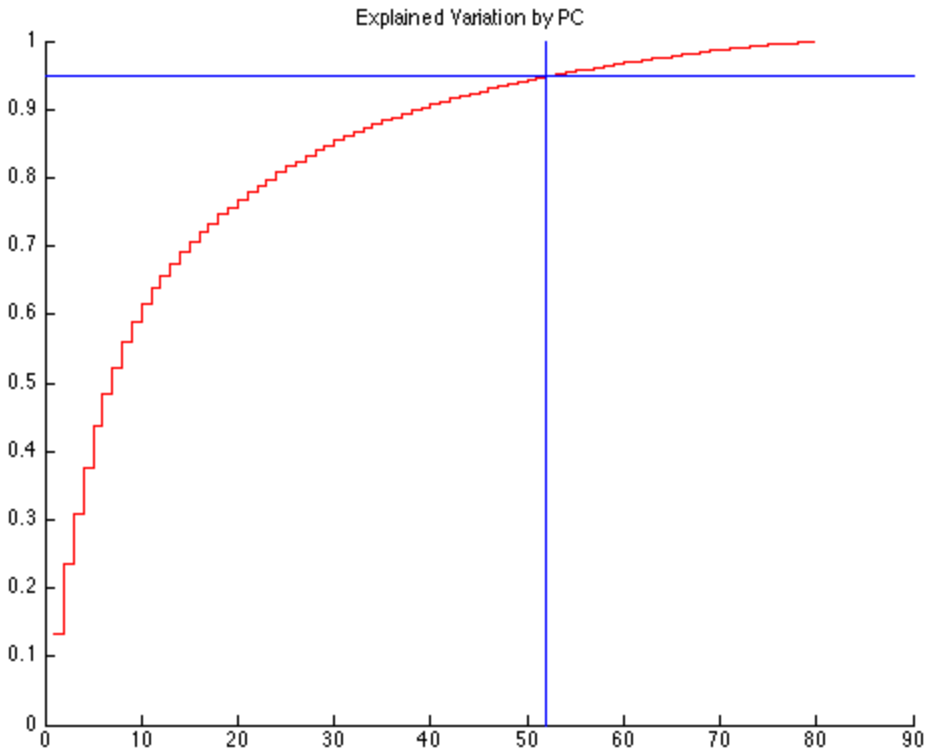
Table of Contents


```
% Sean Bennett
% HW6
```

Question 1

```
load('CancerMicroarray.mat')

numTumors = length(G)
numGenes = length(X)

% There are 2308 different genes sequenced from 83 different tumors.

[coeffPC, scorePC, latentPC] = pca(X);

cumulativeLatent = cumsum(latentPC) / sum(latentPC);

numPC = max(find(cumulativeLatent < 0.95));

% You need 52 or 53 PCs to explain 95% of the data

figure;
hold on;
stairs(cumulativeLatent, 'Color', 'r');
title('Explained Variation by PC');
refline(0, 0.95);
line([52 52],[0 1]);
hold off;


linClass = fitcdiscr(scorePC(:, 1:10), G);
quadClass = fitcdiscr(scorePC(:, 1:10), G, 'DiscrimType', 'quadratic');

treeDiscr = fitctree(scorePC(:, 1:10), G);
```

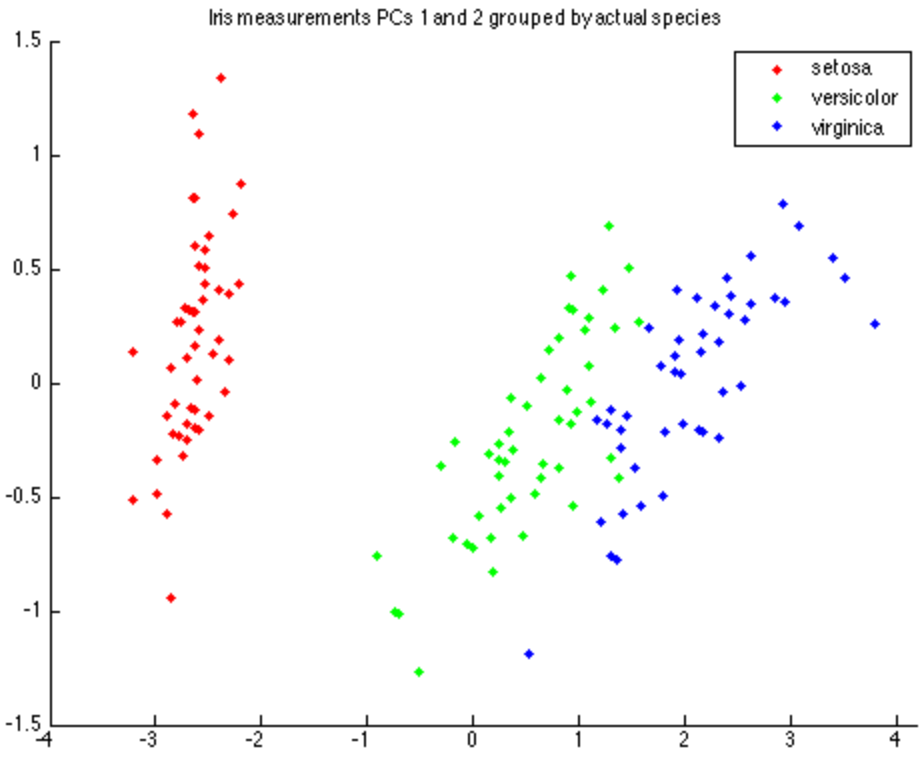
```
numTumors =

    83


numGenes =

        2308
```

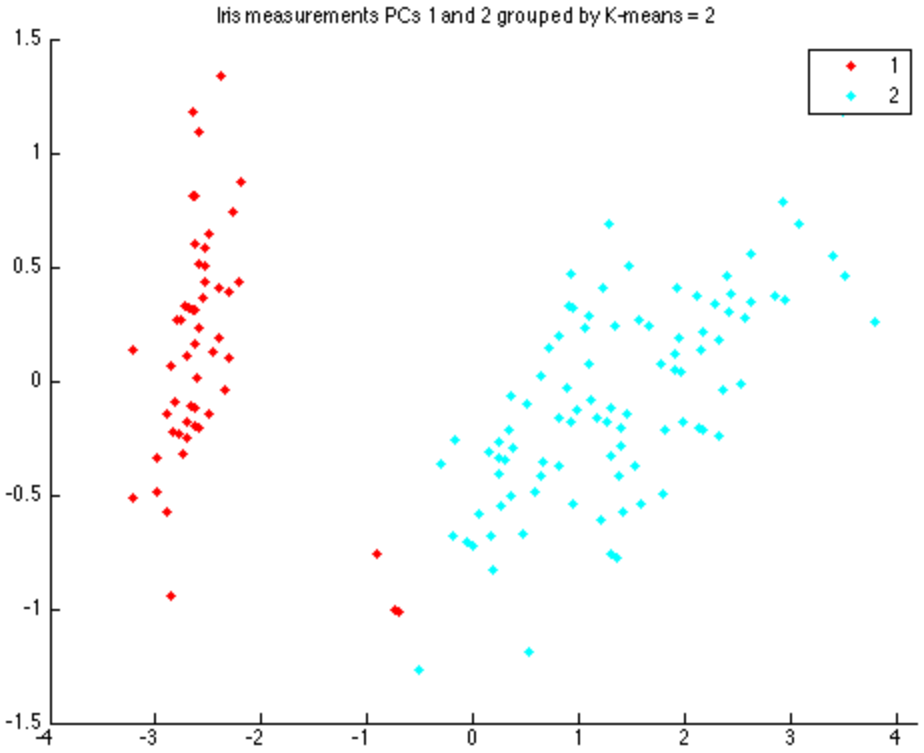


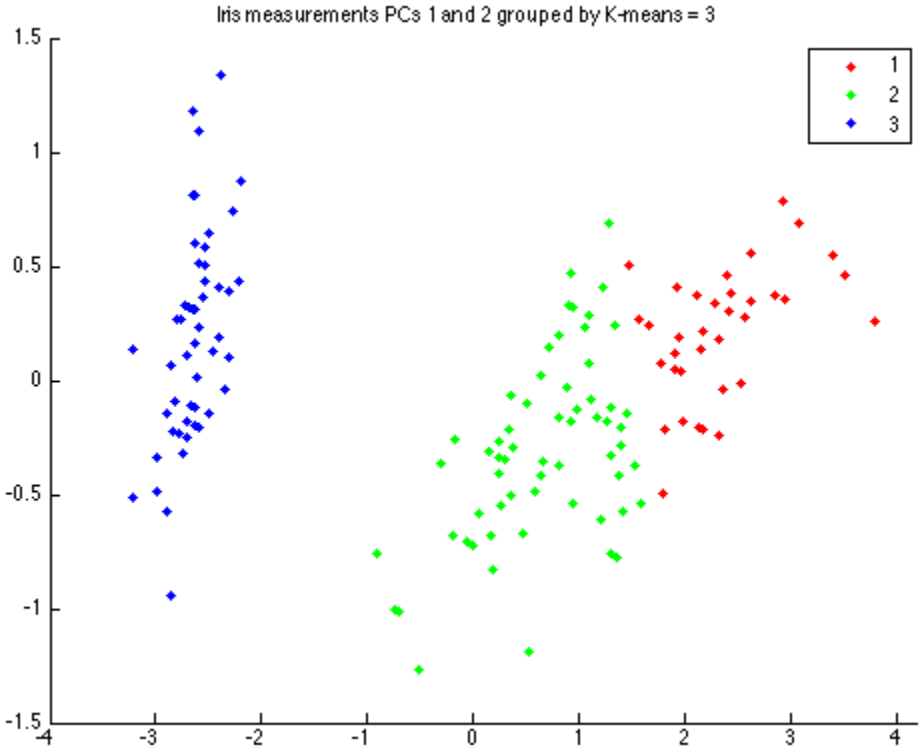
Question 2

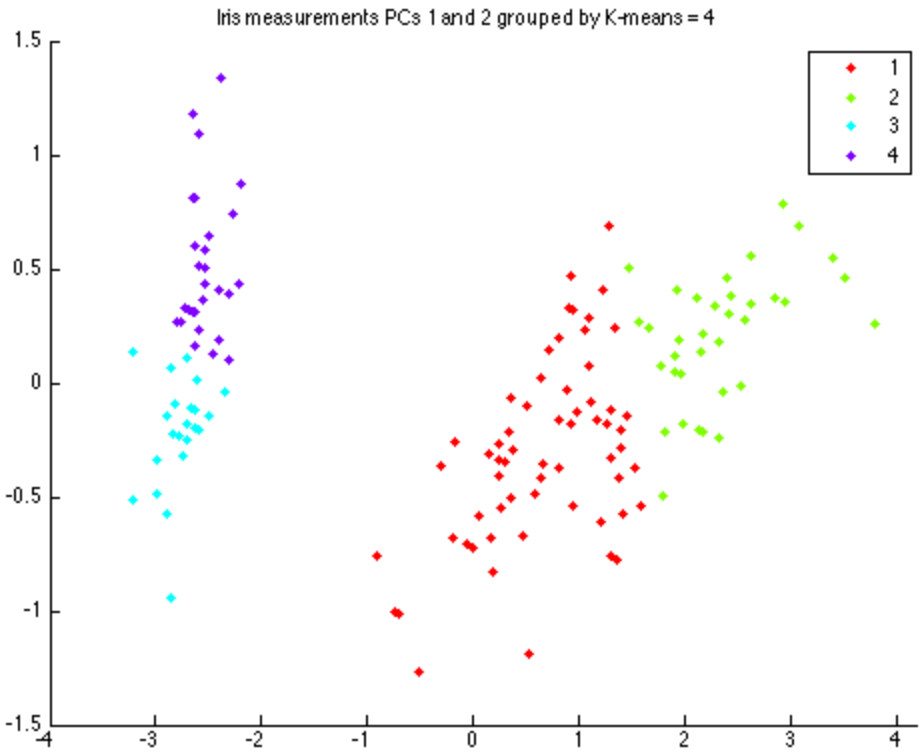
```
load('fisheriris');
numSpecies = length(unique(species));
numIris = length(species);
numMeas = size(meas,2);

% The measruements Ronald Fisher took were Petal Width, Petal Length, Sepal
% Width, and Sepal Length, measured in mm

[coeffPC2, scorePC2, latentPC2] = pca(meas);

figure;
gscatter(scorePC2(:, 1), scorePC2(:, 2), species);
title('Iris measurements PCs 1 and 2 grouped by actual species');

kmeans2 = kmeans(meas, 2);
figure;
gscatter(scorePC2(:, 1), scorePC2(:, 2), kmeans2);
title('Iris measurements PCs 1 and 2 grouped by K-means = 2');

kmeans3 = kmeans(meas, 3);
figure;
gscatter(scorePC2(:, 1), scorePC2(:, 2), kmeans3);
title('Iris measurements PCs 1 and 2 grouped by K-means = 3');
```

```
kmeans4 = kmeans(meas, 4);
figure;
gscatter(scorePC2(:, 1), scorePC2(:, 2), kmeans4);
title('Iris measurements PCs 1 and 2 grouped by K-means = 4');
```



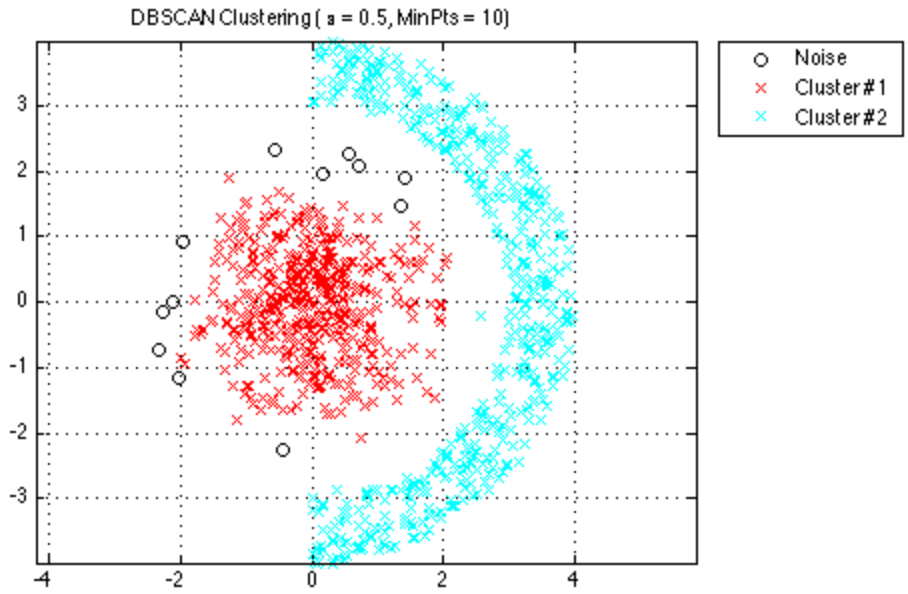

Iris measurements PCs 1 and 2 grouped by K-means = 2
1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4

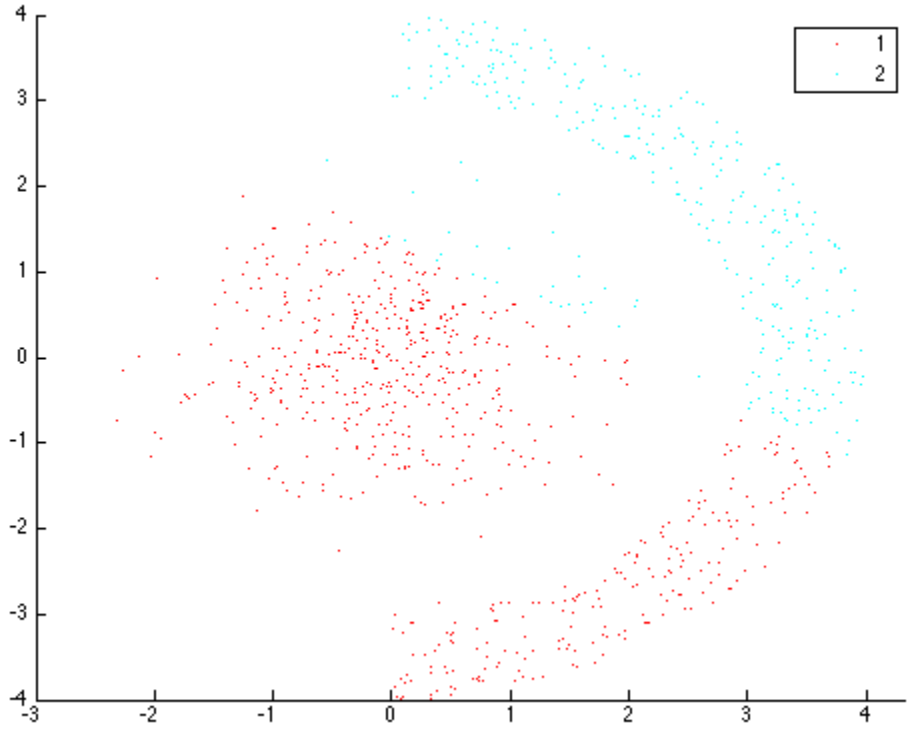

Iris measurements PCs 1 and 2 grouped by K-means = 3
1
2
3
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4



Question 3

```
% Pt a.

% Pt b.

load('mydata');

epsilon=0.5;
MinPts=10;
IDX=DBSCAN(X,epsilon,MinPts);

figure;
PlotClusterinResult(X, IDX);
title(['DBSCAN Clustering (\epsilon = ' num2str(epsilon) ', MinPts = ' num2str(Min


Q3kmeans = kmeans(X, 2);
figure;
gscatter(X(:, 1), X(:, 2), Q3kmeans);

% pt c.

% The DBSCAN algorithm
```


DBSCAN Clustering (ε = 0.5, MinPts = 10)
Noise
Cluster#1
Cluster#2


1
2

Published with MATLAB® R2014a