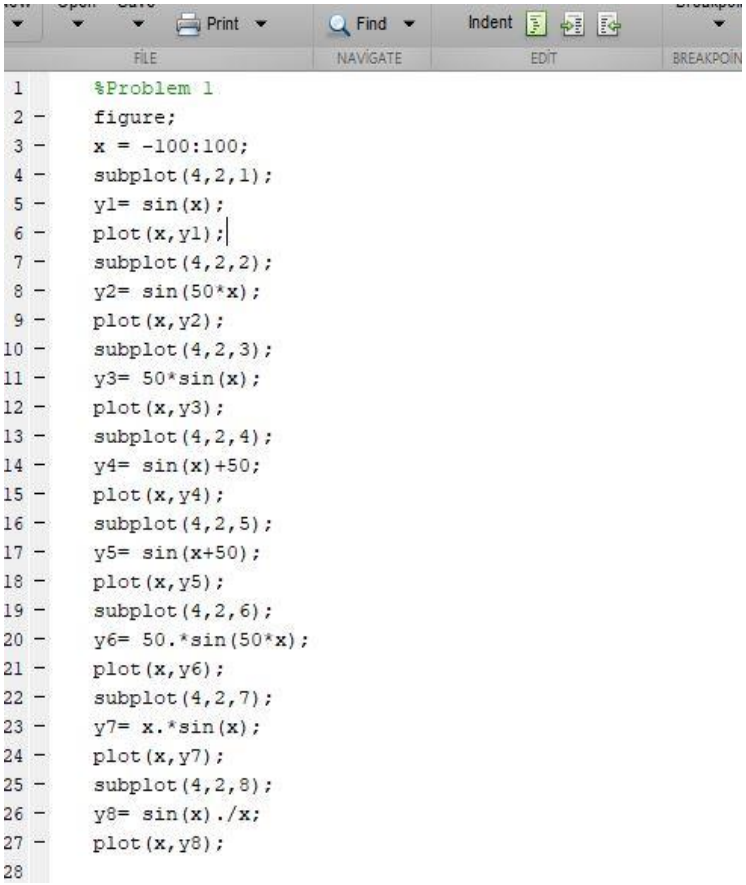
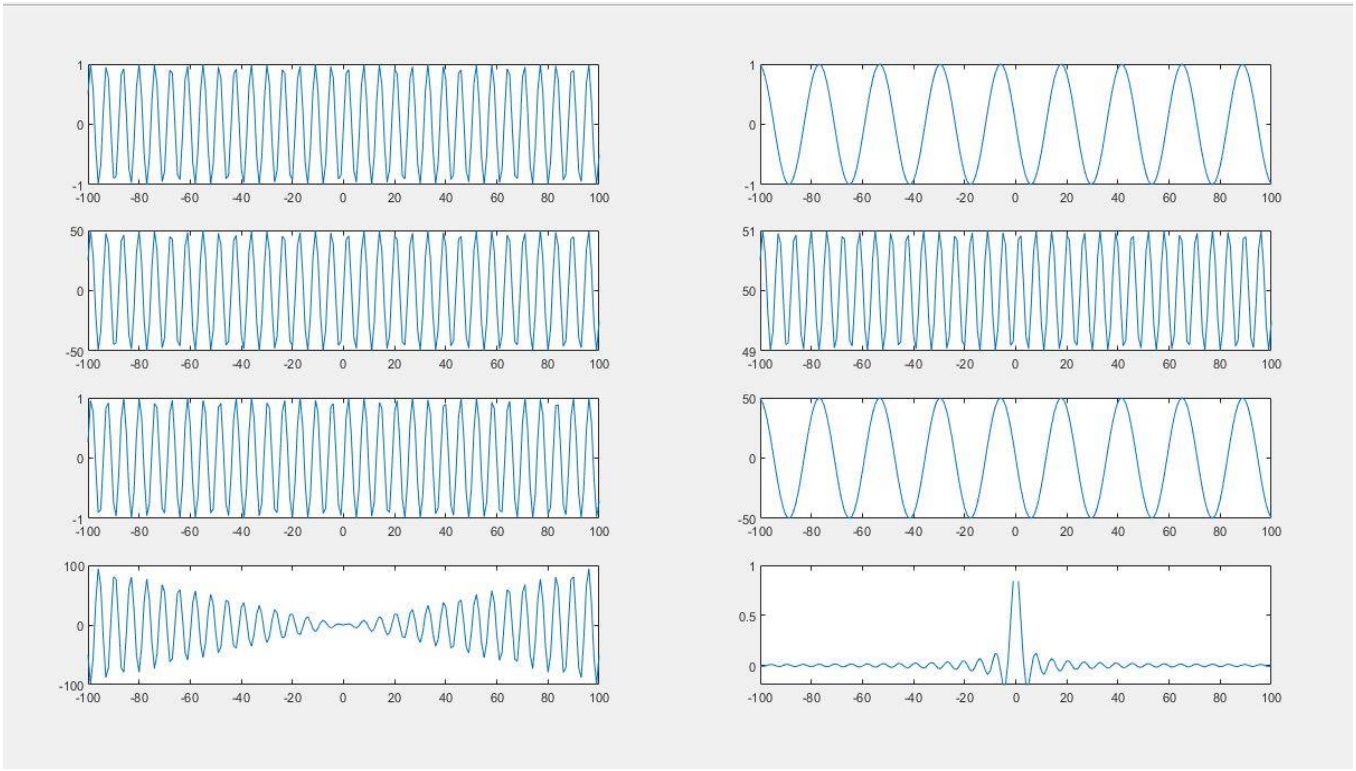
CMPE362-SIGNAL PROCESSING
PROJECT 1
REPORT

İREM UĞUZ
2015400165

Problem 1




```
1     %Problem 1
2 -   figure;
3 -   x = -100:100;
4 -   subplot(4,2,1);
5 -   y1= sin(x);
6 -   plot(x,y1);
7 -   subplot(4,2,2);
8 -   y2= sin(50*x);
9 -   plot(x,y2);
10 -  subplot(4,2,3);
11 -  y3= 50*sin(x);
12 -  plot(x,y3);
13 -  subplot(4,2,4);
14 -  y4= sin(x)+50;
15 -  plot(x,y4);
16 -  subplot(4,2,5);
17 -  y5= sin(x+50);
18 -  plot(x,y5);
19 -  subplot(4,2,6);
20 -  y6= 50.*sin(50*x);
21 -  plot(x,y6);
22 -  subplot(4,2,7);
23 -  y7= x.*sin(x);
24 -  plot(x,y7);
25 -  subplot(4,2,8);
26 -  y8= sin(x)./x;
27 -  plot(x,y8);
28
```

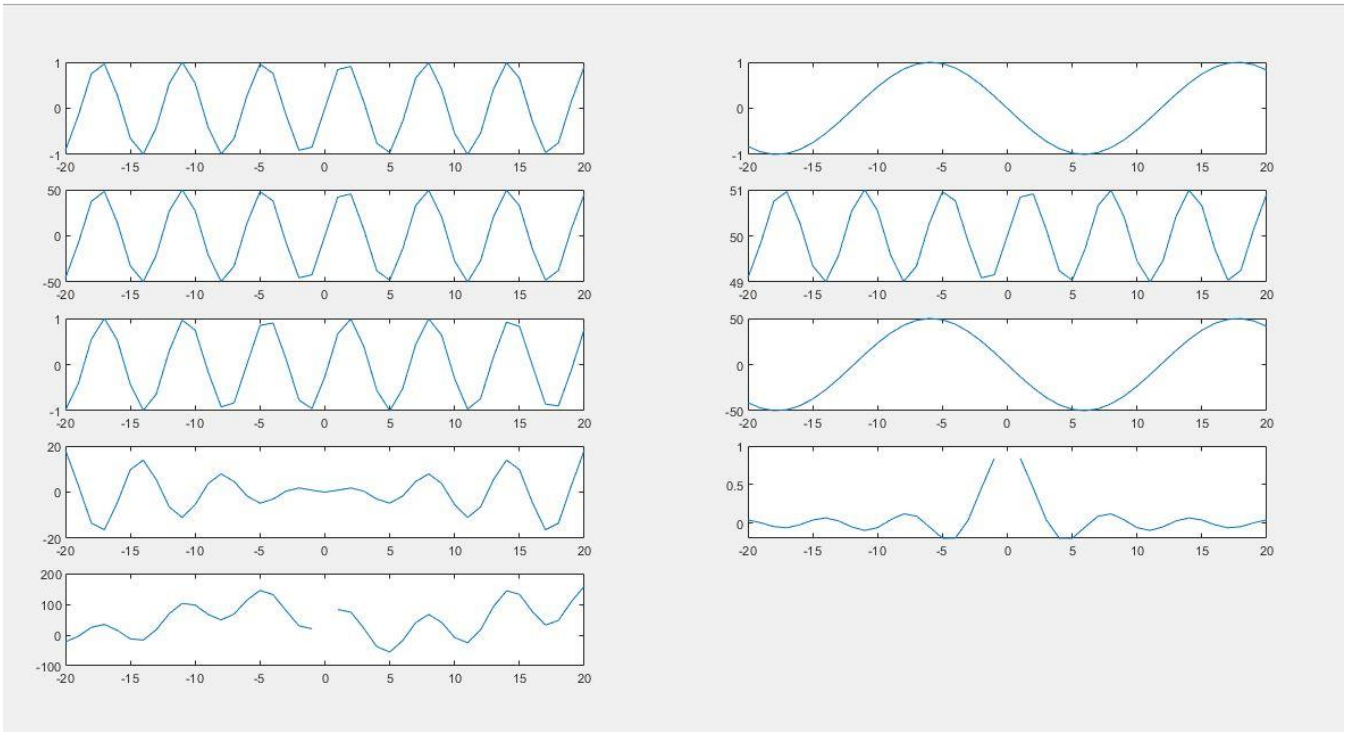


From this question I learned how to do elementary multiplication and division. I didn't know that I needed to use . operator so till I learned that it was a little difficult for me. Also when I saw the plots, I learned how any arithmetic operation affects the sinusoid signal. X*sin(x) and sin(x)/x's plots were especially interesting to see.

Problem 2

```
FILE  NAVIGATE  EDIT  BREAKPOINTS

%Problem 2
figure;
x = -20:20;
subplot(5,2,1);
y1= sin(x);
plot(x,y1);
subplot(5,2,2);
y2= sin(50*x);
plot(x,y2);
subplot(5,2,3);
y3= 50*sin(x);
plot(x,y3);
subplot(5,2,4);
y4= sin(x)+50;
plot(x,y4);
subplot(5,2,5);
y5= sin(x+50);
plot(x,y5);
subplot(5,2,6);
y6= 50.*sin(50*x);
plot(x,y6);
subplot(5,2,7);
y7= x.*sin(x);
plot(x,y7);
subplot(5,2,8);
y8= sin(x)./x;
plot(x,y8);
subplot(5,2,9);
y9= y1+y2+y3+y4+y5+y6+y7+y8;
plot(x,y9);
```

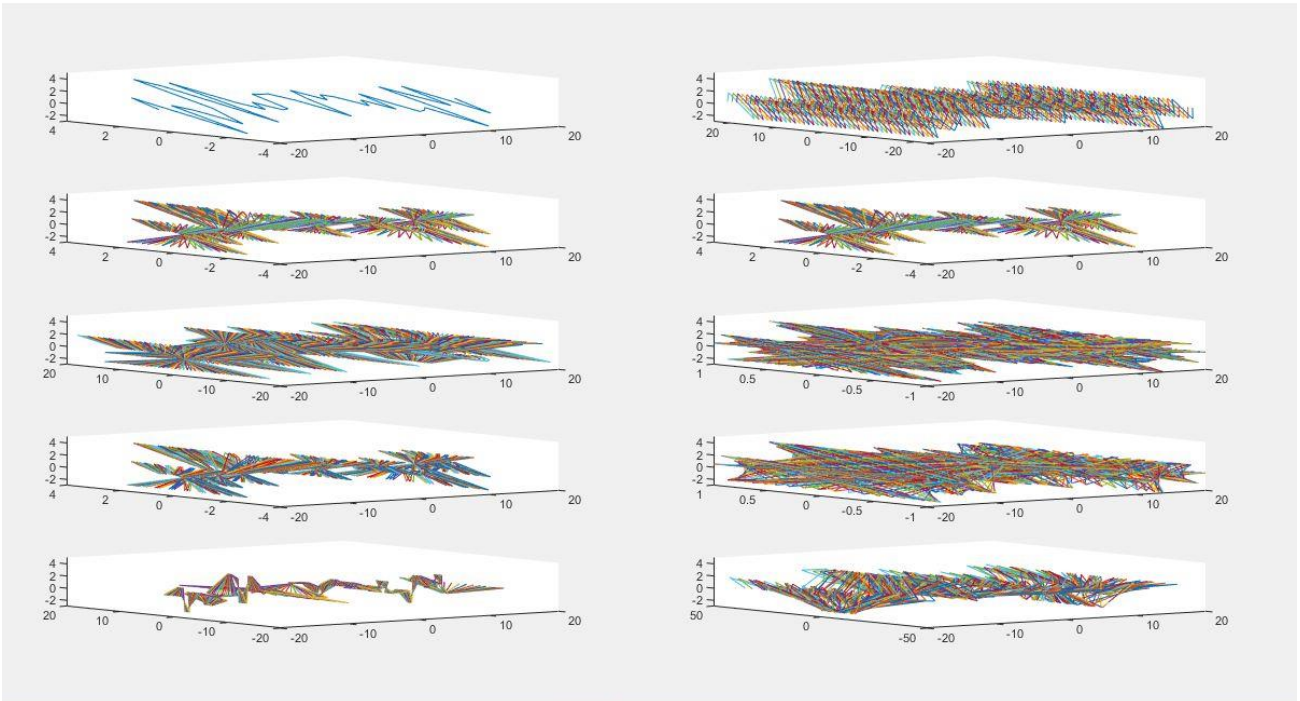


This question was very similar to the previous question. The difference in this question is that the plots here is drawn in a more limited x line, which makes the plots look less

smooth. Because there is less x points here the plots look not circular but as a addition of linear lines.

Problem 3

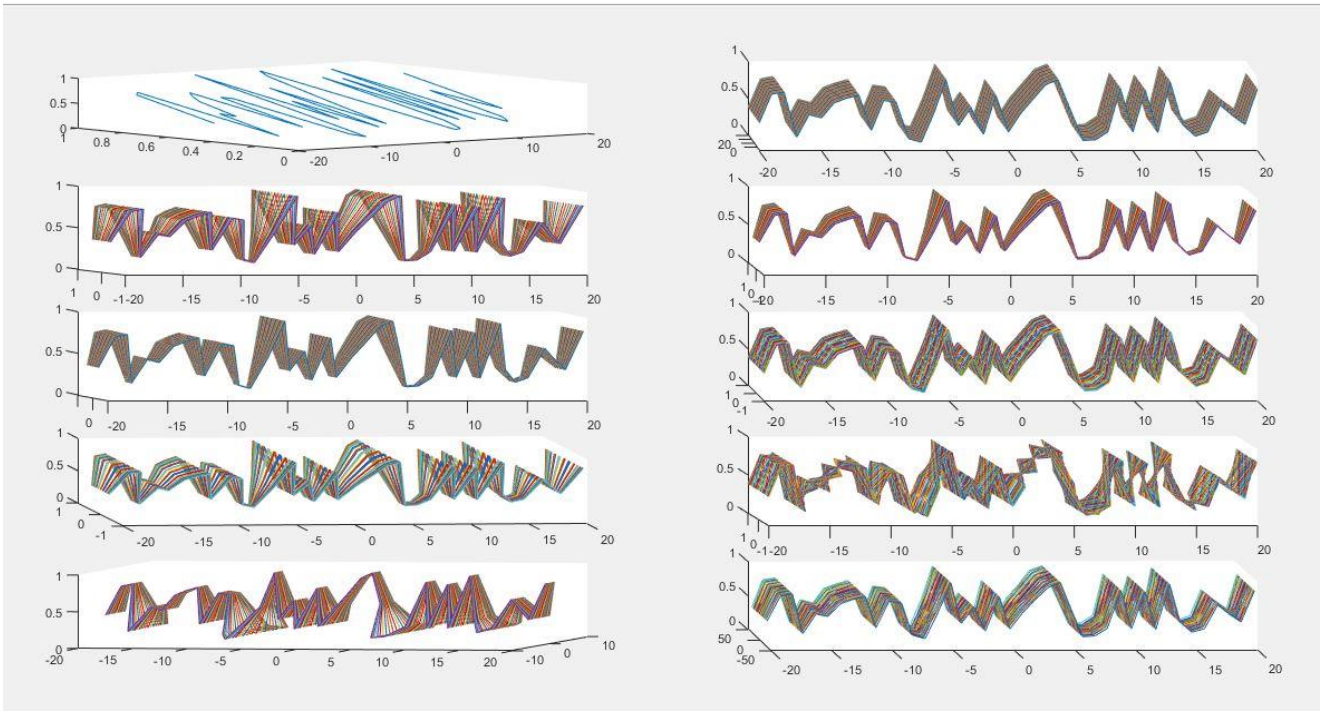
```
5
6       %Problem 3
7 -     figure;
8 -     z=randn(41,1);
9 -     subplot(5,2,1);
0 -     yl0= z;
1 -     plot3(x,yl0,z);
2 -     subplot(5,2,2);
3 -     yl1= x+z;
4 -     plot3(x,yl1,z);
5 -     subplot(5,2,3);
6 -     yl2= z*sin(x);
7 -     plot3(x,yl2,z);
8 -     subplot(5,2,4);
9 -     yl3= z.*sin(x);
0 -     plot3(x,yl3,z);
1 -     subplot(5,2,5);
2 -     yl4= x.*sin(z);
3 -     plot3(x,yl4,z);
4 -     subplot(5,2,6);
5 -     yl5= sin(x+z);
6 -     plot3(x,yl5,z);
7 -     subplot(5,2,7);
8 -     yl6= z.*sin(50*x);
9 -     plot3(x,yl6,z);
0 -     subplot(5,2,8);
1 -     yl7= sin(x+50*z);
2 -     plot3(x,yl7,z);
3 -     subplot(5,2,9);
4 -     yl8= sin(x)./z;
5 -     plot3(x,yl8,z);
6 -     subplot(5,2,10);
7 -     yl9= yl1+yl2+yl3+yl4+yl5+yl6+yl7+yl8;
8 -     plot3(x,yl9,z);
9
```



Here in this question, aside from x and y values, we have an extra variable which is z. z is a unit random variable with a normal distribution, that has mean 0 and it can be created with randn function of MATLAB. Plotting this question was a little tricky as it has three variables. I found the plot3 function of MATLAB, that helps to draw 3d plots. The plot is a little hard to understand as we had three variables to consider.

Problem 4

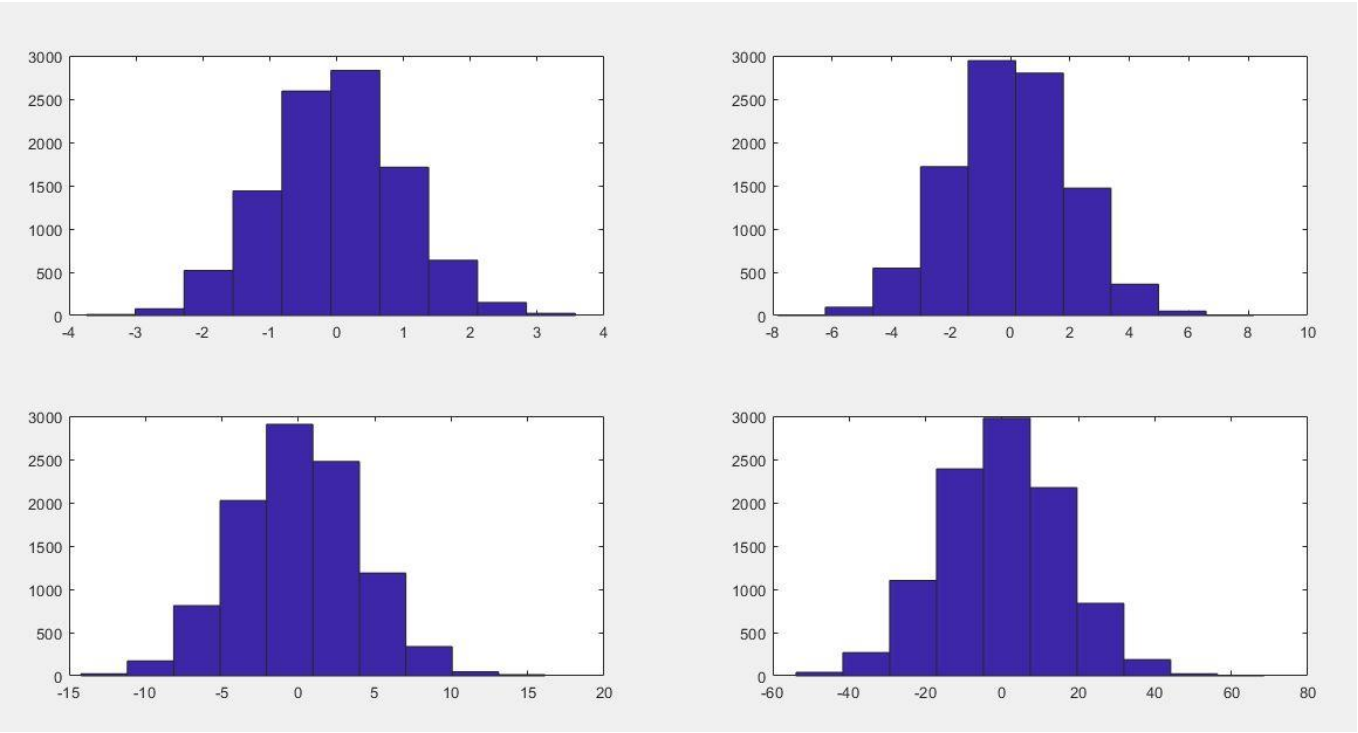
```
%Problem 4
figure;
z=rand(41,1);
subplot(5,2,1);
y20= z;
plot3(x,y20,z);
subplot(5,2,2);
y21= x+z;
plot3(x,y21,z);
subplot(5,2,3);
y22= z*sin(x);
plot3(x,y22,z);
subplot(5,2,4);
y23= z.*sin(x);
plot3(x,y23,z);
subplot(5,2,5);
y24= x.*sin(z);
plot3(x,y24,z);
subplot(5,2,6);
y25= sin(x+z);
plot3(x,y25,z);
subplot(5,2,7);
y26= z.*sin(50*x);
plot3(x,y26,z);
subplot(5,2,8);
y27= sin(x+50*z);
plot3(x,y27,z);
subplot(5,2,9);
y28= sin(x)./z;
plot3(x,y28,z);
subplot(5,2,10);
y29= y21+y22+y23+y24+y25+y26+y27+y28;
plot3(x,y29,z);
```



This question had the similar calculation as the previous questions. But this time we add a uniformly distributed variable z which can be created with rand function to the plot. The smooth sinusoidal signals become a little distorted after addition of z. The question was a good opportunity to analyze the change of the wave.

Problem 5

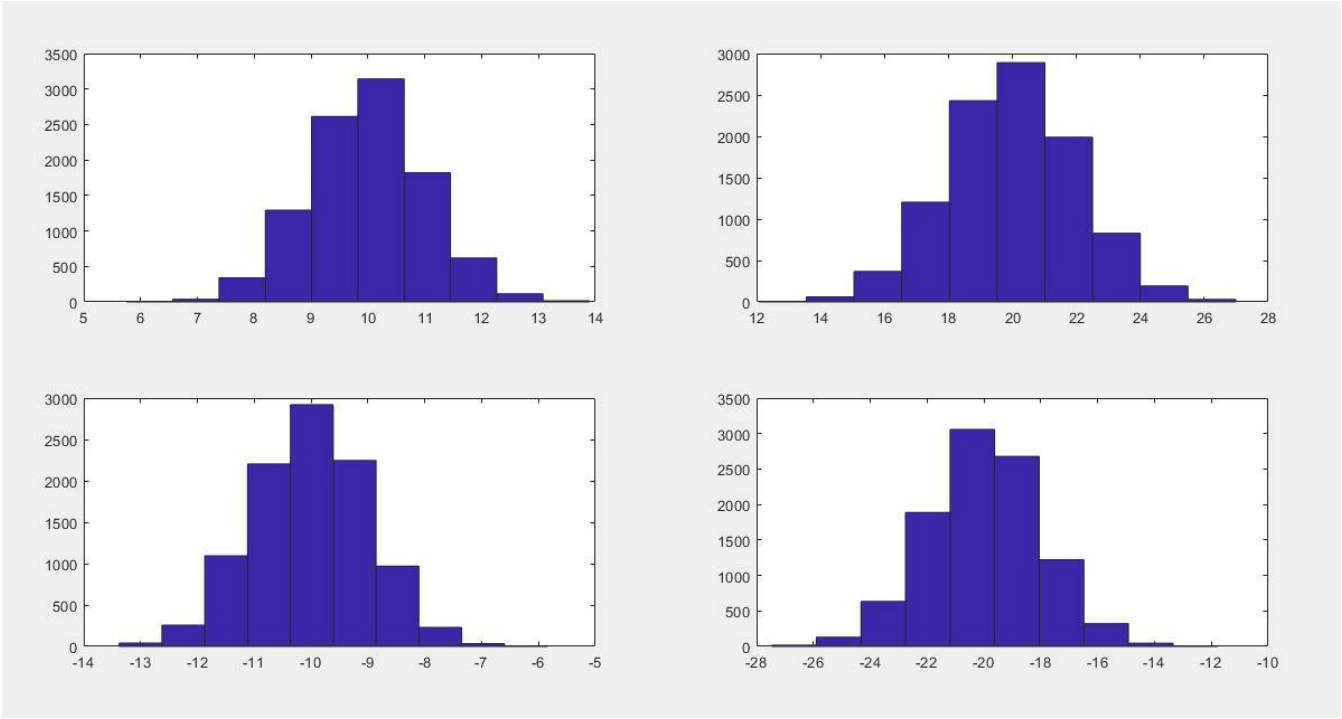
```
63
64      %Problem 5
65 -    figure;
66 -    r1 = 1.*randn(10000,1) + 0;
67 -    r2 = 2.*randn(10000,1) + 0;
68 -    r3 = 4.*randn(10000,1) + 0;
69 -    r4 = 16.*randn(10000,1) + 0;
70 -    subplot(2,2,1);
71 -    hist(r1);
72 -    subplot(2,2,2);
73 -    hist(r2);
74 -    subplot(2,2,3);
75 -    hist(r3);
76 -    subplot(2,2,4);
77 -    hist(r4);
```



In this question, I learned how to create random variables with certain standard deviation with using the randn function which gives a Gaussian distributed variable with mean 0. Also I learned how to use hist function. But in the MathWorks web site, it said that we shouldn't use hist but use a better function that is named histogram. I used hist anyway because it was stated in the project description.

Problem 6

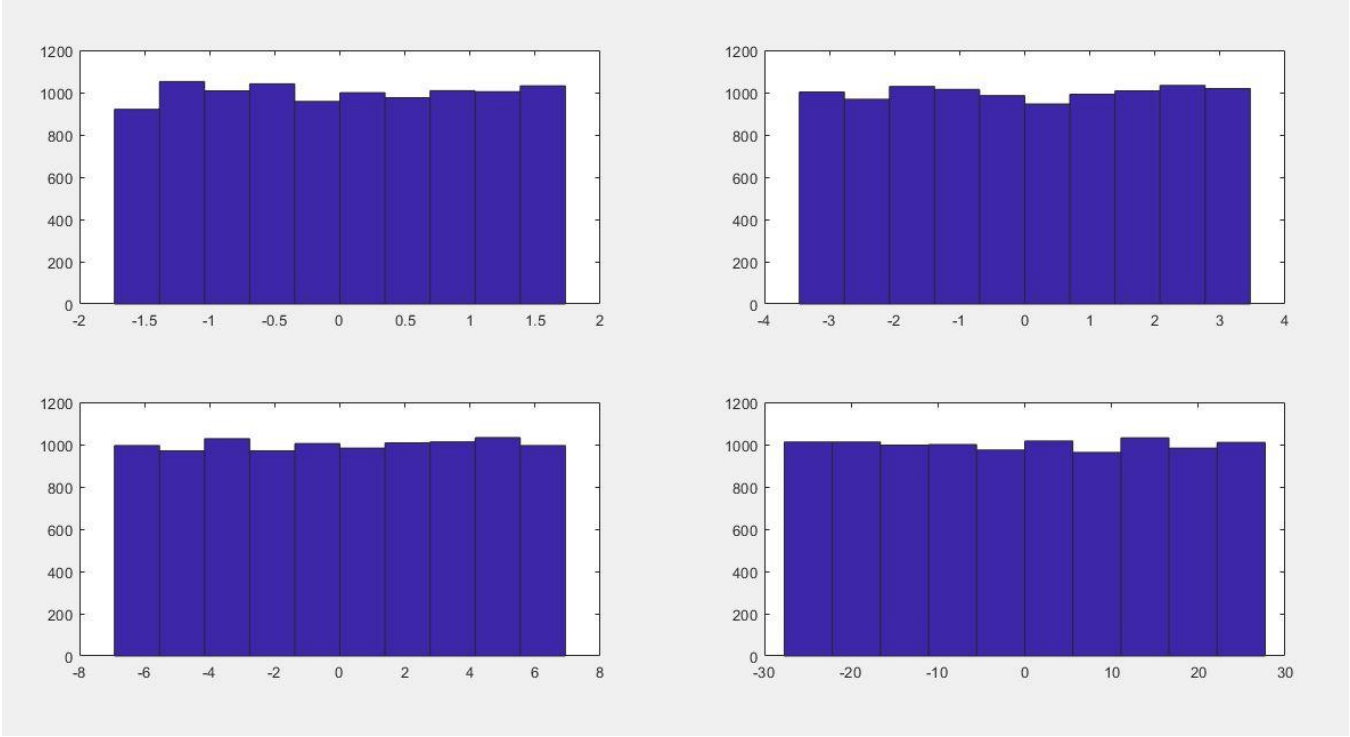
```
179      %Problem 6
180 -    figure;
181 -    r6 = 1.*randn(10000,1) + 10;
182 -    r7 = 2.*randn(10000,1) + 20;
183 -    r8 = 1.*randn(10000,1) - 10;
184 -    r9 = 2.*randn(10000,1) - 20;
185 -    subplot(2,2,1);
186 -    hist(r6);
187 -    subplot(2,2,2);
188 -    hist(r7);
189 -    subplot(2,2,3);
190 -    hist(r8);
191 -    subplot(2,2,4);
192 -    hist(r9);
193
```



In this question, contrary to ne previous question, we are expected to create random variables with certain mean and standard deviation using the Gaussian random variable. In the problem 5, I learned how to manipulate the standard deviation. In this question, it was easier to manipulate the mean.

Problem 7

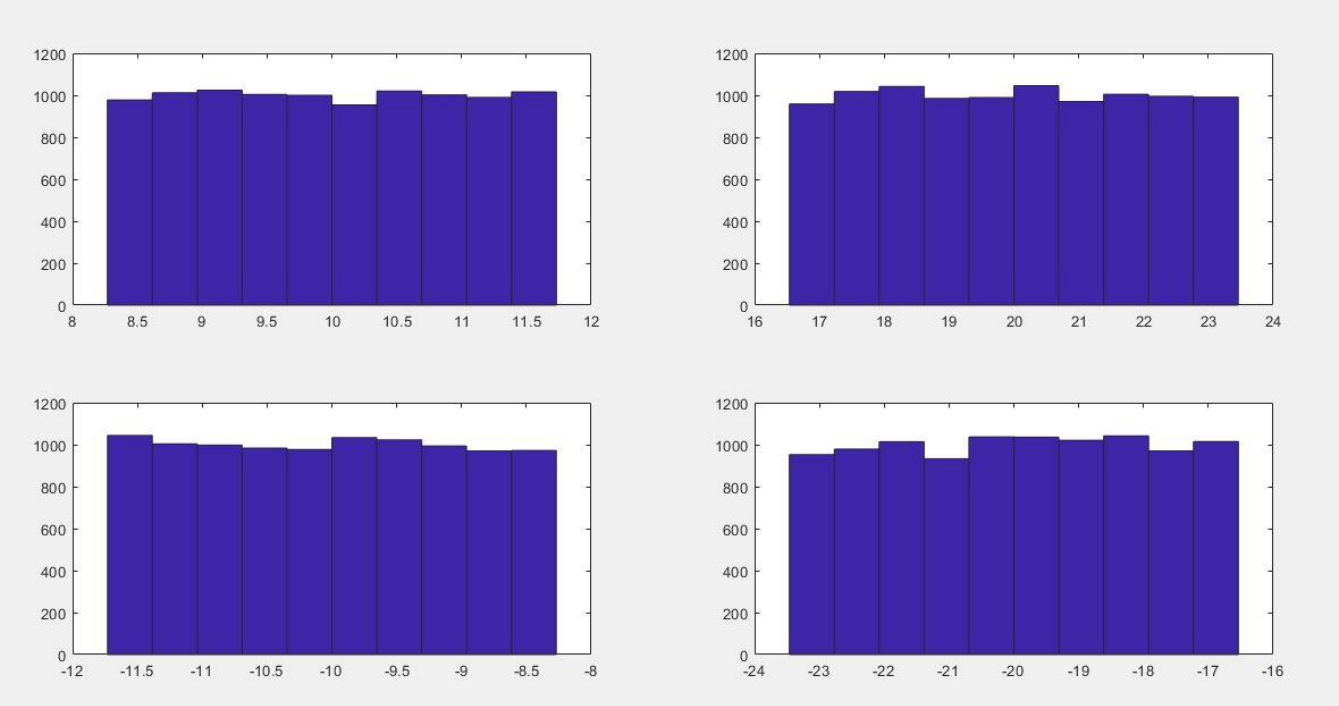
```
193
194      %Problem 7
195 -    figure;
196 -    r11 = 1.*((rand(10000,1)-0.5).*sqrt(12)) + 0;
197 -    r21 = 2.*((rand(10000,1)-0.5).*sqrt(12)) + 0;
198 -    r31 = 4.*((rand(10000,1)-0.5).*sqrt(12)) +0 ;
199 -    r41 = 16.*((rand(10000,1)-0.5).*sqrt(12)) + 0;
200 -    var(r11);
201 -    subplot(2,2,1);
202 -    hist(r11);
203 -    subplot(2,2,2);
204 -    hist(r21);
205 -    subplot(2,2,3);
206 -    hist(r31);
207 -    subplot(2,2,4);
208 -    hist(r41);
209
```



This question and the Problem 8 was the questions that I had the most hardship. At first, I tried to use the same tactics from the previous question but the result was not correct. After doing some research, I learned to manipulate the uniformly distributed variable that is the output of the rand function the have the mean 0 and the variance 1, like the output of the randn function. But since the standard deviation is not 1/12 but something closer to that number, it wasn't the exact 1 as standard deviation but some number that is closer to that.

Problem 8

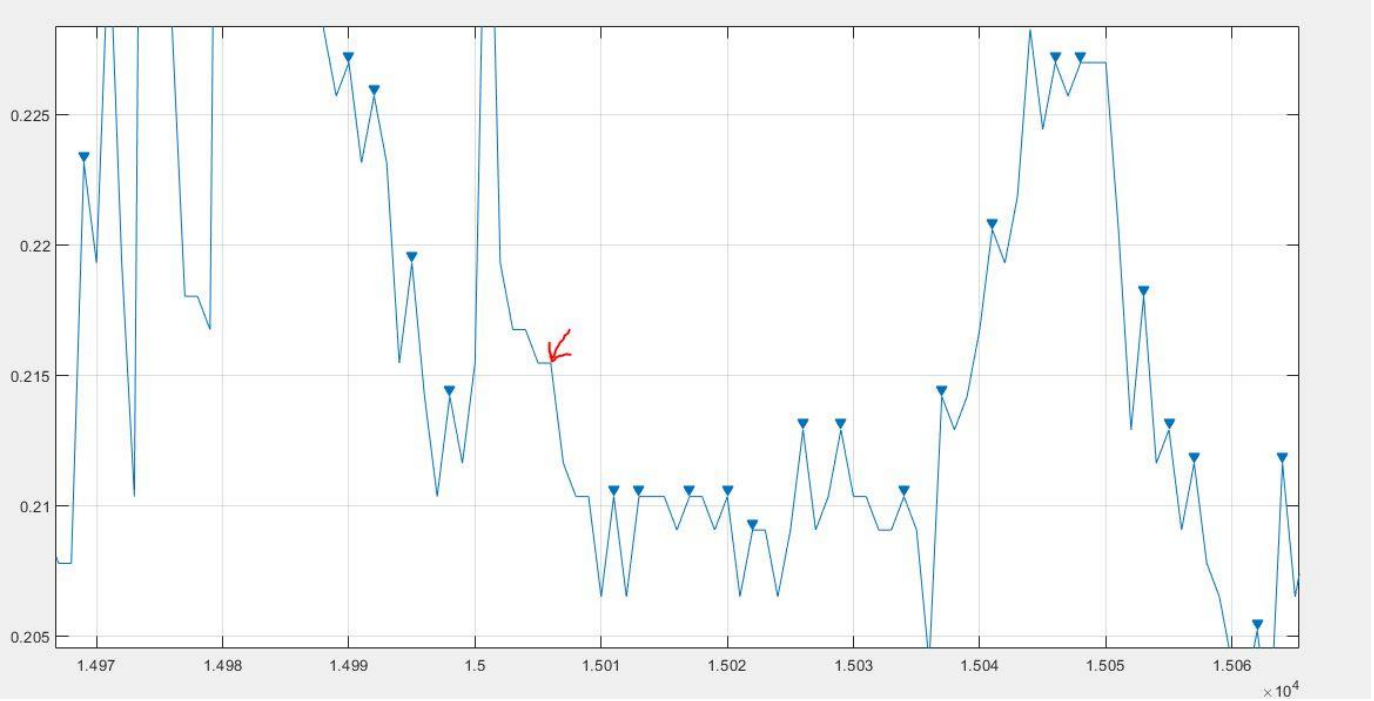
```
%Problem 8
figure;
r61 = 1.*((rand(10000,1)-0.5).*sqrt(12)) + 10;
r71 = 2.*((rand(10000,1)-0.5).*sqrt(12)) + 20;
r81 = 1.*((rand(10000,1)-0.5).*sqrt(12)) - 10;
r91 = 2.*((rand(10000,1)-0.5).*sqrt(12)) - 20;
subplot(2,2,1);
hist(r61);
subplot(2,2,2);
hist(r71);
subplot(2,2,3);
hist(r81);
subplot(2,2,4);
hist(r91);
```



This was one of the most challenging questions of the project. I salved it the same way as the previous question.

Problem 9

```
y=csvread('exampleSignal1.csv');
y=y.';
plot(y);
findpeaks(y);
```

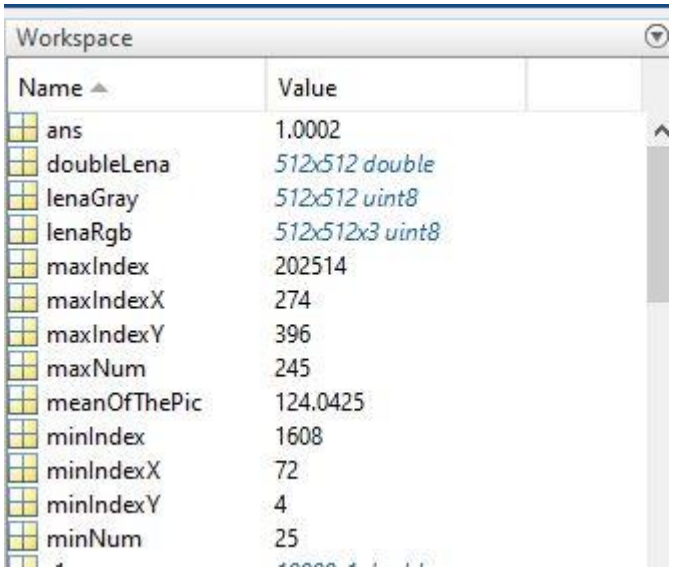


When inspected, findpeaks function seems very good at finding peaks. Most of the peaks were found by findpeaks function. One point that I think find peaks misses is the peaks that has the same value. Like the example that is marked with the red arrow above, that was a local maxima, but I think findpeaks missed it because that point has very close values next to it.

Problem 10

```
lenaRgb = imread('lena.png');
lenaGray = rgb2gray(lenaRgb);
doubleLena=double(lenaGray);
standartDeviation=sqrt(var(doubleLena,0,'all'));
meanOfThePic= mean(doubleLena,'all');
[maxNum,maxIndex]=max(doubleLena(:));
[maxIndexX,maxIndexY]=ind2sub(size(doubleLena),maxIndex);
[minNum,minIndex]=min(doubleLena(:));
[minIndexX,minIndexY]=ind2sub(size(doubleLena),minIndex);
```

Here is the results I found:



And the standart deviation I found is:47.8557. It wasn't in the picture.

Opinions About MATLAB

As a programming language, MATLAB has its pros and cons. In the aspect, ease of writing and reading, I think it is a very easy language. Something, that is interesting is that arrays start with 1, instead of 0. I didn't understand why the designers of the language made it like that. I think everything would be much nicer if the arrays were starting with 0 index. One of the aspects that I liked about MATLAB is that it doesn't have variable declarations. You can create a variable when you want to use it, and you don't have to initialize a declared variable like Java. This makes it easy to write. It looks like Python in that aspect but since it has ;'s at the end of lines, I found MATLAB more likable. Also, reading a picture and a csv file was quite easy. It is also great that there is special functions for matrix operations. They may be very handy. Also special plotting functions were really enjoyable. And I liked the MathWorks page very much. It has very nice explanations and examples about built-in functions. It is a very well documented language and I could find answers to my questions without the help pf StackOverFlow there.