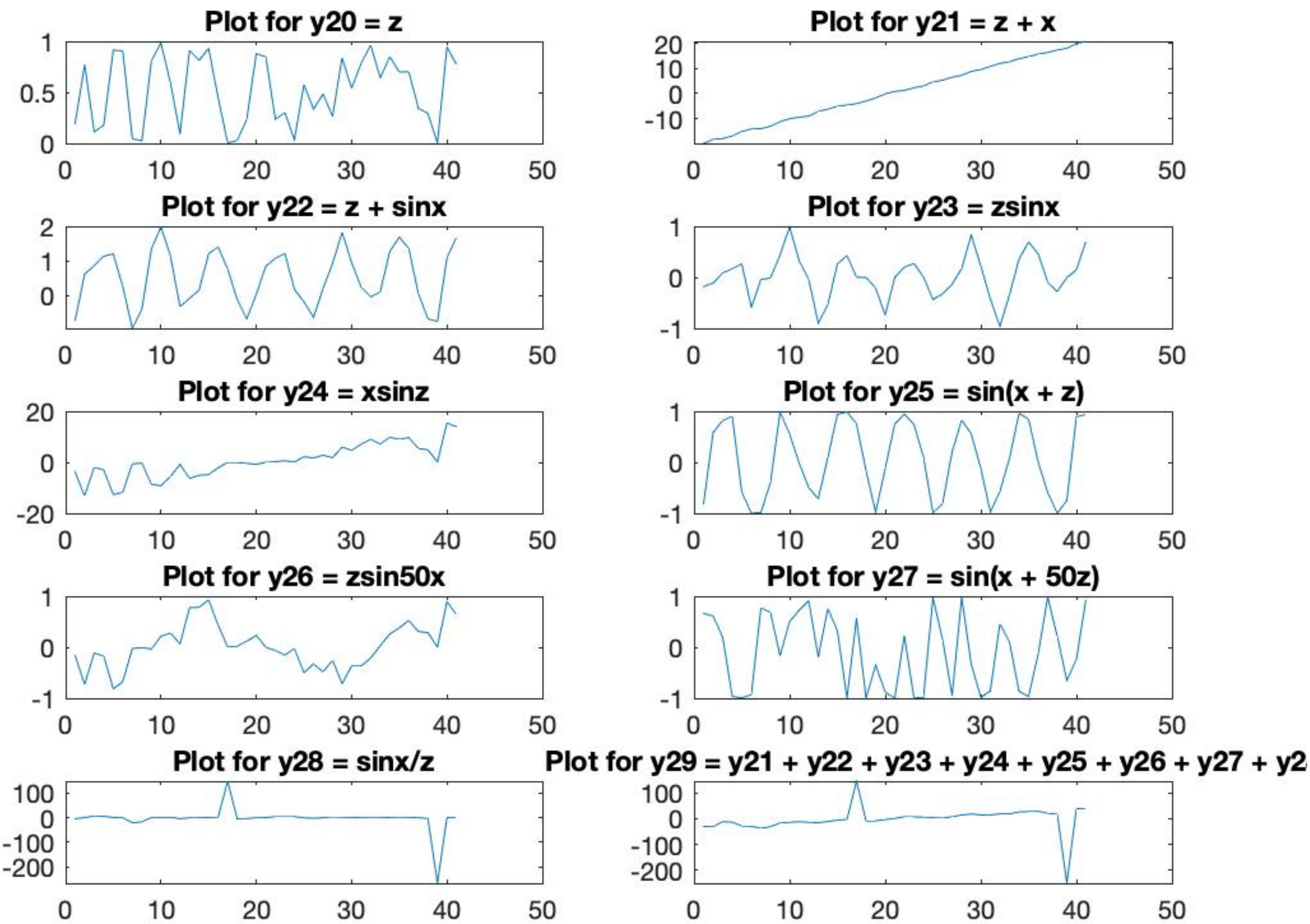
Code:

```
%Problem 3


z = randn(1,41);
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;

figure(3);
subplot(5,2,1);
plot(y10);
title('Plot for y10 = z');

subplot(5,2,2);
plot(y11);
title('Plot for y11 = z + x');

subplot(5,2,3);
plot(y12);
title('Plot for y12 = z + sinx');

subplot(5,2,4);
plot(y13);
title('Plot for y13 = zsinx');

subplot(5,2,5);
plot(y14);
title('Plot for y14 = xsinz');

subplot(5,2,6);
plot(y15);
title('Plot for y15 = sin(x + z)');

subplot(5,2,7);
plot(y16);
title('Plot for y16 = zsin50x');

subplot(5,2,8);
plot(y17);
```

```
title('Plot for y17 = sin(x + 50z)');

subplot(5,2,9);
plot(y18);
title('Plot for y18 = sinx/z');

subplot(5,2,10);
plot(y19);
title('Plot for y19 = y11 + y12 + y13 + y14 + y15 + y16 + y17 + y18');
```

Problem 4:



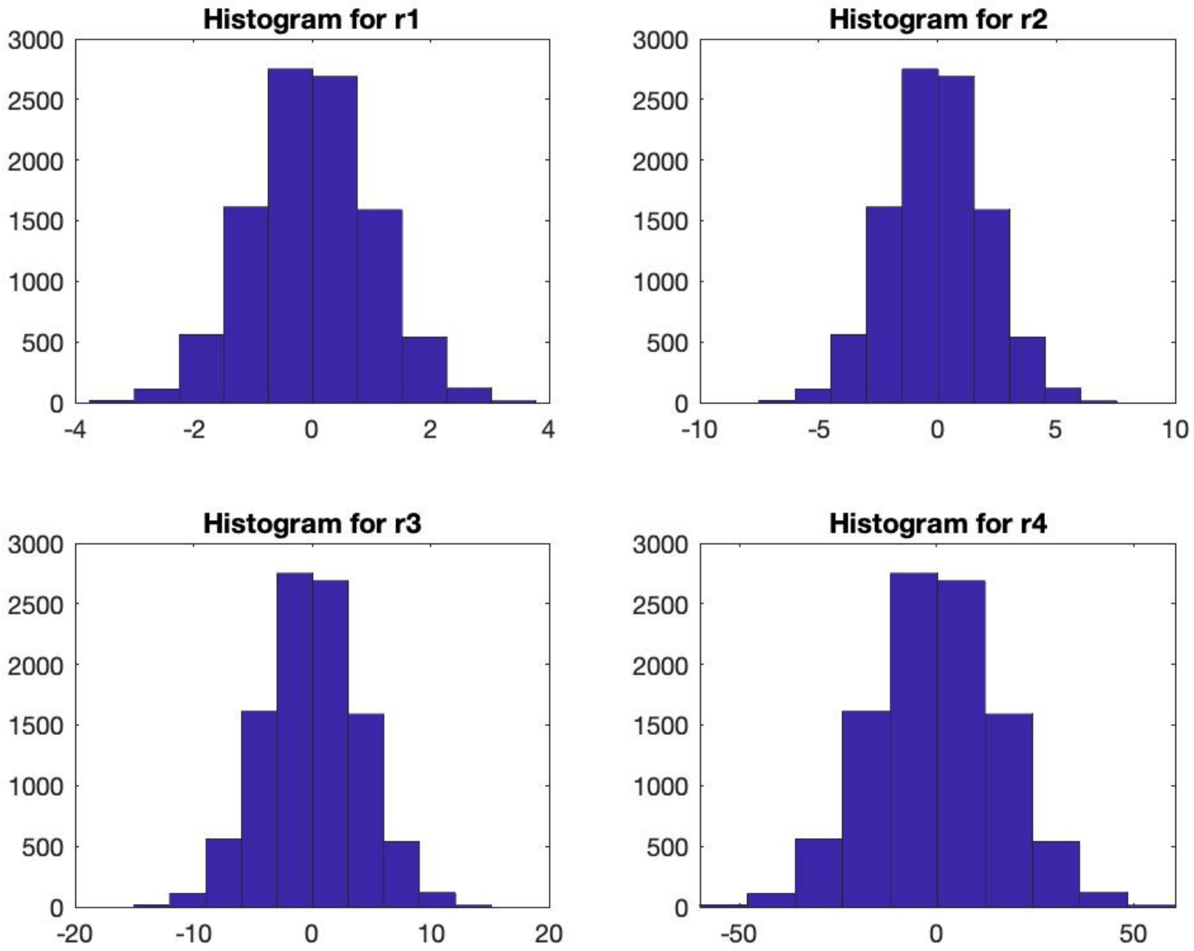
Between problem 3 and 4 we understand the difference between uniform distribution and the normal distribution clearly. In uniform distribution, the values of the random variable is more scattered but in normal distribution, centralizes on the middle.

Code:

```
%Problem 4

z = rand(1,41);

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;

figure(4);
subplot(5,2,1);
plot(y20);
title('Plot for y20 = z');

subplot(5,2,2);
plot(y21);
title('Plot for y21 = z + x');

subplot(5,2,3);
plot(y22);
title('Plot for y22 = z + sinx');

subplot(5,2,4);
plot(y23);
title('Plot for y23 = zsinx');

subplot(5,2,5);
```

```
plot(y24);
title('Plot for y24 = xsinz');

subplot(5,2,6);
plot(y25);
title('Plot for y25 = sin(x + z)');

subplot(5,2,7);
plot(y26);
title('Plot for y26 = zsin50x');

subplot(5,2,8);
plot(y27);
title('Plot for y27 = sin(x + 50z)');

subplot(5,2,9);
plot(y28);
title('Plot for y28 = sinx/z');

subplot(5,2,10);
plot(y29);
title('Plot for y29 = y21 + y22 + y23 + y24 + y25 + y26 + y27 + y28');
```

Problem 5:



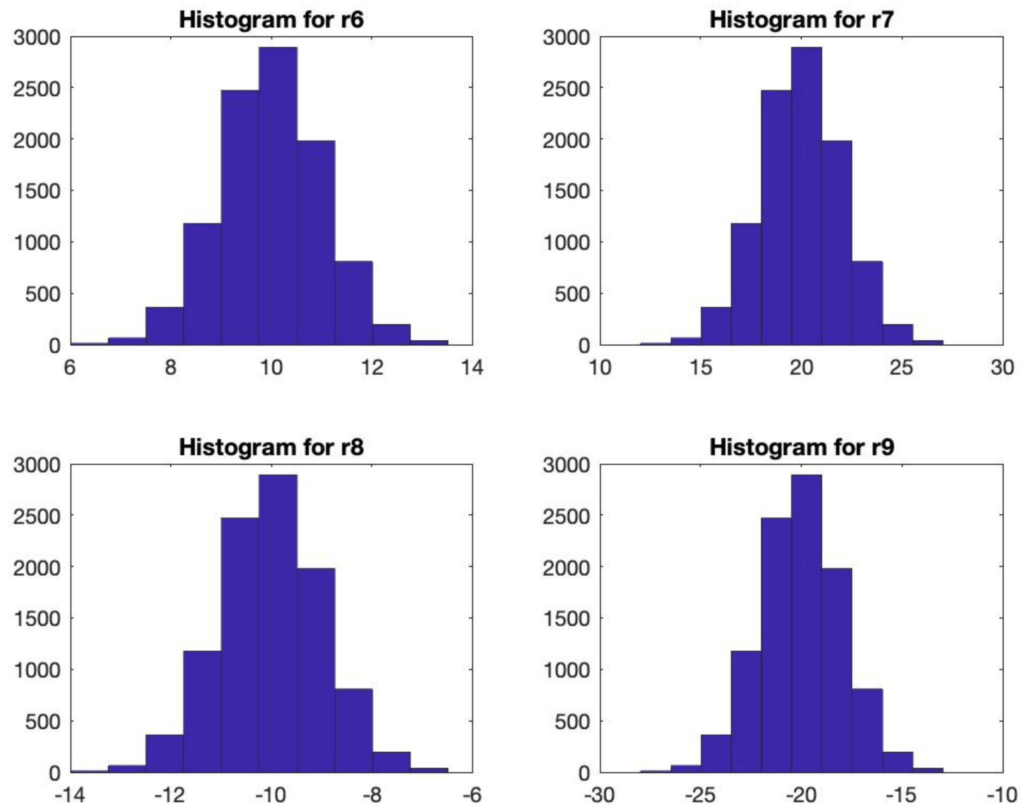
When we increase variance, the values vary more and the histogram becomes wider on the sides, it becomes more balanced through the middle.

Code:

```
%Problem 5
z = randn(1,10000);
r1 = z;
r2 = 2.*z;
r3 = 4.*z;
r4 = 16.*z;

figure(5);
```

```
subplot(2,2,1);
hist(r1);
title('Histogram for r1');

subplot(2,2,2);
hist(r2);
title('Histogram for r2');

subplot(2,2,3);
hist(r3);
title('Histogram for r3');

subplot(2,2,4);
hist(r4);
title('Histogram for r4');
```

Problem 6:



When we change the mean, our histogram shifts towards left or right. If we increase the mean, it shifts right and if we decrease our mean, it shifts left.

Also a general formula that I have used on these two questions is:

$y = ax + b$, x being the distribution that we have on hand.

$\mu_y = a\mu_x + b.$

$\sigma^2{}_y = a^2\sigma^2{}_x.$

Thus a becomes $\sigma_{x\,/}\ \sigma_{y\,.}$

And b becomes $(\mu_y - \sigma_{x\,/}\ \sigma_{y*}\mu_x).$

In normal distribution, μ_x*=0,* $\sigma^2{}_x = 1.$

In uniform distribution μ_x*=1/2,* $\sigma^2{}_x = 1/12.$

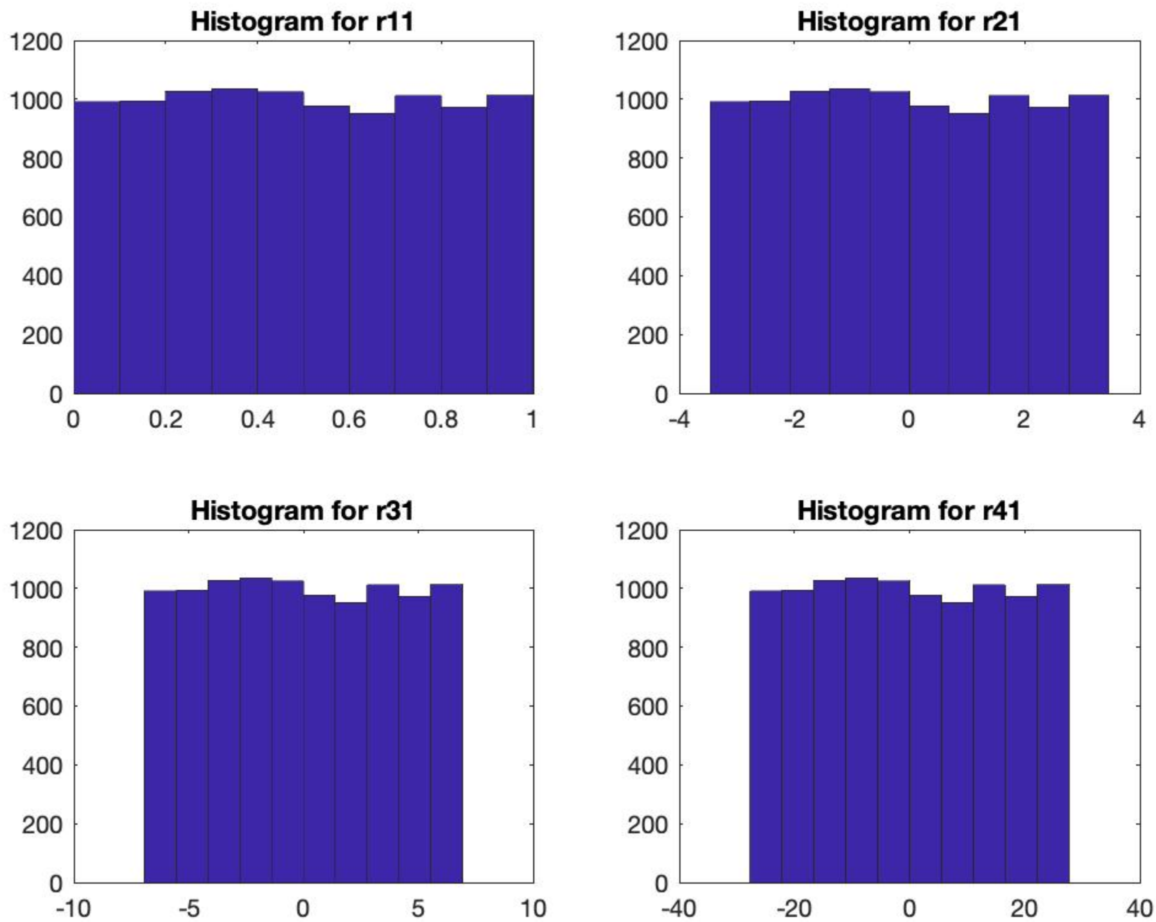
I used these two equations in problems 5,6,7,8.

Code:

```
%Problem 6

z = randn(1,10000);
r6 = z + 10;
r7 = 2.*z + 20;
r8 = z - 10;
r9 = 2.*z - 20;

figure(6);
subplot(2,2,1);
hist(r6);
title('Histogram for r6');

subplot(2,2,2);
hist(r7);
title('Histogram for r7');

subplot(2,2,3);
hist(r8);
```

```
title('Histogram for r8');

subplot(2,2,4);
hist(r9);
title('Histogram for r9’);
```

Problem 7:

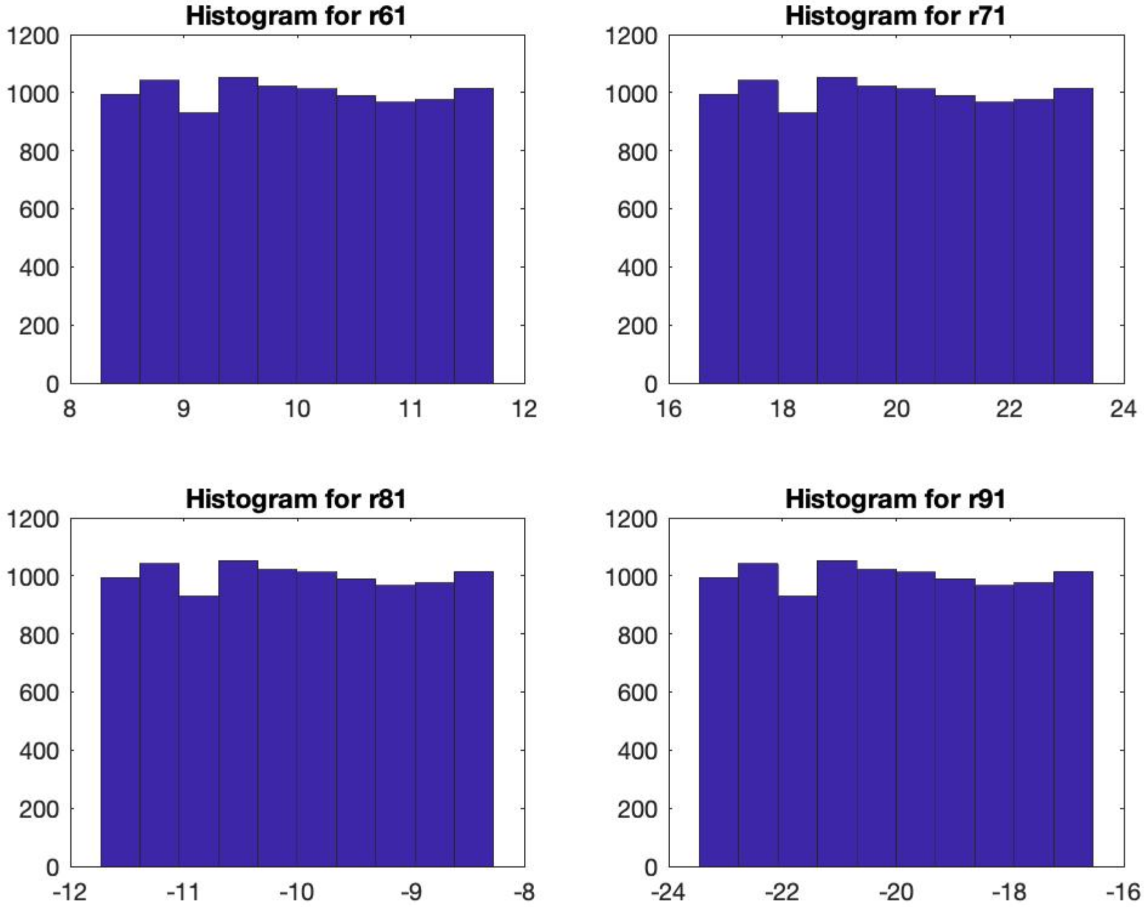


In uniform distribution, our histogram is more flatter and more homogenous. If we increase our variance, the values vary more for example in r41 the values differ between -30 to 30, in r11 the values differ between 0 to 1.

Code:

```
%Problem 7

z = rand(1,10000);
r11 = z;
r21 = sqrt(12*4).*z - sqrt(3)*2;
r31 = sqrt(12*16).*z - sqrt(3)*4;
r41 = sqrt(12*256).*z - sqrt(3)*16;

figure(7);
subplot(2,2,1);
hist(r11);
title('Histogram for r11');

subplot(2,2,2);
hist(r21);
title('Histogram for r21');

subplot(2,2,3);
hist(r31);
title('Histogram for r31');

subplot(2,2,4);
hist(r41);
title('Histogram for r41');
```

Problem 8:



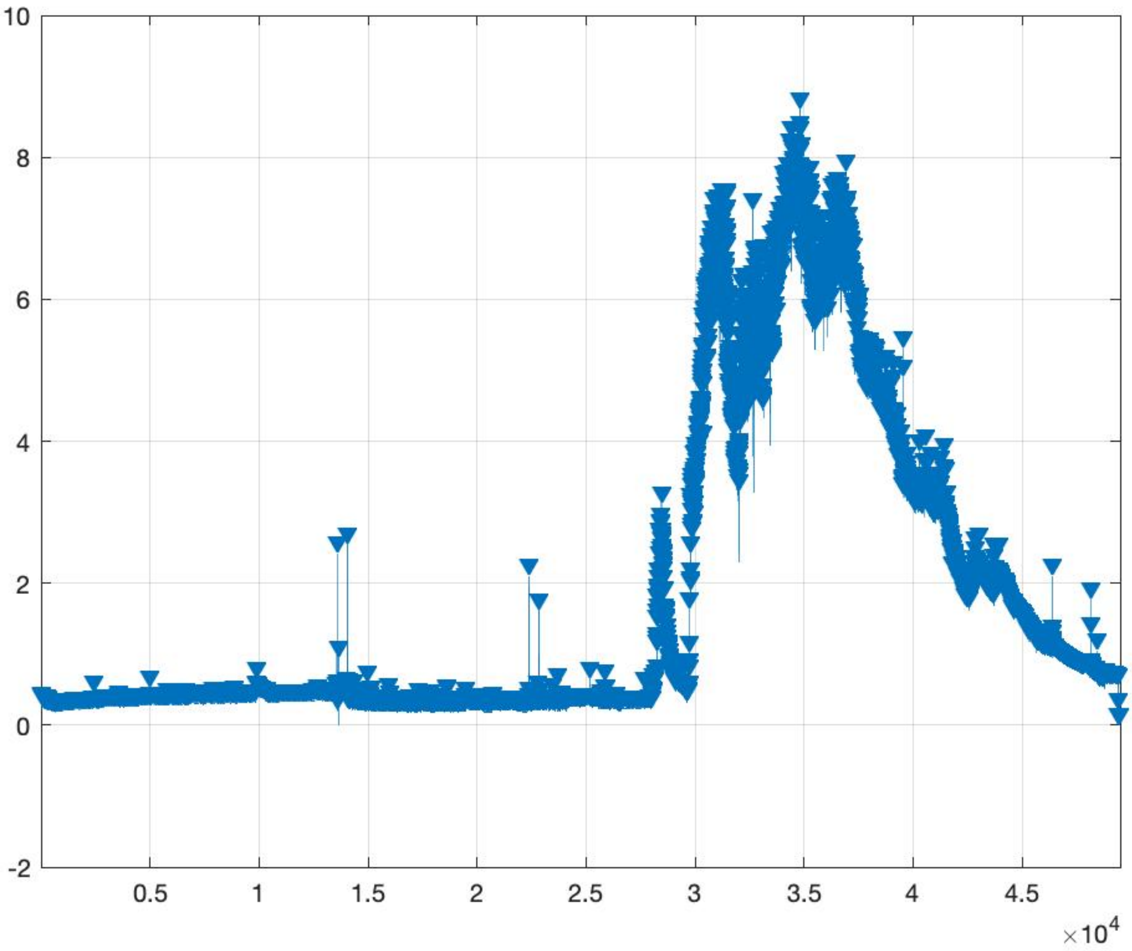
When we change the mean, our histogram shifts towards left or right. If we increase the mean, it shifts right and if we decrease our mean, it shifts left. Also if we increase the variance, our variables vary more.

Code:

```
%Problem 8

z = rand(1,10000);
r61 = sqrt(12).*z + 10 - sqrt(3);
r71 = sqrt(12*4).*z + 20 - sqrt(3)*2;
r81 = sqrt(12).*z -10 - sqrt(3);
r91 = sqrt(12*4).*z -20 - sqrt(3)*2;
```

```
figure(8);
subplot(2,2,1);
hist(r61);
title('Histogram for r61');

subplot(2,2,2);
hist(r71);
title('Histogram for r71');

subplot(2,2,3);
hist(r81);
title('Histogram for r81');

subplot(2,2,4);
hist(r91);
title('Histogram for r91');
```

Problem 9:



In this problem, find peaks method return almost all of the local maxima's since I didn't give any arguments to it such as minimum peak distance etc. But because of 'If a peak is flat, the function returns only the point with the lowest index.' sentence in its definition , it doesn't return all of the flat points of the function even though they count as a peak.

Code:

```
%Problem 9
m = csvread('exampleSignal.csv',3,0);
figure;
plot(m);
figure;
findpeaks(m);
```

Problem 10 Code:

```
%Problem 10
m = imread('lena.png');
g = rgb2gray(m);
disp(strcat('Mean of the picture:' , num2str(mean(g,'all'))));
disp(strcat('Variance of the picture:', num2str(std2(g))));
maxelement = max(g,[],'all');
[maxX,maxY] = find(g == maxelement);
disp(strcat('Max element of the picture:',num2str(maxelement)));
disp(strcat('Location of the max element(row, column):', num2str(maxX) , ',' ,
num2str(maxY)));
minelement = min(g,[],'all');
[minX, minY] = find(g == minelement);
disp(strcat('Min element of the picture:',num2str(minelement)));
disp(strcat('Location of the min element(row, column): ' , num2str(minX) , ',' ,
num2str(minY)));
```

Matlab is very different from any other language that I've used. Maybe it is because of the IDE, I find it very hard to code in Matlab. Also functions are more ambiguous in Matlab since they have a lot of signatures. I have never plotted data before, creating figures in Matlab were very unfamiliar to me. But working with vectors, matrices are very easy and efficient with operations like '.*', './'. Generating random data is also very easy in Matlab. Since I am used to declaring data, the syntax of the Matlab felt strange to me. But overall Matlab is very data-friendly programming language. It handles large datas very efficiently and is very easy to use.