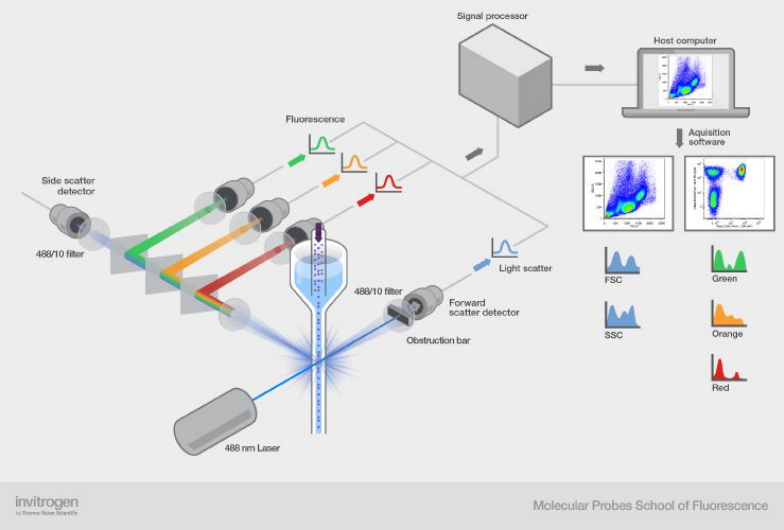
Chapter 19: Clustering

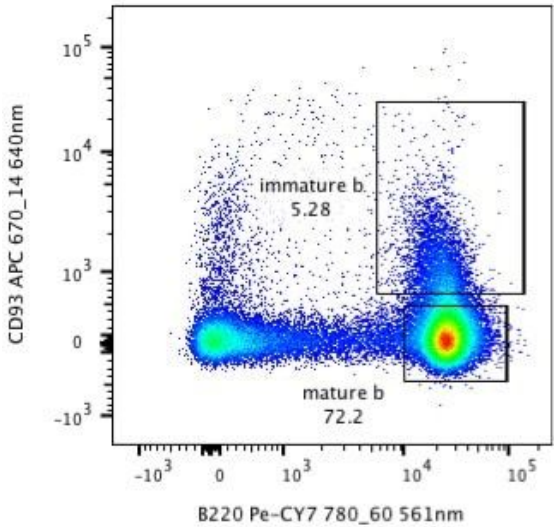
Modern Clinical Data Science
Chapter Guides
Bethany Percha, Instructor

How to Use this Guide

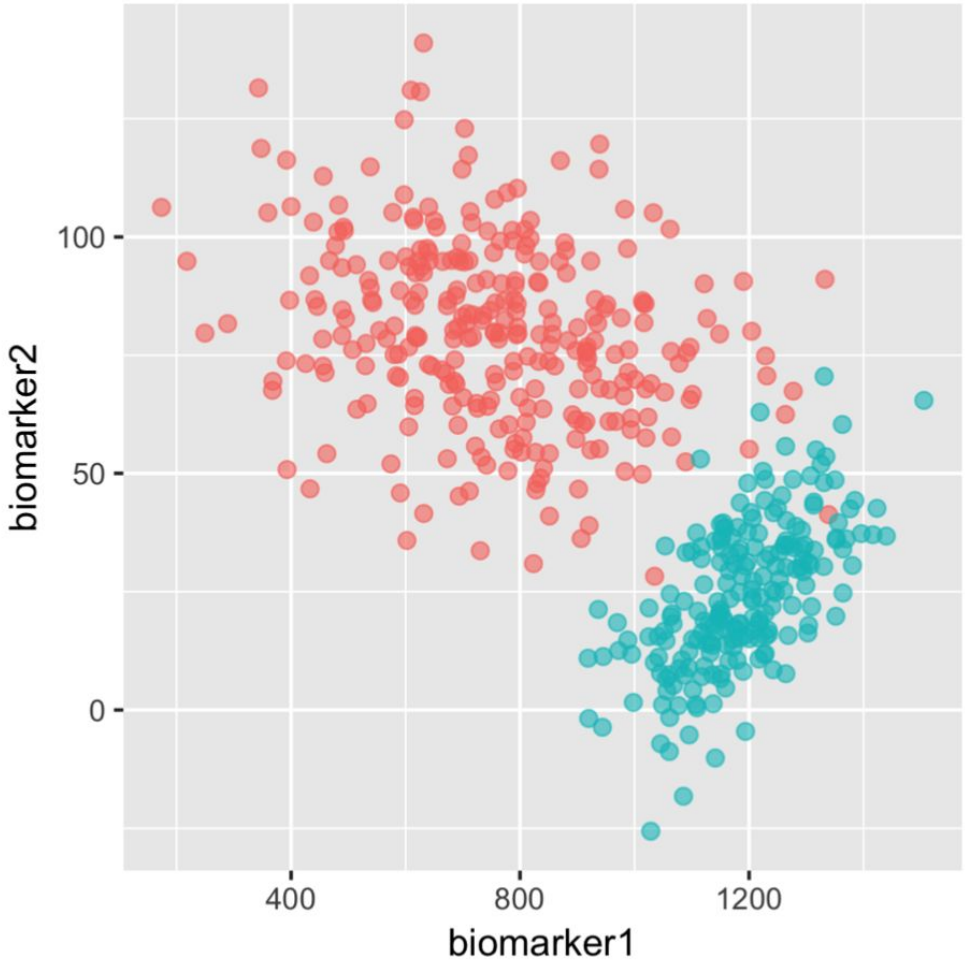
- Read the corresponding notes chapter first
- Try to answer the discussion questions on your own
- Listen to the chapter guide (should be 30 min, max) while following along in the notes

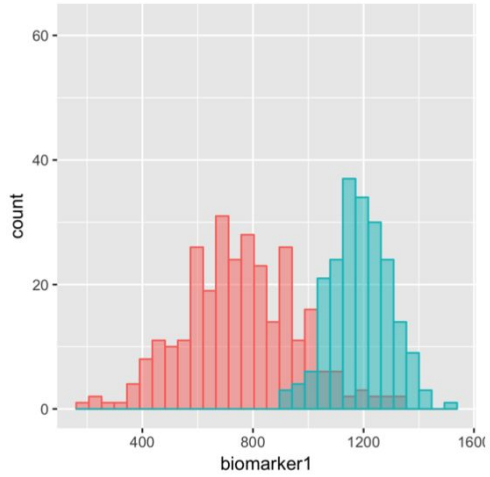
Thought Experiment: Flow Cytometry Data

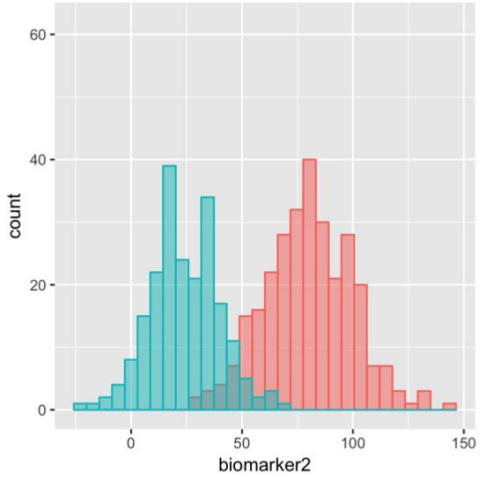




https://bitesizebio.com/26007/spot-the-difference-5-ways-to-improve-the-presentation-of-your-flow-cytometry-data/

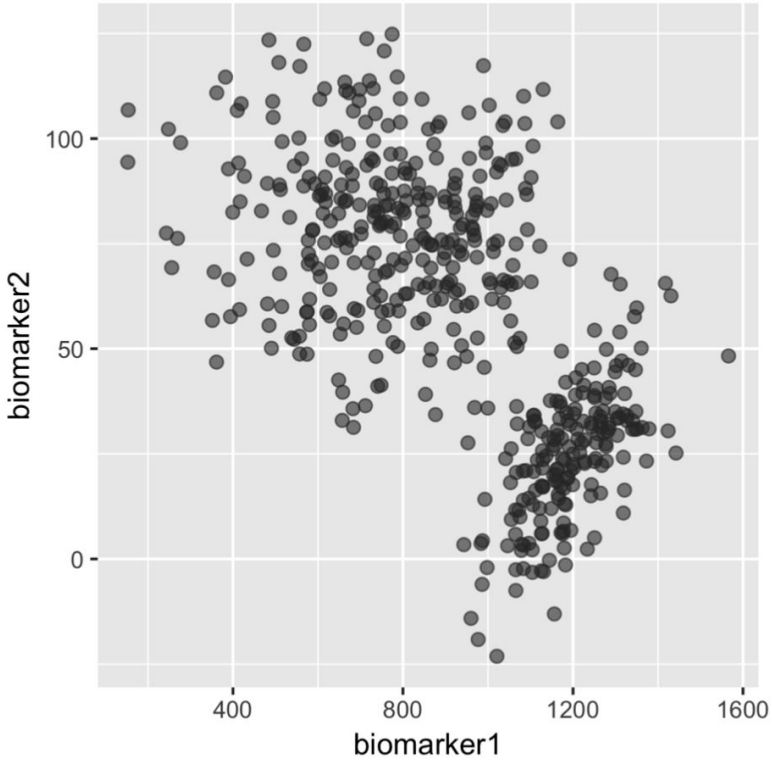

biomarker2
100
50
0
400
800
1200
biomarker1

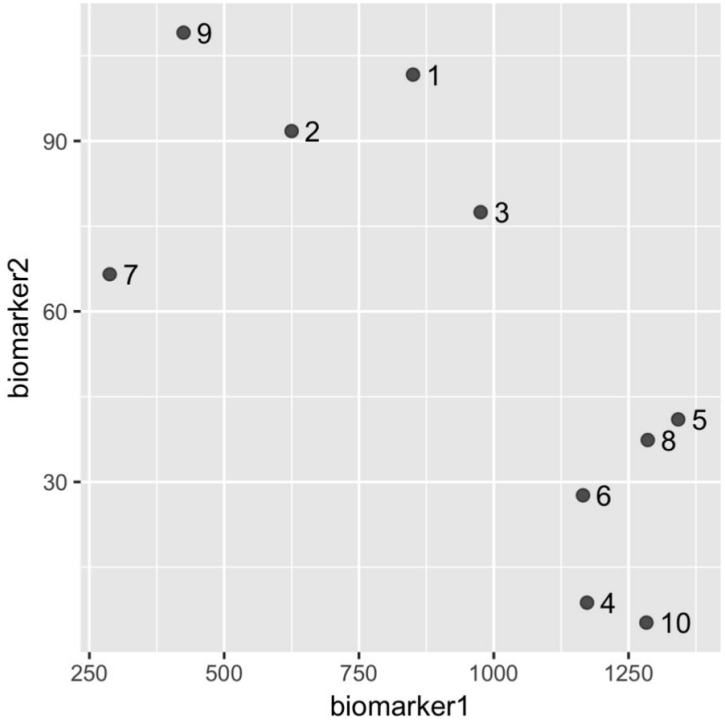

60
40
20
0
count
400
800
1200
biomarker1


60
40
20
0
count
0
50
100
150
biomarker2

Question 19.1

Speculate on some possible approaches for separating data that look like this into groups, or clusters.





i	Biomarker 1 Intensity ($x_1^{(i)}$)	Biomarker 2 Intensity ($x_2^{(i)}$)	Cell Line (z)
1	634.83	110.55	
2	650.06	74.22	
3	788.24	81.52	
4	771.47	84.98	
5	515.81	91.08	
6	1101.23	31.05	
7	649.32	77.05	
8	652.89	97.16	
9	1183.02	11.73	
10	1238.45	33.46	

The K-Means Algorithm

1. Assign each of the n datapoints to a random cluster. You can do this one of three ways:

   (1) Choose a random cluster for each point independently.

   (2) Choose K initial points to be the cluster centers.

   (3) Choose K initial points uniformly within the feature space (not necessarily data point locations) to be the cluster centers.

   After the initial cluster assignments are made, proceed to the update step, below.

2. **Assignment step.** Assign each point to the cluster whose mean is the closest, using Euclidean distance. Mathematically:

$$c_i^{(t)} := \text{argmin}_j \|x^{(i)} - \mu_j\|$$

   where we note that the distance is given by the L_2, or Euclidean, norm.

3. **Update step.** Calculate the means to be the centroids of the points in the clusters.

$$\mu_j^{(t)} := \frac{\sum_{i=1}^{n} x^{(i)} \cdot \mathbb{I}\left\{c^{(i)} = j\right\}}{\sum_{i=1}^{n} \mathbb{I}\left\{c^{(i)} = j\right\}}$$

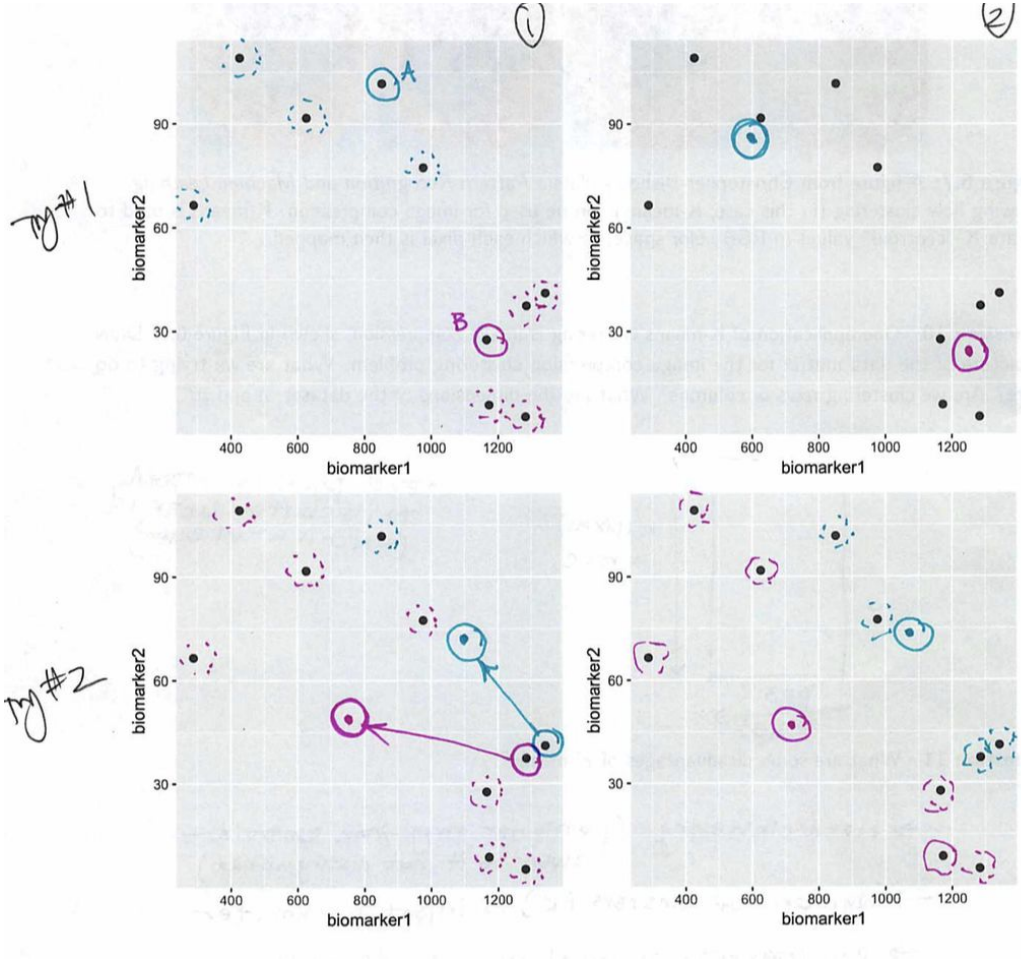
4. Repeat Steps 2 and 3 until no points change clusters (or until convergence).

Question 19.2

Which supervised learning algorithm does this remind you of? What is different between K-means and this algorithm?

Question 19.3

Below is our unlabeled, downsampled flow cytometry dataset. Cluster it into two groups using K-means. You can initialize your clusters however you want. (Note: Assume that the data were standardized in advance so that the spacing between the white lines equals one "unit" for both biomarker 1 and biomarker 2.)


(1)
(2)
A
B
biomarker2
biomarker1
90
60
30
400
600
800
1000
1200

Question 19.4

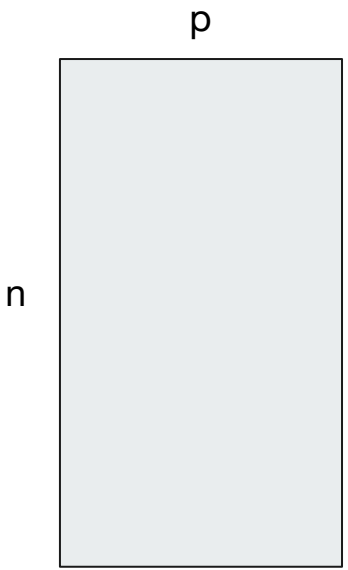
What are some disadvantages of K-means?

- Hard clustering
- Can converge to local minima
- K is input parameter (must choose)
- Assumes Euclidean distance makes sense
- Assumes axes on roughly equal scale

Question 19.5

One application of K-means clustering is image compression, as shown in the figure below (which is from Christopher Bishop's classic book *Pattern Recognition and Machine Learning*).

Draw a picture of the data matrix for the image compression clustering problem. How will the clustering work here? Are we clustering rows or columns? What are the dimensions of the dataset, n and p?

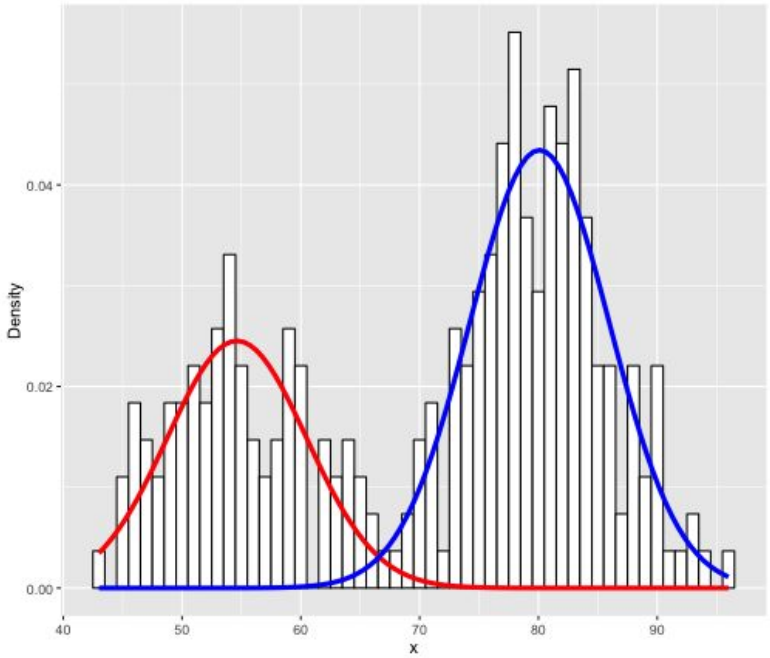


Mixture Models

Mixture models represent data using mixtures of simple probability distributions.

You need to fit:

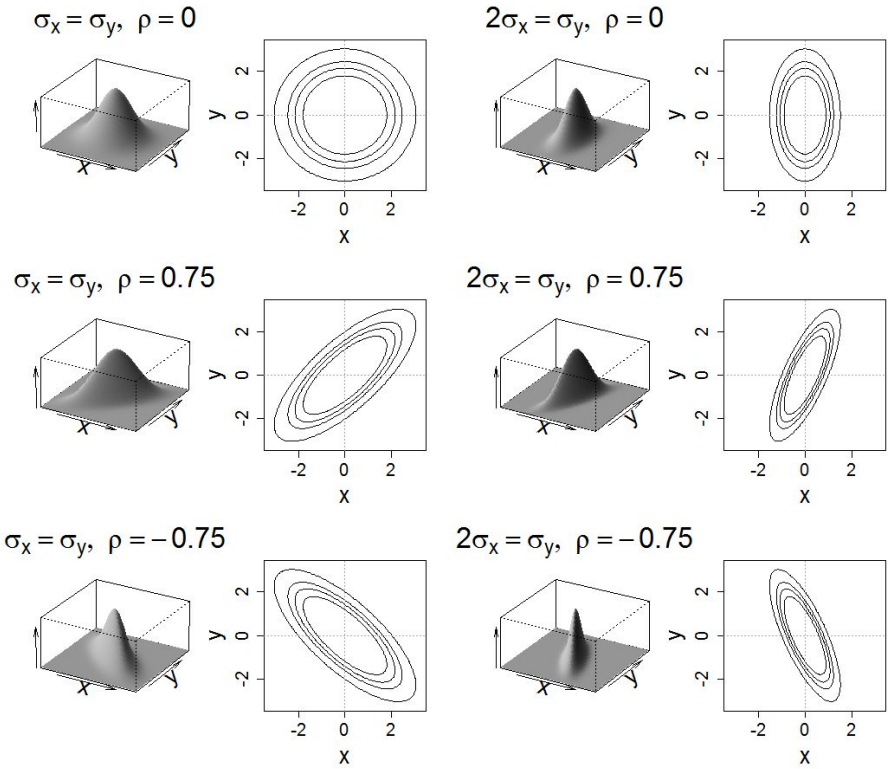
- K different distributions
- The “mixing parameters”



http://tinyheero.github.io/2015/10/13/mixture-model.html

Multivariate Gaussian

Just like the univariate Gaussian... now with more dimensions!

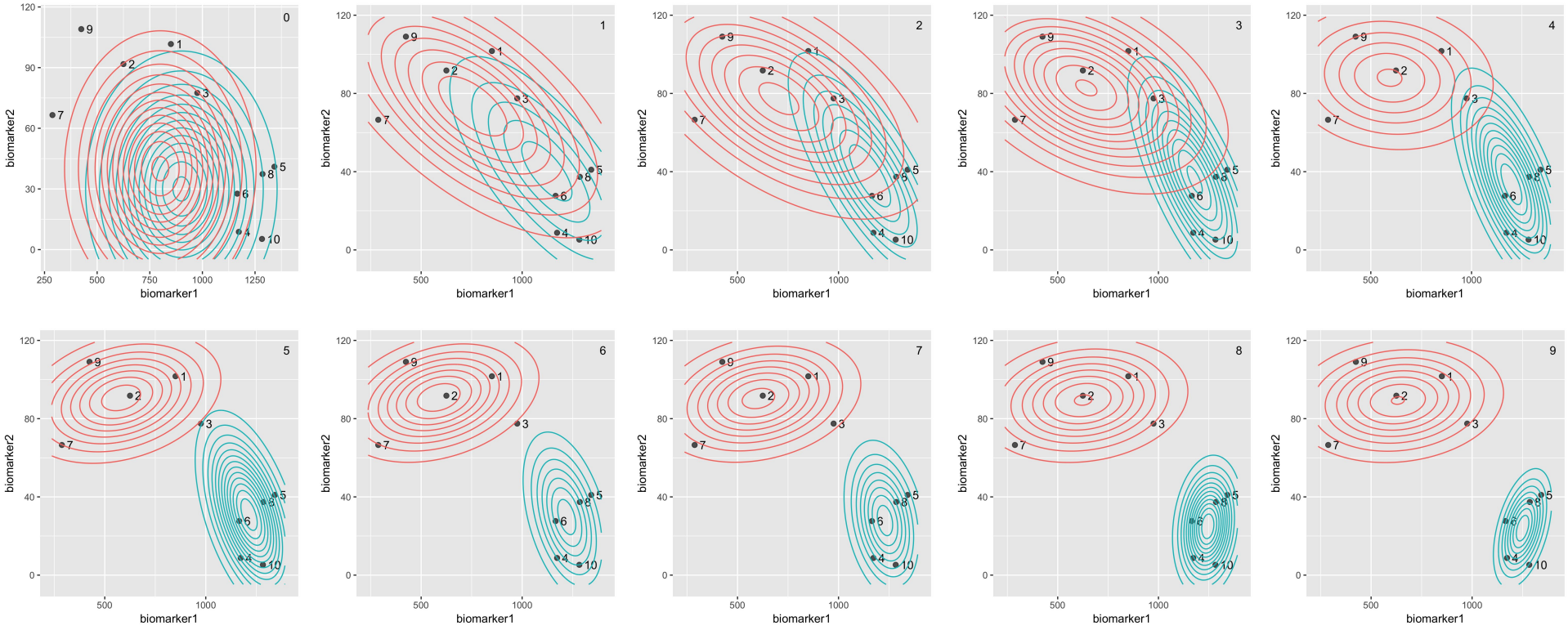


http://www.stat.cmu.edu/~kass/KEB/RHTML/R/bivariateNormalPerspectives.r.html

Gaussian Mixture Models

A Gaussian mixture model fits a set of unlabeled data to a set of K multivariate Gaussians. There are three sets of parameters that the algorithm needs to identify:

1. $\mu_1, \ldots, \mu_K$ (the means of the Gaussians)
2. $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K$ (the covariance matrices of the Gaussians)
3. $\phi_1, \ldots, \phi_K$ (the mixing proportions, which must sum to one)


0
1
2
3
4
5
6
7
8
9
biomarker1
biomarker2

Gaussian Mixtures

1. Initialize the means μ_k, covariances $\mathbf{\Sigma}_k$, and mixing coefficients ϕ_k for all of the Gaussians $k = 1, \ldots, K$.

2. **E step.** Give each point a "voting weight" in each Gaussian equal to the probability (based on current parameter values) that it came from that Gaussian:

$$\begin{aligned} w_j^{(i)} &:= p(z^{(i)} = j | x^{(i)}, \phi, \mu, \mathbf{\Sigma}) \\ &= \frac{\phi_j \cdot \mathcal{N}(x^{(i)} | \mu_j, \mathbf{\Sigma}_j)}{\sum_{k=1}^{K} \phi_k \cdot \mathcal{N}(x^{(i)} | \mu_k, \mathbf{\Sigma}_k)} \end{aligned}$$

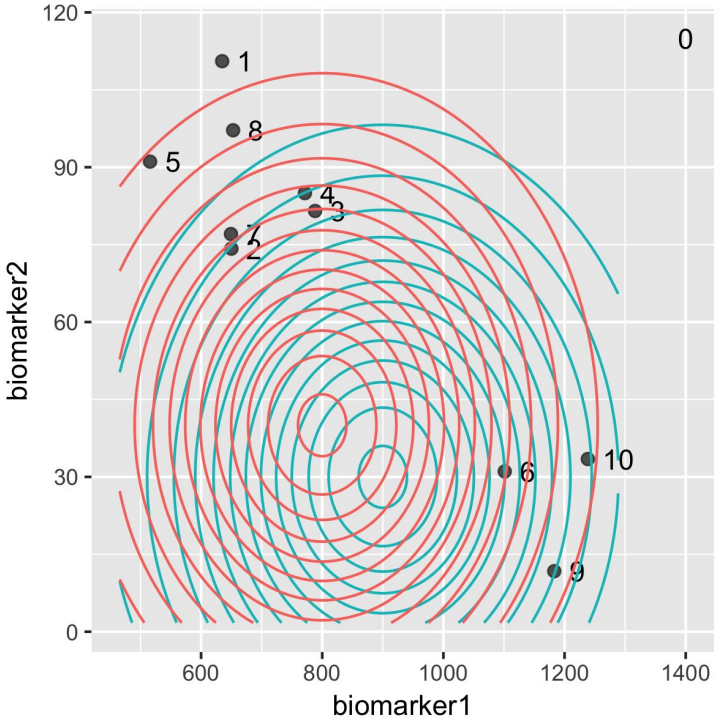
   Note that you will have K different voting weights for each point, and there are n points, so you need to do nK total calculations here.

3. **M step.** Re-estimate the parameters for the different Gaussians by letting each point vote in each Gaussian according to its voting weight.

$$\begin{aligned} \phi_j &:= \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)} \\ \mu_j &:= \frac{\sum_{i=1}^{n} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n} w_j^{(i)}} \\ \mathbf{\Sigma}_j &:= \frac{\sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} w_j^{(i)}} \end{aligned}$$

4. Check for convergence of either the parameters or the log-likelihood[1]. If the convergence criterion is not satisfied, return to step 2.

Step 1: Initialization



Step 1: Initialization

$$\phi_A = 0.5$$

$$\phi_B = 0.5$$

$$\mu_A = \begin{bmatrix} 900 \\ 30 \end{bmatrix}$$

$$\mu_B = \begin{bmatrix} 800 \\ 40 \end{bmatrix}$$

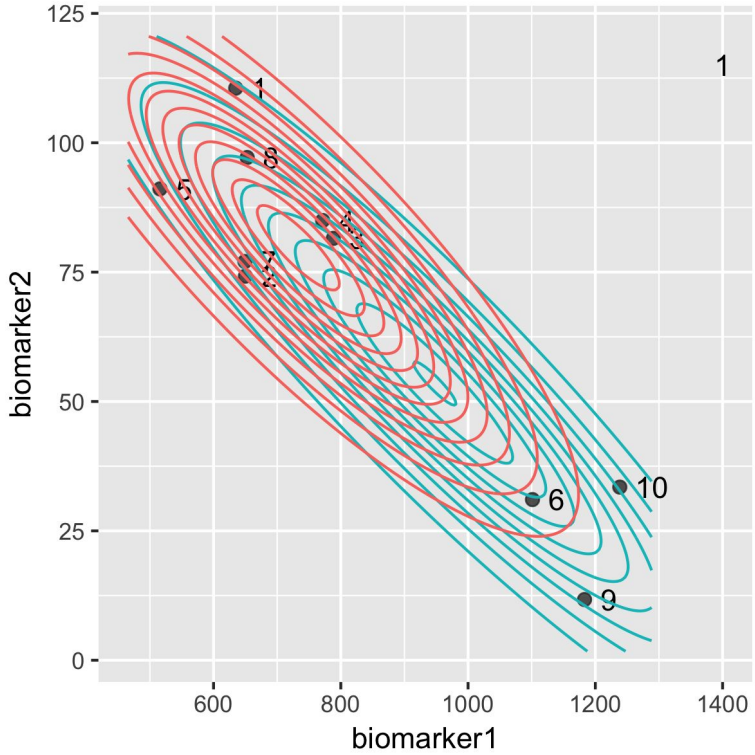
$$\boldsymbol{\Sigma}_A = \begin{bmatrix} 200^2 & 0 \\ 0 & 30^2 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_B = \begin{bmatrix} 200^2 & 0 \\ 0 & 30^2 \end{bmatrix}$$

Step 2: E-step

i	$x_1^{(i)}$	$x_2^{(i)}$	$\mathcal{N}(x^{(i)}\|\mu_A, \mathbf{\Sigma}_A)$	$\mathcal{N}(x^{(i)}\|\mu_B, \mathbf{\Sigma}_B)$	$w_A^{(i)}$	$w_B^{(i)}$
1	634.83	110.55	3e-07	1.2e-06	0.201	0.799
2	650.06	74.22	4.1e-06	1e-05	0.282	0.718
3	788.24	81.52	5.2e-06	1e-05	0.338	0.662
4	771.47	84.98	4e-06	8.5e-06	0.320	0.680
5	515.81	91.08	5.3e-07	2.3e-06	0.189	0.811
6	1101.23	31.05	1.6e-05	8.2e-06	0.662	0.338
7	649.32	77.05	3.5e-06	9.3e-06	0.275	0.725
8	652.89	97.16	1e-06	3.3e-06	0.234	0.766
9	1183.02	11.73	8.1e-06	2.7e-06	0.749	0.251
10	1238.45	33.46	6.3e-06	2.3e-06	0.729	0.271
sum					3.979	6.021

Step 3: M-step


125
100
75
50
25
0
biomarker2
600
800
1000
1200
1400
biomarker1
1
2
3
4
5
6
7
8
9
10

Step 3: M-step

$$\phi_A = 0.398$$

$$\phi_B = 0.602$$

$$\mu_A = \begin{bmatrix} 947.6 \\ 53.5 \end{bmatrix}$$

$$\mu_B = \begin{bmatrix} 733.2 \\ 79.7 \end{bmatrix}$$

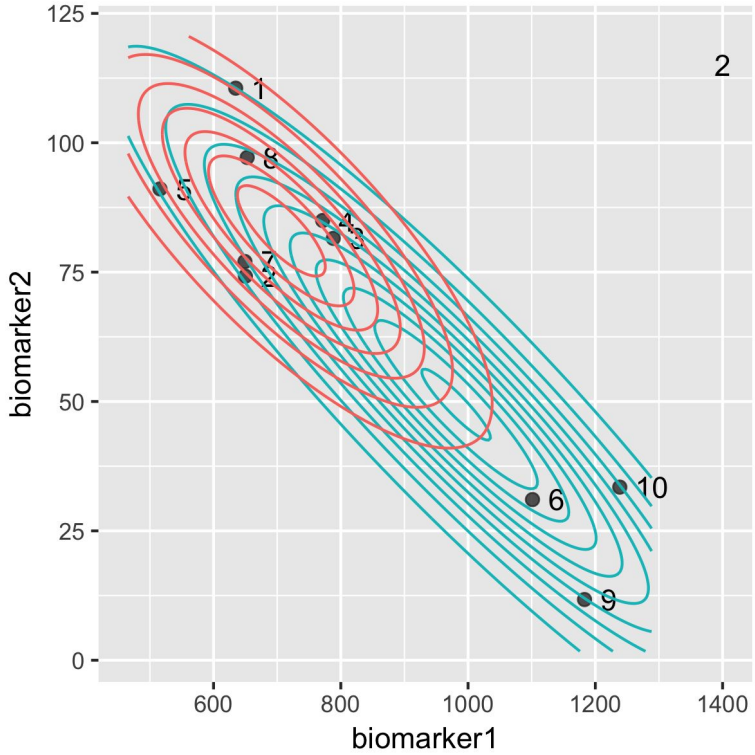
$$\mathbf{\Sigma}_A = \begin{bmatrix} 256.6^2 & -0.925 \cdot 256.6 \cdot 32.3 \\ -0.925 \cdot 256.6 \cdot 32.3 & 32.3^2 \end{bmatrix}$$

$$\mathbf{\Sigma}_B = \begin{bmatrix} 195.4^2 & -0.855 \cdot 195.4 \cdot 24.7 \\ -0.855 \cdot 195.4 \cdot 24.7 & 24.7^2 \end{bmatrix}$$

Step 4: E-step

i	$x_1^{(i)}$	$x_2^{(i)}$	$\mathcal{N}(x^{(i)}\|\mu_A, \mathbf{\Sigma}_A)$	$\mathcal{N}(x^{(i)}\|\mu_B, \mathbf{\Sigma}_B)$	$w_A^{(i)}$	$w_B^{(i)}$
1	634.83	110.55	5.9e-06	1.6e-05	0.193	0.807
2	650.06	74.22	1.4e-05	3.1e-05	0.226	0.774
3	788.24	81.52	3.1e-05	5.1e-05	0.287	0.713
4	771.47	84.98	2.7e-05	4.8e-05	0.271	0.729
5	515.81	91.08	7.2e-06	2.2e-05	0.178	0.822
6	1101.23	31.05	3.9e-05	8.5e-06	0.754	0.246
7	649.32	77.05	1.7e-05	3.8e-05	0.227	0.773
8	652.89	97.16	2e-05	4.6e-05	0.219	0.781
9	1183.02	11.73	1.7e-05	1.5e-06	0.884	0.116
10	1238.45	33.46	1.4e-05	1.8e-06	0.837	0.163
sum					4.078	5.922

Step 5: M-step


125
100
75
50
25
0
600
800
1000
1200
1400
biomarker2
biomarker1
2
1
8
5
3
4
7
6
10
9

Step 5: M-step

$$\phi_A = 0.408$$

$$\phi_B = 0.592$$

$$\mu_A = \begin{bmatrix} 981.2 \\ 49.4 \end{bmatrix}$$

$$\mu_B = \begin{bmatrix} 706.5 \\ 83 \end{bmatrix}$$

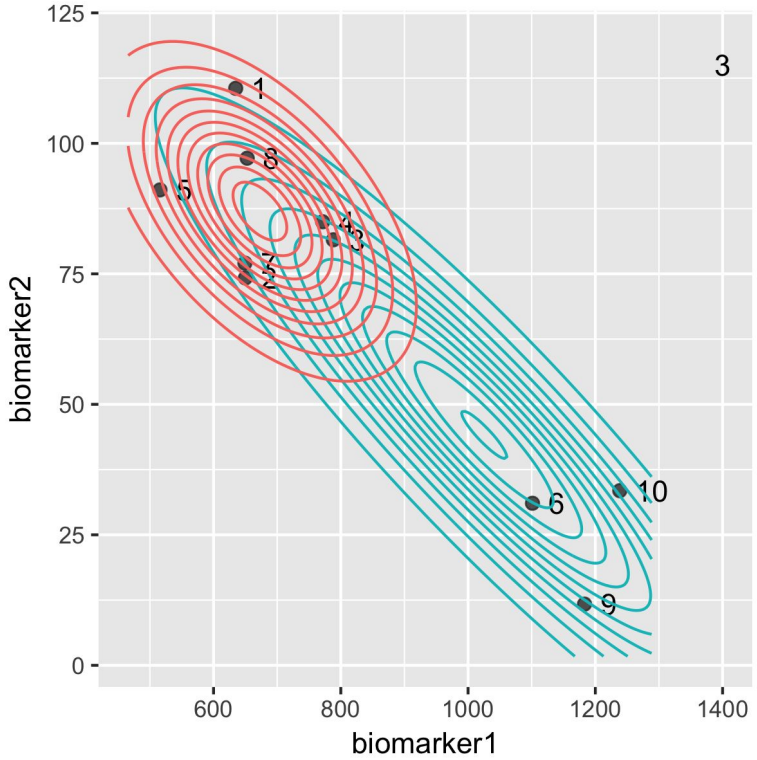
$$\mathbf{\Sigma}_A = \begin{bmatrix} 252.6^2 & -0.924 \cdot 252.6 \cdot 32.1 \\ -0.924 \cdot 252.6 \cdot 32.1 & 32.1^2 \end{bmatrix}$$

$$\mathbf{\Sigma}_B = \begin{bmatrix} 164.3^2 & -0.793 \cdot 164.3 \cdot 20.8 \\ -0.793 \cdot 164.3 \cdot 20.8 & 20.8^2 \end{bmatrix}$$

Step 6: E-step

i	$x_1^{(i)}$	$x_2^{(i)}$	$\mathcal{N}(x^{(i)}\|\mu_A, \mathbf{\Sigma}_A)$	$\mathcal{N}(x^{(i)}\|\mu_B, \mathbf{\Sigma}_B)$	$w_A^{(i)}$	$w_B^{(i)}$
1	634.83	110.55	5e-06	1.9e-05	0.153	0.847
2	650.06	74.22	1.1e-05	3.8e-05	0.171	0.829
3	788.24	81.52	2.8e-05	5.9e-05	0.250	0.750
4	771.47	84.98	2.4e-05	5.6e-05	0.230	0.770
5	515.81	91.08	5.4e-06	2.7e-05	0.122	0.878
6	1101.23	31.05	4.3e-05	2.7e-06	0.917	0.083
7	649.32	77.05	1.4e-05	4.7e-05	0.171	0.829
8	652.89	97.16	1.7e-05	5.7e-05	0.167	0.833
9	1183.02	11.73	2e-05	2.1e-07	0.985	0.015
10	1238.45	33.46	1.6e-05	3.8e-07	0.965	0.035
sum					4.132	5.868

Step 7: M-step


125
100
75
50
25
0
biomarker2
600
800
1000
1200
1400
biomarker1
1
2
3
4
5
6
7
8
9
10

Step 7: M-step

$$\phi_A = 0.413$$

$$\phi_B = 0.587$$

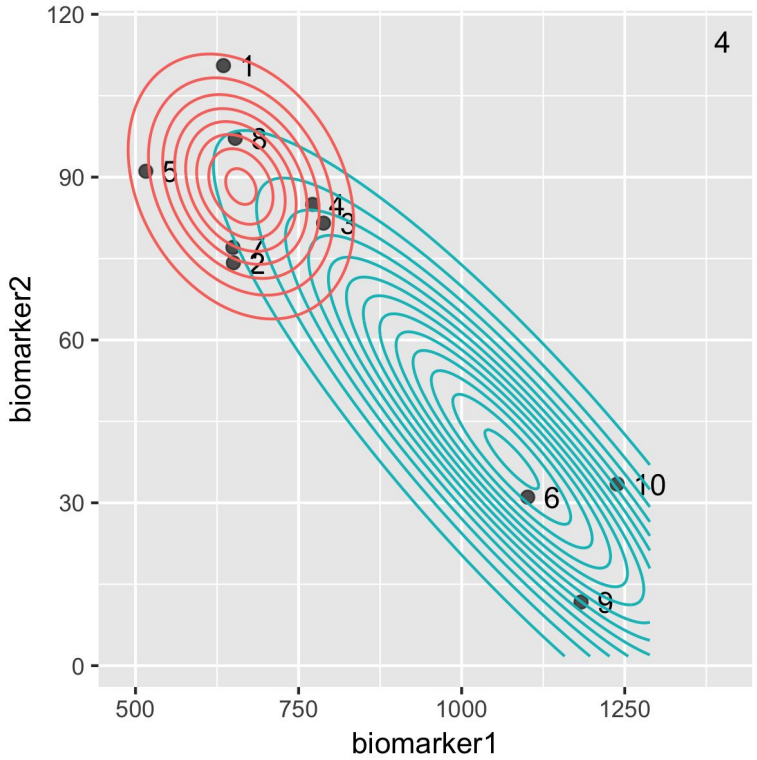
$$\mu_A = \begin{bmatrix} 1025.3 \\ 44.2 \end{bmatrix}$$

$$\mu_B = \begin{bmatrix} 672.9 \\ 87 \end{bmatrix}$$

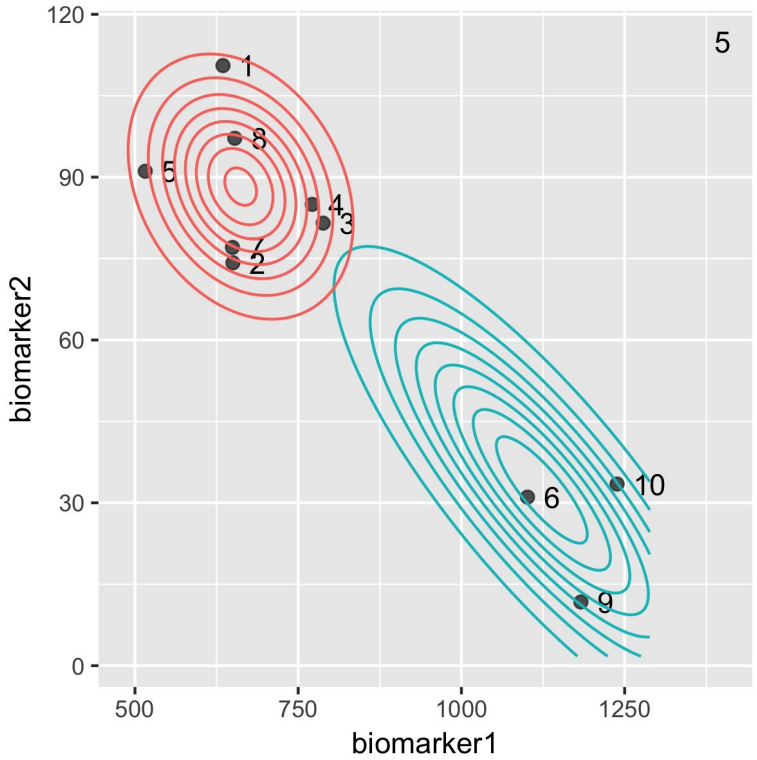
$$\mathbf{\Sigma}_A = \begin{bmatrix} 235.5^2 & -0.916 \cdot 235.5 \cdot 30.3 \\ -0.916 \cdot 235.5 \cdot 30.3 & 30.3^2 \end{bmatrix}$$

$$\mathbf{\Sigma}_B = \begin{bmatrix} 110.6^2 & -0.558 \cdot 110.6 \cdot 14.6 \\ -0.558 \cdot 110.6 \cdot 14.6 & 14.6^2 \end{bmatrix}$$

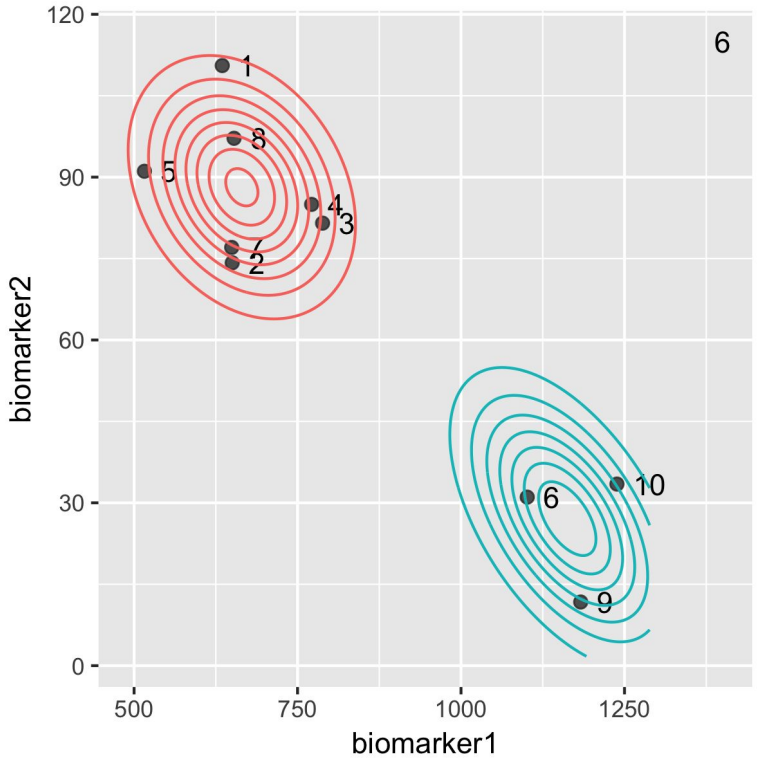
Do EM again...


120
90
60
30
0
biomarker2
500
750
1000
1250
biomarker1
1
2
3
4
5
6
7
8
9
10

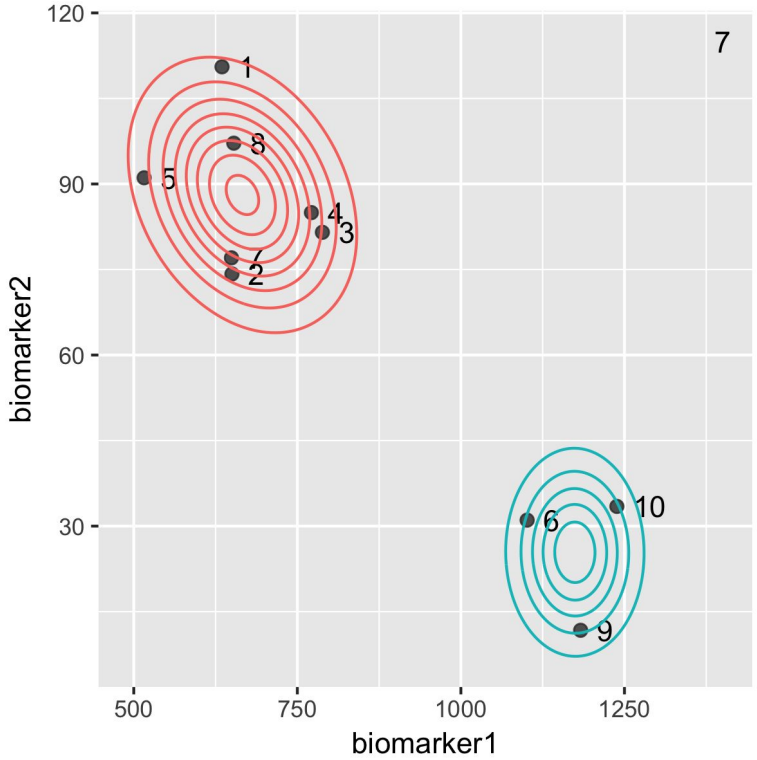
And again...


120
90
60
30
0
biomarker2
500
750
1000
1250
biomarker1
1
8
5
4
3
7
2
6
10
9
5

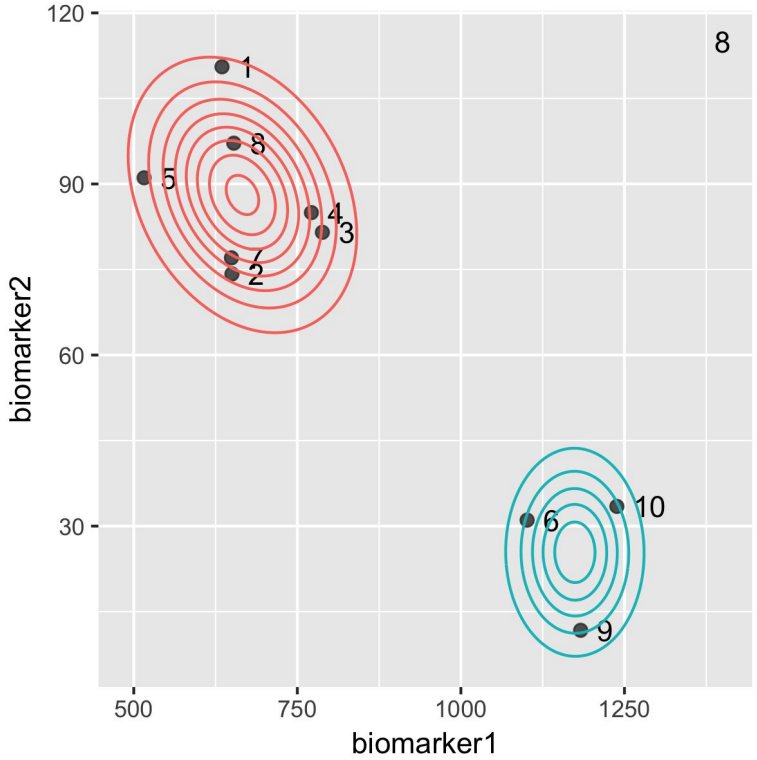
And again...


120
90
60
30
0
500
750
1000
1250
biomarker1
biomarker2
1
8
5
4
3
7
2
6
6
10
9

And again...


120
90
60
30
biomarker2
500
750
1000
1250
biomarker1
1
2
3
4
5
6
7
8
9
10

And again...


120
90
60
30
biomarker2
500
750
1000
1250
biomarker1
1
2
3
4
5
6
7
8
9
10

Final E-Step

i	$x_1^{(i)}$	$x_2^{(i)}$	$\mathcal{N}(x^{(i)}\|\mu_A, \mathbf{\Sigma}_A)$	$\mathcal{N}(x^{(i)}\|\mu_B, \mathbf{\Sigma}_B)$	$w_A^{(i)}$	$w_B^{(i)}$
1	634.83	110.55	1.8e-40	2.6e-05	0.000	1.000
2	650.06	74.22	2.5e-28	7.1e-05	0.000	1.000
3	788.24	81.52	1.6e-21	5.8e-05	0.000	1.000
4	771.47	84.98	2.5e-23	7.7e-05	0.000	1.000
5	515.81	91.08	1.7e-43	3.4e-05	0.000	1.000
6	1101.23	31.05	0.00011	6e-13	1.000	0.000
7	649.32	77.05	5e-29	9.5e-05	0.000	1.000
8	652.89	97.16	2.2e-34	0.00012	0.000	1.000
9	1183.02	11.73	0.00011	6e-18	1.000	0.000
10	1238.45	33.46	0.00011	3.7e-16	1.000	0.000
sum					3.000	7.000

Final Parameters

$$\phi_A = 0.30$$

$$\phi_B = 0.70$$

$$\mu_A = \begin{bmatrix} 1174.2 \\ 25.4 \end{bmatrix}$$

$$\mu_B = \begin{bmatrix} 666.1 \\ 88.1 \end{bmatrix}$$

$$\mathbf{\Sigma}_A = \begin{bmatrix} 56.4^2 & -0.009 \cdot 56.4 \cdot 9.7 \\ -0.009 \cdot 56.4 \cdot 9.7 & 9.7^2 \end{bmatrix} = \begin{bmatrix} 3176.8 & -5.0 \\ -5.0 & 94.6 \end{bmatrix}$$

$$\mathbf{\Sigma}_B = \begin{bmatrix} 84.8^2 & -0.287 \cdot 84.8 \cdot 11.7 \\ -0.287 \cdot 84.8 \cdot 11.7 & 11.7^2 \end{bmatrix} = \begin{bmatrix} 7185.8 & -284.8 \\ -284.8 & 137.5 \end{bmatrix}$$

Mixture models vs. K-means

Similarities:

- The number of clusters, K, is still an input parameter.
- Mixture models can still converge to local minima depending on the initialization.

Differences:

- **Soft clustering** instead of hard clustering
- Points distributed over clusters using **likelihood**, not Euclidean distance.
- Mixture components don’t have to be Gaussian (e.g. topic models).

Question 19.6

Mixture models are examples of **generative models**, which tell a story about how the observed data were generated. Below is the actual code that generated the data for the flow cytometry example.

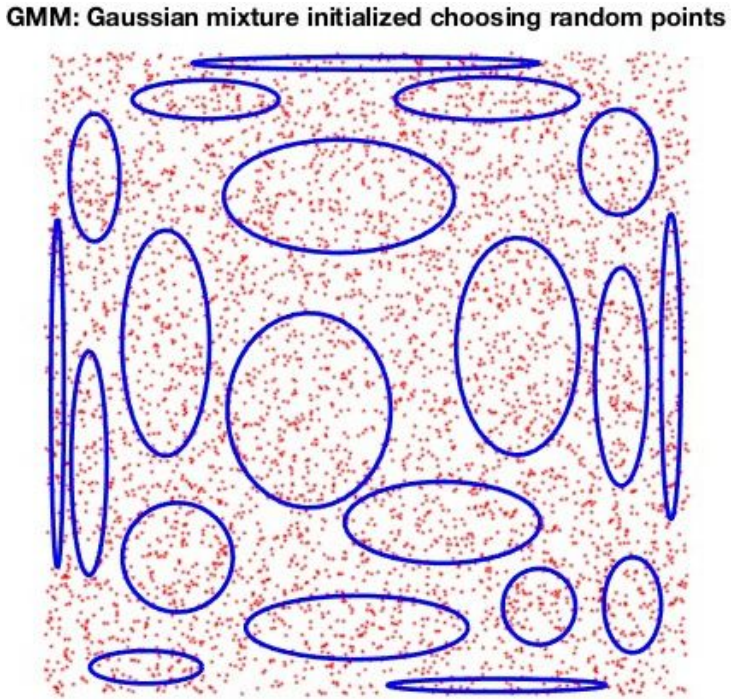
```
```{r Sampling from two cell lines}
cell_line_sample <- function(N) {
  mean.0 <- c(1200, 25)
  mean.1 <- c(750, 80)
  cov.0 <- matrix(c(100^2, 0.6*100*15, 0.6*100*15, 15^2), nrow=2)
  cov.1 <- matrix(c(200^2, -0.4*200*10, -0.4*200*10, 20^2), nrow=2)
  p.group.0 <- 0.4

  group <- runif(N)
  rand.samples = {}
  for(i in 1:N){
      if (group[i] < p.group.0) {
          rand.samples <- rbind(rand.samples, mvrnorm(1, mean.0, cov.0))
      } else {
          rand.samples <- rbind(rand.samples, mvrnorm(1, mean.1, cov.1))
      }
  }
  rand.samples <- as.data.frame(rand.samples)
  names(rand.samples) <- c("biomarker1", "biomarker2")
  rand.samples$group <- group < p.group.0
  rand.samples$name <- seq(1, N)
  return(rand.samples)
}
```
```

It turns out that this code matches the "story" of the Gaussian mixture model perfectly. (With real data, of course, this would not be the case.) What is that story?

Some Final Notes on Mixture Models

Initialization matters. Also, if your data aren't actually separable, you can end up with strange results.



http://www.vlfeat.org/overview/gmm.html

Question 19.7

Compare these parameters to the values from the code that generated the data (Question 19.6). What do you notice?

Question 19.8

Think of 2-3 different unsupervised learning problems from biology or medicine where a mixture model makes sense, conceptually at least, for modeling the data. How would you set up the mixture model in each case?