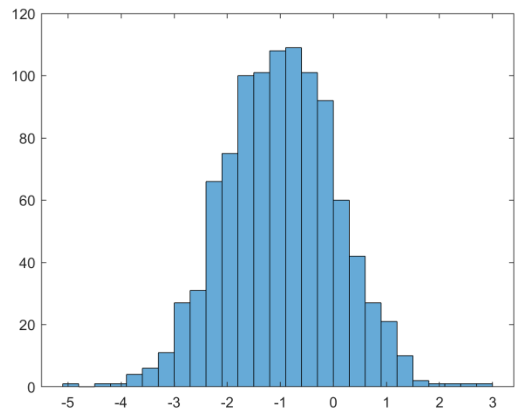
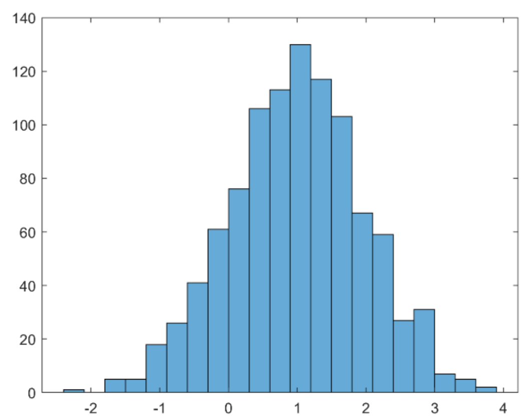
Kwong Yuet Michael Fadillah Wong
CSCI-B 455
1a) Use the csvread.m function to read the data from the Excel _le into a matrix. Be sure that the columns of data in the matrix, match the columns of data in the Excel
file = csvread("in_class2_data.csv")

1b) The distribution for the data in the _rst two columns is unknown. Generate separate histogram plots, one for the data in the _rst column and the other for the data in the second column. Based on the histograms, what family of distribution(s) best describes the two data sets? Be sure to include this answer in your writeup or just display it to the screen.
col1 = histogram(file(:,1))
col2 = histogram(file(:,2))
%Column1 data is leaning towards -1, it is a gaussian normal distribution
%Column2 data is leaning towards 1, it is also a gaussian normal distribution





1c) Using the distribution from the previous part, estimate the expected values (e.g. E[X]) for the data in the _rst two columns using maximum likelihood (ML) estimation. Look at the slides on parameter estimation for details. Note that there should be an expected value for the _rst column of data, and a separate expected value for the second column of data. Display the expected values to the screen.
mleE1 = sum(file(:,1))/length(file(:,1)) %-0.9876
mleE2 = sum(file(:,2))/length(file(:,2)) %0.9876

1d) In a similar fashion, estimate the unknown variances (e.g. V ar[X]) for the data in the _rst two columns using maximum likelihood (ML) estimation. Look at the slides on parameter estimation for details. Use the estimated expected values from above to estimate variance. Note that there should be a variance for the _rst column of data, and a separate variance for the second column of data. Display the variances to the screen.

```
mleVar1 = sum((file(:,1) - mleE1).^2)/length(file(:,1)) %1.0927
mleVar2 = sum((file(:,2) - mleE2).^2)/length(file(:,2)) %0.9447
```

2a) For each data sample from the _rst column, compute the likelihood that it was generated from H0 and from H1. Hence you should have two probability values for each sample from column one. Hint: write a function that computes the probability from the given data sample, mean, and variance. Use the distribution family that was indicated in question 1(b).

```
first = file(:,1)
like1 = []
for i = 1 :length(first)
        like1 = [like1 ; likelihood_x(first(i))]
end
```

2b) Repeat the above step, but use data from the second column of the Excel.

```
second = file(:,2)
like2 = []
for i = 1 :length(second)
        like2 = [like2 ; likelihood_x(second(i))]
end
```

definition for a and b:

```
function f = likelihood_x(x)
        e1 = -0.9876
        e2 = 0.9875
        var1 = 1.0927
        var2 = 0.9447
        p1 = normpdf(x,e1,sqrt(var1))
        p2 = normpdf(x,e2,sqrt(var2))
        f(1) = p1
        f(2) = p2
end
```

2c) Assuming uniform costs and equal priors, compute the likelihood ratio and compare it to the likelihood ratio test, in order to classify each data sample from each column. Store your decision (0 or 1) for each data sample of column one in a variable, and likewise store your decision for the column two data samples in a separate variable. Hint: Use the `>' operator to output 1 when the left-hand side is greater than the right-hand side, and 0 otherwise.

```
decision1 = []
for i = 1 : length(like1)
   temp = like1(i,:)
   left = temp(1)
   right = temp(2)
   decision1 = [decision1 ; right > left]
end
```

```
decision2 = []
for i = 1 : length(like2)
   temp = like2(i,:)
   left = temp(1)
   right = temp(2)
   decision2 = [decision2 ; right > left]
end
```

(results are shown in the matlab for Question 2a ,2b , and 2c)

2d) Compute the decision error (e.g. number of incorrectly classi_ed points divided by the total number of points) for each column, considering that the _rst column of data should be classi_ed as H0 (e.g. 0) and the second column of data should be classi_ed as H1 (e.g. 1). Display these scores to the screen. Hint: It may help to use logic operators (e.g. `==' or `~=').

```
error1 = 0
for i = 1:length(decision1)
   if decision1(i) ~= 0
      error1 = error1 + 1
   end
end

derror1 = error1/length(decision1) = 0.166
error2 = 0

for i = 1:length(decision2)
   if decision2(i) ~= 1
      error2 = error2 + 1
   end
end

derror2 = error2/length(decision2) = 0.157
```

2e) Use the decision errors from each column to compute the probability of error (e.g. P(E)). Display this error to the screen. Hint: assume equal priors.

PE = (derror1+derror2)/2 = 0.1615