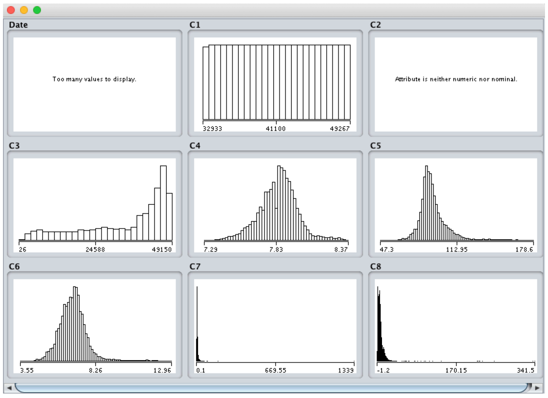
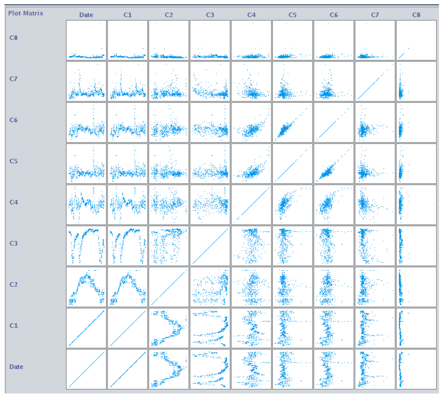
Yangyang Xu 43442401

Prac 1

Q2

There are 9 columns in the mystery2017 dataset. The first column is date time, so name it as “Date”. The rest of columns will be named from left to right, as C1, C2, C3, C4, C5, C6, C7, and C8.

After import the dataset to Weka. I noticed the data file records the information from 1/01/2015 to 31/12/2015, a whole year. The frequency of records is regular, one piece of data per 30 minutes. C2 can’t be displayed, due to its data format is not numerical or nominal. “C1” and “C3" have big-number scales than all other 7 histogram chats, and similar maximum number(C1 Max number: 49267; C3 Max number: 49150).

On the histogram graph. Each data category of “C1” has the almost same frequency. For “C3”, the frequency is similar before 37813, after this point, there is a huge increased frequency. For “C4”, most of the centre of data categories have higher frequency than the data at the left side and right side, and trend to decrease from the centre to the left and right. “C5” and “C6” have the similar histogram graphs; their right-side data categories have more distribution, and the data graphs trend to decrease from the centre of concentration to the left and right sides. “C7” and “C8” also have the similar histogram graphs, most of data concentrate at the top of left, and the frequency of data categories from left to right, have decreasing trend; on the right side of concentration, the data is scattered at different data groups.

On the visualisation graph. “C1” increases along with the “Date”, and the graph has a strong linear association.“Date” Vs. “C2”, and “C1” Vs. “C2”, their graph have moderate quadratic relationships, and there are increase and decrease trends, When “C8” compares to other 8 data, from the “Date” to “C4”, the “C8” has high static association with others. From “C5” to “C7”, the distribution concentrates at the left. “C5” Vs. “C6”, it has the moderate positive linear association, the trend of “C6” is increasing. The scatter graphs of “Date” Vs. all other data columns separately, have strong similar associated distributions, when these 8 scatter plots compare with the other 8 scatter graphs of “C1” vs. “Date” to “C8” separately(except “C1” itself). The scatter graphs of “C2” Vs. all other data columns individually, have moderate similar associated distributions, when these 8 scatter plots compare with the other 8 latter graphs of “C3” Vs. all other data columns. When the scatter plot x-axis are “C4”, “C5”, “C6”, “C7” and “C8”, the most concentrated regions are same as the the trends which are on histogram graphs of the “C4”, “C5”, “C7” and “C8”.

Q6

```matlab

function out = prac1_1(in,n)
    out = [];
    temp = [];
    p = size(in);
    p = p(2);
    form1 = fliplr(in);
    ct=0;

    for x = 1:p
        temp = [temp,form1(x)];
        ct = ct+1;
        if ct == n || x == p
          out = [out,fliplr(temp)];
          ct = 0;
          temp = [];
        end
    end

end
```

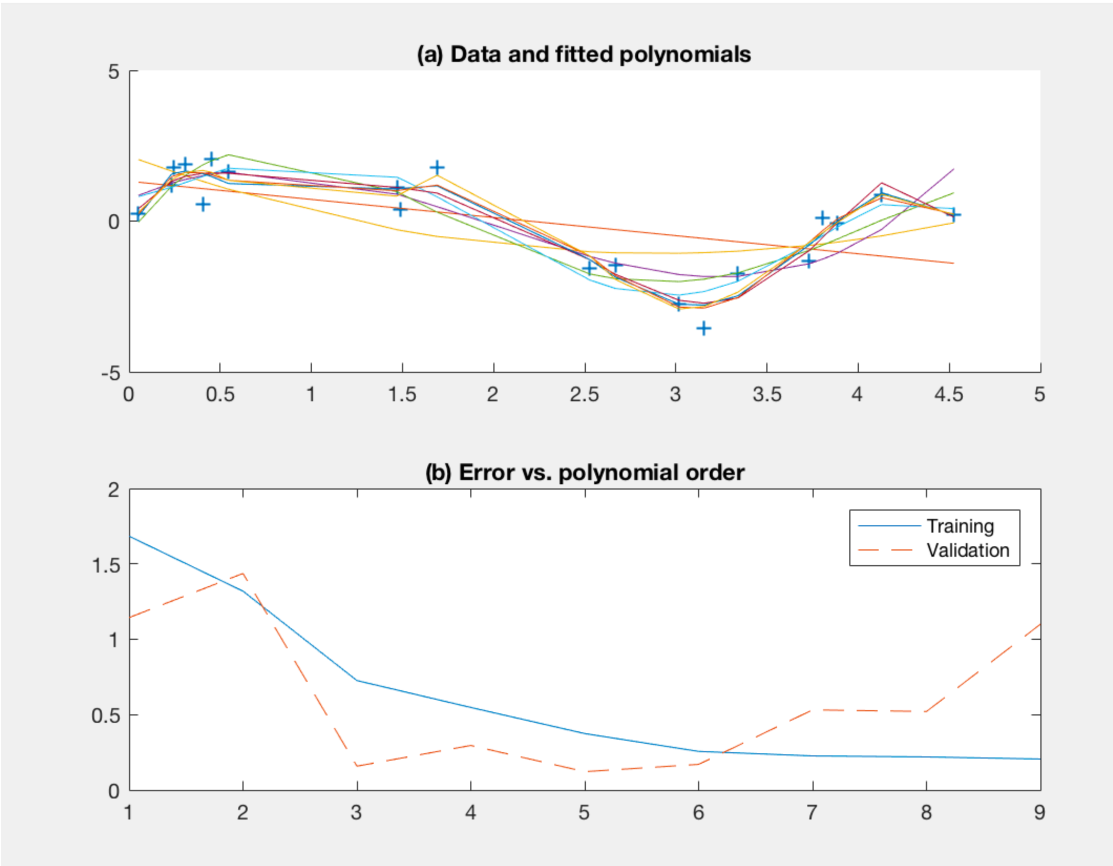
Command Window

```
>> out = prac1_1([1,2,3,4,5],1)

out =

     5     4     3     2     1

>> out = prac1_1([1,2,3,4,5],2)

out =

     4     5     2     3     1

>> out = prac1_1([1,2,3,4,5,6],2)

out =

     5     6     3     4     1     2

>> out = prac1_1([1,2,3,4,5,6],3)

out =

     4     5     6     1     2     3

>> out = prac1_1([1,2,3,4,5,6],4)

out =

     3     4     5     6     1     2

>> out = prac1_1([19,34,59,2,45,83,20],5)

out =

    59     2    45    83    20    19    34
```

Prac 2

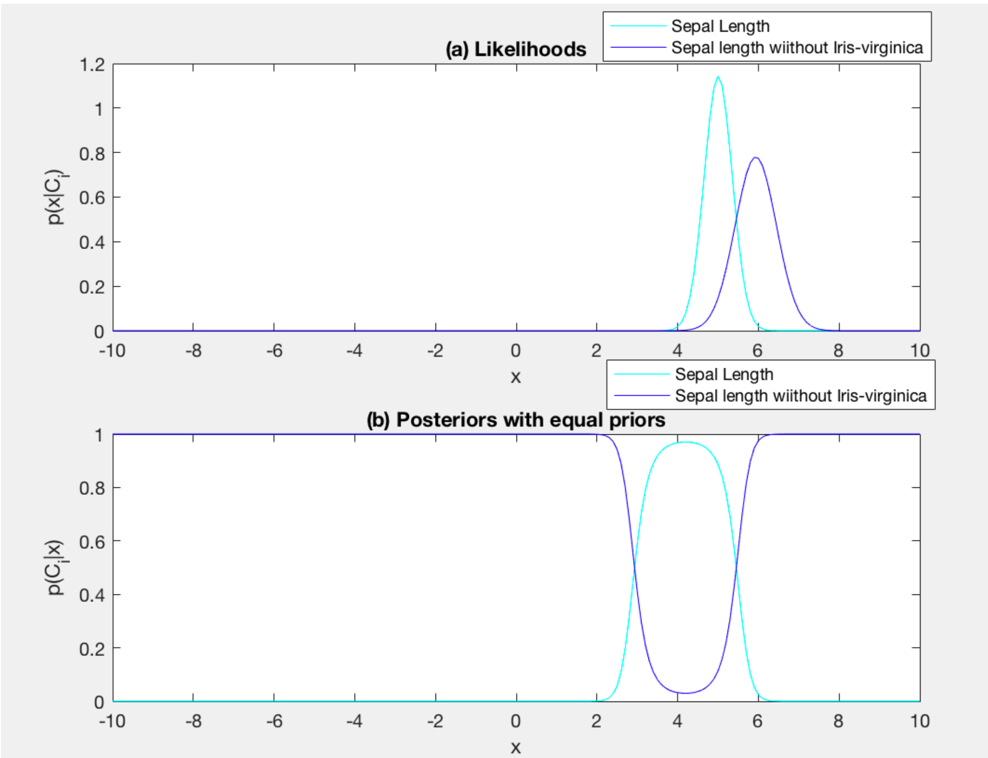
Q1

```
%% Q1
figure;
subplot(2,1,1);
Xt = rand(20,1)*5;
Xt = sort(Xt);
Yt = randn(20,1)+2*sin(1.5*Xt);
% Poly Degree from 1 -> 9
scatter(Xt,Yt,'+');
hold on;
sseT = zeros(9,1);%SSE for Traning
sseV = zeros(9,1);%SSE for Validation
denseInfo = zeros(size(Xt,1),9);%fit error

for j = 1:9
    fitpoly = polyfit(Xt,Yt,j);
    y1 = polyval(fitpoly,Xt);
    sseT(j) = mean((Yt-y1).^2);
    plot(Xt,y1);
end
axis([0 5 -5 5]);
xticks(0:0.5:5);
yticks([-5 0 5]);
title("(a) Data and fitted polynomials");

subplot(2,1,2);
% New validation sets
Xv = rand(20,1)*5;
Xv = sort(Xv);
Yv = 2*sin(1.5*Xv);
for k = 1:9
    fitpoly = polyfit(Xt,Yt,k);
    y2 = polyval(fitpoly,Xv);
    sseV(k) = mean((Yv-y2).^2);
end

plot((1:9),sseT,'-',(1:9),sseV,'--');
legend('Training','Validation')
title("(b) Error vs. polynomial order");
```



Q4

```
f = importdata('Iris.mat');
feature1 = f.VarName1(1:50);
feature3 = f.VarName1(51:100);
x = (-10:0.1:10);%Sample

%% Maximum likelihoods -> C1(sepal length)
ml1 = mle(feature1,'distribution','norm');

%% Maximum likelihoods -> C2(no include Iris-virginica)
ml2 = mle(feature3,'distribution','norm');

%% p(x|C1) -> C1(sepal length)
p1 = normpdf(x,ml1(1),ml1(2));

%% p(x|C2) -> C2(no include Iris-virginica)
p2 = normpdf(x,ml2(1),ml2(2));

%% Class posteriors -> C1(sepal length)
P1 = (p1.*0.5)./(p1.*0.5+p2.*0.5);

%% Class posteriors -> C2(no include Iris-virginica)
P2 = (p2.*0.5)./(p1.*0.5+p2.*0.5);

%% Draw Likelihood functions -> C1 & C2
figure;
subplot(2,1,1);
plot(x,p1,'c',x,p2,'b-');
xlabel('x')
ylabel('p(x|C_i)')
legend('Sepal Length','Sepal length wiithout Iris-virginica')
title('(a) Likelihoods');

%% Draw Class Posteriors functions -> C1 & C2
subplot(2,1,2);
plot(x,P1,'c',x,P2,'b-');
xlabel('x')
ylabel('p(C_i|x)')
legend('Sepal Length','Sepal length wiithout Iris-virginica')
title('(b) Posteriors with equal priors')
```



Prac 3

Q1

Class1: Column #9 = 1
Class2: Column #9 = 0

```
clear;
data = importdata('pimaindiansdiabetes.mat');
iris = importdata('iris.mat');
sample = data(1:500,1:8); %sample
%Mean
sample_mean = mean(sample);
%Covariance
sample_cov = cov(sample);
%Classification Error:loss function
Class1 = sample(data(1:500,9)==1,:);
Class2 = sample(data(1:500,9)==0,:);
```

```
%% Q1: QDA - Sample
% p(x)
sp = mvnpdf(sample,sample_mean,sample_cov);
% P(X|Class1)
sP1 = mvnpdf(sample,mean(Class1),cov(Class1));
% P(X|Class2)
sP2 = mvnpdf(sample,mean(Class2),cov(Class2));
%Likelihood Density P(C1|x)
sP_density1 = sP1.*(182/500)./sp;
%Likelihood Density P(C2|x)
sP_density2 = sP2.*(318/500)./sp;
%Error: compare to class 1
specifier_sClass1 = sP_density1 ./ (sP_density1+sP_density2);
%Get exact specifier for test data
specifier_s = round(specifier_sClass1);
%Find the # of error data from test
serrorSet = specifier_s(data(1:500,9)~=specifier_s(:));
num_error = size(serrorSet);
%Classification Error
classificationError = num_error(1)/500;
```

```
%% Q1: QDA - Test
sample = data(1:500,1:8); %sample
test = data(501:768,1:8);
%Mean
sample_mean = mean(sample);
%Covariance
sample_cov = cov(sample);
%Classification Error:loss function
Class1 = sample(data(1:500,9)==1,:);
Class2 = sample(data(1:500,9)==0,:);
% p(x)
p = mvnpdf(test,sample_mean,sample_cov);
% P(X|Class1)
P1 = mvnpdf(test,mean(Class1),cov(Class1));
% P(X|Class2)
P2 = mvnpdf(test,mean(Class2),cov(Class2));
%Likelihood Density P(C1|x)
P_density1 = P1.*(182/500)./p;
%Likelihood Density P(C2|x)
P_density2 = P2.*(318/500)./p;
%Error: compare to class 1
specifier_Class1 = P_density1 ./ (P_density1+P_density2);
%Get exact specifier for test data
specifier = round(specifier_Class1);
%Find the # of error data from test
errorSet = specifier(data(501:768,9)~=specifier(:));
num_errorTest = size(errorSet);
%Trainning Error
trainingError = num_errorTest(1)/268;
```

(a) the training classification error: 0.2460
(b) the test classification error: 0.2201
(c) the model parameters:

MEAN VECTORS

```
>> mean(Class1)

ans =

    4.7802  140.4890   69.7253   21.7143  102.4286   35.3231    0.5672   36.2692
```

```
>> mean(Class2)

ans =

    3.2516  110.5063   68.1981   19.9591   68.1321   30.0654    0.4397   31.2830
```

COVARIANCE MATRICES

```
>> cov(Class1)

ans =

   1.0e+04 *

    0.0014    0.0008    0.0011   -0.0005   -0.0029   -0.0004   -0.0000    0.0018
    0.0008    0.0968    0.0040    0.0003    0.1135    0.0006    0.0001    0.0060
    0.0011    0.0040    0.0494    0.0078    0.0270    0.0007   -0.0000    0.0060
   -0.0005    0.0003    0.0078    0.0297    0.1239    0.0035    0.0002   -0.0031
   -0.0029    0.1135    0.0270    0.1239    1.9763    0.0031    0.0005    0.0162
   -0.0004    0.0006    0.0007    0.0035    0.0031    0.0056    0.0000   -0.0017
   -0.0000    0.0001   -0.0000    0.0002    0.0005    0.0000    0.0000   -0.0000
    0.0018    0.0060    0.0060   -0.0031    0.0162   -0.0017   -0.0000    0.0115
```

```
>> cov(Class2)

ans =

   1.0e+04 *

    0.0009    0.0011    0.0006   -0.0004   -0.0033    0.0001   -0.0000    0.0019
    0.0011    0.0776    0.0072   -0.0002    0.1061    0.0038    0.0001    0.0099
    0.0006    0.0072    0.0312    0.0050    0.0142    0.0057    0.0000    0.0038
   -0.0004   -0.0002    0.0050    0.0218    0.0613    0.0055    0.0000   -0.0025
   -0.0033    0.1061    0.0142    0.0613    1.0863    0.0242    0.0009   -0.0166
    0.0001    0.0038    0.0057    0.0055    0.0242    0.0064    0.0000    0.0007
   -0.0000    0.0001    0.0000    0.0000    0.0009    0.0000    0.0000    0.0000
    0.0019    0.0099    0.0038   -0.0025   -0.0166    0.0007    0.0000    0.0138
```

Q2

Used a full but common/shared covariance matrix for each class.
Class1: Column #9 = 1
Class2: Column #9 = 0

(a) the training classification error: 0.2380
(b) the test classification error: 0.2015
(c) the model parameters:

MEAN VECTORS

```
>> mean(Class1)

ans =

    4.7802  140.4890   69.7253   21.7143  102.4286   35.3231    0.5672   36.2692
```

```
>> mean(Class2)

ans =

    3.2516  110.5063   68.1981   19.9591   68.1321   30.0654    0.4397   31.2830
```

COVARIANCE MATRICES(SHARED COVARIANCE)

Formula: $S = \sum_i p(C_i) S_i$

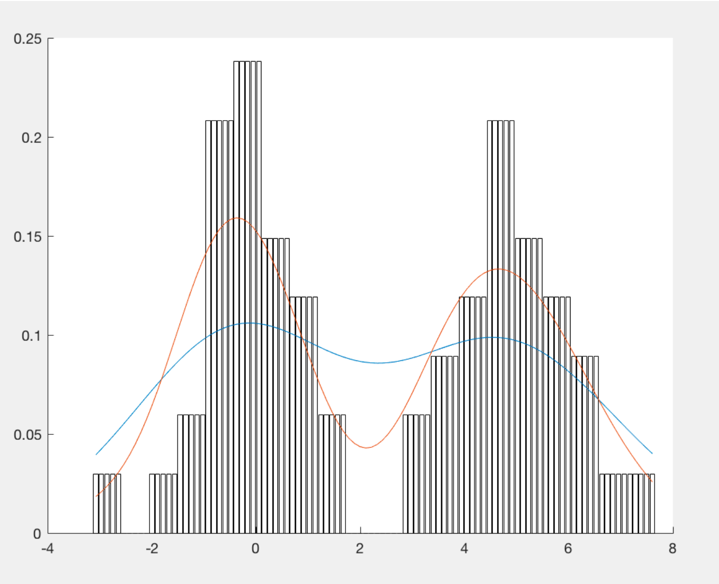
```
>> sample_cov

sample_cov =

   1.0e+04 *

    0.0011    0.0010    0.0008   -0.0004   -0.0032   -0.0001   -0.0000    0.0019
    0.0010    0.0846    0.0060   -0.0000    0.1088    0.0026    0.0001    0.0085
    0.0008    0.0060    0.0378    0.0061    0.0189    0.0039   -0.0000    0.0046
   -0.0004   -0.0000    0.0061    0.0247    0.0841    0.0048    0.0001   -0.0027
   -0.0032    0.1088    0.0189    0.0841    1.4103    0.0165    0.0008   -0.0047
   -0.0001    0.0026    0.0039    0.0048    0.0165    0.0061    0.0000   -0.0002
   -0.0000    0.0001   -0.0000    0.0001    0.0008    0.0000    0.0000    0.0000
    0.0019    0.0085    0.0046   -0.0027   -0.0047   -0.0002    0.0000    0.0130
```

Q5

```

x = [randn(30,1); 5+randn(30,1)];
test_pts = linspace(min(x),max(x),100);
M = 0.5.*normpdf(test_pts,0)+0.5.*normpdf(test_pts,5);
% Histogram -> Density Probability based 20_bin bin_width
[N,center]= hist(x,20);
H = histogram(x,20);
H_head = H.BinLimits(1);
H_end = H.BinLimits(2);
H_density = (H.Values/60)/H.BinWidth;

% Histogram -> Scale new #100 bin_width based on the center density
new_intervals =  linspace(0.6,20.4,100); %intervals got from H0 = bar(H_density,'w')
H1 = [];
for i = new_intervals
    H1 = [H1,(H_density(round(i)))];
end
figure;


[f1,x1,b] = ksdensity(x,test_pts);%K1
[f2,x2] = ksdensity(x,test_pts,'Bandwidth',b/2);%K2

hold on;
bar(x1,H1,'w');
hold on;
plot(x1,f1)
hold on;
plot(x2,f2)

```

```

%KL divergences: M -> H1
size(H1(H1==0))
H1(H1==0) = 0.000000001;
Process = M.*log(M./H1);
M_H1 = sum(Process(isfinite(Process)));
%KL divergences: M -> K1
M_K1 = sum(M.*log(M./f1));
%KL divergences: M -> K2
M_K2 = sum(M.*log(M./f2));

```

Because when I set the number of 60 data into the histogram with 20 bins, not all of intervals can be full filled. And when I scaled the 20 bins into 100 bins, the empty intervals were left and denoted as 0 in H1 density data matrix. But log function of the KL-divergence can only allow positive numbers, thus, I set “0” to “0.000000001”, which can approach to 0.



H1: White
K1: Blue
K2: Red

M_H1	5.8237
M_K1	2.6101
M_K2	0.9753