

Math 444: Project 2

Levent Batakci

February 28, 2021

The work in this project focuses on using Linear Discriminant Analysis as a means to analyze how well suited data sets are to be clustered. All of the mentioned implementations are coded in MATLAB and can be found in my [public GitHub Repository](#).

Algorithm Background

A natural question that arises in data science is whether or not a given data set lends itself to clustering. That is, do the recorded attributes provide some kind of insight into a meaningful grouping of the data? When provided with annotated data - that is, data where each point is labeled as being in some cluster - we can approach this problem through Linear Discriminant Analysis (LDA).

The key concept is that if the data attributes provide insight on the labeling, we should be able to "look" at the data in a way that makes clusters seem compact within themselves but far from one another.

For instance, consider a 3D scatter plot of data with clear clusters. As the view is rotated, the data is essentially being projected onto different planes to create the visual. If we're smart about how we rotate the view, we can make the clustering much more apparent. This is an intuitive way to understand the goal of LDA.

Formally, we should be able to find separating directions to project the data on. We want to choose these directions in a way so that the scattering within the clusters is low and the scattering between clusters is high. This will produce a visual effect of tightly bound clusters that are far from one another.

When the data is not centered, we essentially waste 1 separating direction because we have to consider the data relative to the origin. For this reason, we generally find it wise to center the data. Given k clusters, we compute $k - 1$ separating directions. The separating directions q are computed in the code by the function $LDA(X, I)$.

LDA Showcase

Before analyzing real data sets, we want to showcase the LDA algorithm. To do so, we present the results of running the LDA algorithm on generated data with 3 distinct clusters. Each cluster consists of 200 points $x^{(j)} \in \mathbf{R}^3$ generated by a Gaussian distribution. The clusters all have covariance equal to the identity and are centered at $(-3, -3, -3)$, $(2, 0, 0)$, and $(4, 4, -6)$.

Before computing the separating directions, we centered the data. As discussed in the previous section, we did this in order to avoid wasting a separating direction. The LDA algorithm returned two separating directions for the centered data. We projected the data onto these two directions, and the result can be seen in **Figure 1** on the next page.

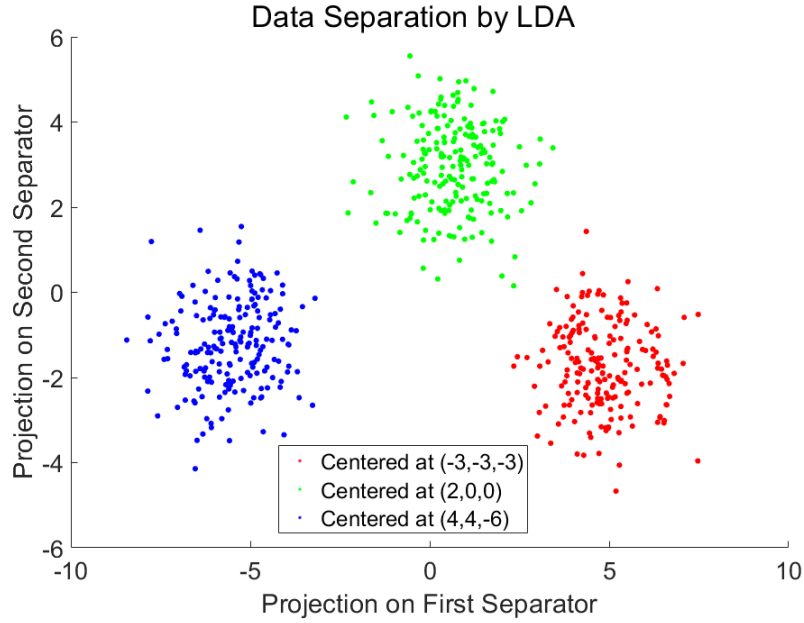


Figure 1: LDA Results for Generated Data

The LDA performed phenomenally on this test data. The three clusters, which are color-coded, are visually separated beyond any reasonable doubt. Each cluster is somewhat tightly packed while being separated from each of the others. It makes sense that LDA did well with this data - the data attributes reflect the Gaussian from which a given point originates. In the following sections, we will explore whether the recorded attributes carry a non-trivial insight as to the real-world grouping.

Iris Data

The first real data set we analyzed was the [Iris data set](#). This data set consists of data from three different subspecies of Iris (flowers) - *Iris setosa*, *Iris virginica*, and *Iris versicolor*. We are interested to see if the data lends itself to a natural clustering which can distinguish between these three subspecies. The data is of the form

$$x^{(j)} = \begin{pmatrix} \text{petal length in cm} \\ \text{sepal width in cm} \\ \text{petal length in cm} \\ \text{petal width in cm} \end{pmatrix},$$

where $1 \leq j \leq 150$. The data is encoded as a matrix $X \in \mathbb{R}^{4 \times 150}$ and is loaded from the file *IrisData.mat*. The data came alongside an annotation matrix $I \in \mathbb{R}^{1 \times 150}$ denoting the subspecies to which each data point belongs.

As a first step, we chose to center the data. Then, we computed the two best separating directions by running the LDA algorithm on the data. To visualize the separation, we projected the data onto these two directions. The resulting graphic can be seen in **Figure 2** on the next page.

Overall, the LDA did a relatively good job of separating the clusters. It is immediately apparent that the Iris Setosa cluster (red) is significantly more distinguishable than the other two. The Iris Versicolor and Iris Virginica clusters are also distinguishable, but much less so.

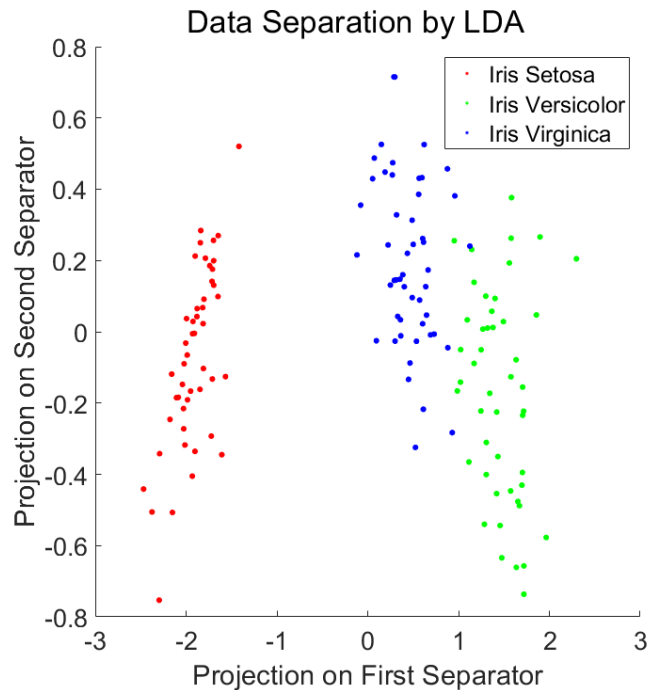


Figure 2: LDA Results for Iris Data

A natural question that arises is how important each of the separators is. Looking at **Figure 2**, it's apparent that there's a lot of overlap in the vertical component of the clusters. Indeed, most of the visual separation is clearly horizontal. We interpreted this as indicating that the first separator is essentially the only important one.

Another question that arises is how the LDA compares to the Principal Component Analysis (PCA). **Figure 3** on the right shows the first two principal components of the Iris Data. Immediately, one notices that the LDA and PCA results look very similar. Indeed, the LDA does seem to be akin to a mirrored and slightly rotated (see the red group) version of the PCA. The mirroring is unimportant - but the rotation is not. The rotation essentially makes the Versicolor and Virginica (green and blue) clusters significantly easier to distinguish.

Indeed, the LDA does a much better job of separating these two clusters. Furthermore, as stated before, these clusters separate well along the first separator and not the second. This is also somewhat true in the PCA - the clusters separate better along the first principal component than the second.

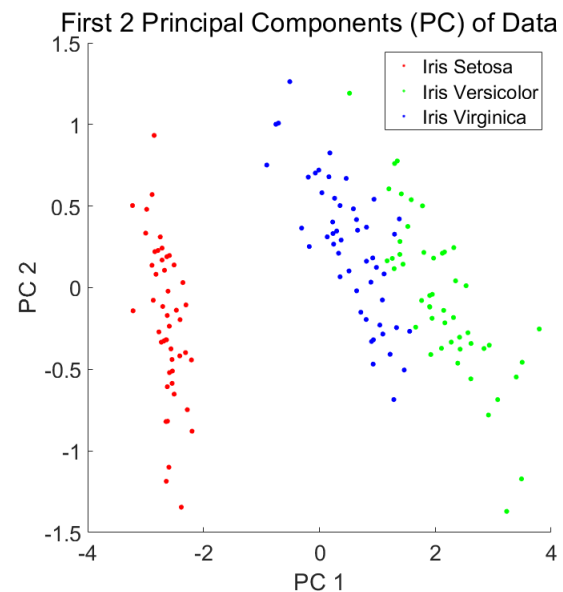


Figure 3: PCA of Iris Data

Overall, the LDA results can be viewed as an improvement of the results of the PCA. Given the projection of a new data point along the first separator, we can make relatively confident conclusions about which cluster it belongs to.

Breast Cancer Data

The next data we analyzed was the [Breast Cancer Wisconsin data set](#). Particularly, we are interested to see if the data attributes provide insight on a clustering of benign and malignant tumors. The data is of the form $x^{(j)} \in \mathbb{R}^{30}$ where $1 \leq j \leq 699$, and each of the components of data components is a integer between 1 and 10. So the data is stored in a matrix $X \in \mathbb{R}^{30 \times 699}$. The data is also accompanied by an annotation matrix I which places each data point into one of two clusters - benign and malignant.

Again, we first centered the data. Then, we computed the LDA. Since there are two clusters, we computed 1 separating direction. The results of the LDA are seen in **Figure 4** as a histogram.

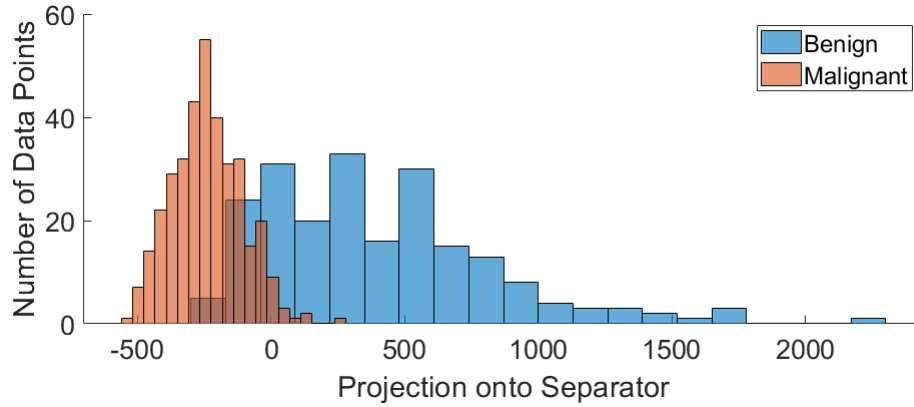


Figure 4: LDA results on Breast Cancer Data

We also performed a principal component analysis of the data. The results can be seen in **Figure 5**. Note that the absolute value of the first principal component is plotted.

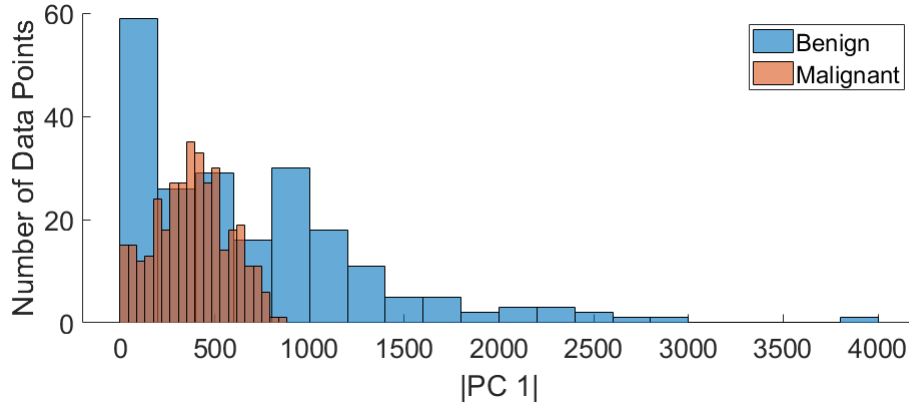


Figure 5: PCA results on Breast Cancer Data

There are some basic similarities and differences between the results of the LDA and the PCA. As a similarity, we notice that the benign tumors are significantly more spread out than the malignant ones in both the PCA and the LDA. Moreover, we notice that both have an overlap between the two. That being said, the overlap is much greater in the PCA. In fact, all of the malignant cluster is contained in the range of the benign cluster for the PCA, whereas this is not the case for the LDA.

Clearly, the LDA does a better job of distinguishing the two groups. That being said, there is still a significant overlap.

We chose to analyze the separating direction to determine if there were any unimportant attributes. We decided that an attribute would be deemed as "unimportant" if it was no more than 10 percent

of every separator. Since the separators are normalized to have magnitude 1, this meant that we eliminated attributes of magnitude less than or equal to $(0.1)^{\frac{1}{2}} \approx 0.3162$. We did this in MATLAB as follows:

```
%Remove insignificant components
perc = 0.1;
Q(abs(Q) <= perc^(1/2)) = 0;
Xc(Q(:,1)==0,:) = [];
```

Specifically, we ended up removing all but 3 attributes! Afterwards, since the data was significantly modified, we recomputed the separators. The result of the second LDA is shown in **Figure 6**.

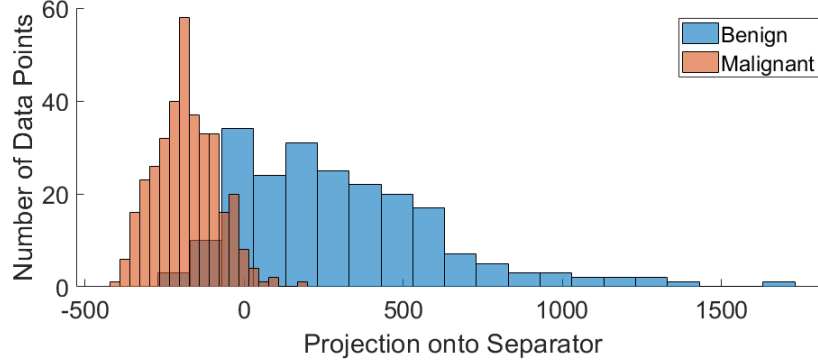


Figure 6: Caption

It's immediately apparent that the effect of removing these attributes was somewhat minimal. That being said, it can clearly be seen that there is in fact slightly less overlap between the two clusters. This indicates that removing the other attributes was marginally helpful. Overall, the LDA did a fair job of separating this data. With a context so important, a doctor's diagnosis would still be needed.

Wine Data

The next data set we analyzed consists of the [chemical analysis of wines](#) originating from three different cultivars. Specifically, the data is of the form

$$x^{(j)} = \begin{pmatrix} \text{Alcohol} \\ \text{Malic acid} \\ \text{ash} \\ \text{Alkalinity of ash} \\ \text{Magnesium} \\ \text{Total Phenols} \\ \text{Flavonoids} \\ \text{Nonflavonoid phenols} \\ \text{Proanthocyanins} \\ \text{Color intensity} \\ \text{Hue} \\ \text{OD280/OD315 of diluted wines} \\ \text{Proline} \end{pmatrix},$$

where the entries are concentrations/levels. The data consisted of 178 data points and was stored in a matrix $X \in \mathbb{R}^{13 \times 178}$. Furthermore, the data is accompanied by an annotation matrix $I \in \mathbb{R}^{178 \times 1}$ which denotes the cultivar from which each data point originated.

Again, we took the first step of centering the data. After that, we ran the LDA algorithm and computed the first two separators. The projections of the data onto these direction can be seen in **Figure 7**. Overall, the separation is clear but could be improved. Most of the issues occur where clusters meet.

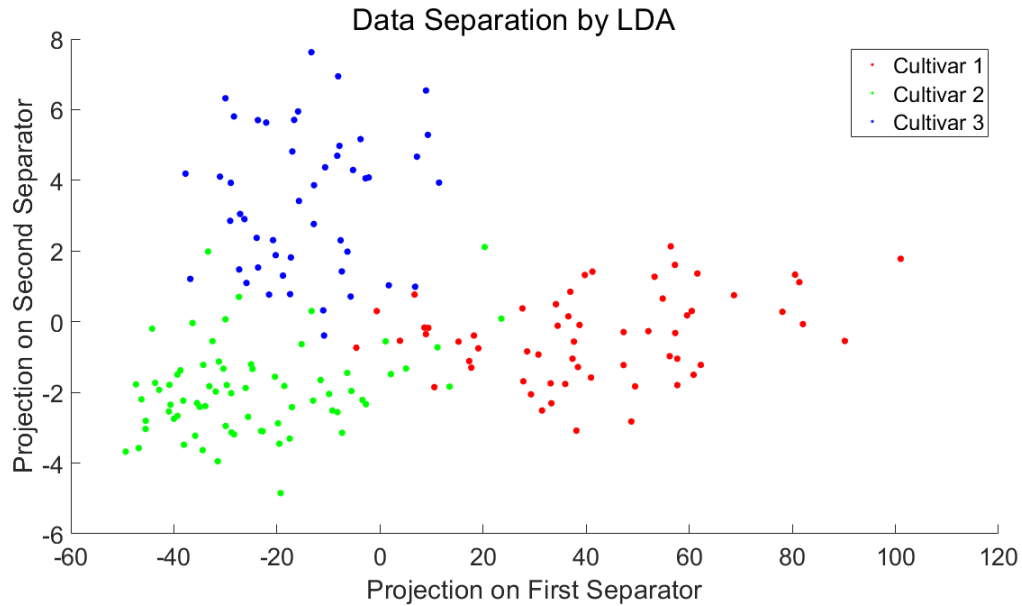


Figure 7: LDA Results on Wine Data

Again, we chose to analyze the separating direction to determine if there were any unimportant attributes. The removed attributes were Malic acid, ash, Magnesium, Total Phenols, Nonflavaonoid phenols, Proanthocyanin, Hue, and Proline. Afterwards, since the data was significantly modified, we recomputed the separators. The result of the second LDA is shown in **Figure 8**.

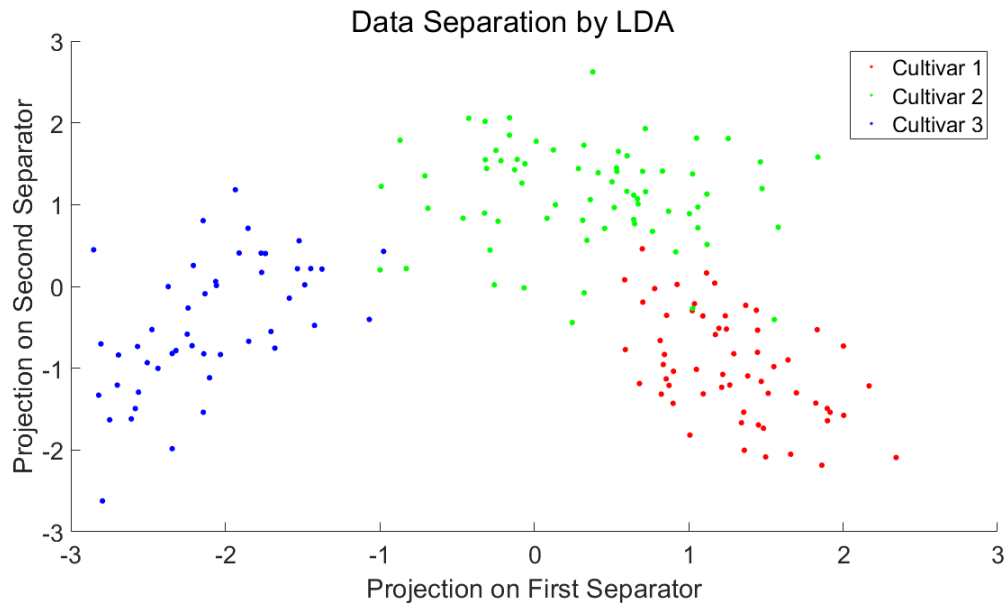


Figure 8: LDA Results on Reduced Wine Data

Immediately, it's apparent that the separators changed. Furthermore, the data actually seems to be better separated! While it may initially come as a surprise, this actually makes perfect sense. The inconsequential attributes act as noise, and so we get better accuracy by removing them.

There's still a bit of overlap between the cultivar 1 and 2 clusters, but the overall separation is very clear. The attributes of the Wine data provide insight into which cultivar the wines originate from, but some of the attributes are unnecessary and cause noise.

Forest Spectra Data

The last data set we analyzed was the forest spectra data. This data set is simulation of a helicopter-borne microwave scatterometer probing a forest, where the backscattering amplitude is recorded as a function of distance. The data is stored in a matrix $X \in \mathbb{R}^{50 \times 780}$. Furthermore, it is accompanied by an annotation I which places each data point into one of four group - Pine, Birch, Fir, and Shrub. Our analysis focused on whether the recorded backscattering amplitude provides insight as to the type of vegetation.

Again, we first centered the data. After that, we computed the LDA. This time, there were 3 separating directions since there are 4 clusters. The resulting pairwise scatterplots can be seen in **Figure 9**.

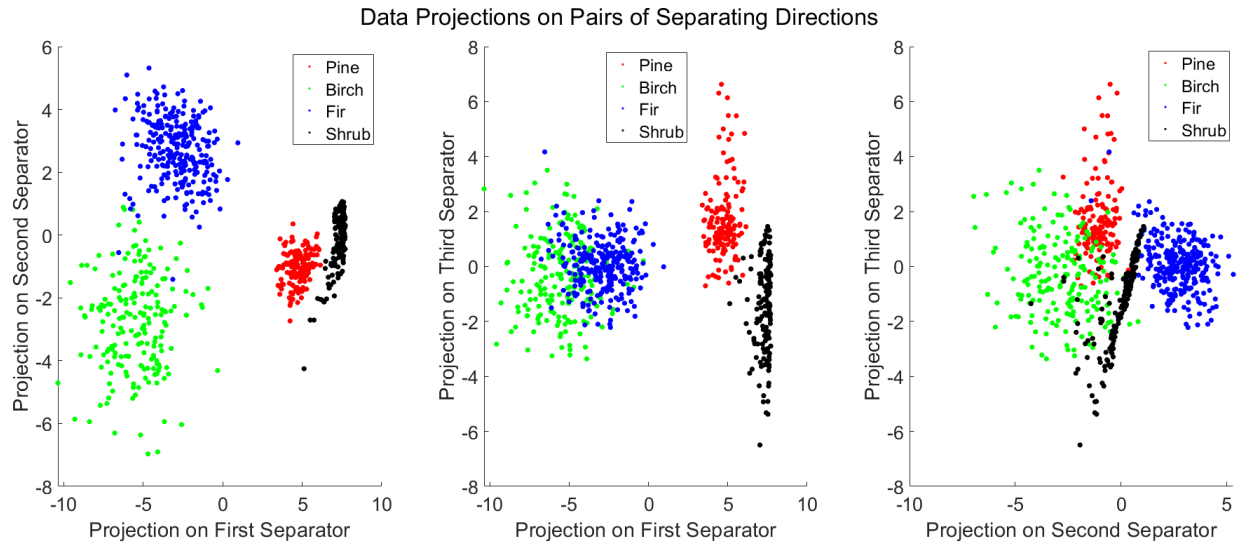


Figure 9: Pairwise LDA Results on Forest Data

It's immediately apparent that the first of these scatter plots, the one which displays the first separator versus the second, does a phenomenal job of separating the groups. There is very little overlap, and the clusters are all somewhat compact (some more than others). Looking at the other two scatter plots, there is unfortunately little to no insight to be gained.

Performing a deeper analysis, we see that the first separator does a good job of distinguishing between red (pine), black (shrub), and green/blue (birch/fir). The second separator does a good job of separating the green and blue groups (birch and fir). For this reason, the scatterplot of the projections on the first separator versus the second yield a visually compelling graphic in which all groups are separated.

Overall, the backscattering amplitude as a function of distance proves to be an effective way of identifying vegetation types.

Appendix

As stated before, please feel free to visit the [public GitHub Repository](#).

1 Code - LDA

```
function [Q, Sw_eps, Sb, eps] = LDA(X, I)
%LDA returns the leading directions to project the data

%Get the dimensions and labels
[n, p] = size(X);
labels = unique(I);
k = numel(labels);

P = zeros(1,k); %Number of elements per cluster
for i = 1:k
    g = labels(i);
    P(1,i) = nnz(I == g);
end

%Compute the group centers
C = getCenters(X, I, k);
c0 = sum(X, 2) / p; %Get the overall center

%Compute the within cluster scatter
Sw = zeros(n, n); %Within cluster scatter
for i = 1:k
    Xi = X(:, I==i) - C(:,i) * ones(1, nnz(I==i));
    Sw = Sw + Xi*Xi';
end

%Adjust Sw
E = eigs(Sw);
d1 = E(1);
tau = 10^-10;
eps = tau*(d1^2);
Sw_eps = Sw + eps.*eye(n,n);

%Compute Sb, the between-cluster scatter matrix
Sb = zeros(n,n);
for i = 1:k
    c_ = C(:,i)-c0;
    Sb = Sb + P(1,i) * c_*(c_');
end

%Compute the Cholesky factorization of Sw,eps
K = chol(Sw_eps);

[W, E] = eigs((K'\Sb)/K, k-1);
```



```

    Q = K \ W;
end

```

2 Code - Test Data

```

%Levent Batakci
%Test the stuff !

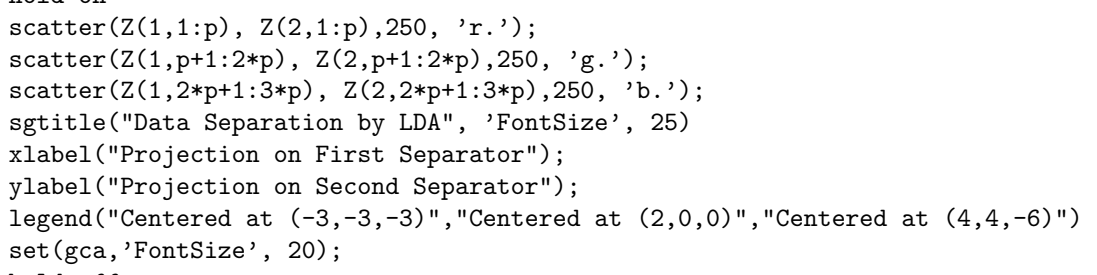
clear
clc
close all

%Parameters
k=3;

%Generate data
c1 = [-3 -3 -3]';
c2 = [2 0 0]';
c3 = [4 4 -6]';
Sigma = eye(3);
X = [mvnrnd(c1, Sigma, 200)' mvnrnd(c2, Sigma, 200)' mvnrnd(c3, Sigma, 200)'];
xc = sum(X,2)/600;
Xc = X - xc * ones(1, size(X,2));
I = [ones(1, 200) 2*ones(1, 200) 3*ones(1, 200)];
k = numel(unique(I));

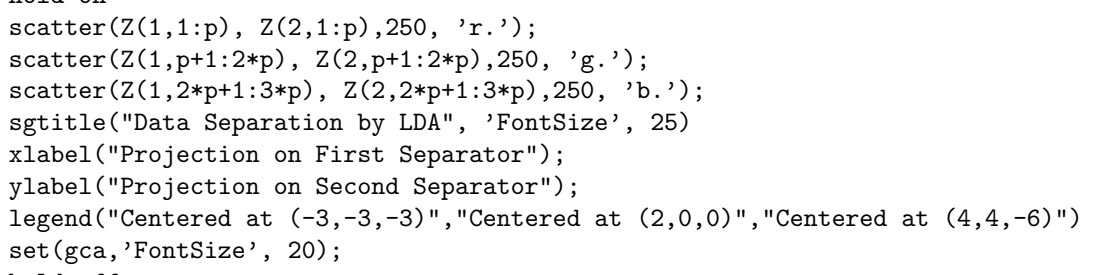
Q = LDA(Xc, I);
for i= 1:k-1
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

figure(1)
p = 200;
hold on
scatter(Z(1,1:p), Z(2,1:p),250, 'r.');
```



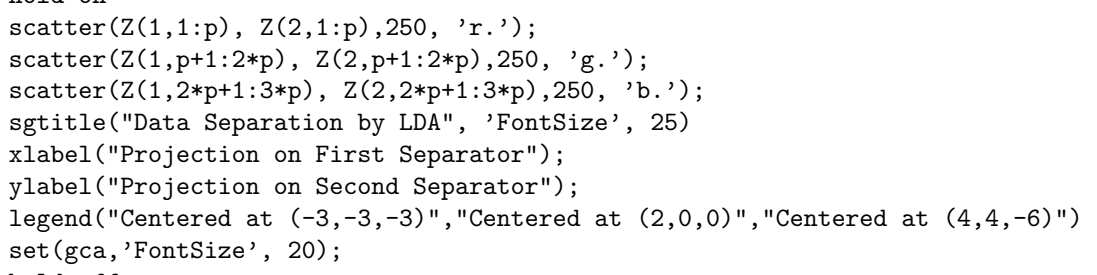
```

scatter(Z(1,p+1:2*p), Z(2,p+1:2*p),250, 'g.');
```



```

scatter(Z(1,2*p+1:3*p), Z(2,2*p+1:3*p),250, 'b.');
```



```

sgtitle("Data Separation by LDA", 'FontSize', 25)
xlabel("Projection on First Separator");
ylabel("Projection on Second Separator");
legend("Centered at (-3,-3,-3)","Centered at (2,0,0)","Centered at (4,4,-6)")
set(gca,'FontSize', 20);
hold off

%Testing purposes
figure(2)
scatter3(X(1,:), X(2,:), X(3,:));

```

3 Code - Iris Data

```
clear all

%Load Iris
load IrisData.mat
Xc = X - sum(X,2)/size(X,2) * ones(1, size(X,2)); %Center the data
I = [1 * ones(1, 50) 2 * ones(1, 50) 3 * ones(1, 50)];
k = numel(unique(I));
Q = LDA(Xc, I);
for i= 1:k-1
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

figure(1)
k1 = (I==1);
k2 = (I==2);
k3 = (I==3);
hold on
scatter(Z(1,k1), Z(2,k1),250, 'r. ');
scatter(Z(1,k2), Z(2,k2),250, 'g. ');
scatter(Z(1,k3), Z(2,k3),250, 'b. ');
sgtitle("Data Separation by LDA", 'FontSize', 25)
xlabel("Projection on First Separator");
ylabel("Projection on Second Separator");
legend("Iris Setosa","Iris Versicolor","Iris Virginica");
set(gca,'FontSize', 20);
hold off

%PCA!!
figure(2)
[U,V,D] = svd(Xc);
Z = [U(:,1) U(:,2)]' * Xc;
hold on
scatter(Z(1,k1), Z(2,k1),250, 'r. ');
scatter(Z(1,k2), Z(2,k2),250, 'g. ');
scatter(Z(1,k3), Z(2,k3),250, 'b. ');
sgtitle("First 2 Principal Components (PC) of Data", 'FontSize', 25)
xlabel("PC 1");
ylabel("PC 2");
legend("Iris Setosa","Iris Versicolor","Iris Virginica");
set(gca,'FontSize', 20);
hold off
```

4 Code - Breast Cancer Data

```
clear all

%Load Iris
```

```

load WisconsinBreastCancerData_Unpacked.mat

X = Data_WCD_Matrix;
I = I_Label;
k = numel(unique(I));

Xc = X - sum(X,2)/size(X,2) * ones(1, size(X,2)); %Center the data
Q = LDA(Xc, I);
for i= 1:k-1
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

figure(1)
hold on
benign = Z(:,I==1);
malignant = Z(:,I==2);
histogram(benign,'NumBins', 20)
% sgtitle("LDA", 'FontSize', 30);
set(gca,'FontSize', 20)
histogram(malignant,'NumBins', 20)
xlabel("Projection onto Separator")
ylabel("Number of Data Points")
set(gca,'FontSize', 20)
legend("Benign", "Malignant");
hold off

%PCA !
[U,D,V] = svd(Xc);
pc1 = U(:,1)' * Xc;

figure(2)
hold on
histogram(abs(pc1(:,I==1)), 'NumBins', 20)
histogram(abs(pc1(:,I==2)), 'NumBins', 20)
xlabel("|PC 1|", 'FontSize', 30)
ylabel("Number of Data Points")
set(gca,'FontSize', 20)
legend("Benign", "Malignant");
hold off

%Remove insignificant components
perc = 0.1;
Q(abs(Q) <= perc^(1/2)) = 0;
Q
Q(:,1)==0
Xc(Q(:,1)==0,:) = [];
Q = LDA(Xc, I);
for i= 1:size(Q,2)
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

```

```

figure(3)
hold on
benign = Z(:,I==1);
malignant = Z(:,I==2);
histogram(benign,'NumBins', 20)
% sgttitle("LDA", 'FontSize', 30);
histogram(malignant,'NumBins', 20)
xlabel("Projection onto Separator")
ylabel("Number of Data Points")
set(gca,'FontSize', 20)
legend("Benign", "Malignant");

```

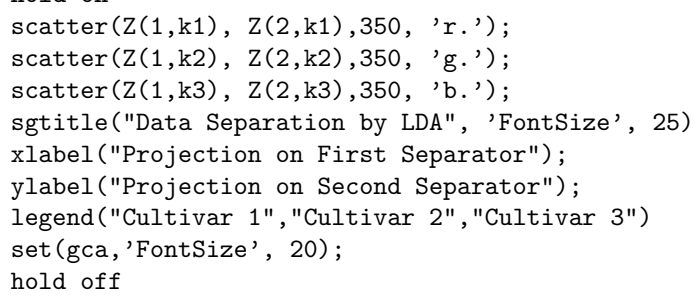
5 Code - Wine Data

```

clear all

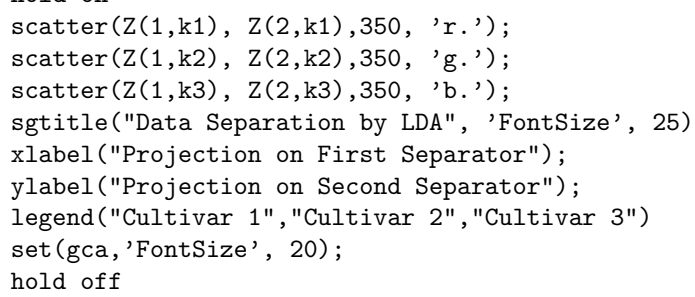
%Load Wine
load WineData.mat
k = numel(unique(I));
Xc = X - sum(X,2)/size(X,2) * ones(1,size(X,2));
Q = LDA(Xc, I);
for i= 1:k-1
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

figure(1)
k1 = (I'==1);
k2 = (I'==2);
k3 = (I'==3);
hold on
scatter(Z(1,k1), Z(2,k1),350, 'r.');
```



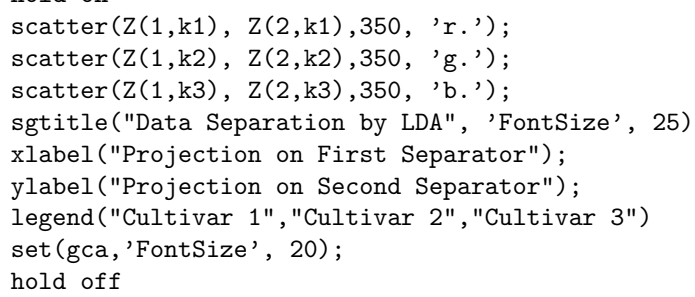
```

scatter(Z(1,k2), Z(2,k2),350, 'g.');
```



```

scatter(Z(1,k3), Z(2,k3),350, 'b.');
```



```

sgttitle("Data Separation by LDA", 'FontSize', 25)
xlabel("Projection on First Separator");
ylabel("Projection on Second Separator");
legend("Cultivar 1","Cultivar 2","Cultivar 3")
set(gca,'FontSize', 20);
hold off

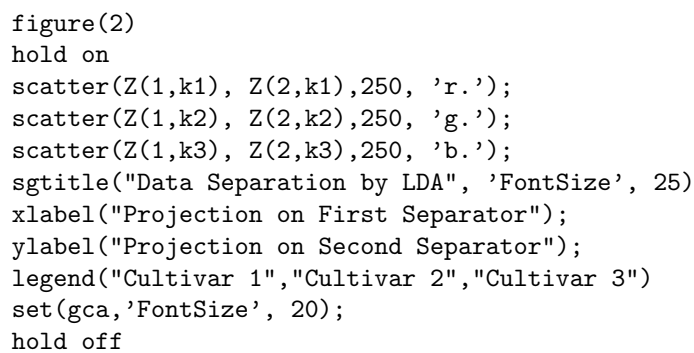
%Remove insignificant components
perc = 0.1;
Q(abs(Q) <= perc^(1/2)) = 0;
Q(:,1)==0 & Q(:,2)==0
Xc(Q(:,1)==0 & Q(:,2)==0,:) = [];
Q = LDA(Xc, I);
for i= 1:size(Q,2)
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end

```

```

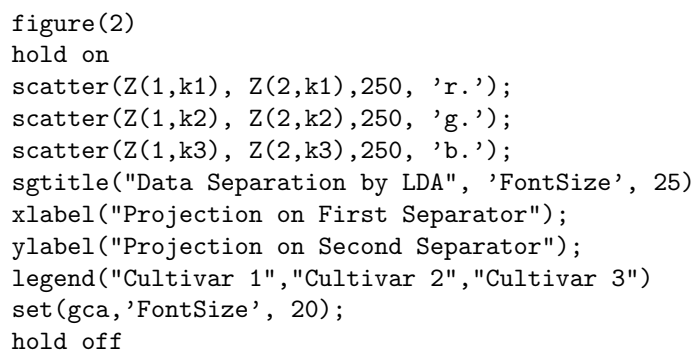
Z = Q' * Xc;

figure(2)
hold on
scatter(Z(1,k1), Z(2,k1),250, 'r.');
```



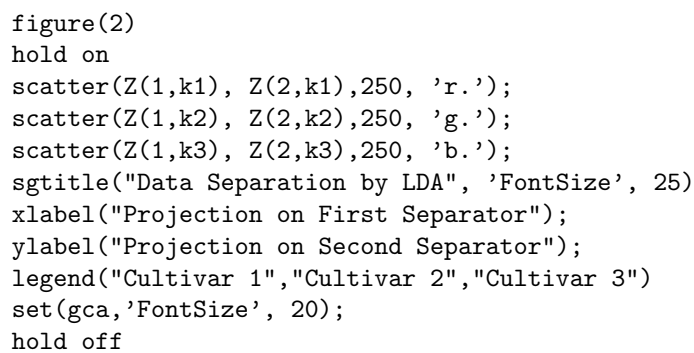
```

scatter(Z(1,k2), Z(2,k2),250, 'g.');
```



```

scatter(Z(1,k3), Z(2,k3),250, 'b.');
```



```

sgtitle("Data Separation by LDA", 'FontSize', 25)
xlabel("Projection on First Separator");
ylabel("Projection on Second Separator");
legend("Cultivar 1","Cultivar 2","Cultivar 3")
set(gca,'FontSize', 20);
hold off
```

6 Code - Forest Data

```

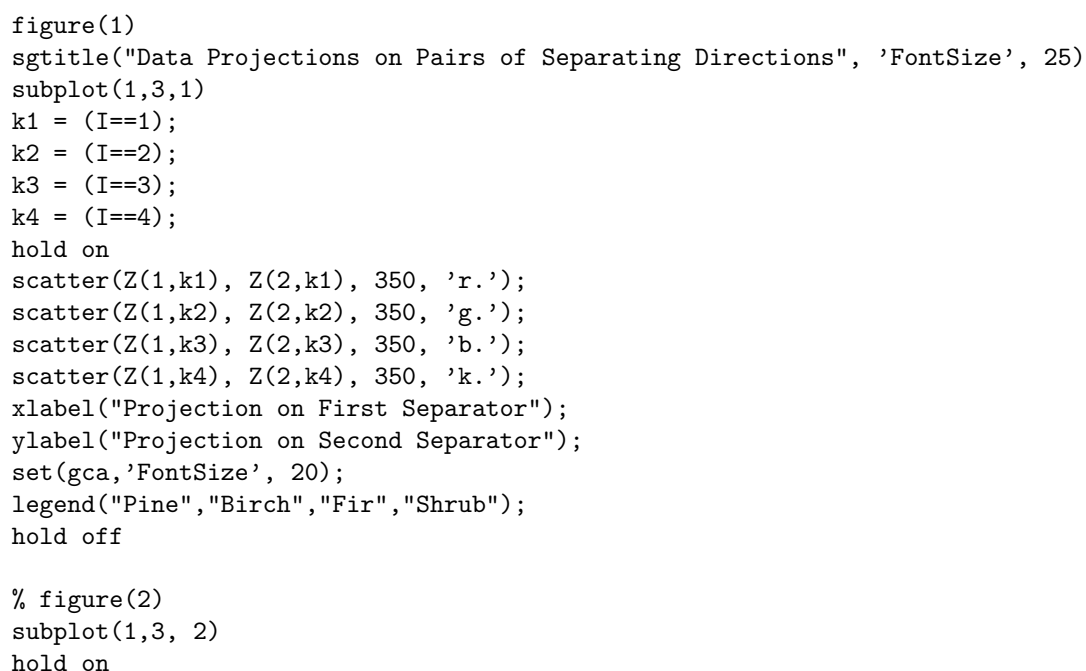
clear all

%Load Forest
load ForestSpectra.mat

I = Itype;
k = numel(unique(I));

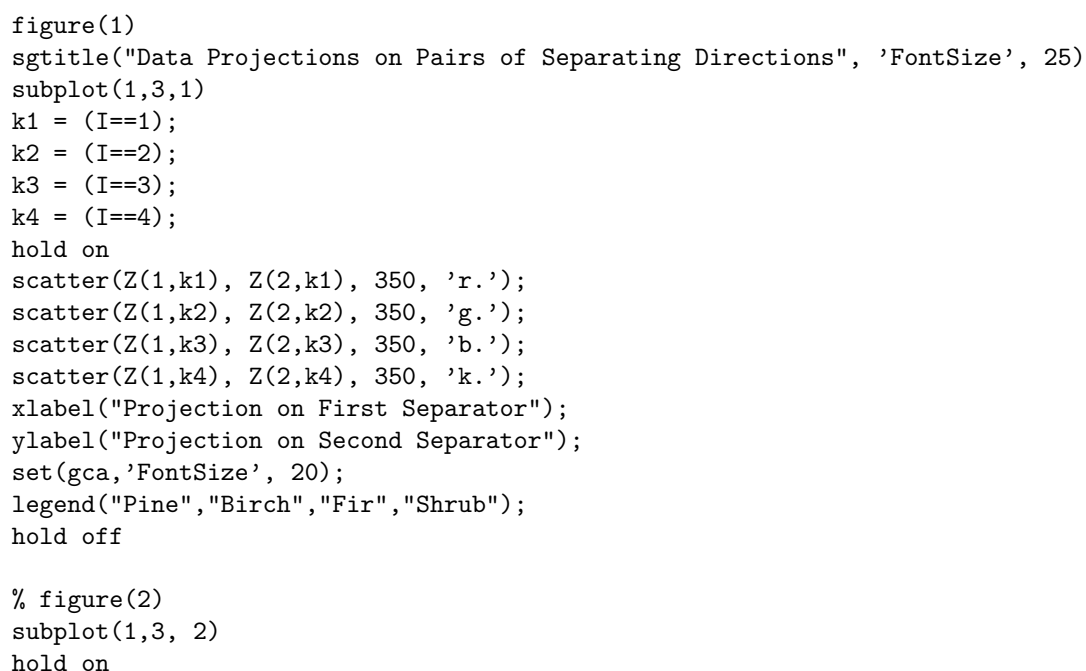
Xc = X - sum(X,2)/size(X,2) * ones(1,size(X,2)); %Center the data
Q = LDA(Xc, I);
for i= 1:k-1
    Q(:,i) = Q(:,i) / norm(Q(:,i));
end
Z = Q' * Xc;

figure(1)
sgtitle("Data Projections on Pairs of Separating Directions", 'FontSize', 25)
subplot(1,3,1)
k1 = (I==1);
k2 = (I==2);
k3 = (I==3);
k4 = (I==4);
hold on
scatter(Z(1,k1), Z(2,k1), 350, 'r.');
```



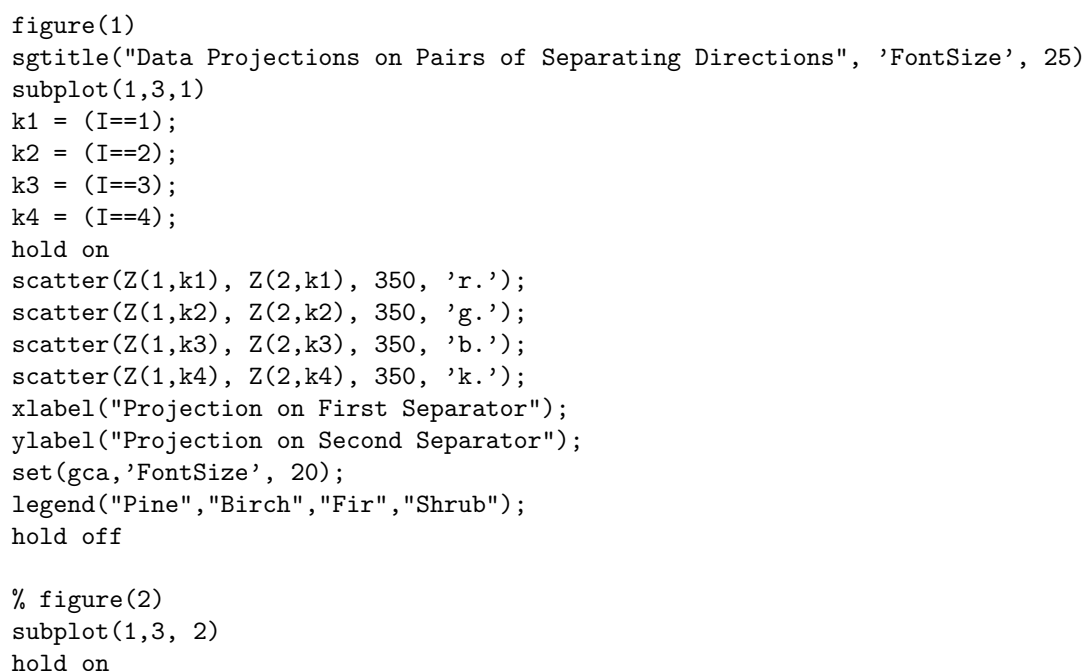
```

scatter(Z(1,k2), Z(2,k2), 350, 'g.');
```



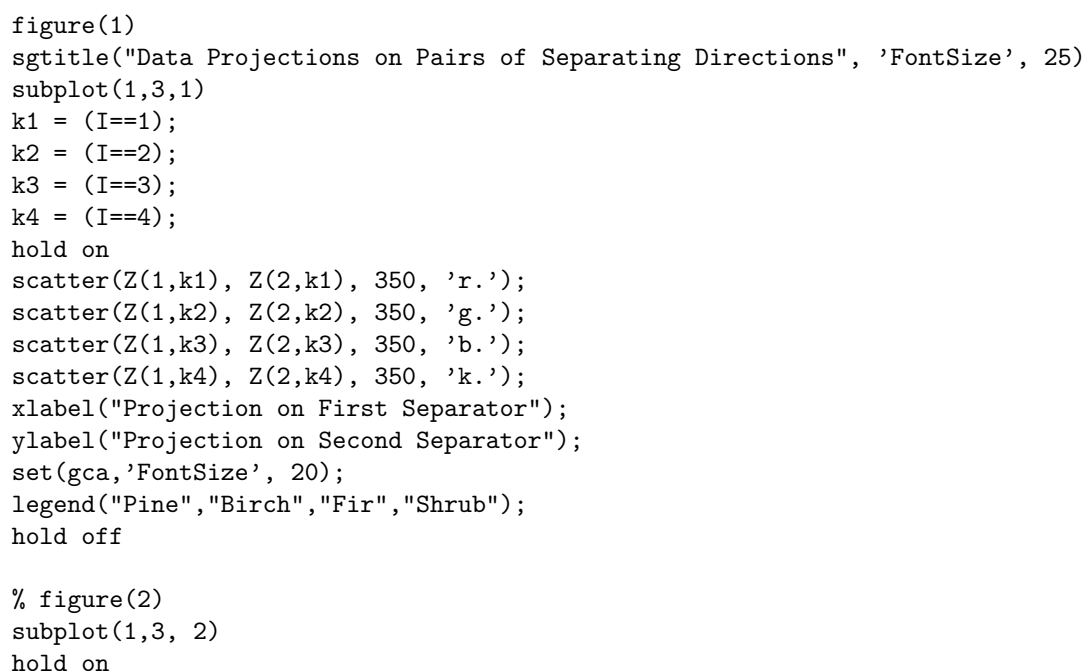
```

scatter(Z(1,k3), Z(2,k3), 350, 'b.');
```



```

scatter(Z(1,k4), Z(2,k4), 350, 'k.');
```



```

xlabel("Projection on First Separator");
ylabel("Projection on Second Separator");
set(gca,'FontSize', 20);
legend("Pine","Birch","Fir","Shrub");
hold off

% figure(2)
subplot(1,3, 2)
hold on
```

```

scatter(Z(1,k1), Z(3,k1), 350, 'r. ');
scatter(Z(1,k2), Z(3,k2), 350, 'g. ');
scatter(Z(1,k3), Z(3,k3), 350, 'b. ');
scatter(Z(1,k4), Z(3,k4), 350, 'k. ');
xlabel("Projection on First Separator");
ylabel("Projection on Third Separator");
set(gca,'FontSize', 20);
legend("Pine","Birch","Fir","Shrub");
hold off
%
%
%
% figure(3)
subplot(1,3, 3)
hold on
scatter(Z(2,k1), Z(3,k1), 350, 'r. ');
scatter(Z(2,k2), Z(3,k2), 350, 'g. ');
scatter(Z(2,k3), Z(3,k3), 350, 'b. ');
scatter(Z(2,k4), Z(3,k4), 350, 'k. ');
xlabel("Projection on Second Separator");
ylabel("Projection on Third Separator");
set(gca,'FontSize', 20);
legend("Pine","Birch","Fir","Shrub");
hold off

```