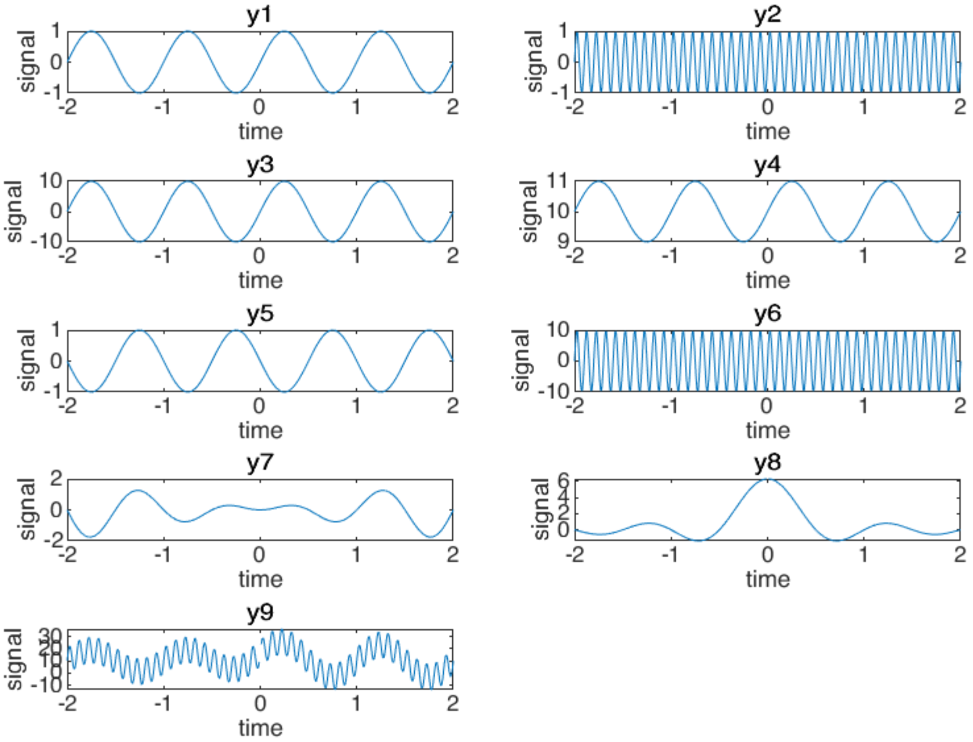
CmpE 362
Introduction to Signals
for Computer Engineers
Spring 2020
Homework 1

Emilcan ARICAN - 2016400231

Solution 1

First I cleared old variables. Then, created a time vector, and created vectors for the values of the functions in those times. After that, plotted them in a figure as subplots.

I observed how the signals differ when amplitude, frequency, phase, etc. changes applied.



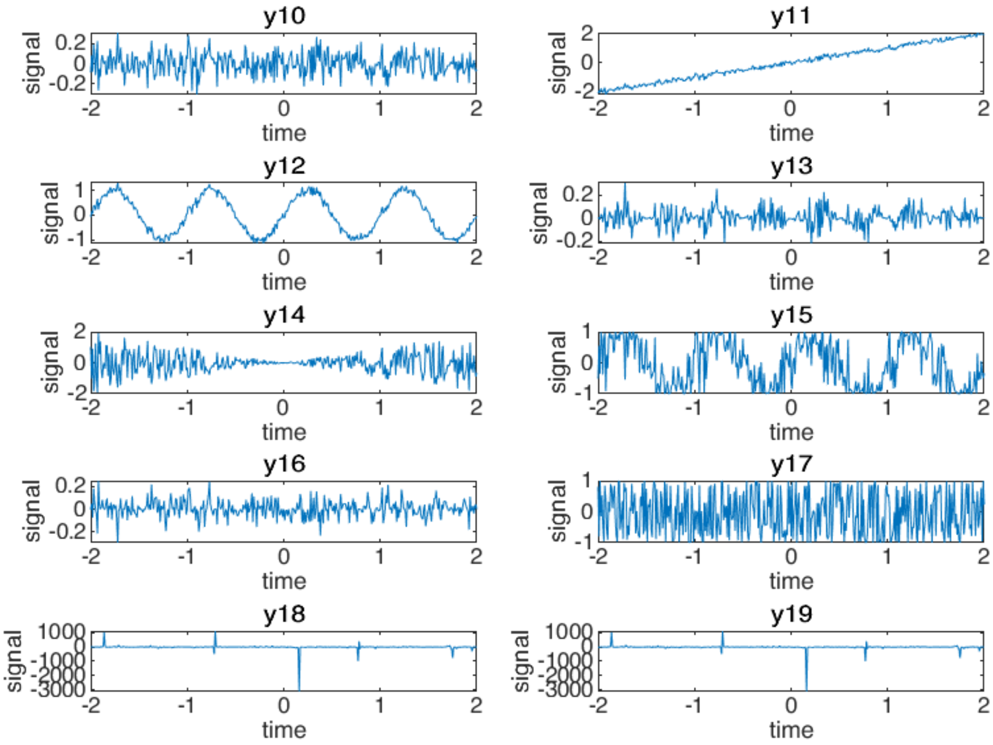
Solution 1 - Code

```
%% START: Question-1
%clear old variables
clear; clc;

%create time vector
t = -2:0.01:2;

%create vectors for function values
y1 = sin(2*pi*t);
y2 = sin(2*pi*10*t);
y3 = 10*sin(2*pi*t);
y4 = sin(2*pi*t)+10;
y5 = sin(2*pi*(t-0.5));
y6 = 10*sin(2*pi*10*t);
y7 = t.*sin(2*pi*t);
y8 = sin(2*pi*t)./t;
y9 = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8;

%create a figure with Name "Question-1"
figure(’Name’, ’Question-1’, ’NumberTitle’, ’off’);
%plot the functions as subplots with specified titles and labels.
subplot(5,2,1), plot(t,y1), title(’y1’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,2), plot(t,y2), title(’y2’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,3), plot(t,y3), title(’y3’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,4), plot(t,y4), title(’y4’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,5), plot(t,y5), title(’y5’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,6), plot(t,y6), title(’y6’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,7), plot(t,y7), title(’y7’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,8), plot(t,y8), title(’y8’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,9), plot(t,y9), title(’y9’), xlabel(’time’), ylabel(’signal’);
% END: Question-1
```

Solution 2

First I cleared old variables. Then, created a time vector, and vector of random variables multiplied by 0.1. Created vectors for function values, y1 and y2 is from the question 1 however they are used in this question too. Since this section might be run separately, y1 and y2 created at here as well. After that, plotted functions that are specified in the question, in a figure as subplots.

In this question we have added noise to some signals. And observed how this effect the signals. This noise (created by gaussian random variable) spoils the signal more than the one in the third question. Since gaussian random variable is not uniformly distributed.



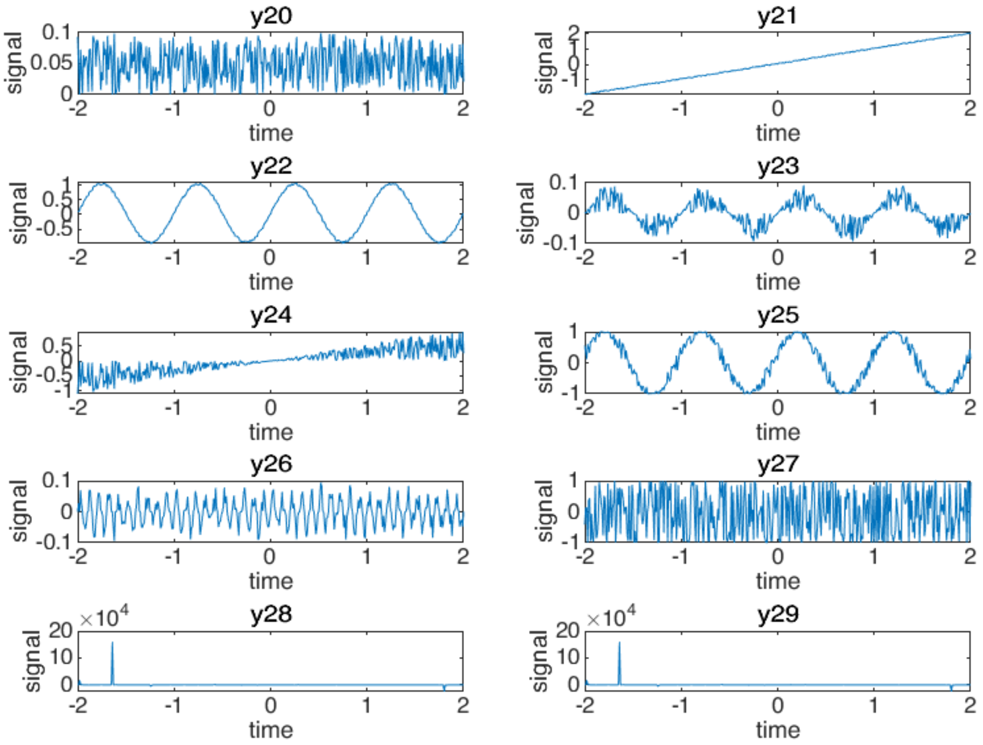
Solution 2 - Code

```
%% START: Question-2
%clear old variables
clear; clc;

%create time vector
t = -2:0.01:2;

%create gaussian random numbers
z = randn(1,401);
z = z*0.1;

%create vectors for function values
%these two functions are from question-1
y1 = sin(2*pi*t);
y2 = sin(2*pi*10*t);

%create vectors for function values
y10 = z;
y11 = z+t;
y12 = z + y1;
y13 = z .* y1;
y14 = t.*sin(2*pi*z);
y15 = sin(2*pi*(t+z));
y16 = z .* y2;
y17 = sin(2*pi*(t+10*z));
y18 = y1 ./ z;
y19 = y10 + y11 + y12 + y13 + y14 + y15 + y16 + y17 + y18;

%create a figure with Name "Question-2"
figure(’Name’, ’Question-2’, ’NumberTitle’, ’off’);
%plot the functions as subplots with specified titles and labels.
subplot(5,2,1), plot(t,y10), title(’y10’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,2), plot(t,y11), title(’y11’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,3), plot(t,y12), title(’y12’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,4), plot(t,y13), title(’y13’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,5), plot(t,y14), title(’y14’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,6), plot(t,y15), title(’y15’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,7), plot(t,y16), title(’y16’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,8), plot(t,y17), title(’y17’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,9), plot(t,y18), title(’y18’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,10), plot(t,y19), title(’y19’), xlabel(’time’), ylabel(’signal’);
% END: Question-2
```

Solution 3

First I cleared old variables. Then, created a time vector, and vector of random variables multiplied by 0.1. Created vectors for function values, y1 and y2 is from the question 1 however they are used in this question too. Since this section might be run separately, y1 and y2 created at here as well. After that, plotted functions that are specified in the question, in a figure as subplots.

In this question we have added noise to some signals. And observed how this effect the signals. This noise spoils the signal but not as much as the one in the second question. Since uniformly distributed random variable is used in this question.



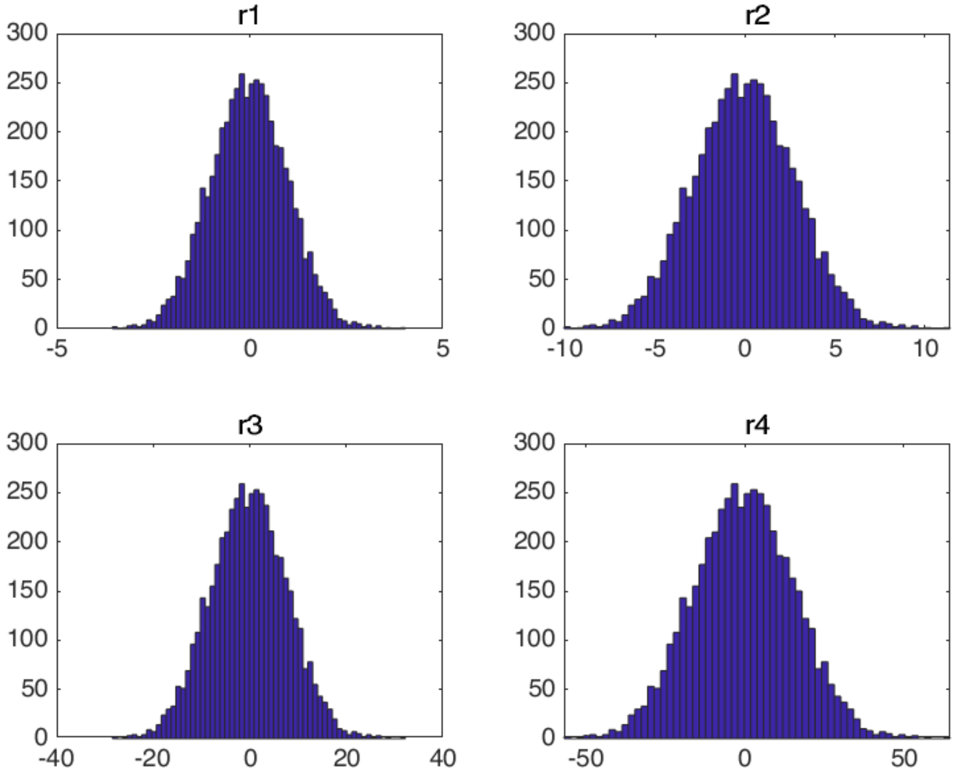
Solution 3 - Code

```
%% START: Question-3
%clear old variables
clear; clc;

%create time vector
t = -2:0.01:2;

%create random numbers
z = rand(1,401);
z = z*0.1;

%create vectors for function values
%these two functions are from question-1
y1 = sin(2*pi*t);
y2 = sin(2*pi*10*t);

%create vectors for function values
y20 = z;
y21 = z+t;
y22 = z+y1;
y23 = z.*y1;
y24 = t.*sin(2*pi*z);
y25 = sin(2*pi*(t+z));
y26 = z.*y2;
y27 = sin(2*pi*(t+10*z));
y28 = y1./z;
y29 = y21 + y22 + y23 + y24 + y25 + y26 + y27 + y28;

%create a figure with Name "Question-1"
figure(’Name’, ’Question-3’, ’NumberTitle’, ’off’);
%plot the functions as subplots with specified titles and labels.
subplot(5,2,1), plot(t,y20), title(’y20’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,2), plot(t,y21), title(’y21’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,3), plot(t,y22), title(’y22’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,4), plot(t,y23), title(’y23’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,5), plot(t,y24), title(’y24’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,6), plot(t,y25), title(’y25’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,7), plot(t,y26), title(’y26’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,8), plot(t,y27), title(’y27’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,9), plot(t,y28), title(’y28’), xlabel(’time’), ylabel(’signal’);
subplot(5,2,10), plot(t,y29), title(’y29’), xlabel(’time’), ylabel(’signal’);
% END: Question-3
```

Solution 4

First I cleared old variables. Then, created vector of gausssian random variables. After that changed their mean and variance according to $z = (x - \mu)/\sigma$ formula. Since, we started with z, and we want to get x; we reorganize the formula so it becomes $x = \mu + z.\sigma$. After creating the distributions, their histograms are plotted as subplots.

I learned how to create gaussian distributions with specified variance. I also observed that; as the variance is increased, floor of the distribution widens.



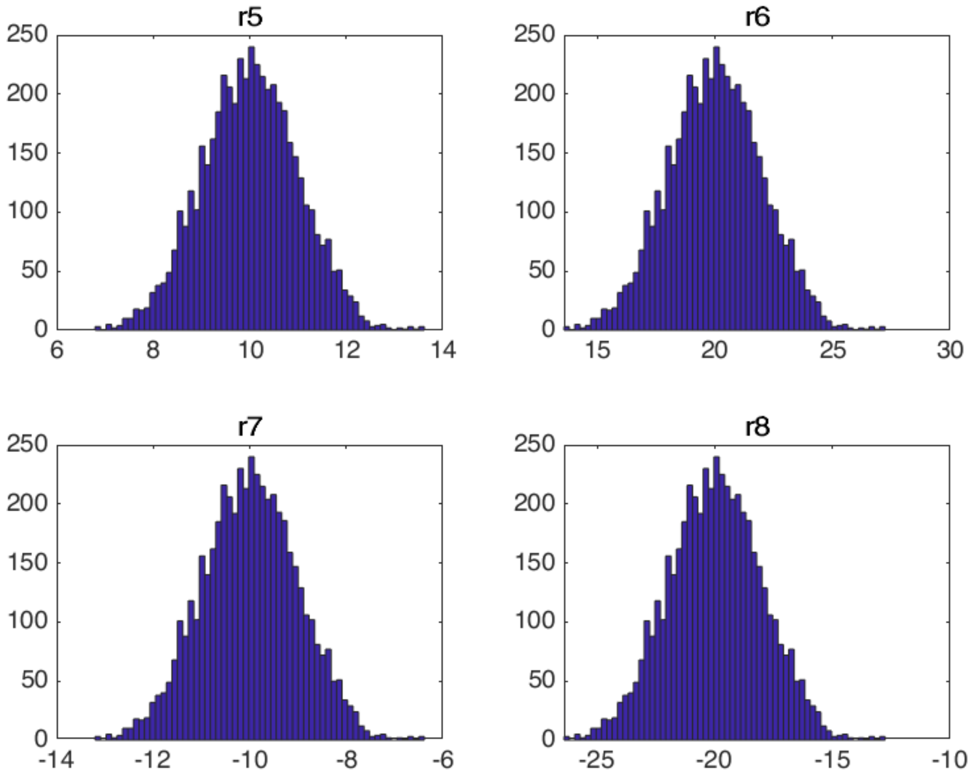
Solution 4 - Code

```
%% START: Question-4
%clear old variables
clear; clc;

%create gaussian random numbers
z = randn(1,5000);

%create a distribution with 0 mean and 1 variance
r1 = 0 + sqrt(1)*z;
%create a distribution with 0 mean and 8 variance
r2 = 0 + sqrt(8)*z;
%create a distribution with 0 mean and 64 variance
r3 = 0 + sqrt(64)*z;
%create a distribution with 0 mean and 256 variance
r4 = 0 + sqrt(256)*z;

%create a figure with Name "Question-4"
figure(’Name’, ’Question-4’, ’NumberTitle’, ’off’);
%plot the histograms of random numbers as subplots
%with specified titles and labels.
subplot(2,2,1), hist(r1,60), title(’r1’);
subplot(2,2,2), hist(r2,60), title(’r2’);
subplot(2,2,3), hist(r3,60), title(’r3’);
subplot(2,2,4), hist(r4,60), title(’r4’);
% END: Question-4
```

Solution 5

First I cleared old variables. Then, created vector of gausssian random variables. After that changed their mean and variance according to $z = (x - \mu)/\sigma$ formula. Since, we started with z, and we want to get x; we reorganize the formula so it becomes $x = \mu + z.\sigma$. After creating the distributions, their histograms are plotted as subplots.

I learned how to create gaussian distributions with specified variance and mean. I observed that; as the variance is increased, floor of the distribution widens. Also observed that when mean is changed distribution moves on the horizontal axis.



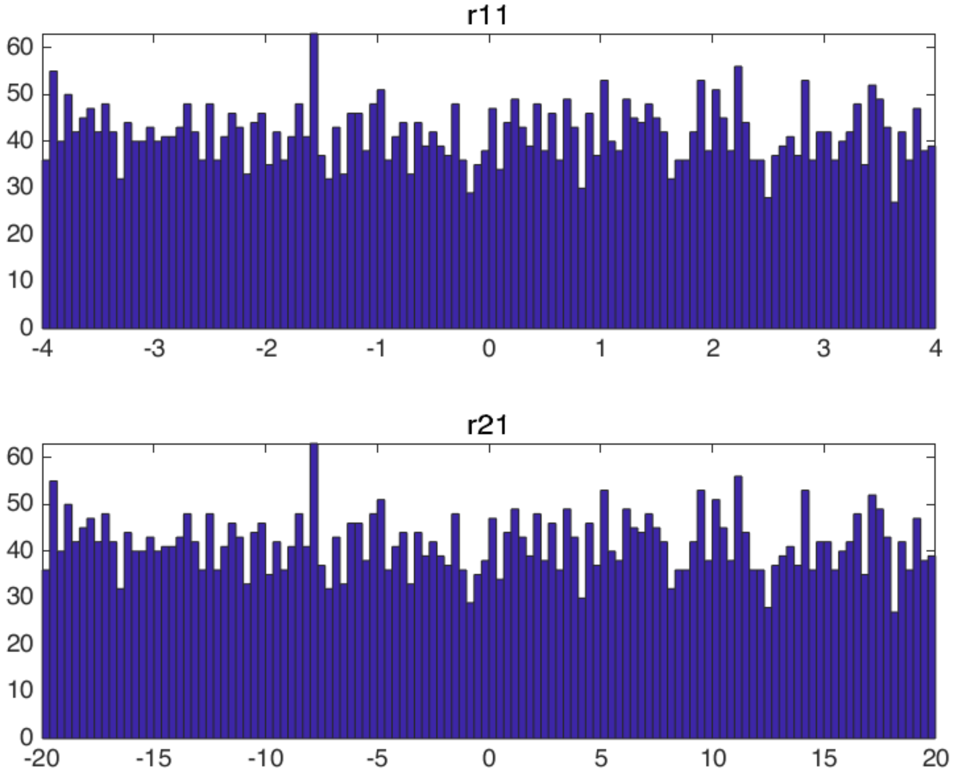
Solution 5 - Code

```
%% START: Question-5
%clear old variables
clear; clc;

%create gaussian random numbers
z = randn(1,5000);

%create a distribution with 10 mean and 1 variance
r5 = 10 + sqrt(1)*z;
%create a distribution with 20 mean and 4 variance
r6 = 20 + sqrt(4)*z;
%create a distribution with -10 mean and 1 variance
r7 = -10 + sqrt(1)*z;
%create a distribution with -20 mean and 4 variance
r8 = -20 + sqrt(4)*z;

%create a figure with Name "Question-5"
figure('Name', 'Question-5', 'NumberTitle', 'off');
%plot the histograms of random numbers as subplots
%with specified titles and labels.
subplot(2,2,1), hist(r5,60), title('r5');
subplot(2,2,2), hist(r6,60), title('r6');
subplot(2,2,3), hist(r7,60), title('r7');
subplot(2,2,4), hist(r8,60), title('r8');
% END: Question-5
```

Solution 6

First I cleared old variables. Then, created vector of random variables that are between 0 and 1. For the first part, multiplied them by 8, so they are between 0 and 8. After subtracted 4, they are between 4 and -4. Similar procedure is applied for the second part. Multiplied with 40 and subtracted 20. After that, their histograms are plotted as subplots.

I learned that how to generate random numbers in a specified range by using randomly generated numbers between 0 and 1.



Solution 6 - Code

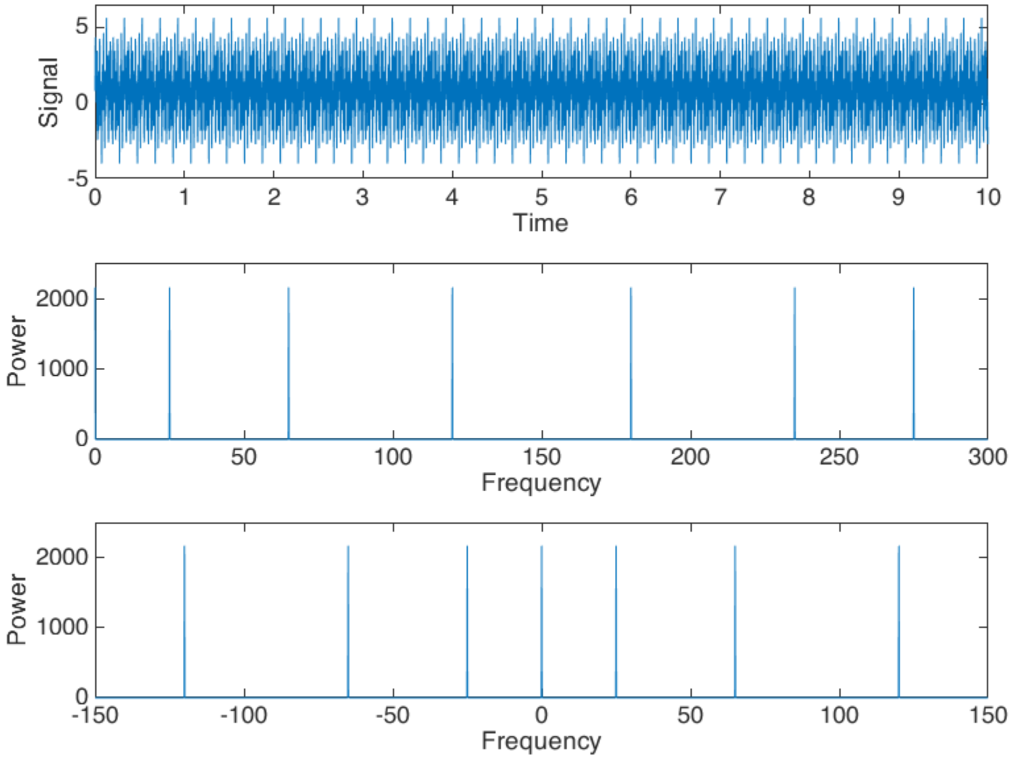
```
%% START: Question-6
clear; clc;

%create random numbers
z = rand(1,5000);

%create random numbers between 4 and -4
r11 = (z*8)-4;
%create random numbers between 20 and -20
r21 = (z*40)-20;

%create a figure with Name "Question-6"
figure(’Name’, ’Question-6’, ’NumberTitle’, ’off’);
%plot the histograms of random numbers as subplots
%with specified titles and labels.
subplot(2,1,1), hist(r11,120), title(’r11’);
subplot(2,1,2), hist(r21,120), title(’r21’);
% END: Question-6
```

Solution 7

First I cleared old variables. Loaded the workspace. Then, applied fourrier transformation in order to move the signal to frequency domain. After that, centered it on the 0. Finally plotted the signal in time domain and in frequency domain (shifted one as well).

Spikes are appears at 0, 25, 65, and 120. So the mathematical formula of the signal is:

x(t) $= \mathrm{A}_0.cos(2.\pi.0.t + \theta_0) + A_1.cos(2.\pi.25.t - 1.5708\theta_1) + A_2.cos(2.\pi.65.t + \theta_2)$
$+ A_3.cos(2.\pi.120.t - \theta_3)$

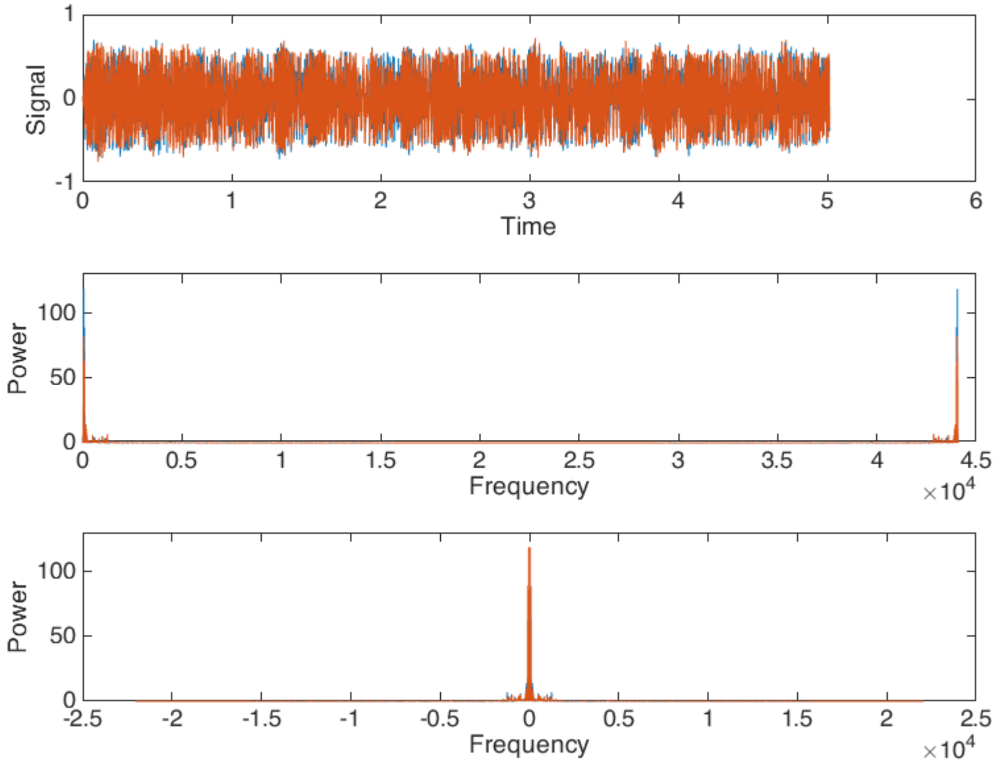


Solution 7 - Code

```
%% START: Question-7
%clear old variables
clear; clc;

%load provided workspace
load(’workspace/mysignal.mat’);

Y = fft(x);              %discrete fourier transform
n = length(x);           %number of samples
f = (0:n-1)*(fs/n);      %frequency range
p = abs(Y).^2/n;         %power of the DFT

Y0 = fftshift(Y);        %shift Y values
f0 = (-n/2:n/2-1)*(fs/n); %0-centered frequency range
p0 = abs(Y0).^2/n;       %0-centered power

%plot signal with specified labels
subplot(3,1,1), plot(t,x), xlabel(’Time’), ylabel(’Signal’), ylim([-5 6.5]);
subplot(3,1,2), plot(f,p), xlabel(’Frequency’), ylabel(’Power’), ylim([0 2500]);
subplot(3,1,3), plot(f0,p0), xlabel(’Frequency’), ylabel(’Power’), ylim([0 2500]);
% END: Question-7
```

Solution 8

First I cleared old variables. Then read audio file, and get its size. There are two audio channels. Calculated the unit time and created an time vector. Moved the signal time domain to frequency domain. And finally, plotted them.



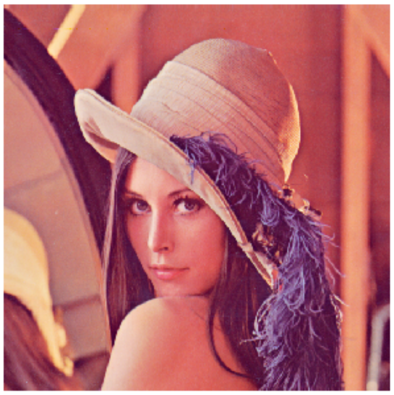
Solution 8 - Code

```
%% START: Question-8
%clear old variables
clear;clc;

%read the audio file
[y,fs] = audioread('audio/nyan-cat.m4a');

%get the dimentions of the audio
[m,n] = size(y);

%there are two channels
y1 = y(:,1);
y2 = y(:,2);

%find unit time
dt = 1/fs;
%create time vector
t = dt*(0:m-1);

Y = fft(y);              %discrete fourier transform
f = (0:m-1)*(fs/m);      %frequency range
p = abs(Y).^2/m;         %power of the DFT

Y0 = fftshift(Y);        %shift Y values
f0 = (-m/2:m/2-1)*(fs/m); %0-centered frequency range
p0 = abs(Y0).^2/m;       %0-centered power

subplot(3,1,1), plot(t,y), xlabel('Time'), ylabel('Signal');
subplot(3,1,2), plot(f,p), xlabel('Frequency'), ylabel('Power');
subplot(3,1,3), plot(f0,p0), xlabel('Frequency'), ylabel('Power');
```

Solution 9

First I cleared old variables. I read the image, and took its dimentions. Then, created its gray form. Calculated mean, and standard deviation. After that calculated min and max values and their indexes. However since these were linear indexes, converted them into row and column form.

$$\begin{aligned}
mean =& 124.0425 \\
standard - deviation =& 47.8557 \\
\\
minimum =& 25 \\
\text{linear index of minimum} =& 1608 \\
(\text{row of minimum}, \text{column of minimum}) =& (72, 4) \\
\\
maximum =& 245 \\
\text{linear index of maximum} =& 202514 \\
(\text{row of maximum}, \text{column of maximum}) =& (274, 396)
\end{aligned}$$



Solution 9 - Code

```
%% START: Question-9
%clear old variables
clear;clc;

%read the image
img = imread('image/lena.png');

%take dimentions of the image
[m,n,p] = size(img);

%take the gray version of the image
img_gray = rgb2gray(img);

%calculate the mean of the matrix
matrix_mean = mean(img_gray(:));

%calculate the standard deviation of elements of the matrix
matrix_std = std(double(img_gray(:)));

%find value and linear index of the maximum element of the gray image matrix
[matrix_max, max_idx] = max(img_gray(:));
%convert linear index to row and column location
[max_row, max_col] = ind2sub(size(img_gray), max_idx);

%find value and linear index of the maximum element of the gray image matrix
[matrix_min, min_idx] = min(img_gray(:));
%convert linear index to row and column location
[min_row, min_col] = ind2sub(size(img_gray), min_idx);

%shows initial and gray forms of the image
subplot(1,2,1), imshow(img);
subplot(1,2,2), imshow(img_gray);
% END: Question-9
```