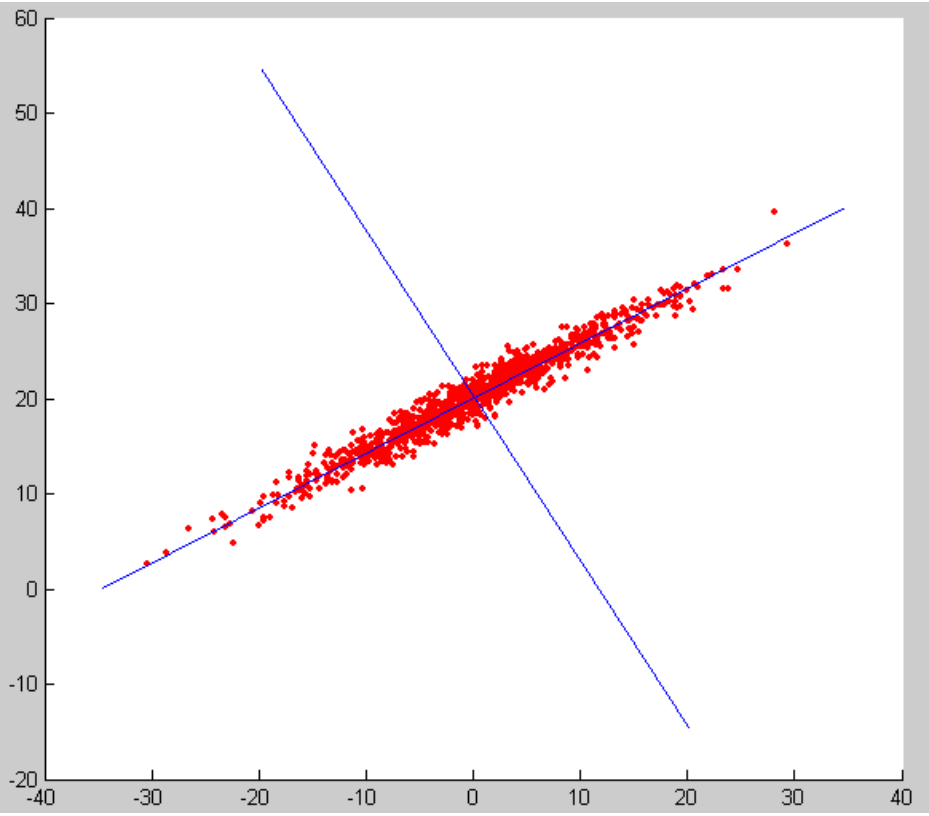
Παράδειγμα εφαρμογής της μεθόδου PCA (Principal Component Analysis) στο MATLAB

```
% Δημιουργία συνόλου δεδομένων (1000 τυχαία σημεία με αναλογία κατανομής 1/10
% στις δύο διαστάσεις , περιστροφή του συνόλου κατά 30 μοίρες και μετατόπιση
% πάνω κατά 20 μονάδες).
a=10*randn(1000,1);
b=randn(1000,1);
R=[cosd(30) -sind(30) ; sind(30) cosd(30)];
G=R*[a,b]';
x=G(1,:);
y=G(2,:)+20;
hold on
plot(x,y,'r.')

% Εφαρμογή της μεθόδου PCA (επιστροφή των συντελεστών του μετασχηματισμού)
[COEFF] = princomp([x;y]')

% Εμφάνιση των principal components από τα ιδιοδιανύσματα:
hold on
mu(1)=mean(x);
mu(2)=mean(y);
scale = 40;
pc1 = line([mu(1) - scale * R(1,1) mu(1) + scale * R(1,1)],[mu(2) - scale * R(2,1)
mu(2) + scale * R(2,1)]);
pc2 = line([mu(1) - scale * R(1,2) mu(1) + scale * R(1,2)],[mu(2) - scale * R(2,2)
mu(2) + scale * R(2,2)]);
```



Παράδειγμα εφαρμογής της μεθόδου LDA (Linear Discriminant Analysis) στο MATLAB

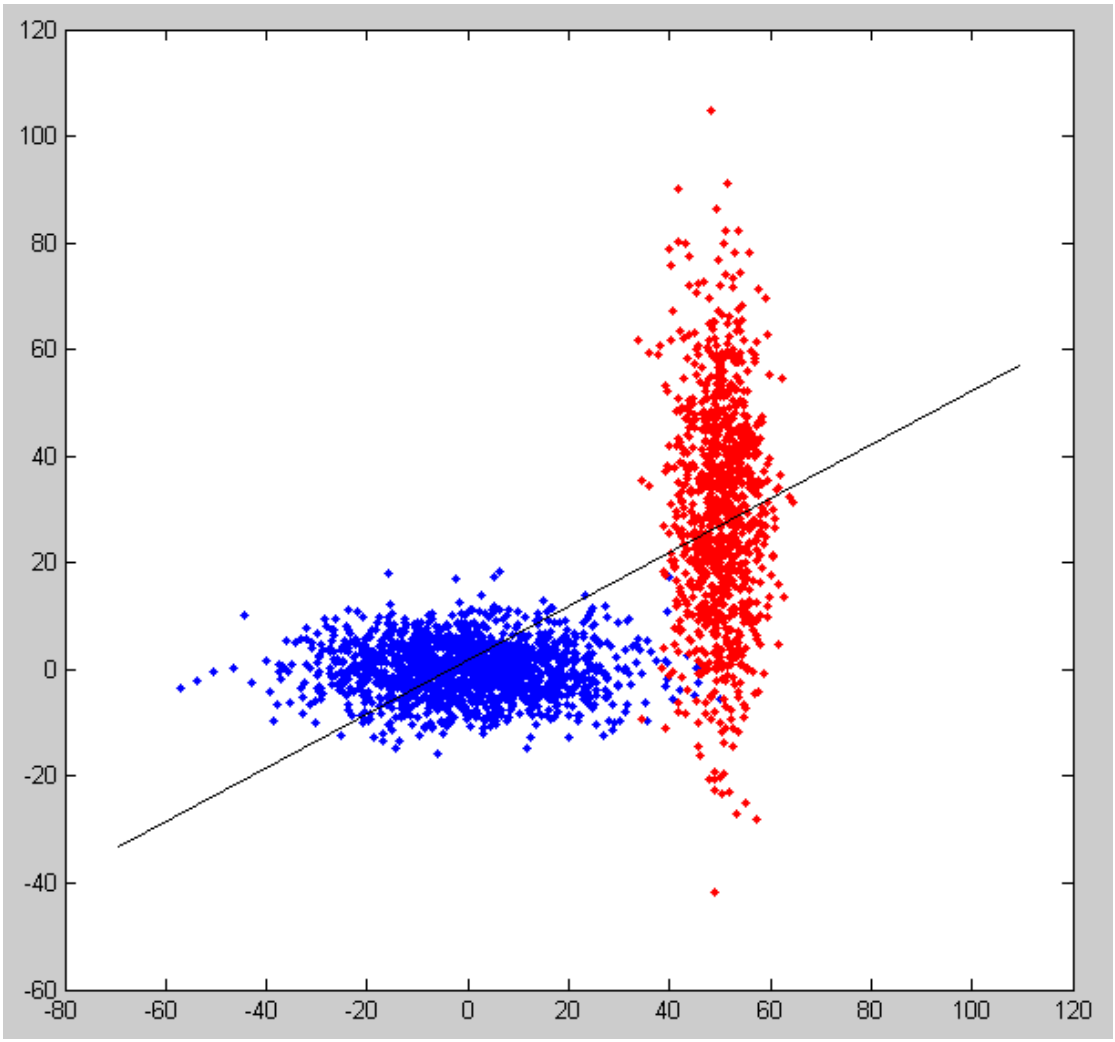
```
% Δημιουργία συνόλου δεδομένων δύο κλάσεων (στην πρώτη κλάση έχουμε 1500
% τυχαία σημεία με αναλογία κατανομής 1/3 στις δύο διαστάσεις, ενώ στην δεύτερη
% έχουμε 1000 τυχαία σημεία με αναλογία κατανομής 1/4 στις δύο διαστάσεις και
% μετατόπιση 50 μονάδες δεξιά και 30 επάνω).
a=15*randn(1500,1);
b=5*randn(1500,1);
Group_X(:,1)=a;
Group_X(:,2)=b;
plot(Group_X(:,1),Group_X(:,2),'b.');
hold on

c=5*randn(1000,1)+50;
d=20*randn(1000,1)+30;
Group_Y(:,1)=c;
Group_Y(:,2)=d;
plot(Group_Y(:,1),Group_Y(:,2),'r.');
hold on

% Δημιουργία ενιαίου συνόλου δεδομένων και ετικετών των κλάσεων
All_data = [Group_X;Group_Y];
All_data_label = [];
for k=1:length(All_data)
   if k<=length(Group_X)
      All_data_label = [All_data_label; 1 ];
   else
      All_data_label = [All_data_label; 2 ];
   end
end

% Εφαρμογή της μεθόδου LDA στα δεδομένα (επιστροφή των ιδιοδιανυσμάτων
% και ιδιοτιμών στους πίνακες V,D.
classes = max(All_data_label);
mu_total = mean(All_data);
mu = [ mean(Group_X); mean(Group_Y) ];
Sw = (All_data - mu(All_data_label,:))'*(All_data - mu(All_data_label,:));
Sb = (ones(classes,1) * mu_total - mu)' * (ones(classes,1) * mu_total - mu);
 [V, D] = eig(Sw\Sb);
% ταξινόμηση κατά φθίνουσα σειρά:
[D, i] = sort(diag(D), 'descend');
V = V(:,i);

% Εμφάνιση της ευθείας πάνω στην οποία οι προβολές των σημείων διαχωρίζονται
% όσο το δυνατόν καλύτερα (1st component)
scale = 100;
pc1 = line([mu_total(1)-scale*V(1,1) mu_total(1)+scale*V(1,1)], [mu_total(2)-
scale*V(2,1)  mu_total(2)+scale*V(2,1)]);
set(pc1, 'color','k');
```


120
100
80
60
40
20
0
-20
-40
-60
-80
-60
-40
-20
0
20
40
60
80
100
120