

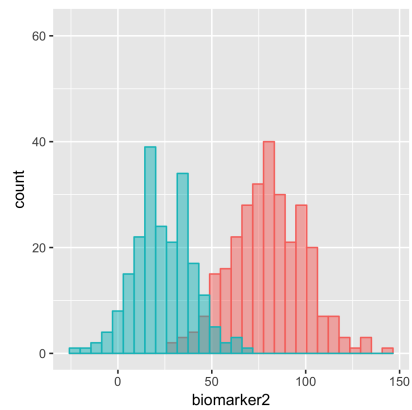
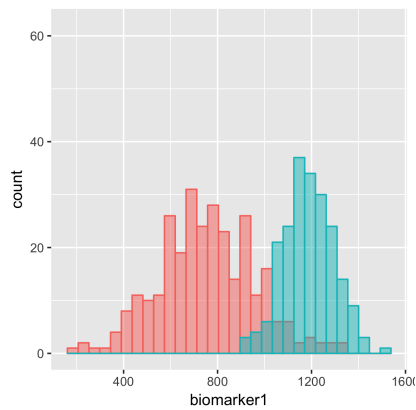
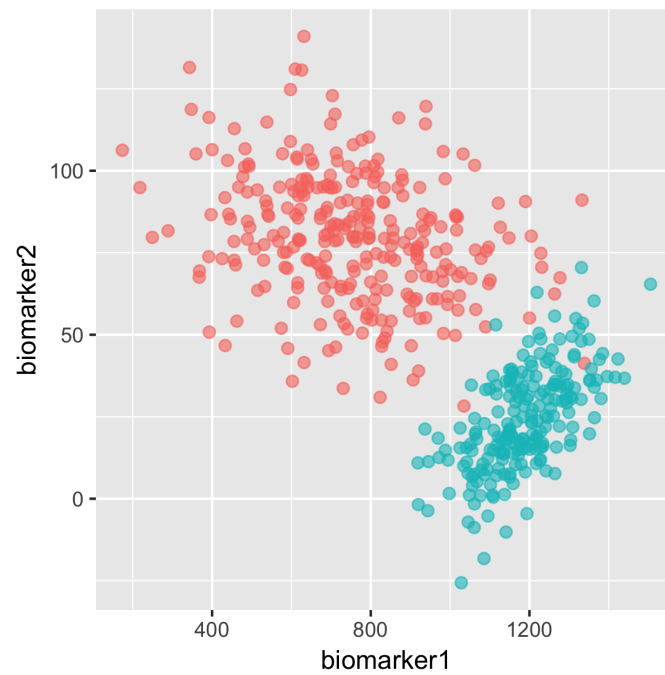
## Chapter 19

# Clustering

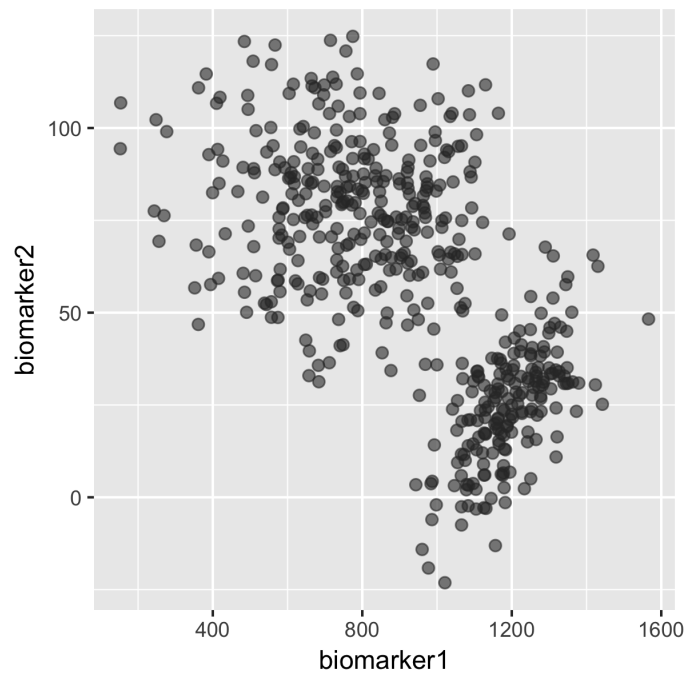
Our discussions so far have focused on the problem of supervised learning, in which we create a mapping between a set of inputs and an output. In this chapter, we turn our attention to the problem of **unsupervised learning**, in which our goal is to uncover hidden structure in a dataset. There is no “special” outcome variable in these types of problems. Way back in Chapter 1, examples 7, 8, and 14 were examples of unsupervised learning problems.

### 19.1 A Thought Experiment: Flow Cytometry Data

Imagine you have data on 500 cells from two different cell lines. For each cell, you record the fluorescence intensity of two different biomarkers, biomarker 1 ( $x_1$ ) and biomarker 2 ( $x_2$ ). The data are shown below.



In real life, you would not have the labels for the two cell lines. You would have no idea which distribution(s) the data were drawn from or what the probabilities associated with the various distributions were. Real flow cytometry data would instead look like this:



The human eye can distinguish structure in a plot like this, but it's harder to train a computer to see it, and it's even harder to prove that the structure the computer finds is real. Over the years, people have tried many different strategies.

#### Question 19.1

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

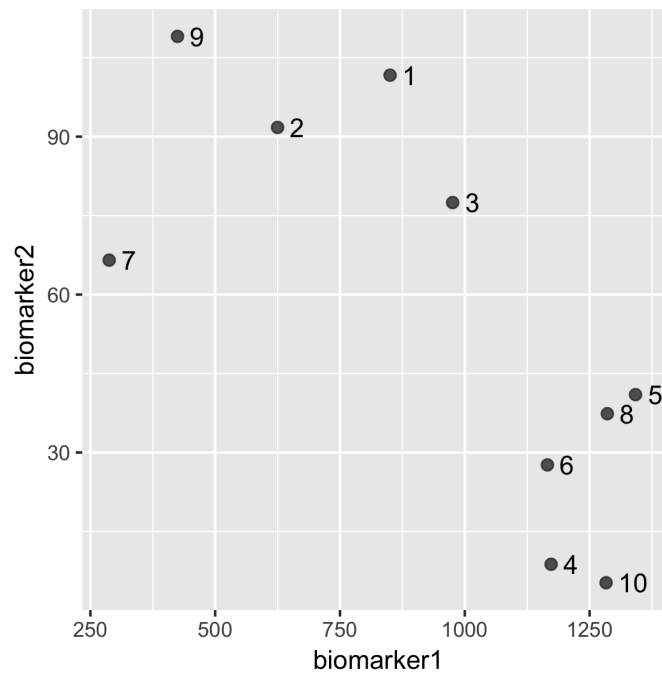
### 19.1.1 Downsampled Dataset

For the rest of this workshop, we need to be able to calculate things by hand, so we will downsample the flow cytometry data, above, to just 10 datapoints. The downsampled, labeled data are shown here:

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

Without their labels, the data look like this:

$i$	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	



## 19.2 K-Means

Let's first consider a very simple unsupervised machine learning algorithm called **K-means**. The goal of K-means is to cluster [unlabeled] data into groups so that the distances between points within a group are minimized.

Assume you have a dataset  $\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ , where each vector  $x^{(i)}$  is of length  $p$ , and you want to cluster these  $n$  vectors into  $K$  distinct groups. Here is 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:

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

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.

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

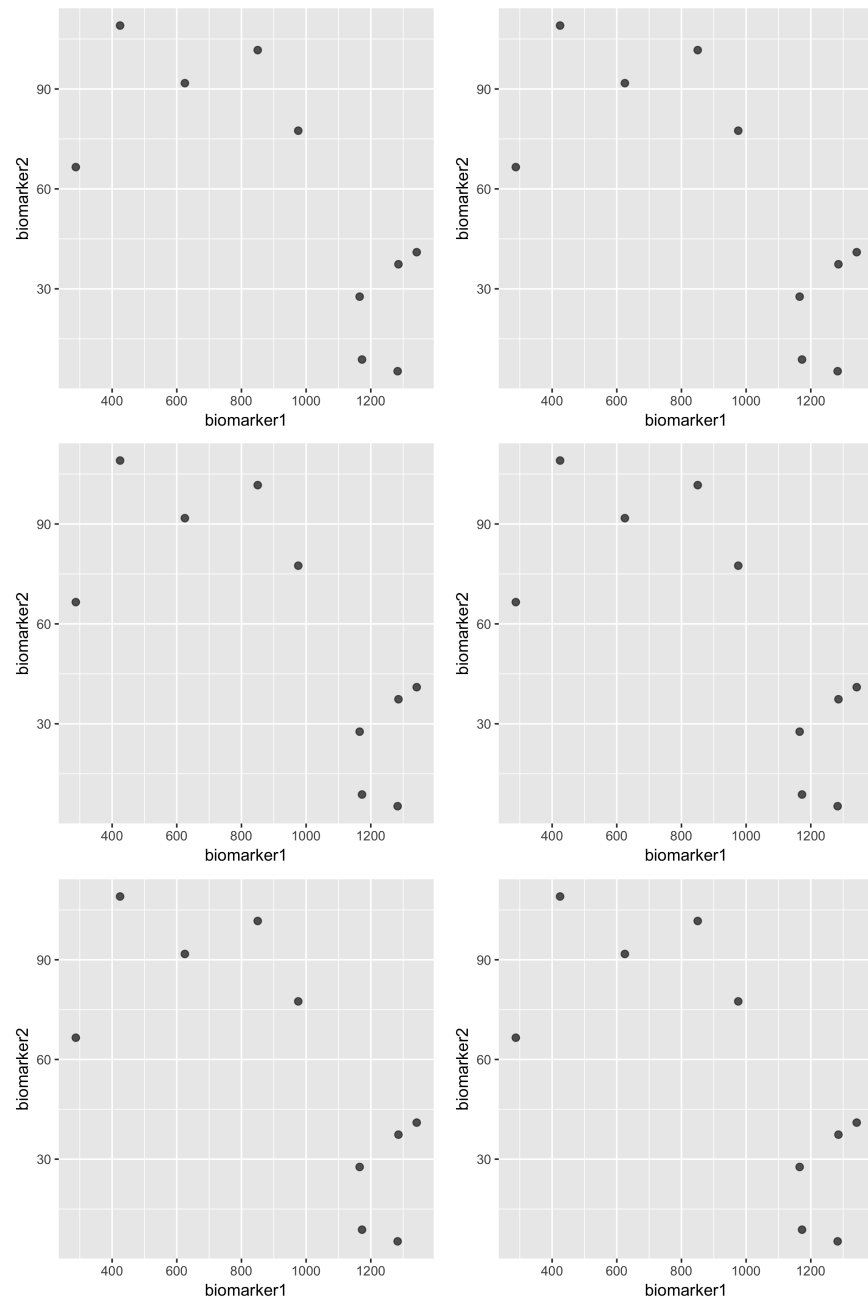
4. Repeat Steps 2 and 3 until no points change clusters (or until convergence).

#### 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.)



#### Question 19.4

What are some disadvantages of K-means?

### 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  $p$ ?

## 19.3 Mixture Models

Mixture models represent data using mixtures of simple probability distributions. They are a step up in methodological rigor from K-means and are much more flexible, although the basic idea is the same.

Here are the key similarities:

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

Here are the key differences:

1. Mixture models use a **soft clustering** instead of a hard clustering. Each datapoint is assigned a probability distribution over potential clusters.



2. Instead of a distance metric like Euclidean distance, points are distributed over clusters by considering probability densities of the various clusters and how likely it is that the point came from each probability density.
3. No need to assume axes are on an equal scale; the individual probability distributions can account for this.

We'll examine mixtures of Gaussians today because they're easy to visualize, but in reality the mixture components can be any type of well-behaved probability distribution.

## 19.4 Multivariate Gaussian Distribution

We already saw the univariate Gaussian in Chapter 4. The **multivariate Gaussian** is an extension of the Gaussian to multiple dimensions.

### 19.4.1 Probability Density and Parameters

The  $m$ -dimensional **multivariate Gaussian** probability distribution is given by:

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

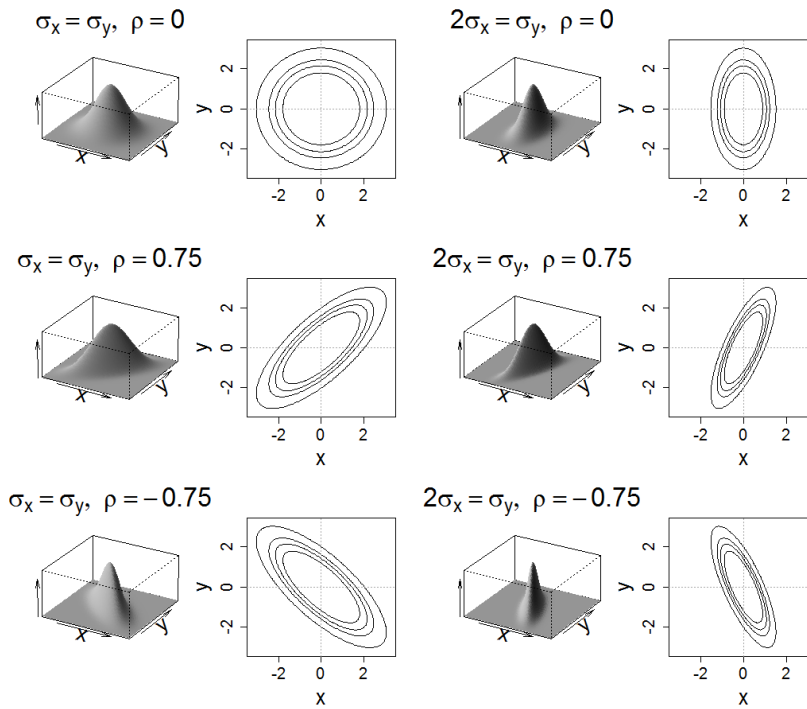
where  $x \in \mathbb{R}^m$  (a vector of length  $m$ ) and the mean,  $\mu$ , is also a vector of length  $m$ . The variance of a univariate Gaussian,  $\sigma$ , is replaced by a covariance matrix of dimension  $m \times m$ :

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \dots & \rho_{1m}\sigma_1\sigma_m \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{m1}\sigma_m\sigma_1 & \rho_{m2}\sigma_m\sigma_2 & \dots & \sigma_m^2 \end{bmatrix}$$

where  $\rho_{ij}$  is the Pearson correlation of  $X_i$  and  $X_j$ . Or alternatively, the  $ij$ th element of the covariance matrix is  $\text{cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$ .

In two dimensions, the cross-sections of the multivariate Gaussian distribution are ellipses. The axes of the ellipses are given by the eigenvectors of the covariance matrix,  $\Sigma$ . The first and second eigenvalues of the covariance matrix give the variance of the data along the major and minor axes of the ellipses, respectively. Some examples of bivariate normal distributions are shown below. This figure and the code that generated it can be found here:

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



## 19.4.2 Maximum Likelihood Estimates

To fit data to a multivariate Gaussian distribution, we once again use our old friend, maximum likelihood estimation (Chapter 5). Here are the maximum likelihood estimates for  $\mu$  and  $\Sigma$  for the multivariate Gaussian:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x^{(i)} \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (x^{(i)} - \hat{\mu})(x^{(i)} - \hat{\mu})^T$$

## 19.5 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.  $\mu_1, \dots, \mu_K$  (the means of the Gaussians)
2.  $\Sigma_1, \dots, \Sigma_K$  (the covariance matrices of the Gaussians)
3.  $\phi_1, \dots, \phi_K$  (the mixing proportions, which must sum to one)

### 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?

Here is the algorithm for fitting a Gaussian mixture model:

1. Initialize the means  $\mu_k$ , covariances  $\Sigma_k$ , and mixing coefficients  $\phi_k$  for all of the Gaussians  $k = 1, \dots, K$ .
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:

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

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.

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

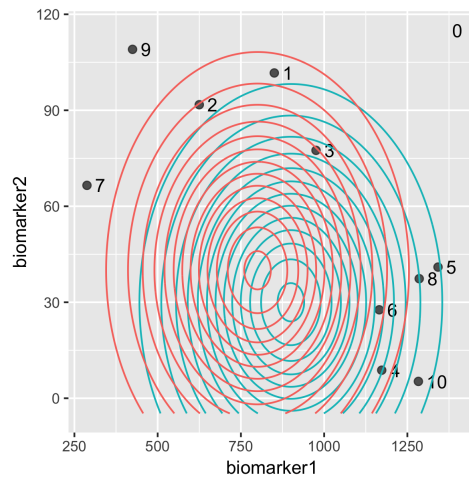
4. Check for convergence of either the parameters or the log-likelihood. If the convergence criterion is not satisfied, return to step 2.

Note that you are *not* guaranteed to get the right answer; the final Gaussians can change depending on how you initialize the model.

## 19.6 A Gaussian Mixture Model for Flow Cytometry Data

We will now follow the steps from the algorithm above to fit a Gaussian mixture model to the dataset from our flow cytometry example.

1. Initialize the means  $\mu_A$  and  $\mu_B$ , the covariances  $\Sigma_A$  and  $\Sigma_B$ , and the mixing coefficients  $\phi_A$  and  $\phi_B$ .

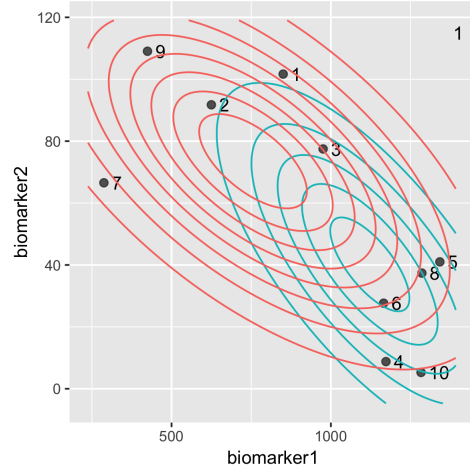


$$\begin{aligned}\phi_A &= 0.5 \\ \phi_B &= 0.5 \\ \mu_A &= \begin{bmatrix} 900 \\ 30 \end{bmatrix} \\ \mu_B &= \begin{bmatrix} 800 \\ 40 \end{bmatrix} \\ \Sigma_A &= \begin{bmatrix} 200^2 & 0 \\ 0 & 30^2 \end{bmatrix} \\ \Sigma_B &= \begin{bmatrix} 200^2 & 0 \\ 0 & 30^2 \end{bmatrix}\end{aligned}$$

2. Do E step for round 1.

| $i$ | $x_1^{(i)}$ | $x_2^{(i)}$ | $\mathcal{N}(x^{(i)} \mu_A, \Sigma_A)$ | $\mathcal{N}(x^{(i)} \mu_B, \Sigma_B)$ | $w_A^{(i)}$ | $w_B^{(i)}$ |
|-----|-------------|-------------|--|--|-------------|-------------|
| 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       |

3. Do M step for round 1.



$$\phi_A = 0.398$$

$$\phi_B = 0.602$$

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

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

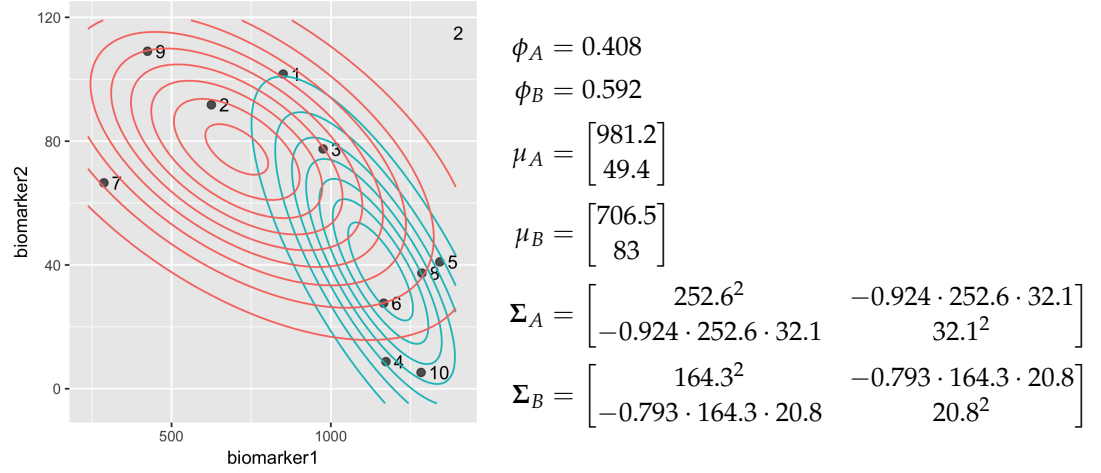
$$\Sigma_A = \begin{bmatrix} 256.6^2 & -0.925 \cdot 256.6 \cdot 32.3 \\ -0.925 \cdot 256.6 \cdot 32.3 & 32.3^2 \end{bmatrix}$$

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

4. Do E step for round 2.

| $i$ | $x_1^{(i)}$ | $x_2^{(i)}$ | $\mathcal{N}(x^{(i)} \mu_A, \Sigma_A)$ | $\mathcal{N}(x^{(i)} \mu_B, \Sigma_B)$ | $w_A^{(i)}$ | $w_B^{(i)}$ |
|-----|-------------|-------------|--|--|-------------|-------------|
| 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       |

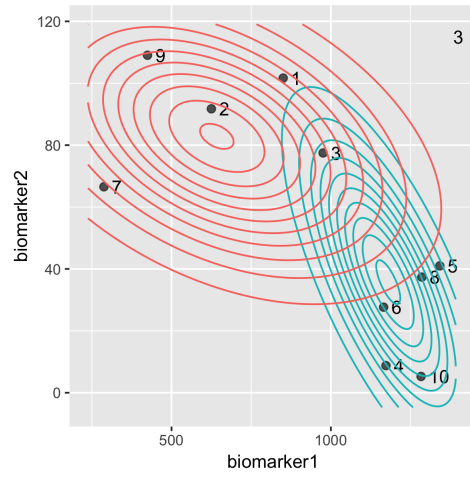
5. Do M step for round 2. Calculate the new log-likelihood.



6. Do E step for round 3.

| $i$ | $x_1^{(i)}$ | $x_2^{(i)}$ | $\mathcal{N}(x^{(i)} \mu_A, \Sigma_A)$ | $\mathcal{N}(x^{(i)} \mu_B, \Sigma_B)$ | $w_A^{(i)}$ | $w_B^{(i)}$ |
|-----|-------------|-------------|--|--|-------------|-------------|
| 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       |

7. Do M step for round 3.



$$\phi_A = 0.413$$

$$\phi_B = 0.587$$

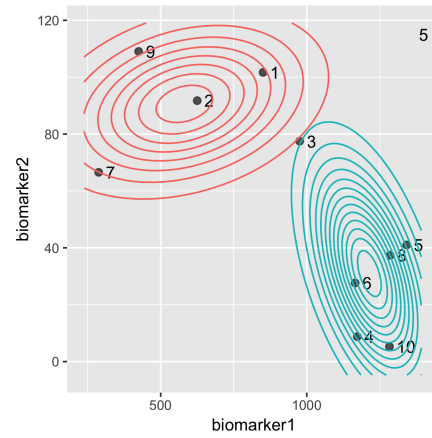
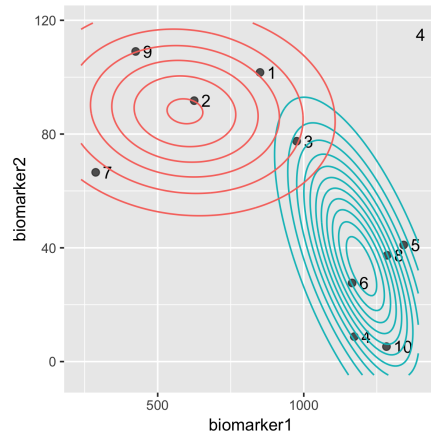
$$\mu_A = \begin{bmatrix} 1025.3 \\ 44.2 \end{bmatrix}$$

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

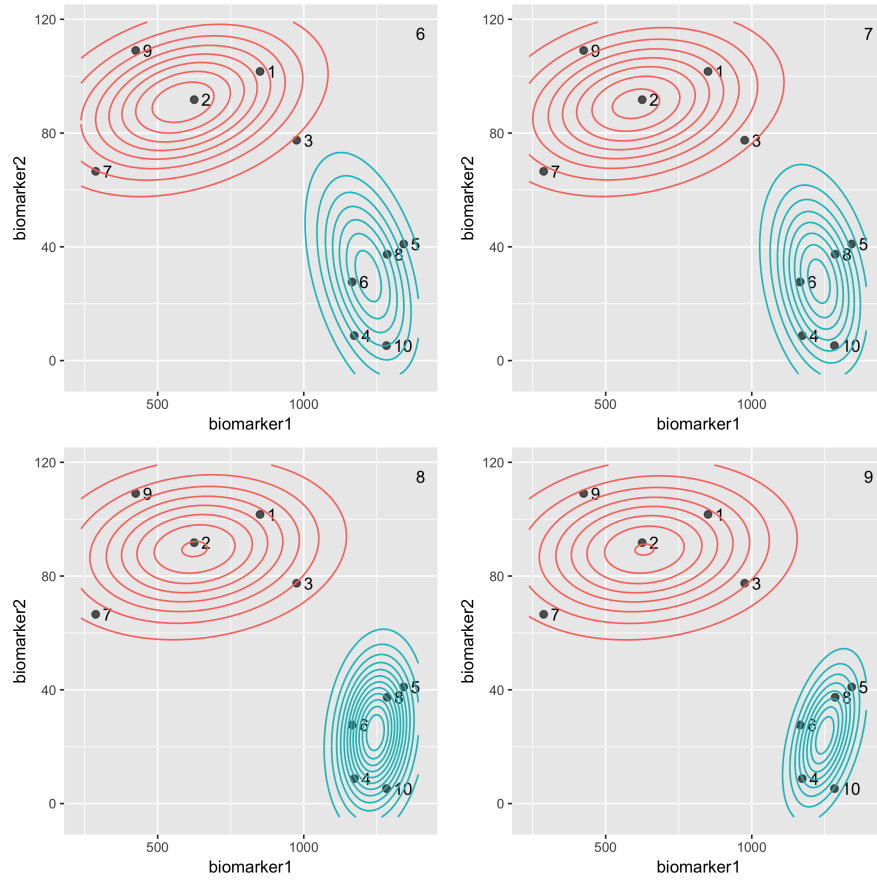
$$\Sigma_A = \begin{bmatrix} 235.5^2 & -0.916 \cdot 235.5 \cdot 30.3 \\ -0.916 \cdot 235.5 \cdot 30.3 & 30.3^2 \end{bmatrix}$$

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

8. Do EM rounds 4 through 9.







Here is the table of calculations for the E-step after round 9:

| $i$ | $x_1^{(i)}$ | $x_2^{(i)}$ | $\mathcal{N}(x^{(i)} \mu_A, \Sigma_A)$ | $\mathcal{N}(x^{(i)} \mu_B, \Sigma_B)$ | $w_A^{(i)}$ | $w_B^{(i)}$ |
|-----|-------------|-------------|--|--|-------------|-------------|
| 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       |

The values of the final parameters are:

$$\phi_A = 0.30$$

$$\phi_B = 0.70$$

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

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

$$\Sigma_A = \begin{bmatrix} 56.4^2 & -0.009 \cdot 56.4 \cdot 9.7 \\ -0.009 \cdot 56.4 \cdot 9.7 & 9.7^2 \end{bmatrix}$$

$$= \begin{bmatrix} 3176.8 & -5.0 \\ -5.0 & 94.6 \end{bmatrix}$$

$$\Sigma_B = \begin{bmatrix} 84.8^2 & -0.287 \cdot 84.8 \cdot 11.7 \\ -0.287 \cdot 84.8 \cdot 11.7 & 11.7^2 \end{bmatrix}$$

$$= \begin{bmatrix} 7185.8 & -284.8 \\ -284.8 & 137.5 \end{bmatrix}$$

#### 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?

## 19.7 The Expectation-Maximization Algorithm

Given a joint distribution  $p(x, z|\theta)$  over observed variables  $X$  and latent variables  $Z$ , governed by parameters  $\theta$ , the goal of the EM algorithm is to maximize the likelihood function  $p(x|\theta)$  with respect to  $\theta$ . Mixture models are an

example of the EM algorithm. Here is its general form:

1. Choose an initial setting for the parameters  $\theta$ .
2. **E step.** For each  $i$ , set

$$Q_i(z^{(i)}) := p(z^{(i)} | x^{(i)}, \theta)$$

3. **M step.** Set

$$\theta := \arg \max_{\theta} \sum_{i=1}^n \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)} | \theta)}{Q_i(z^{(i)})}$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, return to step 2.

Why does this work? See Andrew Ng's lecture notes on the EM algorithm from CS229 at Stanford – they contain the clearest explanation I've seen and use the same notation as us.

## 19.8 Extensions

There are many different clustering algorithms. You can find a good summary of all the different ones here: [https://en.wikipedia.org/wiki/Cluster\\_analysis](https://en.wikipedia.org/wiki/Cluster_analysis). Some topics you might choose to investigate independently (or that we might look at together in the future) include:

- Hierarchical clustering (e.g. phylogenetic trees)
- Methods for choosing  $K$  (the number of clusters)
- Bayesian methods that introduce priors on the parameters in mixture models
- Biclustering

There are also many different types of mixture models that all use the EM algorithm for maximum likelihood optimization. You may want to check out the following (and many more):

- Hidden Markov Models (Baum-Welch algorithm)
- Inside-outside algorithm for induction of probabilistic context-free grammars
- Leicht and Newman's 2007 PNAS paper on finding network clusters using mixture models
- Alignment algorithms for genetic sequence data using HMMs