Q1)
```
%QUESTION 1
a = 10;
b = 2.5 * 10^25;
c = 4 + 3*1i;
d = exp(1i*2*pi/3);
```

Q2)
```
%QUESTION 2
aVec = [3.14 16 19 26];
bVec = [2.71; 8; 28; 182];
t = 1;
for k = 5.0:-0.2:-5.0
    cVec(t)=k;
    t = t+1;
end
dVec = logspace(0,1,101);
eVec = 'Hello';
```
Here for loop is used to space numbers from 5 to -5 however linspace could also be used.

Q3)
```
%QUESTION 3
Ones = ones(9);
aMat = Ones + Ones;
ve = [1 2 3 4 5 4 3 2 1];
bMat = diag(ve);
rshp = 1:100;
cMat = reshape(rshp,[10,10]);
dMat = NaN(3,4);
eMat = [[13 -1 5];[-22 10 -87]];
fMat = -3+(3+3)*rand(5,3);
fMat = floor(fMat);
```

Q4)
```
%QUESTION 4
x = 1 / ( 1 + exp( (15-a)/6 ) );
y = (sqrt(a) + b^(1/21))^pi;
z = log(real((c+d)*(c-d))*sin(a*pi/3))/(c*conj(c));
```
For denominators, some parentheses are used to maintain precedence.

Q5)
```
%QUESTION 5
xVec = 1 / (sqrt(2*pi*2.5^2)) * exp(-1*cVec.^2/(2*2.5^2));
yVec = sqrt(transpose(a).^2 + bVec.^2);
zVec = log10(1./dVec);
```

Q6)

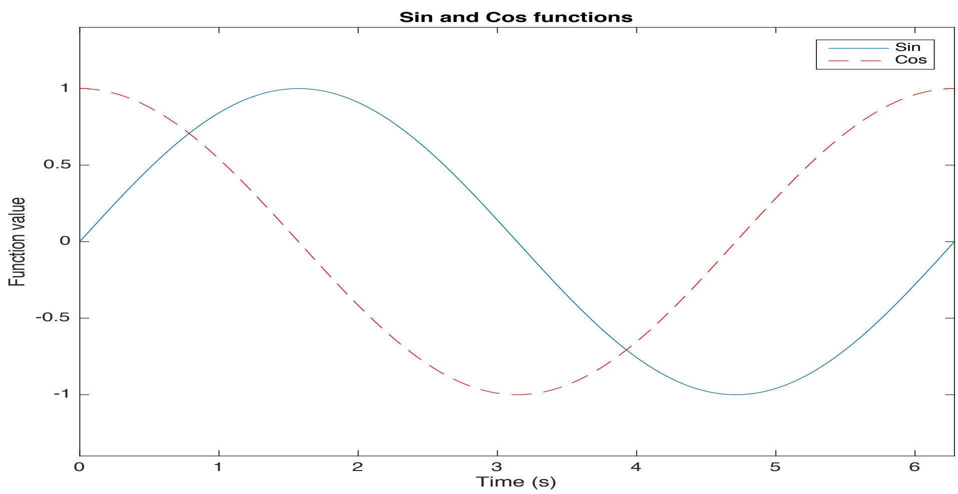
```
%QUESTION 6
xMat = (aVec * bVec)*aMat^2;
yMat = bVec * aVec;
zMat = det(cMat) * transpose(aMat * bMat);
```

Q7)

```
%QUESTION 7
cSum = sum(cMat);
eMean = mean(eMat,2);
eMat(1,:) = [1 1 1];
cSub = cMat(2:9,2:9);
lin = linspace(1,20,20);
for q = 2:2:20
    lin(q)=-1*q;
end
r = rand(1,5);
smalls = find(r<0.5,5);
r(smalls) = 0;
```

Q8)

```
figure
t = linspace(0,2*pi,1000);
y = sin(t);
z = cos(t);
plot(t,y,t,z,'--r');
xlabel('Time (s)');
ylabel('Function value');
title('Sin and Cos functions');
legend('Sin','Cos');
xlim([0 2*pi]);
ylim([-1.4 1.4]);
```

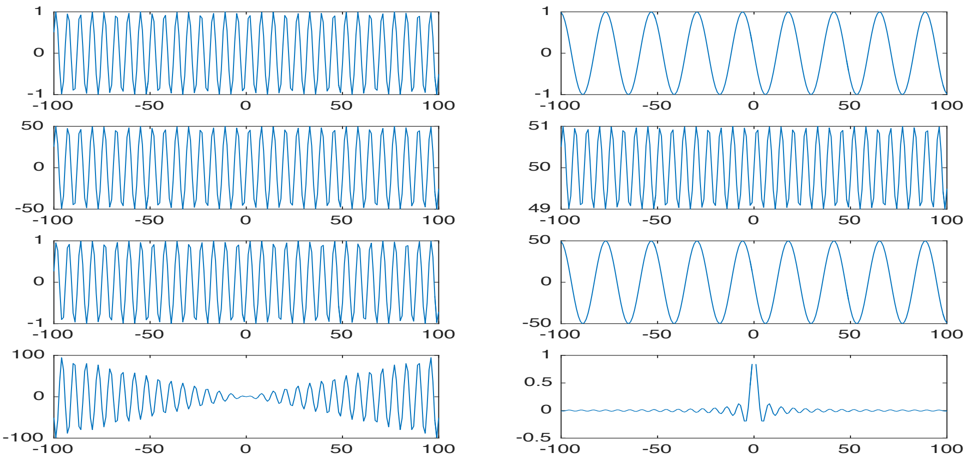


Q9)

```
load('classGrades.mat');
grades = namesAndGrades(1:end,2:end);
%meanGrades = mean(grades);
meanGrades = nanmean(grades);
meanMatrix = ones(size(grades,1),1)*meanGrades;
curvedGrades = 3.5 * (grades ./ meanMatrix);
k = find(curvedGrades>5);
curvedGrades(k) = 5;
totalGrade = nanmean(curvedGrades,2);
totalGrade = ceil(totalGrade);
letters = 'FDCBA';
letterGrades = letters(totalGrade);
alfa = ['Grades: ',letterGrades];
disp(alfa);
```

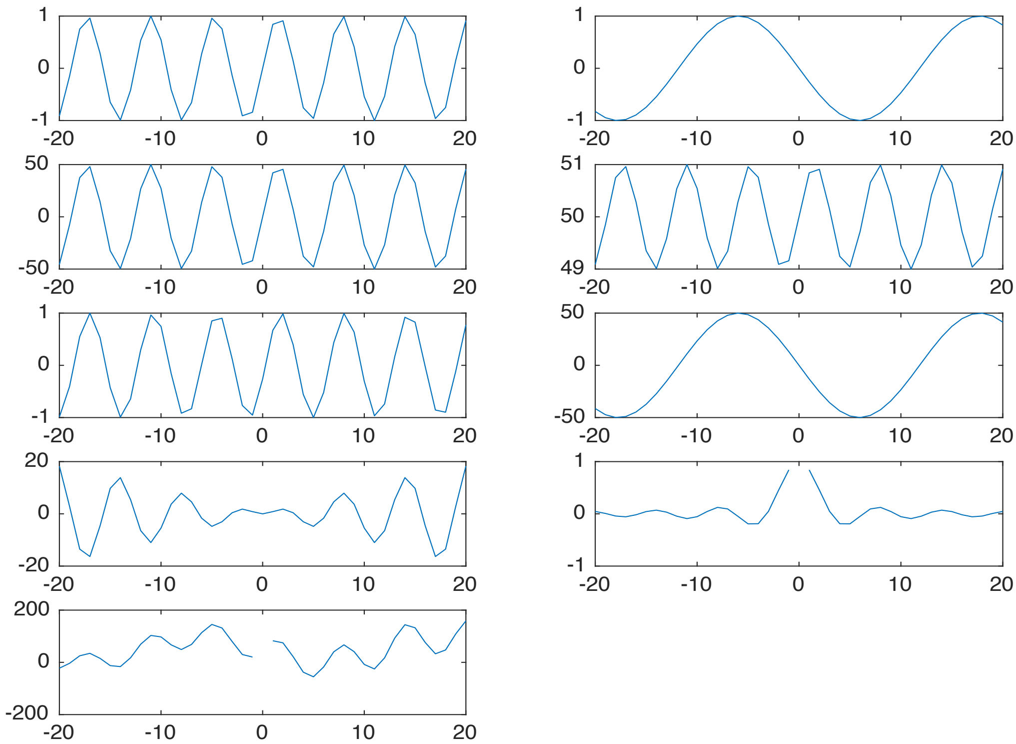
Q10)

```
%QUESTION 10
x = -100:100;
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;
figure
subplot(4,2,1); plot(x,y1)
subplot(4,2,2); plot(x,y2)
subplot(4,2,3); plot(x,y3)
subplot(4,2,4); plot(x,y4)
subplot(4,2,5); plot(x,y5)
subplot(4,2,6); plot(x,y6)
subplot(4,2,7); plot(x,y7)
subplot(4,2,8); plot(x,y8)
```



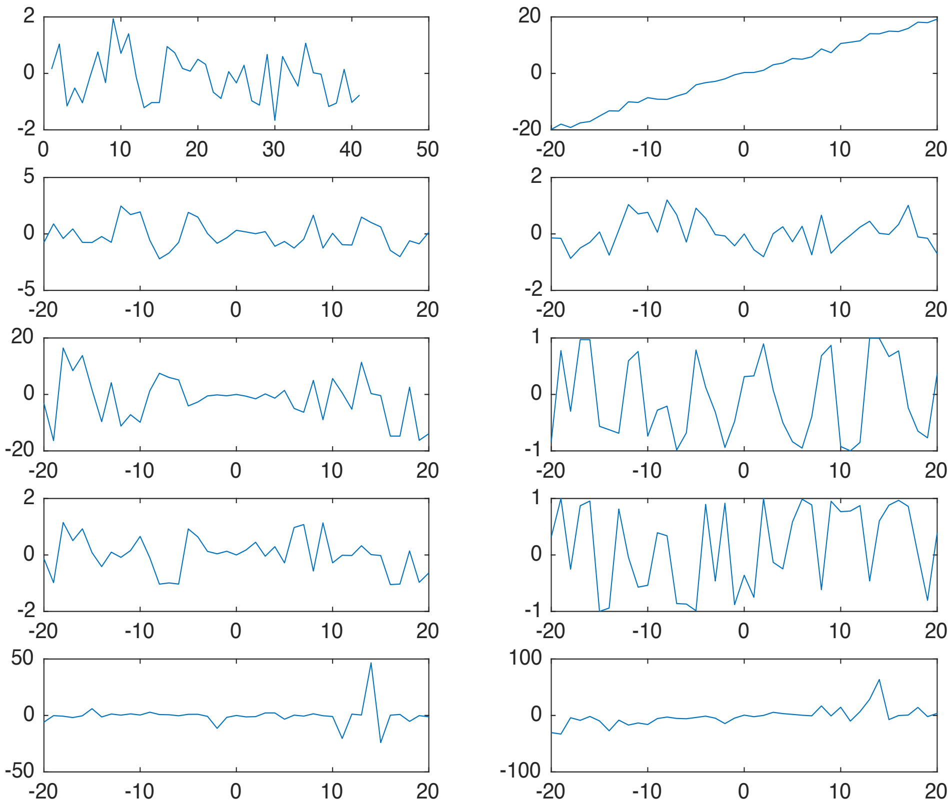
Q11)

```
x = -20:20;
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;
figure
subplot(5,2,1); plot(x,y1)
subplot(5,2,2); plot(x,y2)
subplot(5,2,3); plot(x,y3)
subplot(5,2,4); plot(x,y4)
subplot(5,2,5); plot(x,y5)
subplot(5,2,6); plot(x,y6)
subplot(5,2,7); plot(x,y7)
subplot(5,2,8); plot(x,y8)
subplot(5,2,9); plot(x,y9)
```



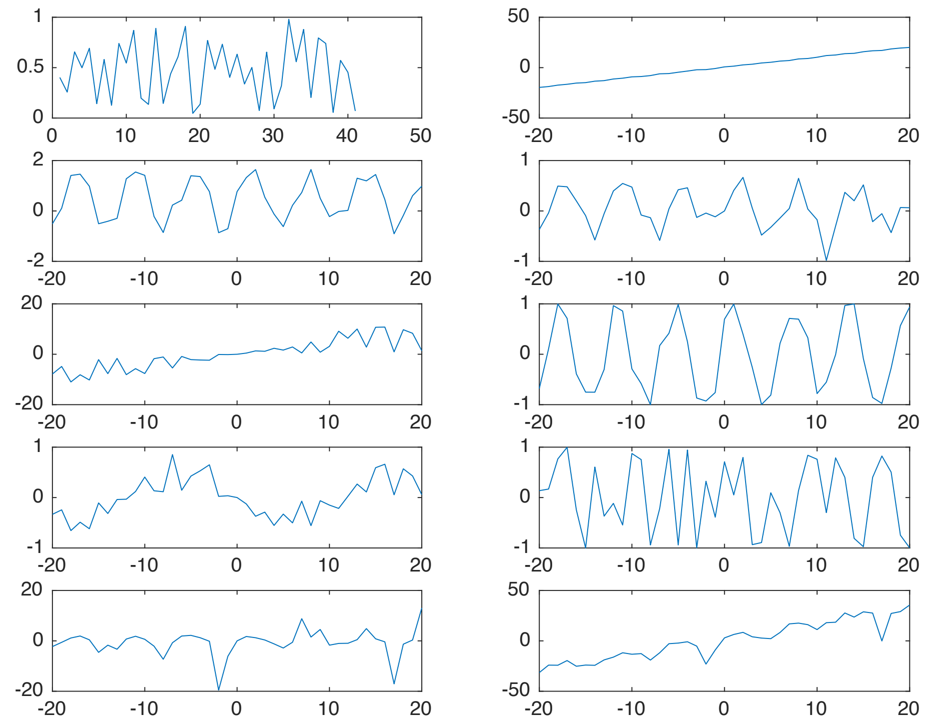
Q12)

```
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
subplot(5,2,1); plot(y10)
subplot(5,2,2); plot(x,y11)
subplot(5,2,3); plot(x,y12)
subplot(5,2,4); plot(x,y13)
subplot(5,2,5); plot(x,y14)
subplot(5,2,6); plot(x,y15)
subplot(5,2,7); plot(x,y16)
subplot(5,2,8); plot(x,y17)
subplot(5,2,9); plot(x,y18)
subplot(5,2,10); plot(x,y19)
```



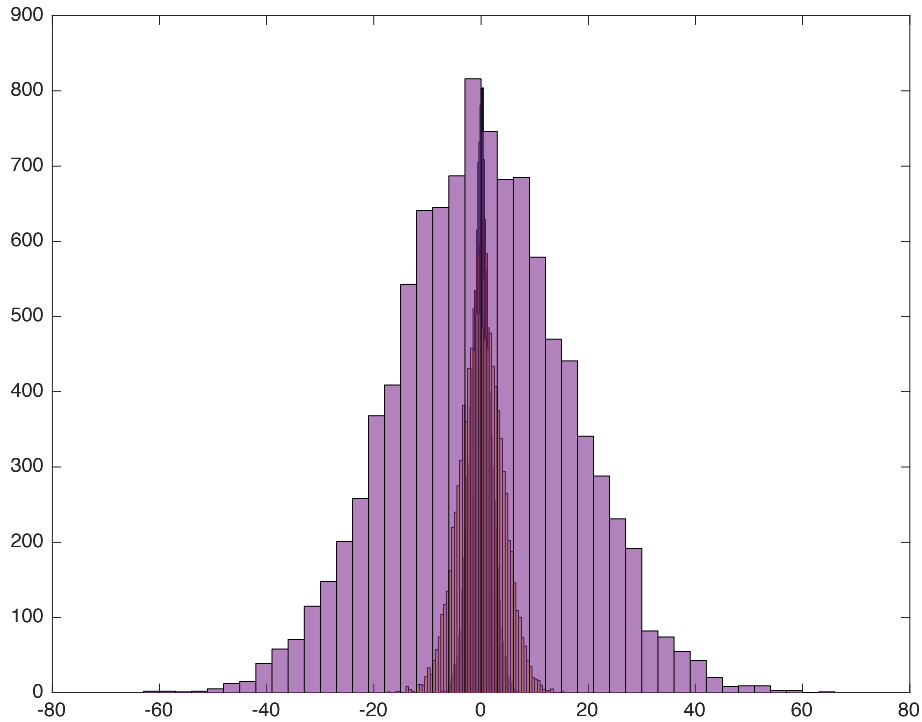
Q13)

```
%QUESTION 13
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
subplot(5,2,1); plot(y20)
subplot(5,2,2); plot(x,y21)
subplot(5,2,3); plot(x,y22)
subplot(5,2,4); plot(x,y23)
subplot(5,2,5); plot(x,y24)
subplot(5,2,6); plot(x,y25)
subplot(5,2,7); plot(x,y26)
subplot(5,2,8); plot(x,y27)
subplot(5,2,9); plot(x,y28)
subplot(5,2,10); plot(x,y29)
```



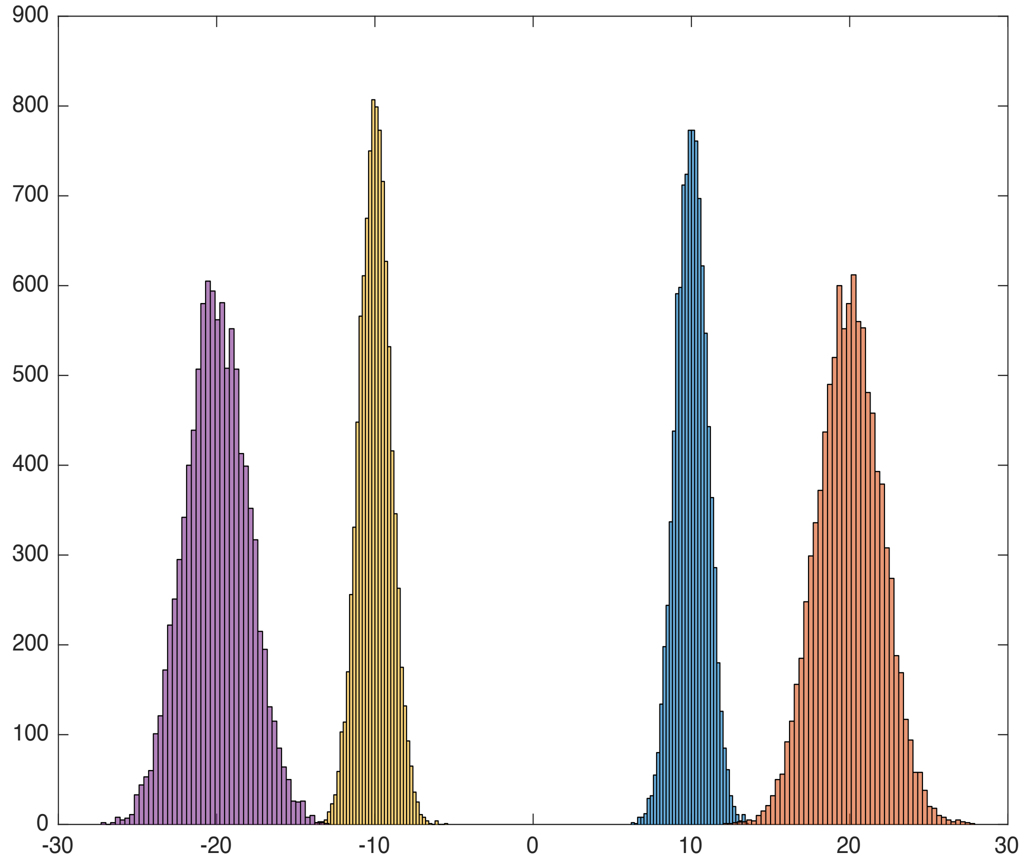
Q14)

```
mu = 0;
s1 = 1;
s2 = 2;
s3 = 4;
s4 = 16;
r1 = s1 .* randn(10000,1) + mu;
r2 = s2 .* randn(10000,1) + mu;
r3 = s3 .* randn(10000,1) + mu;
r4 = s4 .* randn(10000,1) + mu;
figure
h1 = histogram(r1);
hold on
h2 = histogram(r2);
hold on
h3 = histogram(r3);
hold on
h4 = histogram(r4);
```



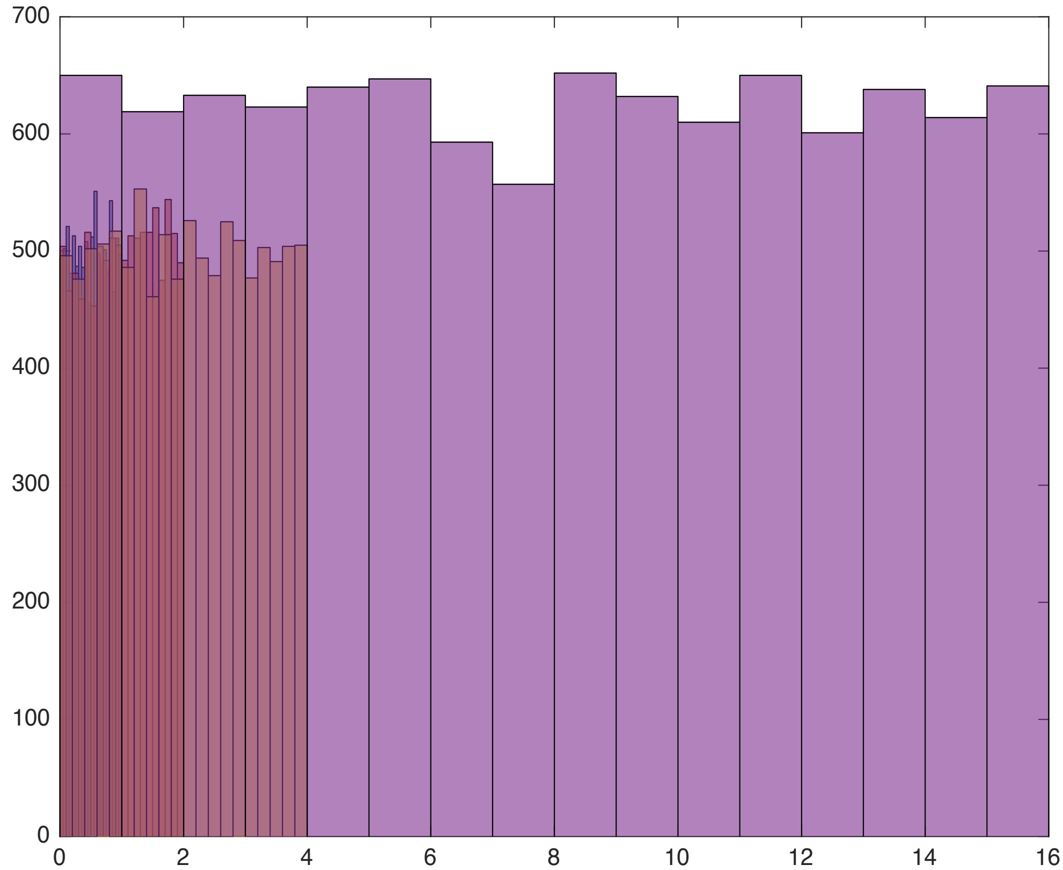
Q15)

```
%QUESTION 15
mu6 = 10;
mu7 = 20;
mu8 = -10;
mu9 = -20;
s6 = 1;
s7 = 2;
s8 = 1;
s9 = 2;
r6 = s6 .* randn(10000,1) + mu6;
r7 = s7 .* randn(10000,1) + mu7;
r8 = s8 .* randn(10000,1) + mu8;
r9 = s9 .* randn(10000,1) + mu9;
figure
h6 = histogram(r6);
hold on
h7 = histogram(r7);
hold on
h8 = histogram(r8);
hold on
h9 = histogram(r9);
```



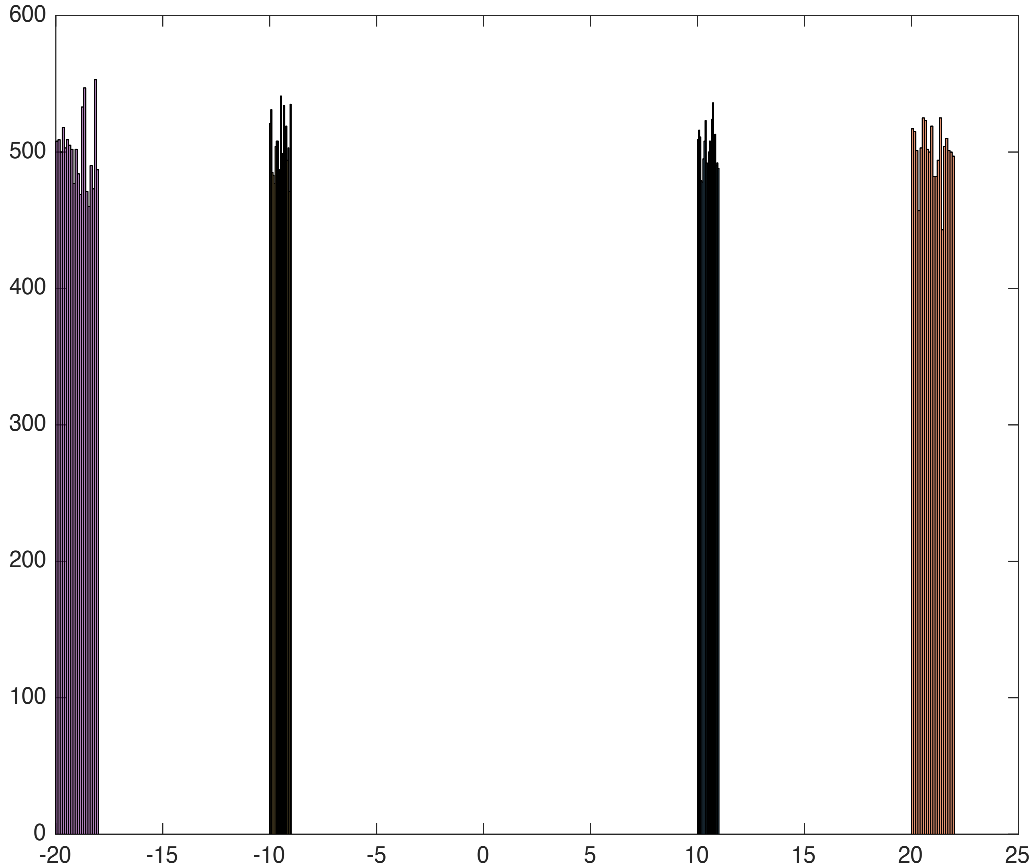
Q16)

```
%QUESTION 16
mu = 0;
s1 = 1;
s2 = 2;
s3 = 4;
s4 = 16;
r11 = s1 .* rand(10000,1) + mu;
r21 = s2 .* rand(10000,1) + mu;
r31 = s3 .* rand(10000,1) + mu;
r41 = s4 .* rand(10000,1) + mu;
figure
h11 = histogram(r11);
hold on
h21 = histogram(r21);
hold on
h31 = histogram(r31);
hold on
h41 = histogram(r41);
```



Q17)

```
mu61 = 10;
mu71 = 20;
mu81 = -10;
mu91 = -20;
s61 = 1;
s71 = 2;
s81 = 1;
s91 = 2;
r61 = s61 .* rand(10000,1) + mu61;
r71 = s71 .* rand(10000,1) + mu71;
r81 = s81 .* rand(10000,1) + mu81;
r91 = s91 .* rand(10000,1) + mu91;
figure
h61 = histogram(r61);
hold on
h71 = histogram(r71);
hold on
h81 = histogram(r81);
hold on
h91 = histogram(r91);
```



Q18)
From questions 10-13, I have learnt that as x goes to 0, sinx/x approaches to 1 and if we decrease the sample size, the plot becomes more unrealistic.
From question 14-17, if we increase the variance, then the plot gets wider, and most of the random numbers get around the mean.

Q19)
I have learnt the fundamentals of MATLAB such as matrix operations, random number generation and plotting functions. The challenging part was syntax of for loop.
For the advantages, there are matrix operations, exponential, plotting functions and gaussian random number generation. In the matrix operations, one needs to deal with several for loops in the other languages such as Java. In the exponential, MATLAB can do it with one symbol; however, in Java one needs a long expression. Plotting functions also requires less code lines than other languages. For gaussian, one needs to write its formula to generate random numbers in other languages, however MATLAB can do it with one function.
For the disadvantages, the first thing is semicolon. If one forgets semicolon, then after running the code, the command window gets filled with the values of the variables. The second point is for loop syntax.