Unsupervised Classification

(Cluster Analysis Group the observations into *k* distinct natural groups)

Hierarchical clustering:

We have a dataset with *n* observations and we want to group the observations into *k* distinct natural groups of similar observations.

We distinguish three stages of cluster analysis:

Input Stage
Algorithm stage
Output stage

Input Stage

1. Scaling:

a) Divide variables by the standard deviation.

b) Spherize the data: Invariance under afine transformations.

$Z = A Y$; $A = \text{Chol}(S)^{-1}$ or the symmetric square root $S^{-1/2}$;

c) Spherize the data with the within variance.

$T = W + B$

To obtain W use iteration.

Hierarchical clustering

2. Similarity and dissimilarity measures.

Clustering methods require the definition of a similarity or dissimilarity measure. For example an inter-point distance d(x1,x2) and an inter-cluster distance d*(C1,C2) are examples of dissimilarity.

The inter point distance is often taken to be the Euclidean distance or Mahalanobis distance. Some times we may use the Manhattan distance.

When the data is not metric we may define any distance or similarity measure from characteristics of the problem. For example for binary data given any two vector observations we construct the table

	1	0	Total
1	a	b	a+b
0	c	d	c+d
Total	A+c	b+d	P

Then we define distance as the square root of the χ^2 statistic.

Also d = 1 - (a+d)/p or d = 1- a/(a+b+c)

Hierarchical clustering

Build a hierarchical tree

- Inter point distance is normally the Euclidean distance
  (some times we may use Manhattan distance).

- Inter cluster distance:
  - Single Linkage: distance between the closes two points
  - Complete Linkage: distance between the furthest two points
  - Average Linkage: Average distance between every pair of points
  - Ward: R^2 change.
- Build a hierarchical tree:
  1. Start with a cluster at each sample point
  2. At each stage of building the tree the two closest clusters joint to form a new cluster.

Hierarchical clustering: Ward's method

At any stage we construct the dissimilarity matrix (or distance matrix) reflecting all the inter-cluster distances between any pair of categories.

We build a hierarchical tree starting with a cluster at each sample point, and at each stage of the tree

Build dissimilarity matrix

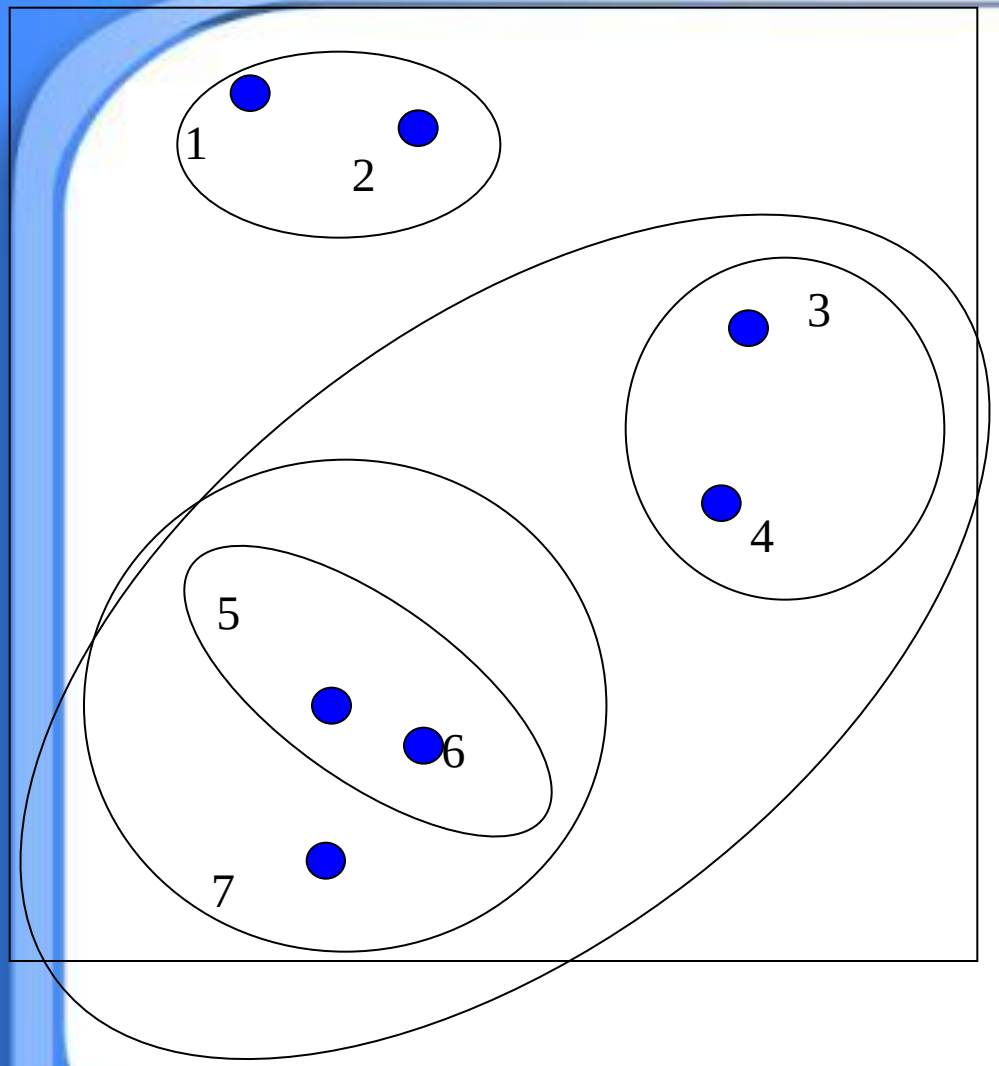
The two closest clusters joint to form a new cluster.

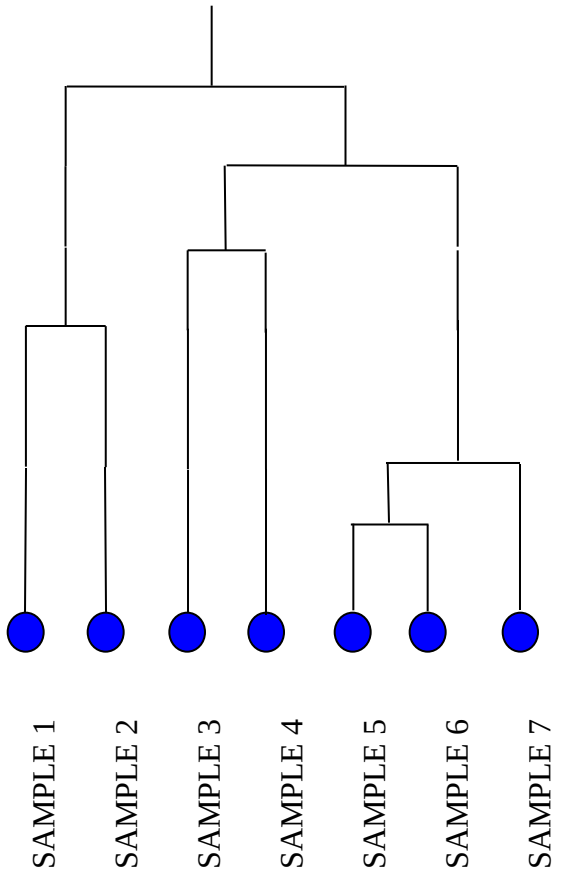
Once we finish building the tree the question becomes: "how many clusters do we chose?"

One way of making this determination is by inspecting the hierarchical tree and finding a reasonable point to break the clusters. We can also plot the criteria function for the different number of cluster and visually look for unusually large jumps. In the example below with *WARD's* clustering method we stop at the first place where the R^2 change (percent-wise) is large.

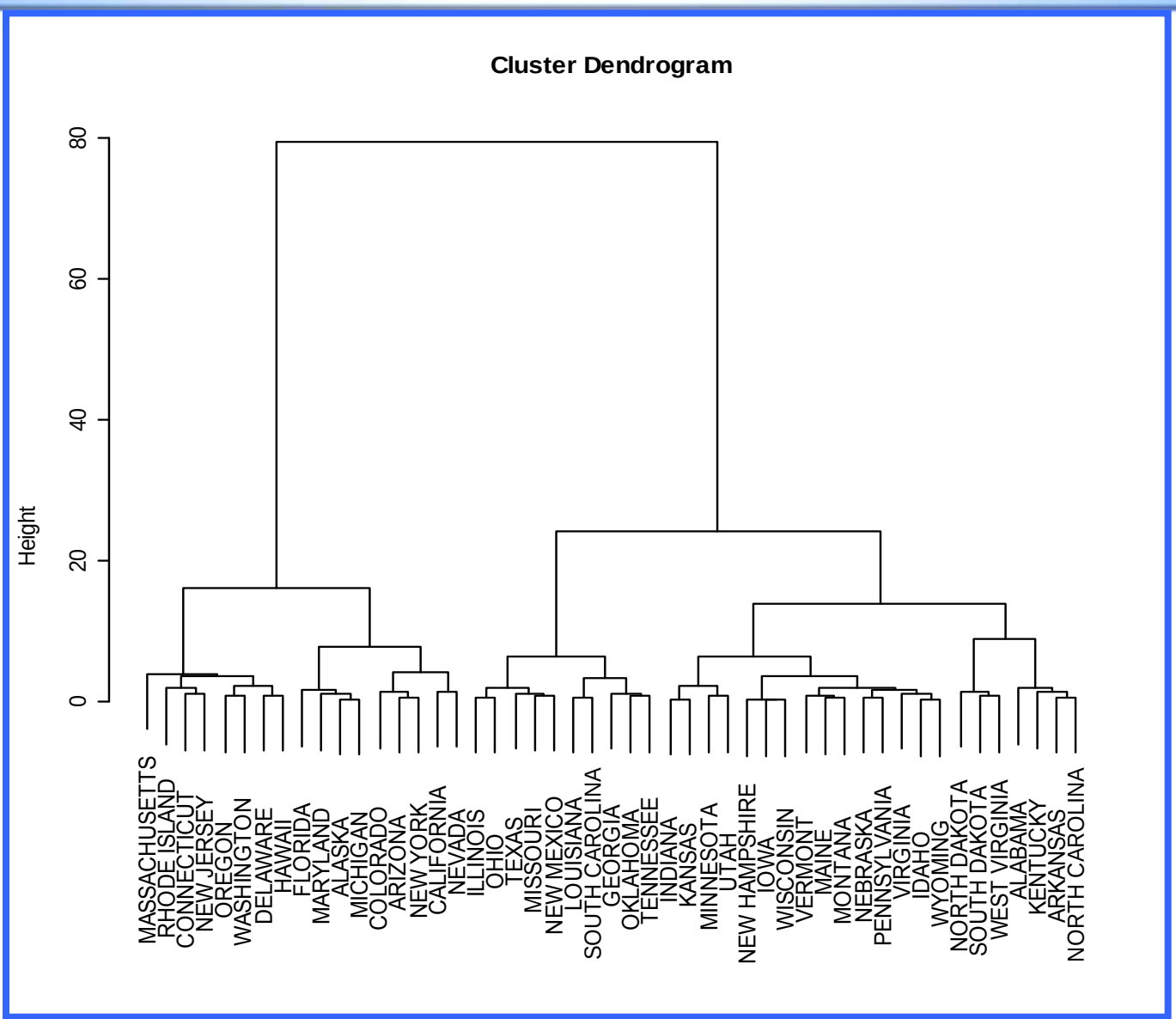
10	CL45	CL15	24	0.008555	0.824891
9	CL25	CL16	84	0.009749	0.815142
8	CL23	CL13	49	0.009836	0.805306
7	CL8	CL22	67	**0.009713**	0.795593
6	CL17	CL11	134	**0.037362**	0.758231
5	CL9	CL14	102	0.037383	0.720848

Hierarchical Cluster Example

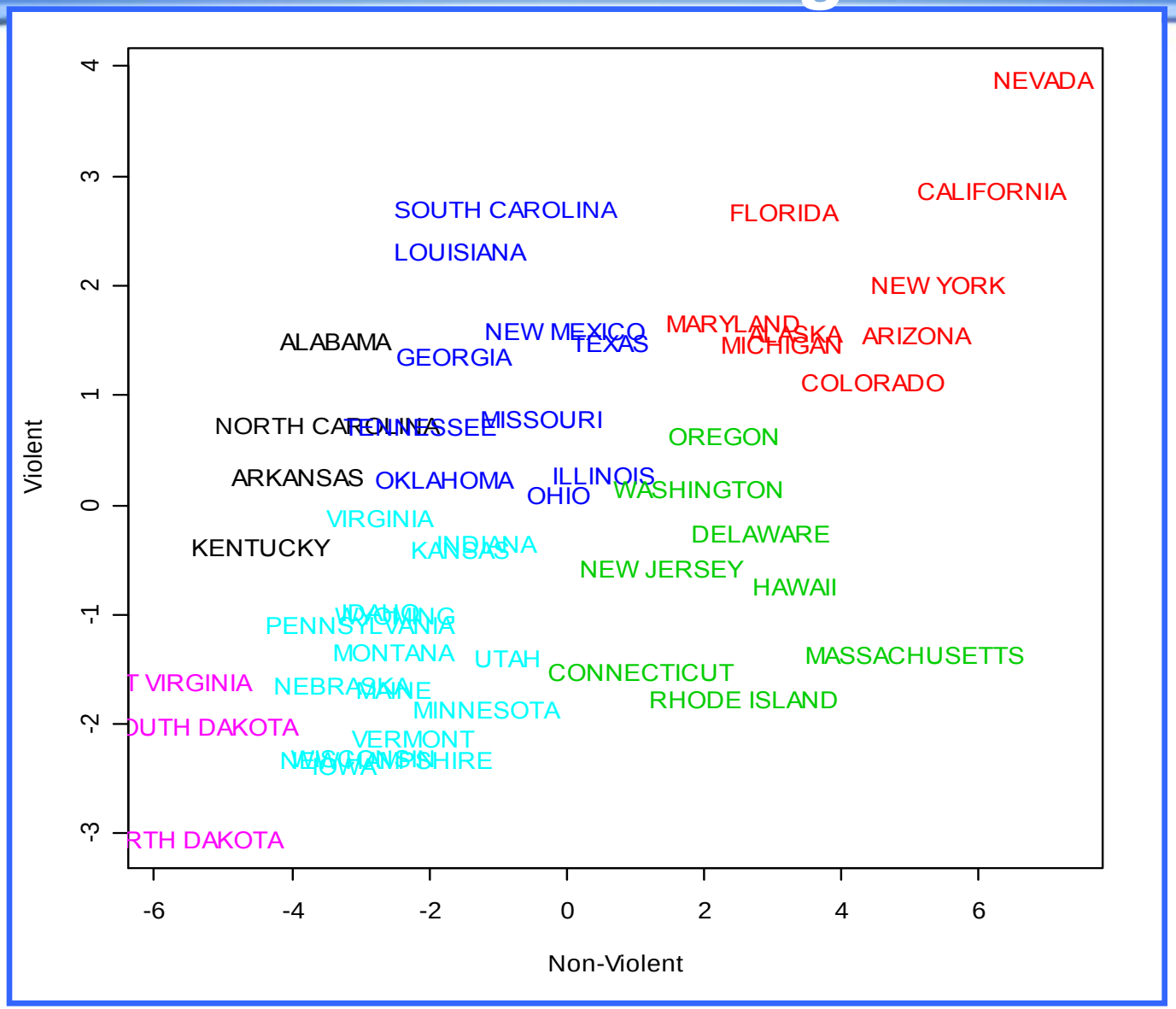




Cluster Analysis : Dendrogram using Ward's method



Cluster Analysis : 6 clusters selected using Ward's method



Non Hierarchical clustering: k-means

Centroid methods. *k-means* algorithm:

We start with a choice of *k* clusters and a choice of distance.

a. Determine the initial set of *k* clusters. *k* seed points are chosen and the data is distributed among *k* clusters.

b. Calculate the centroids of the *k* clusters and move each point to the cluster whose centroid is closest.

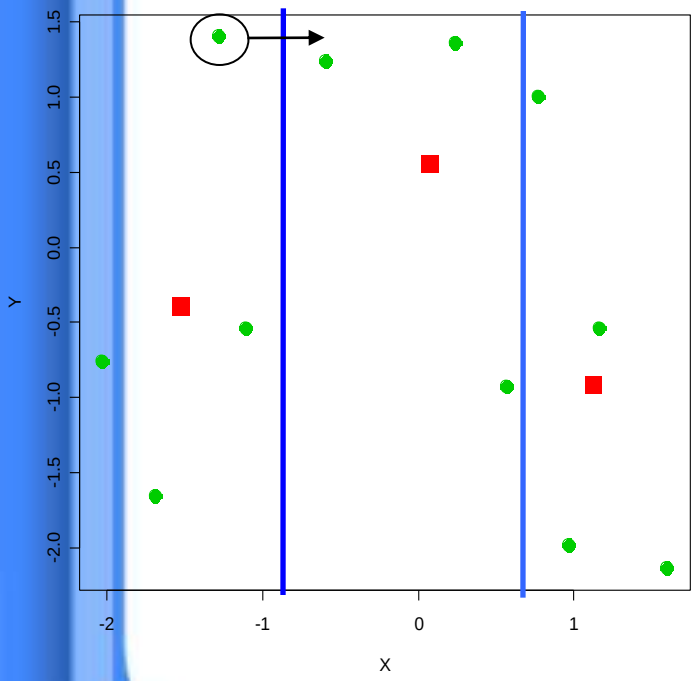
c. Repeat step b. until no change is observed.

This is the same as optimizing the R^2 criteria. At each stage of the algorithm one point is moved to the cluster that will optimize the criteria function. This is iterated until convergence occurs. The final configuration has some dependence on the initial configuration so it is important to take a good start.

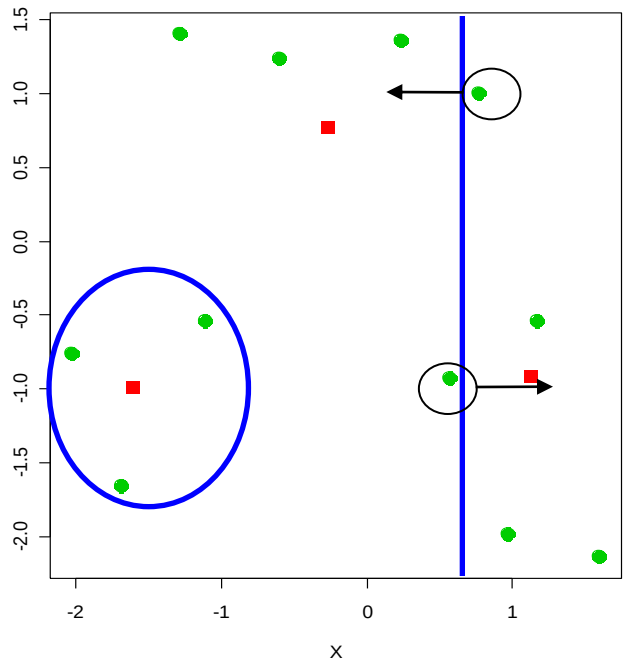
One possibility is to run *WARD*'s method and use the outcome as initial configuration for *k-means*.

Centroid methods: K-means algorithm. .

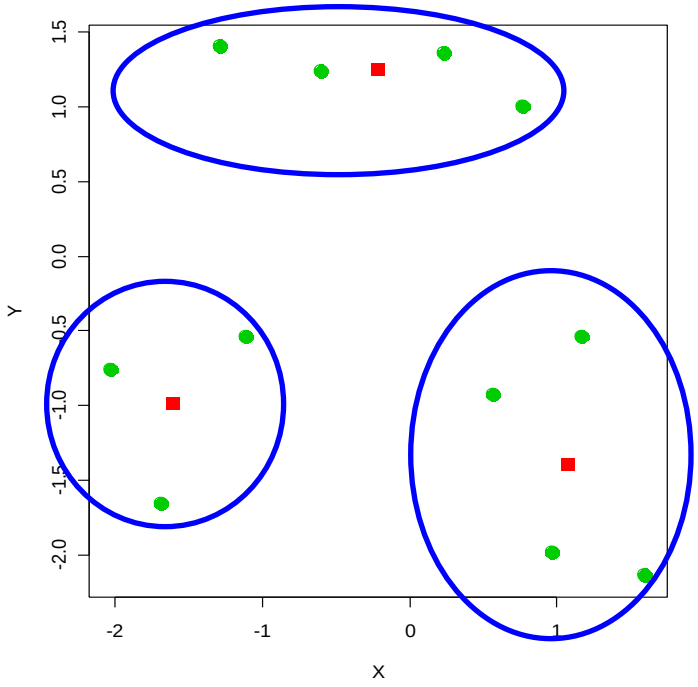
1. K seed points are chosen and the data is distributed among k clusters.
2. At each step we switch a point from one cluster to another if the R^2 is increased.
3. Then the clusters are slowly optimized by switching points until no improvement of the R^2 is possible



Step 1



Step 2



Step n

Non Hierarchical clustering: PAM

PAM

Pam is a robust version of *k-means*.

It used the medioids as the center and L_1 distance (Manhattan) and it is otherwise the same as K-means.

The cluster ***R*** package contains the ***pam*** function.

Model Based Hierarchical Clustering

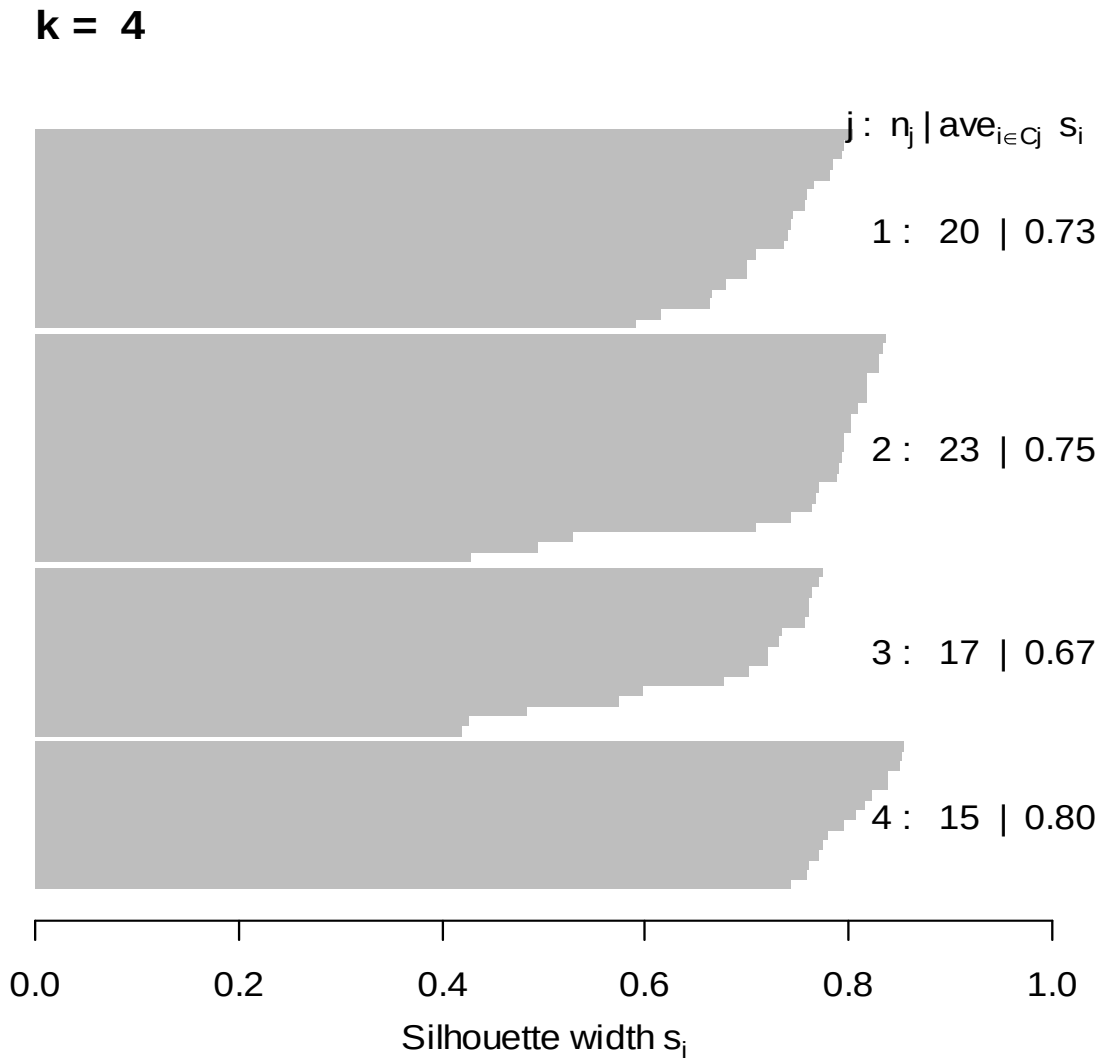
Another approach to hierarchical clustering is model-based clustering, which is based on the assumption that the data are generated by a mixture of underlying probability distributions.

The ***mclust*** function fits model-based clustering models. It also fits models based on heuristic criteria similar to those used by ***pam***.

The R package ***mclust*** and the function of the same name are available from CRAN. The ***mclust*** function is separate from the cluster library, and has somewhat different semantics than the methods discussed previously.

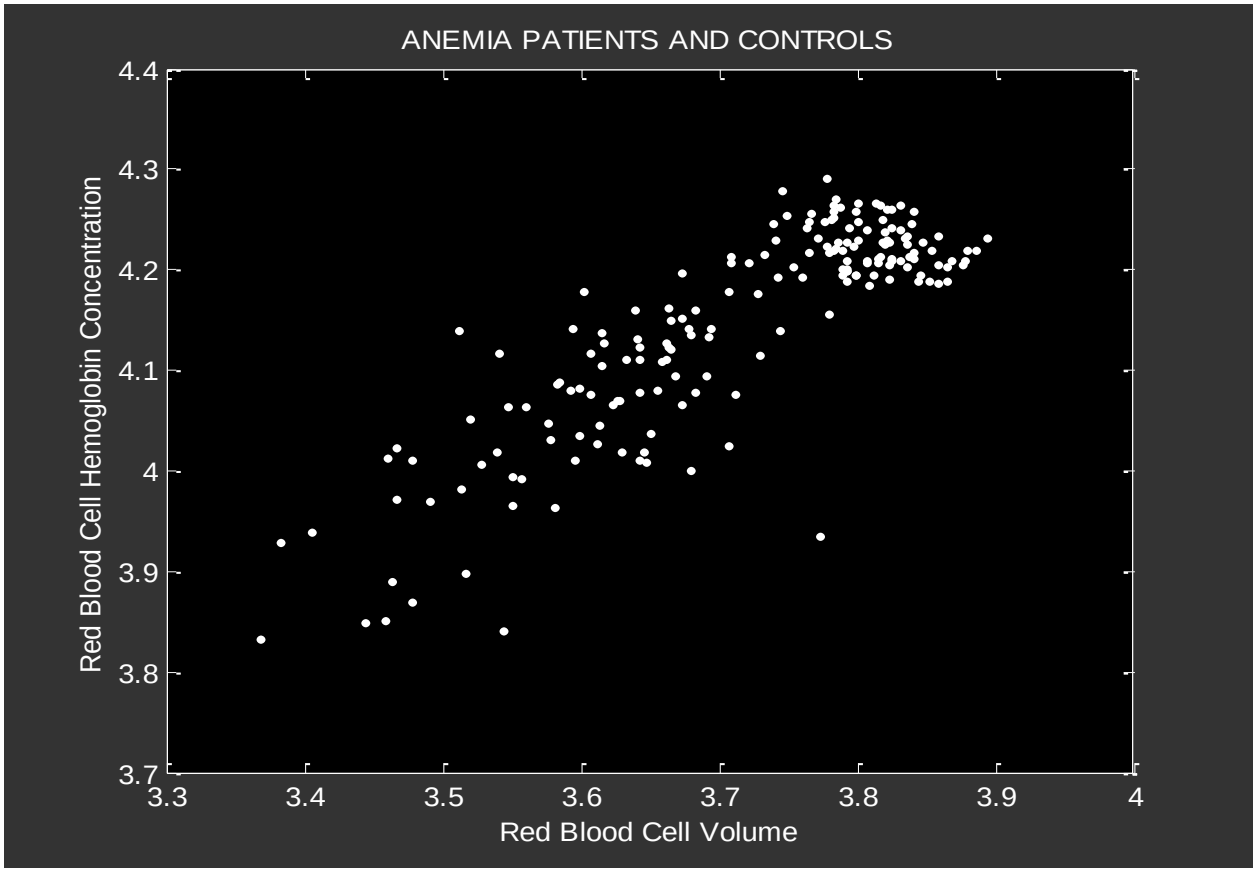
Detecting the number of clusters: silhouette graphs

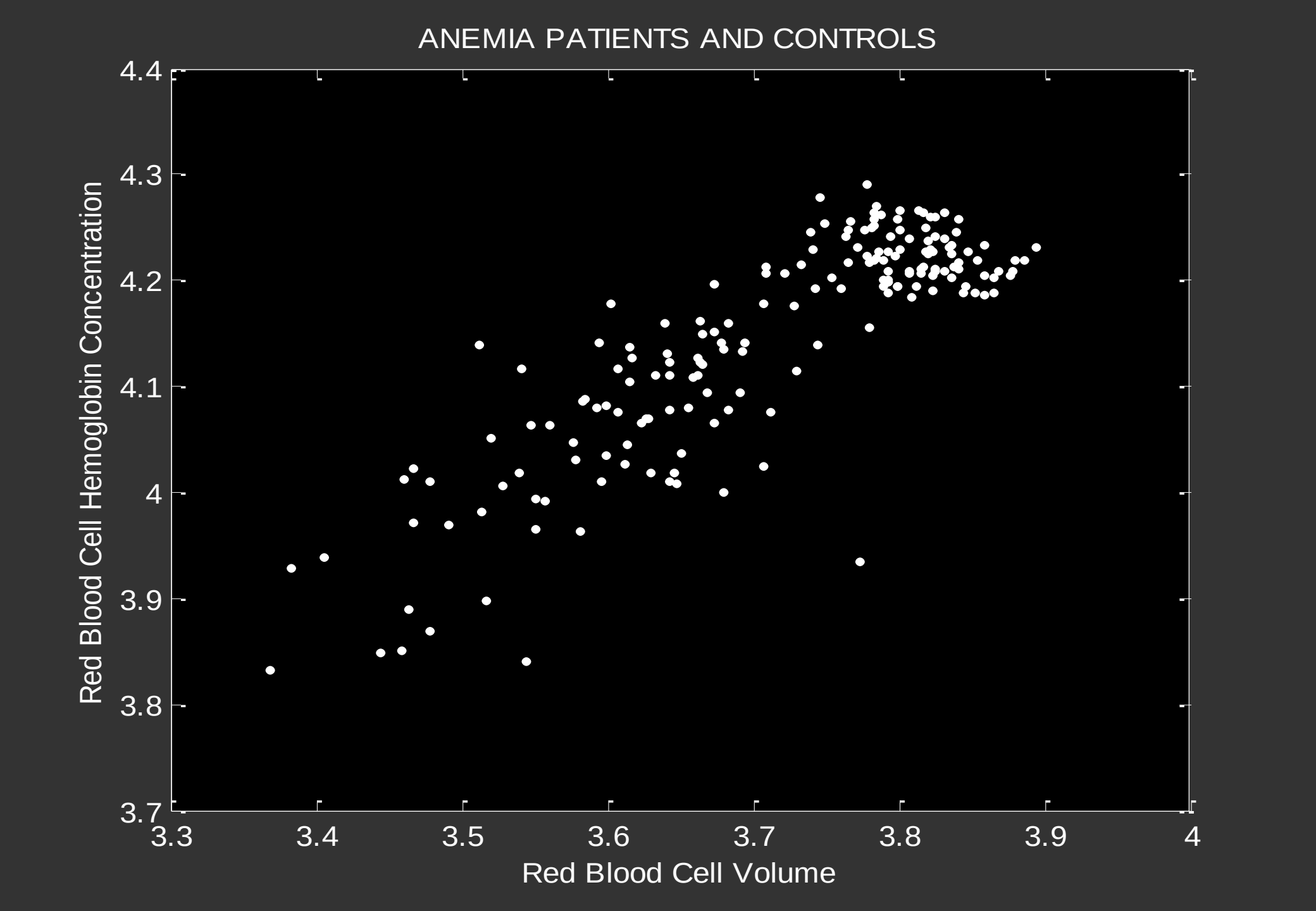
```
library(cluster); data(ruspini);
plot(silhouette(pam(ruspini, k=4)), main = paste("k = ",4), do.n.k=FALSE)
```

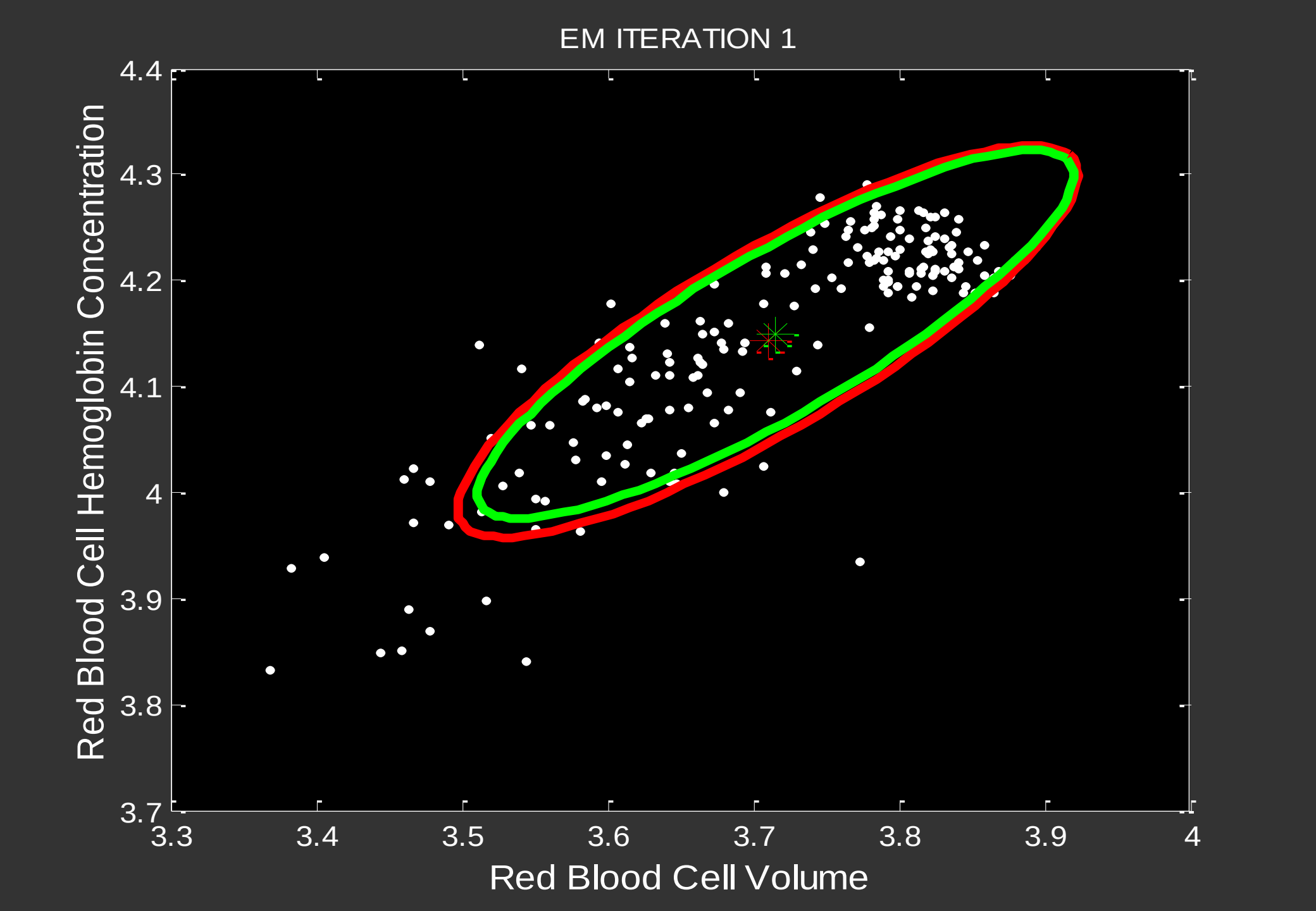


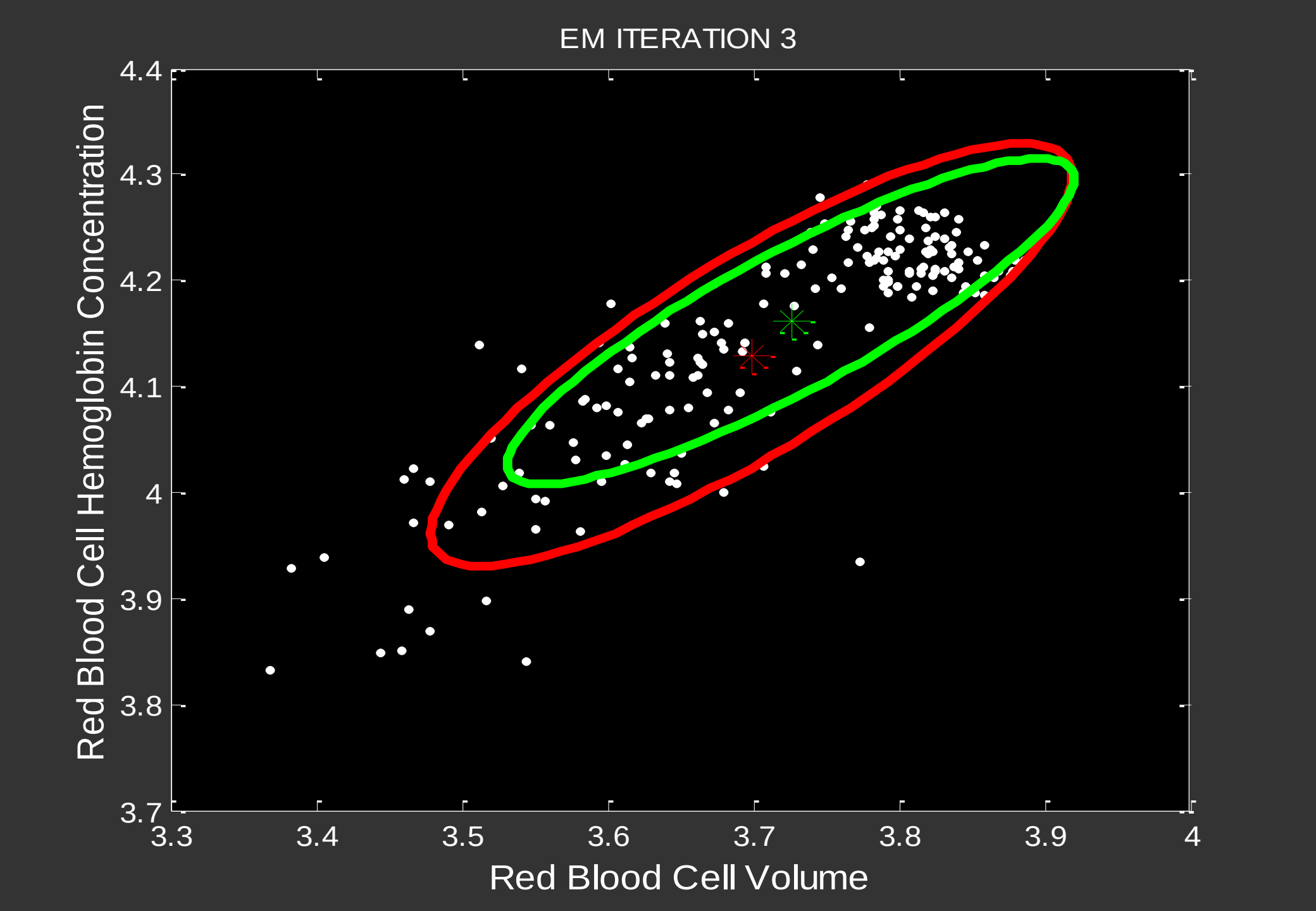
Model-based Clustering

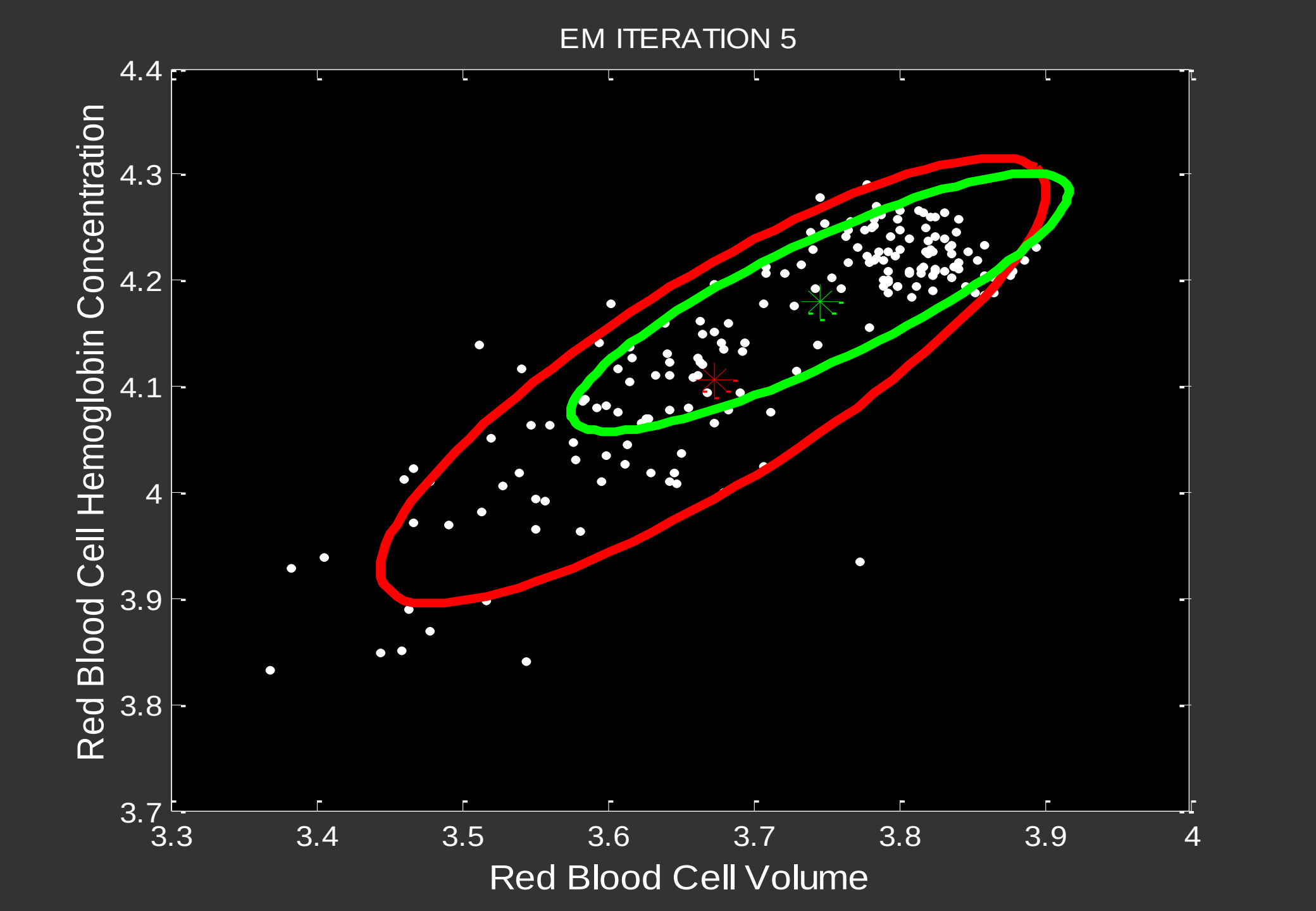
$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x;\theta_k)$$

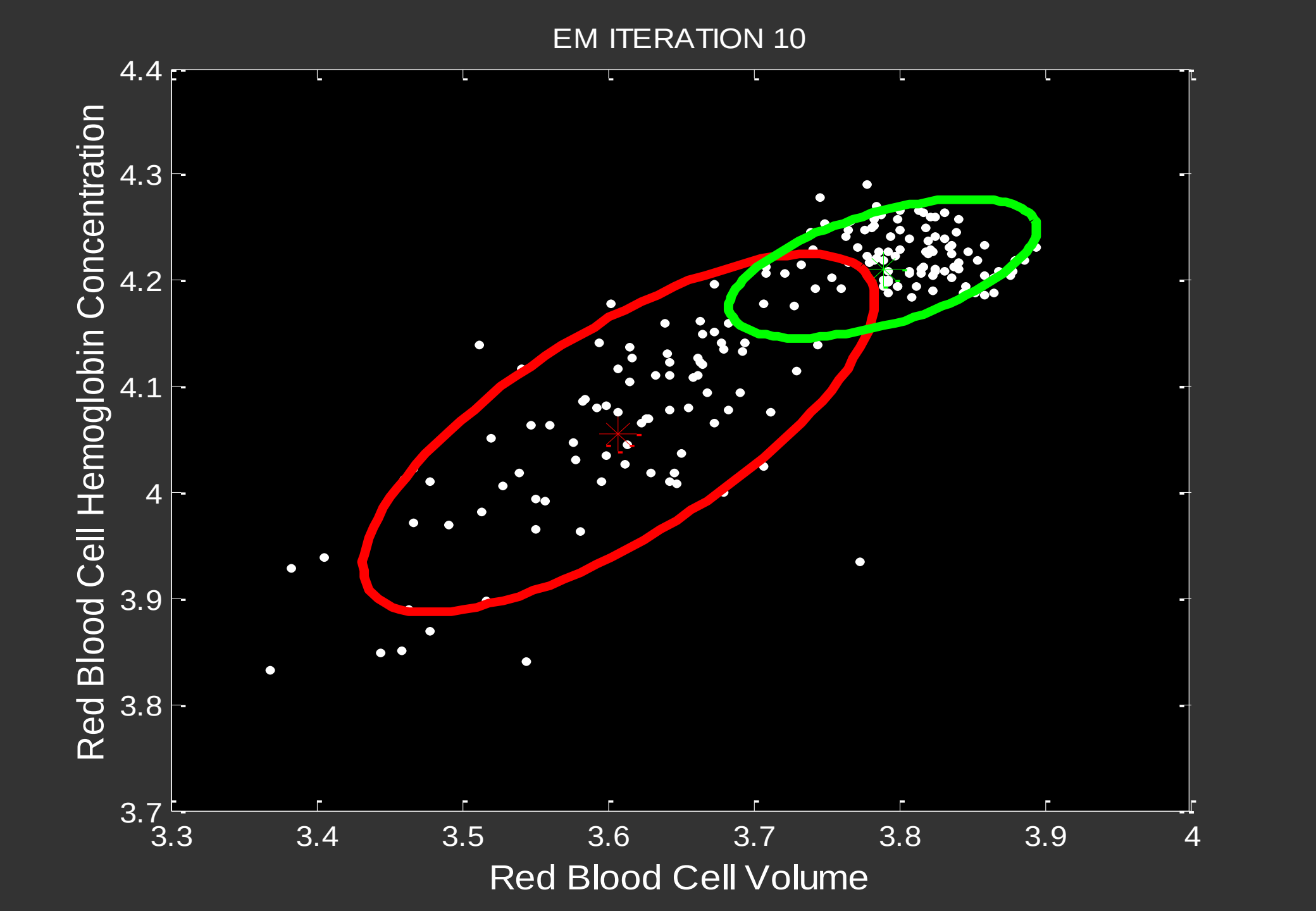


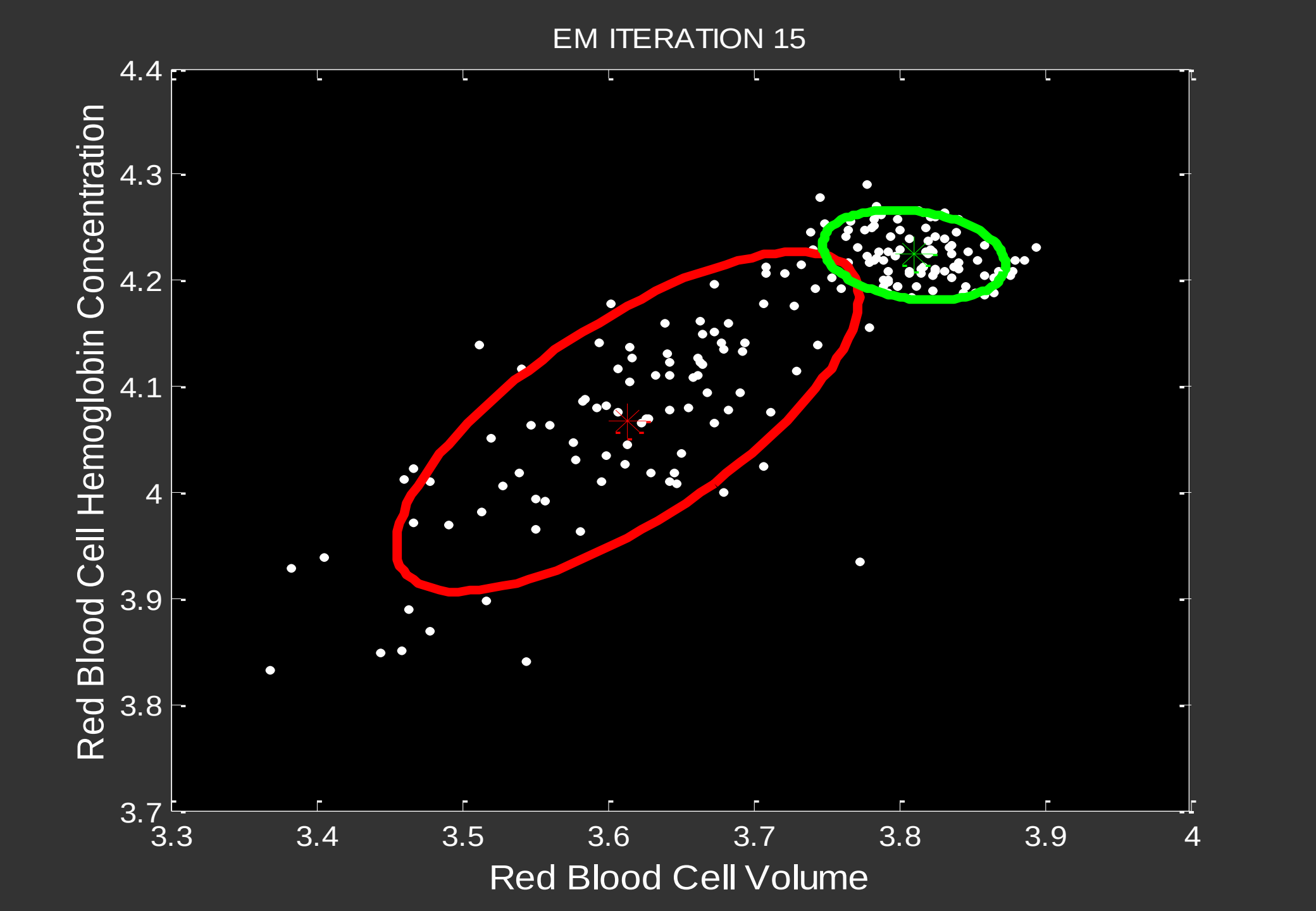

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

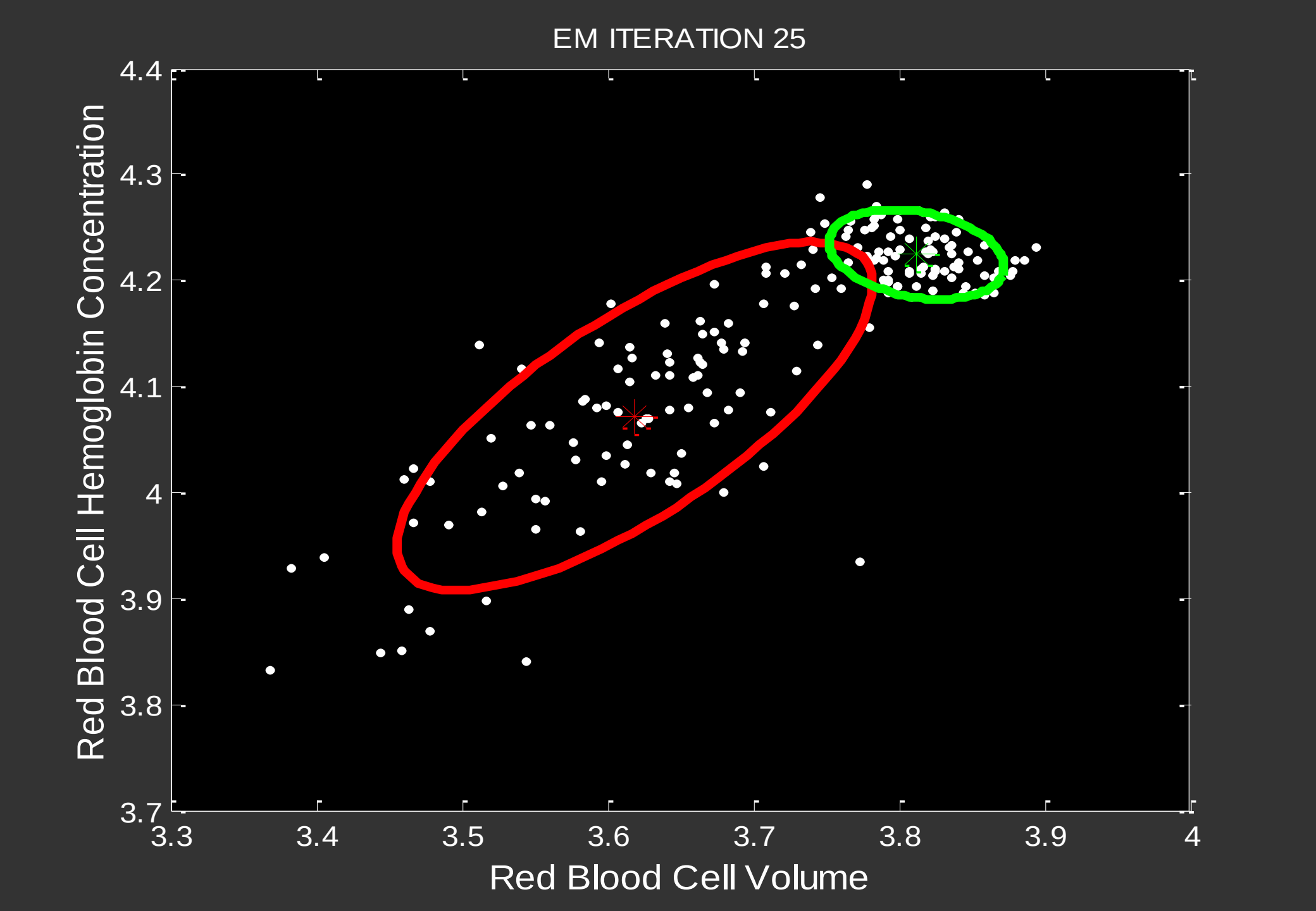

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

Mixtures of {Sequences, Curves, …}

$$p(D_i) = \sum_{k=1}^{K} p(D_i \mid c_k)\,\alpha_k$$

Generative Model

- select a component c_k for individual i

- generate data according to $p(D_i \mid c_k)$

- $p(D_i \mid c_k)$ can be very general

- e.g., sets of sequences, spatial patterns, etc

[Note: given $p(D_i \mid c_k)$, we can define an EM algorithm]

Megavariate data: ABC clustering

1. *A Bootstrap approach called ABC*
   Refers to the Bagging of genes and samples from Microarray data. Genes are bagged using weights proportional to their variances.

2. *By creating new datasets out of subsets of columns and genes we are able to create estimates of the class response several hundred times.*

3. *These estimates are then used to obtain a dissimilarity (distance) measure between the samples of the original data.*

4. *This dissimilarity matrix is then adopted to cluster the data.*

Data

Gene	S1	S2	S3	S4	S5	S6
G8521	1003	1306	713	1628	1268	1629
G8522	890	705	566	975	883	1005
G8523	680	749	811	669	724	643
G8524	262	311	336	1677	1286	1486
G8525	254	383	258	1652	1799	1645
G8526	81	140	288	298	241	342
G8527	4077	2557	2600	3394	2926	2755
G8528	2571	1929	1406	2439	1613	5074
G8529	55	73	121	22	141	44
G8530	1640	1693	1517	1731	1861	1550
G8531	168	229	284	220	310	315
G8532	323	258	359	345	308	315
G8533	12131	11199	14859	11544	11352	11506
G8534	11544	11352	12131	11199	14859	12529
G8535	1929	1406	2439	254	383	258
G8536	191	140	288	298	241	342
G8537	4077	2557	2600	3394	2926	2755
G8538	2571	1613	5074	1652	1799	1645
G8539	55	73	121	22	91	24
G8540	1640	1693	1517	1731	1861	1750
G8541	168	229	284	220	312	335
G8542	323	258	359	345	298	325
G8543	2007	1878	1502	1758	2480	1731
G8544	2480	1731	2007	1878	1502	1758
G8545	1652	1799	1645	254	383	258
G8546	298	241	342	81	150	298
G8547	2607	3394	2926	2755	3077	2227
G8548	2571	1929	1406	2439	1613	5074
G8549	121	22	55	730	201	35
G8550	1640	1693	1517	1731	1861	1550

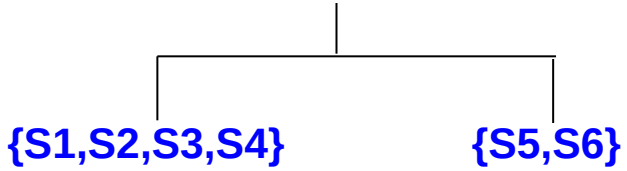
Select n samples and g genes

Gene	S1	S2	S4	S5	S6
G8523	680	749	669	724	643
G8524	262	311	1677	1286	1486
G8528	2571	1929	2439	1613	5074
G8530	1640	1693	1731	1861	1550
G8537	4077	2557	3394	2926	2755
G8545	1652	1799	254	383	258
G8547	2607	3394	2755	3077	2227

Compute similarity

Similarity	S1	S2	S3	S4	S5	S6
S1	0	6	7	7	0	0
S2	6	0	5	5	1	1
S3	7	5	0	8	0	0
S4	7	5	8	0	2	2
S5	0	2	0	2	0	10
S6	0	2	0	2	10	0

Final Clusters



Examples

For each data set:

 # Genes Selected= √G, *# Simulations = 500*

Genes Bagged By Variance

	Armstrong	Colon	Tao	Golub	Iris
BagWeight	**0.01**	**0.1**	**0.2**	**0.17**	**0.05**
BagEquiWeight	0.07	0.48	0.2	0.36	0.11
BagWholeData	0.08	0.48	0.3	0.4	**0.05**
NoBagWeight	**0.01**	**0.1**	**0.2**	**0.17**	0.08
NoBagEquiWeight	0.03	0.37	0.2	0.4	0.13
Ward	0.1	0.48	0.4	0.29	0.09
Kmeans	0.06	0.48	0.4	0.21	0.11