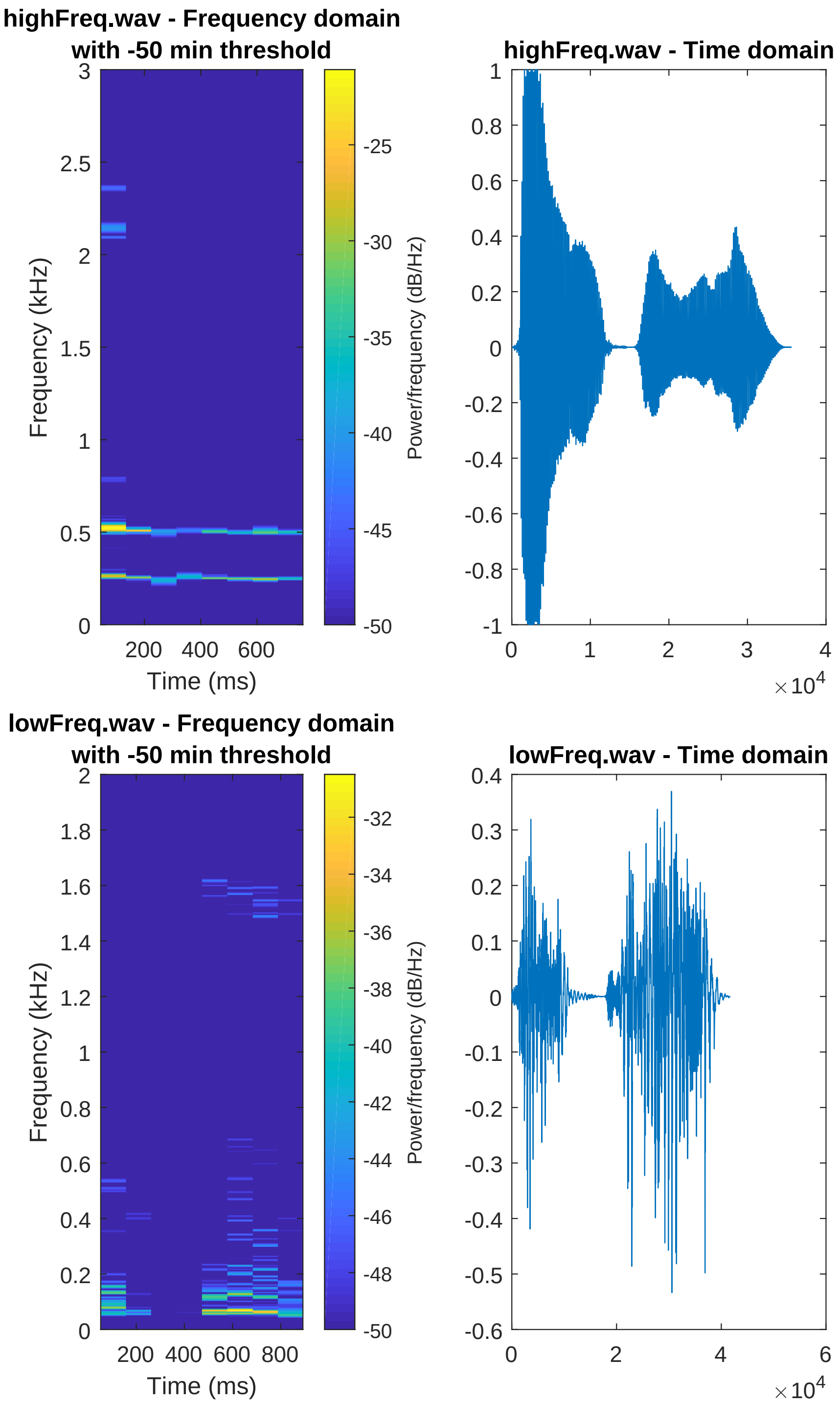
CmpE 362
2018 Spring
Homework 1 : Getting Started with MATLAB

Baran Kılıç
2014400123

March 3, 2018

1 Problem 1


highFreq.wav - Frequency domain
with -50 min threshold
Frequency (kHz)
Time (ms)
Power/frequency (dB/Hz)
highFreq.wav - Time domain
×10^4
lowFreq.wav - Frequency domain
with -50 min threshold
Frequency (kHz)
Time (ms)
Power/frequency (dB/Hz)
lowFreq.wav - Time domain
×10^4

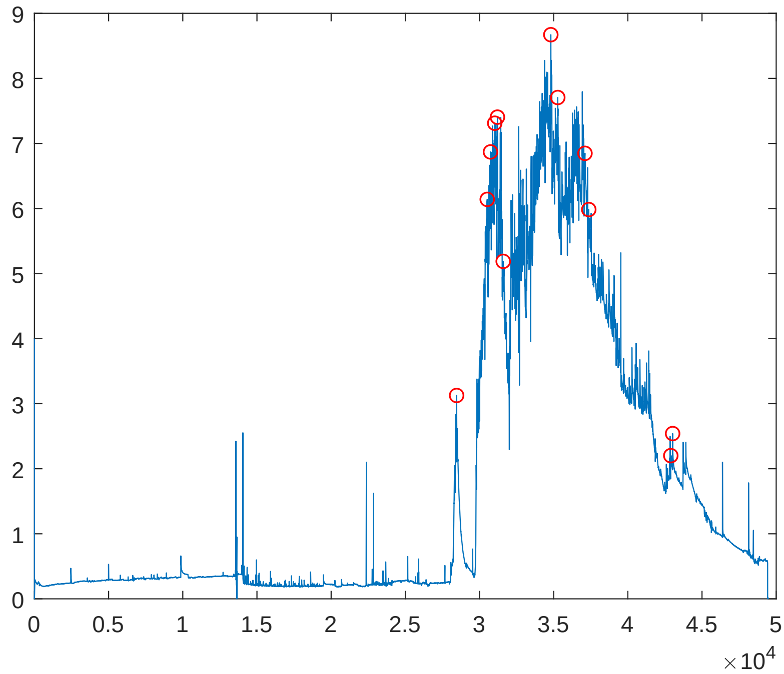
Voice spectra: 46 - 2363 Hz (output of problem1.m script)

1.1 Code

```
[highY,highfs] = audioread('highFreq.wav');
highLeft=highY(:,1); % Left channel of the sound

[lowY,lowfs] = audioread('lowFreq.wav');
lowLeft=lowY(:,1); % Left channel of the sound


figure;
subplot(1,2,1);
spectrogram(highLeft,[],[],[],highfs,'yaxis','MinThreshold',-50)

ax=gca;
ylim(ax, [0,3]); % show only between 0 and 3kHz
title({'highFreq.wav - Frequency domain','with -50 min threshold'})

subplot(1,2,2);
plot(highLeft);
title('highFreq.wav - Time domain')


figure;

subplot(1,2,1);
spectrogram(lowLeft,[],[],[],lowfs,'yaxis','MinThreshold',-50)

ax=gca;
ylim(ax, [0,2]); % show only between 0 and 2kHz
title({'lowFreq.wav - Frequency domain','with -50 min threshold'})

subplot(1,2,2);
plot(lowLeft);
title('lowFreq.wav - Time domain')

% for highFreq
threshold = 50;
[s,f,t] = spectrogram(highLeft,[],[],[],highfs,'yaxis');
ms = abs(s); % magnitude of short time fourier transform
ms(ms<threshold)=0; % make values below threshold zero
[~,column] = size(ms); % # of column of stft
freq = []; % largest freq in each column
```

```
for col = 1:column
    % index of nonzero value that has greatest index
    index = find(ms(:,col),1,'last');
    % pushes the frequency corresponding to this index to freq array
    freq(end+1) = f(index);
end
maxfreq = round(max(freq)); % the max frequency among all freq array


% for lowFreq
[s2,f2,t2] = spectrogram(lowLeft,[],[],[],lowfs,'yaxis');
ms2 = abs(s2); % magnitude of short time fourier transform
ms2(ms2<threshold)=0; % make values below threshold zero
[~,column] = size(ms2); % # of column of stft
freq = []; % smallest freq in each column of stft
for col = 1:column
    % if not all values are zero
    if ~isempty(find(ms2(:,col),1))
        % index of nonzero value that has greatest index
        index = find(ms2(:,col),1);
        % pushes the frequency corresponding to this index to freq array
        freq(end+1)= f2(index);
    end
end
minfreq = round(min(freq)); % the min frequency among all freq array

disp(['Voice spectra: ',num2str(minfreq),' - ',num2str(maxfreq),' Hz']);
```

2 Problem 2



Input method: The script prompts for folder name.

At the beginning of the signal, there are small peaks. Then, there are some larger peaks with small widths. Then, there is a big rise and fall with a lot of noise.

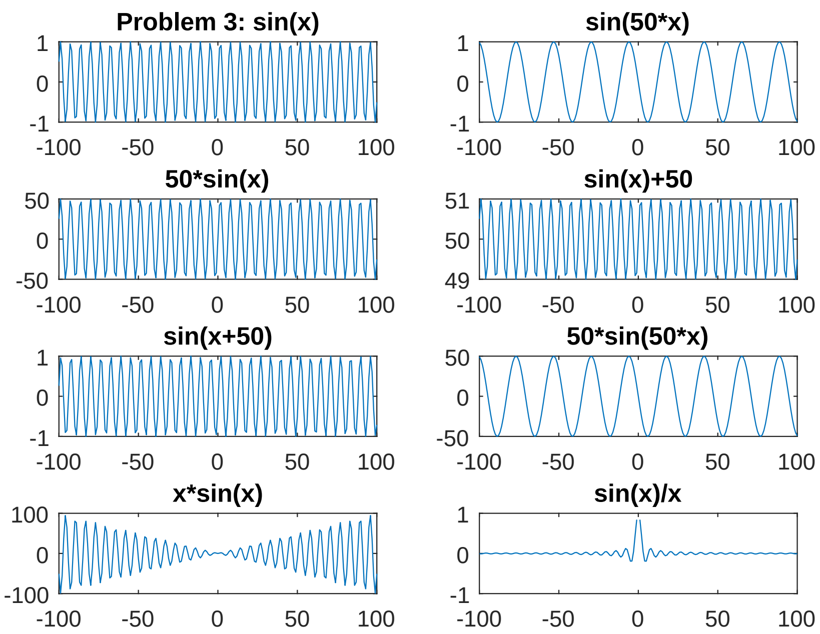
My algorithm found most of the important peaks in my opinion. Until my first peak, there are not any peaks. There are some rises but since their width is small, I think that they are noise. After my first peak, there is a rise and my algorithm counted it as 4 peaks because of the noise. After that, my algorithm found a peak (6th one) but it does not look like a peak. After that, there is again a rise. My algorithm only marked the largest one as peak in this rise. The intermediate ones are not counted as peak because I used a min peak width filter and they have small width. The three peaks that my algorithm found after that, are not very good peaks. The value actually decreases around them. There are two small bumps towards the end. My algorithm found one of them as marked it as two peaks but could not find the other one because my min peak width filter.

2.1 Code

```
prompt = 'Enter the folder name in the working directory (e.g. signal.csv): ';
foldername = input(prompt,'s');
```

```
allcvsfiles = dir( strcat(foldername,'/**/*.csv') );

for file = allcvsfiles'
    csvdata = csvread(file.name);

    figure;
    [pks,locs,w,p] = findpeaks(csvdata);
    [pks,locs] = findpeaks(csvdata,'MinPeakProminence', ...
        mean(p)+std(p),'MinPeakWidth',mean(w)+std(w));
    plot(csvdata,'-o','MarkerIndices',locs,'MarkerEdgeColor','r')
end
```

3 Problem 3



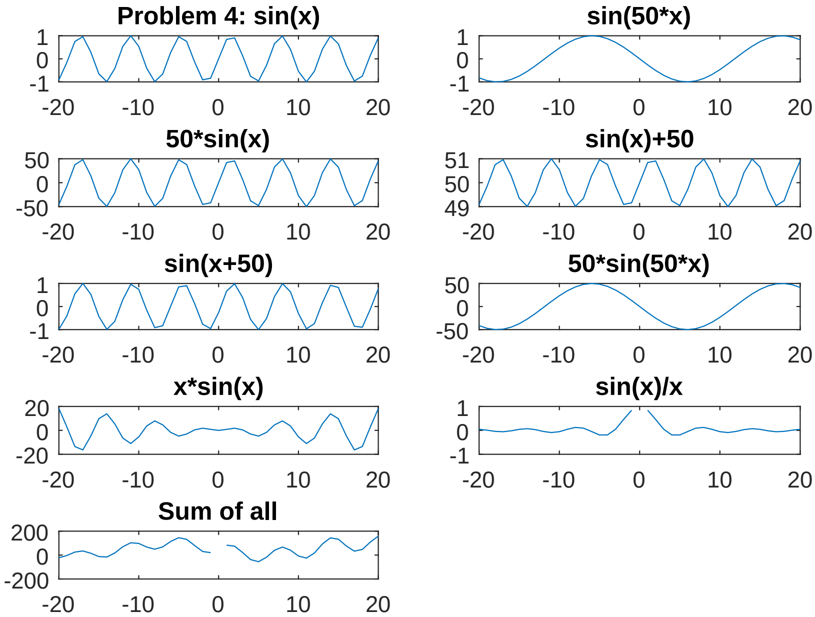
The sine function in the first subplot is not so smooth because we only provide one point to plot for every one unit. The sin(50*x) function looks wrong. If we multiply the inside of the sine function with a constant greater than one, its frequency should increase but instead it decreases because the distance between points that we plot increases. The 50*sin(x) function is correct. Multiplication increases the amplitude as expected. The sin(x)+50 function is also correct. Adding a positive number shifts the graph upwards. The 50*sin(50*x) is wrong because of the same reason with sin(50*x) function. If we multiply the sin(x) function with x, the sine function's amplitude increases as it goes from zero to both sides. If we divide it by x, its amplitude decreases as it goes from zero to both sides. The value of the function at 0 is blank because division by zero is undefined.

3.1 Code

```
figure;
x = -100:100;

subplot(4,2,1);
y1 = sin(x);
plot(x,y1);
title('Problem 3: sin(x)')

subplot(4,2,2);
y2 = sin(50.*x);
plot(x,y2);
title('sin(50*x)')

subplot(4,2,3);
y3 = 50.*sin(x);
plot(x,y3);
title('50*sin(x)')

subplot(4,2,4);
y4 = sin(x)+50;
plot(x,y4);
title('sin(x)+50')

subplot(4,2,5);
y5 = sin(x+50);
plot(x,y5);
title('sin(x+50)')

subplot(4,2,6);
y6 = 50.*sin(50.*x);
plot(x,y6);
title('50*sin(50*x)')

subplot(4,2,7);
y7 = x.*sin(x);
plot(x,y7);
title('x*sin(x)')
```

```
subplot(4,2,8);
y8 = sin(x)./x;
plot(x,y8);
title('sin(x)/x')
```

4 Problem 4



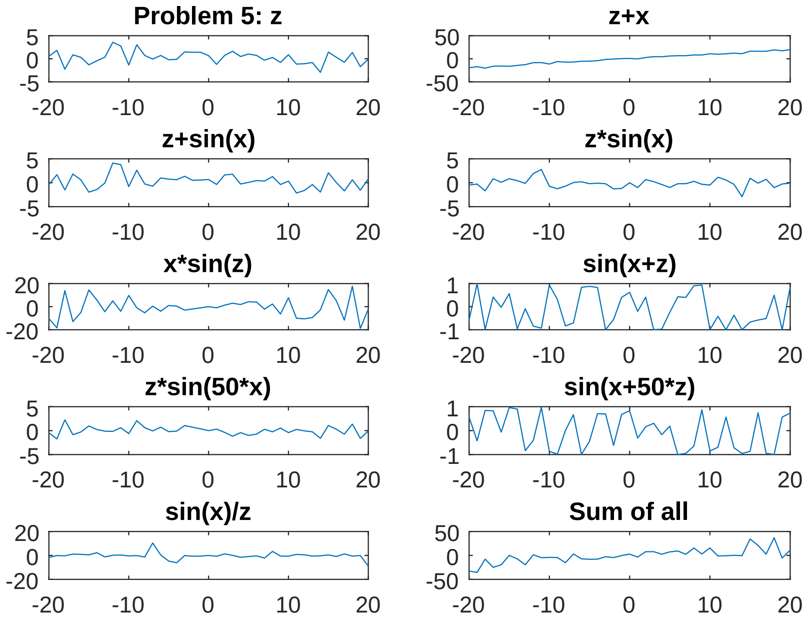
I made the same observation with Problem 3. Except, in this problem, changes are more visible since the domain in this problem is smaller than domain in previous problem. For example, the shift in sin(x+50) is visible.

4.1 Code

```
figure;
x = -20:20;

subplot(5,2,1);
y1 = sin(x);
plot(x,y1);
title('Problem 4: sin(x)')

subplot(5,2,2);
y2 = sin(50.*x);
plot(x,y2);
```

```
title('sin(50*x)')

subplot(5,2,3);
y3 = 50.*sin(x);
plot(x,y3);
title('50*sin(x)')

subplot(5,2,4);
y4 = sin(x)+50;
plot(x,y4);
title('sin(x)+50')

subplot(5,2,5);
y5 = sin(x+50);
plot(x,y5);
title('sin(x+50)')

subplot(5,2,6);
y6 = 50.*sin(50.*x);
plot(x,y6);
title('50*sin(50*x)')

subplot(5,2,7);
y7 = x.*sin(x);
plot(x,y7);
title('x*sin(x)')

subplot(5,2,8);
y8 = sin(x)./x;
plot(x,y8);
title('sin(x)/x')

subplot(5,2,9);
y9 = y1+y2+y3+y4+y5+y6+y7+y8;
plot(x,y9);
title('Sum of all')
```

5 Problem 5



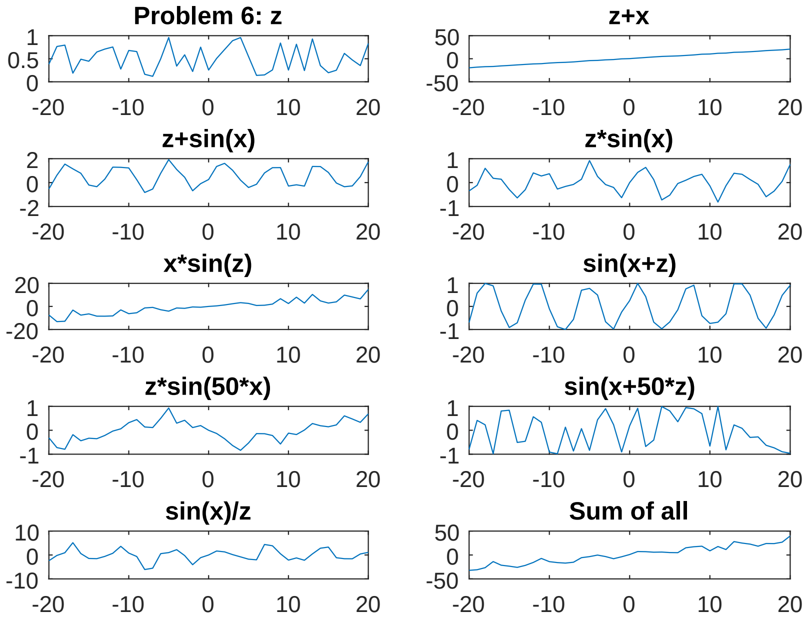
z vector has random values. z+x is almost a linear function. In all other graphs, the sine function is unrecognizable. They all look like random signals.

5.1 Code

```
figure;

z = randn(1,41);

subplot(5,2,1);
y10 = z;
plot(x,y10);
title('Problem 5: z')

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

subplot(5,2,3);
y12 = z+sin(x);
```

```
plot(x,y12);
title('z+sin(x)')

subplot(5,2,4);
y13 = z.*sin(x);
plot(x,y13);
title('z*sin(x)')

subplot(5,2,5);
y14 = x.*sin(z);
plot(x,y14);
title('x*sin(z)')

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

subplot(5,2,7);
y16 = z.*sin(50.*x);
plot(x,y16);
title('z*sin(50*x)')

subplot(5,2,8);
y17 = sin(x+50.*z);
plot(x,y17);
title('sin(x+50*z)')

subplot(5,2,9);
y18 = sin(x)./z;
plot(x,y18);
title('sin(x)/z')

subplot(5,2,10);
y19 = y11+y12+y13+y14+y15+y16+y17+y18;
plot(x,y19);
title('Sum of all')
```

6 Problem 6



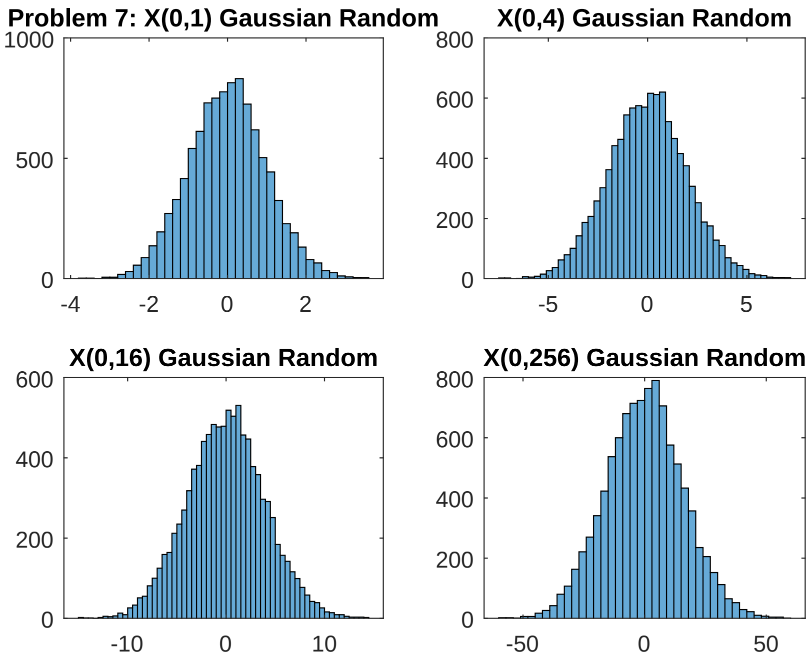
z vector has uniformly distributed random number as values. z+x is almost a linear function. z+sin(x), z*sin(x) and sin(x+z) function seems like periodic. The other graphs looks like they are random.

6.1 Code

```
figure;

z = rand(1,41);

subplot(5,2,1);
y20 = z;
plot(x,y20);
title('Problem 6: z')

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

subplot(5,2,3);
y22 = z+sin(x);
```

```
plot(x,y22);
title('z+sin(x)')

subplot(5,2,4);
y23 = z.*sin(x);
plot(x,y23);
title('z*sin(x)')

subplot(5,2,5);
y24 = x.*sin(z);
plot(x,y24);
title('x*sin(z)')

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

subplot(5,2,7);
y26 = z.*sin(50.*x);
plot(x,y26);
title('z*sin(50*x)')

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

subplot(5,2,9);
y28 = sin(x)./z;
plot(x,y28);
title('sin(x)/z')

subplot(5,2,10);
y29 = y21+y22+y23+y24+y25+y26+y27+y28;
plot(x,y29);
title('Sum of all')
```

7 Problem 7



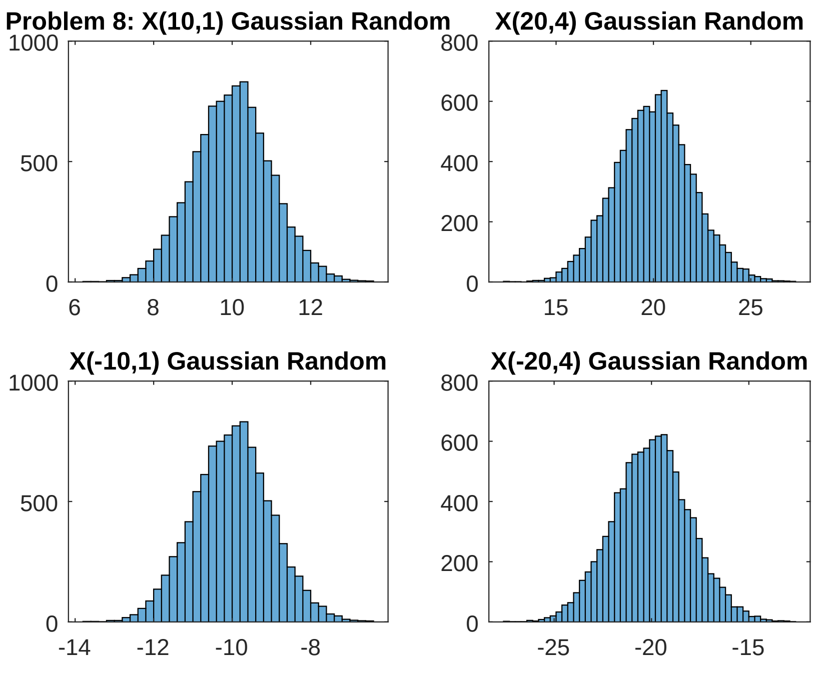
Gaussian random variable creates a histogram with bell shape. Most of the values are in the middle and have a value 0. If we increase the variance, the foots of the bell expands to the both sides.

7.1 Code

```
figure;

z = randn(1,10000);

r1 = z;
r2 = 2.*z;
r3 = 4.*z;
r4 = 16.*z;

subplot(2,2,1);
histogram(r1);
title('Problem 7: X(0,1) Gaussian Random')

subplot(2,2,2);
histogram(r2);
title('X(0,4) Gaussian Random')
```

```
subplot(2,2,3);
histogram(r3);
title('X(0,16) Gaussian Random')

subplot(2,2,4);
histogram(r4);
title('X(0,256) Gaussian Random')
```

8 Problem 8



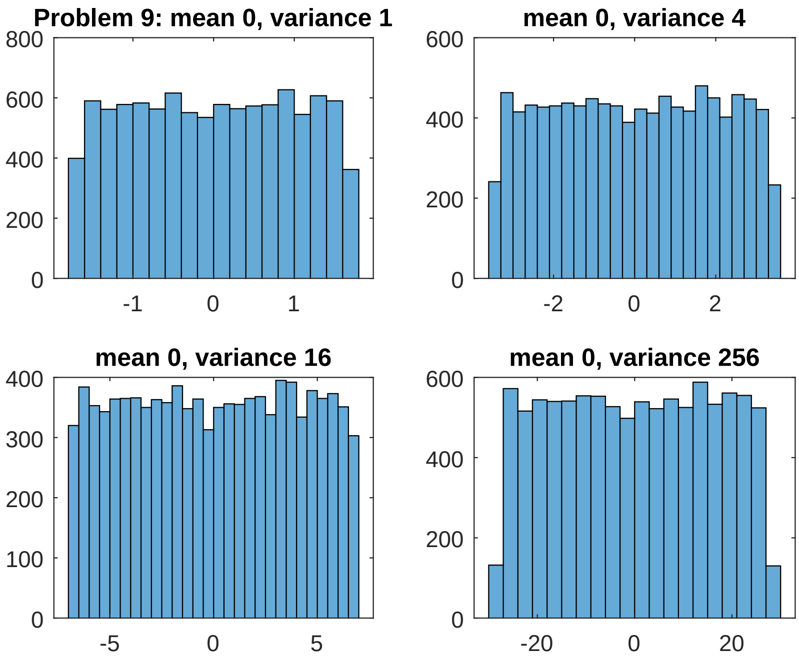
Gaussian random variable creates a histogram with bell shape. Most of the values have the value of the mean. If the mean is 10, the most of the values are around 10. If we increase the variance, the foots of the bell expands to the both sides.

8.1 Code

```
figure;

r6 = 10 + z;
r7 = 20 + 2.*z;
r8 = -10 + z;
r9 = -20 + 2.*z;
```

```
subplot(2,2,1);
histogram(r6);
title('Problem 8: X(10,1) Gaussian Random')

subplot(2,2,2);
histogram(r7);
title('X(20,4) Gaussian Random')

subplot(2,2,3);
histogram(r8);
title('X(-10,1) Gaussian Random')

subplot(2,2,4);
histogram(r9);
title('X(-20,4) Gaussian Random')
```

9 Problem 9

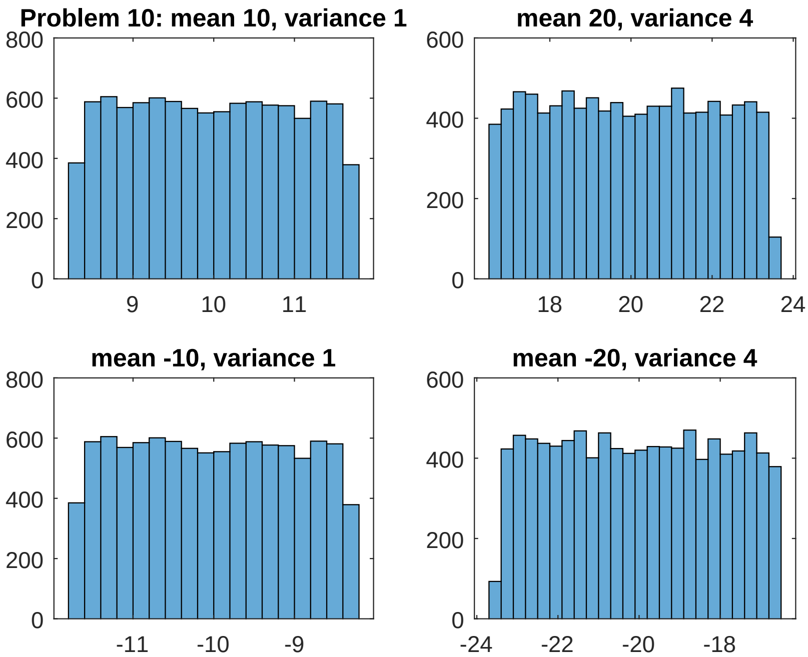


Uniformly distributed random variable creates a histogram in which the bins have mostly equal heights. Only the first and the last bin has smaller height. The bin in the middle of the histogram has the value of the mean. Almost one fourth of the values are smaller than the negative standard deviation. Almost one fourth of the values are between the negative standard deviation and the mean. Almost one fourth of the values are between the positive standard deviation and the mean. Almost one fourth of the values are bigger than the positive standard deviation.

9.1 Code

```
figure;

z = sqrt(12).*(rand(1,10000)-0.5);

r11 = z;
r21 = 2.*z;
r31 = 4.*z;
r41 = 16.*z;

subplot(2,2,1);
histogram(r11);
title('Problem 9: mean 0, variance 1')

subplot(2,2,2);
histogram(r21);
title('mean 0, variance 4')

subplot(2,2,3);
histogram(r31);
title('mean 0, variance 16')

subplot(2,2,4);
histogram(r41);
title('mean 0, variance 256')
```

10 Problem 10



Uniformly distributed random variable creates a histogram in which the bins have mostly equal heights. Only the first and the last bin has smaller height. The bin in the middle of the histogram has the value of the mean. Almost one fourth of the values are smaller than the negative standard deviation. Almost one fourth of the values are between the negative standard deviation and the mean. Almost one fourth of the values are between the positive standard deviation and the mean. Almost one fourth of the values are bigger than the positive standard deviation.

10.1 Code

```
figure;

z = sqrt(12).*(rand(1,10000)-0.5);

r61 = z + 10;
r71 = 2.*z +20;
r81 = z - 10;
r91 = 2.*z -20;

subplot(2,2,1);
histogram(r61);
```

```
title('Problem 10: mean 10, variance 1')

subplot(2,2,2);
histogram(r71);
title('mean 20, variance 4')

subplot(2,2,3);
histogram(r81);
title('mean -10, variance 1')

subplot(2,2,4);
histogram(r91);
title('mean -20, variance 4')
```

11 What I learned about Matlab?

I learned how to

- read a wav file
- read a cvs file
- show several plots in a figure
- plot time vs frequency graph of sound file using a spectrogram
- change the length of axes of a graph
- detect important peaks of a graph
- create a vector of Gaussian random numbers
- create a vector of uniformly distributed random numbers
- plot a histogram
- find mean and variance of an array
- plot graph of sinusoidal functions
- add title to subplots

12 Challenges I faced using Matlab

- Using spectrogram was very hard because I didn't know the meaning of parameters it wanted and the output of it. For instance, I didn't know what window is, what noverlap is, what number of DFT points(nfft) is, what short time fourier transform is, how to find the frequency of a certain time point.

- When I plotted a 4x2 subplot with 8 graphs and then a 5x2 subplot 9 graph, the last grid in the figure was not empty and it showed a graph from the 4x2 subplot. Then I learned about the figure function. It creates a new figure for the sub figures. So is my problem solved.
- When I first tried to visualize the sound with spectrogram, Matlab gave error. Later, I found that my sound file is stored as dual channel. When I selected one of the channel in the matrix of my sound file, I could draw the sound in the frequency domain.
- Finding a way to find all the files with a certain format inside all the sub folders was hard.
- At first I couldn't understand what "Starting with z (0,1) Gaussian(Normal) Random variable." means. Later, I found out that we need to create a random array and use it for later calculations.

13 Differences between Matlab and the other programming languages

Matlab has a big library and function for all kinds of things so manipulating and visualizing data is very easy in Matlab. For example, you plot the spectrum of sound file with only two commands. Writing the code for this in another programming language would be very hard and time consuming.One advantage of Matlab is that it has up-to-date and detailed documentation. In addition, it has some tutorials and examples.

Matlab has its own challange too. For example, it is very easy to forget the dot before multiplication if you want to multiply with a scalar and you end up with matrix multiplication without noticing it. Another example is the Problem 3. In this problem, when I multiplied sin function and plotted it, I got a wrong graph because with multiplication the distance between points to plot increases. We lose the details between this points.