Clustering, Mixture Models, and the EM Algorithm

Rebecca C. Steorts

Agenda

- Clustering
- Two Component Mixture Model
- Latent Variable
- EM Algorithm

Clustering

Clustering is an **unsupervised method** that divides up data into groups (clusters), so that points in any one group are more "similar'' to each other than to points outside the group

Clustering methods (that we have covered)

- Fuzzy clustering (such as deterministic entity resolution or blocking)
- Overlapping clustering (such as locality sensitive hashing)
- Fellegi Sunter method (technically a mixture model, stay tuned)

Clustering methods (that we have not covered)

- Mixture models
- K-means
- Hierarchical clustering
- Others

Application areas

- Clustering temperatures to identify weather patterns, grouping individuals based on height and weight similarities.
- Clustering customers based on satisfaction levels (ordinal) or grouping individuals based on gender (categorical).
- Clustering GPS coordinates to identify spatial patterns, grouping locations on a map based on features.
- Clustering users in a social network or data based based on their connections (edge structure), grouping academic papers based on citation patterns.

History

- First proposed by Karl Pearson (1984) and analyzed on crab data.
- Applications: "agriculture, astronomy, bioinformatics, biology, economics, engineering, genetics, imaging, marketing, medicine, neuroscience, psychiatry, and psychology, among many other fields in the biological, physical, and social sciences". McLachlan et. al (2019).
- One of the methods in machine learning is **topic modeling**, which identifies "topics" in collections of documents/webpages.
- Topic modeling relies on mixtures models.

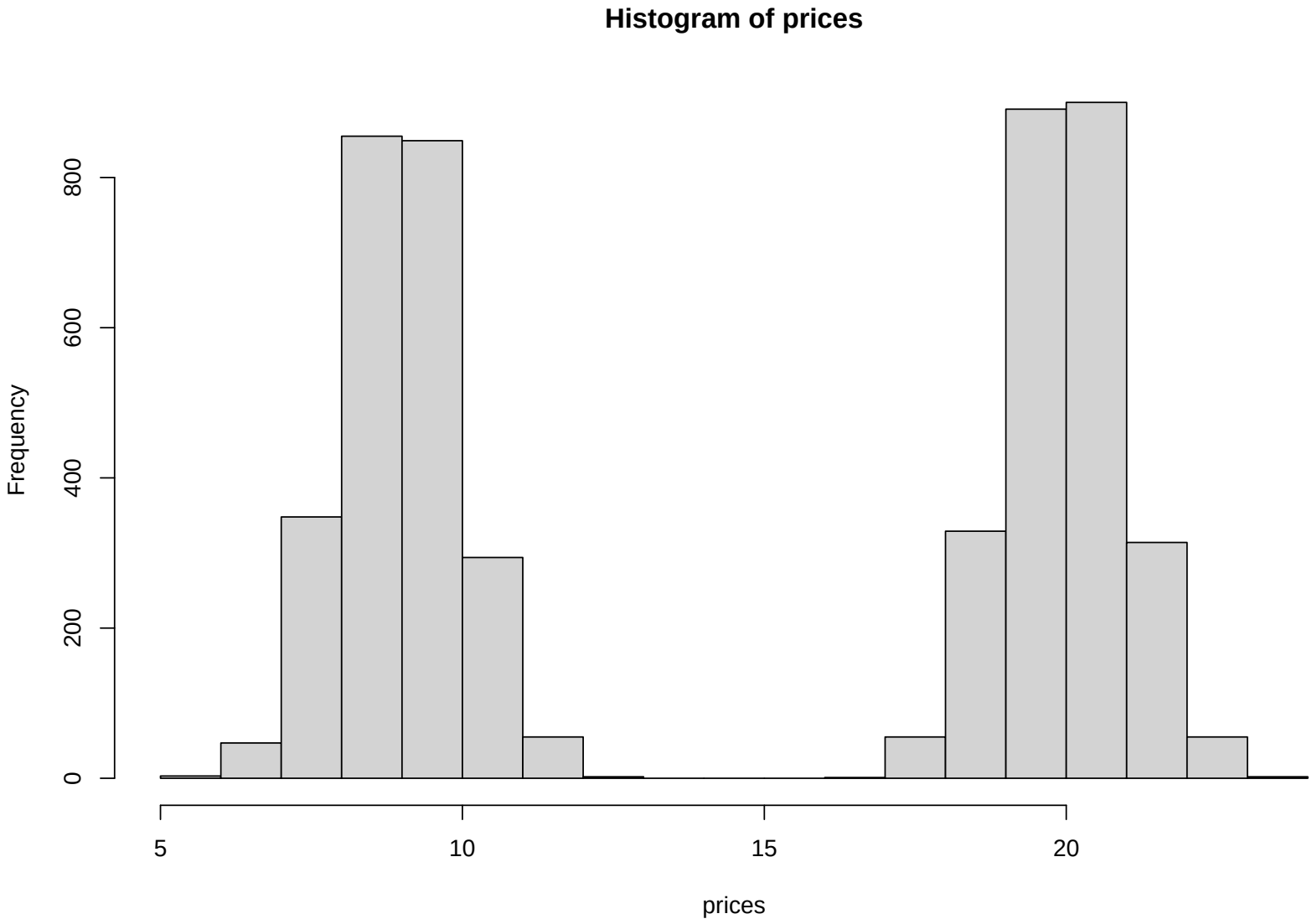
Motivation

- Suppose we want to simulate the price of a randomly chosen book.
- Paperbacks are often cheaper than hardbacks, so let's model them separately.
- Model the price of a book as a mixture model.
- There will be two components (or clusters) in our model – one for paperbacks and one for hardbacks.

Model

- Paperback distribution: $N(9, 1)$
- Hardback distribution: $N(20, 1)$
- Assume that there's a there is a 50% chance of choosing a paperback and 50% of choosing hardback.

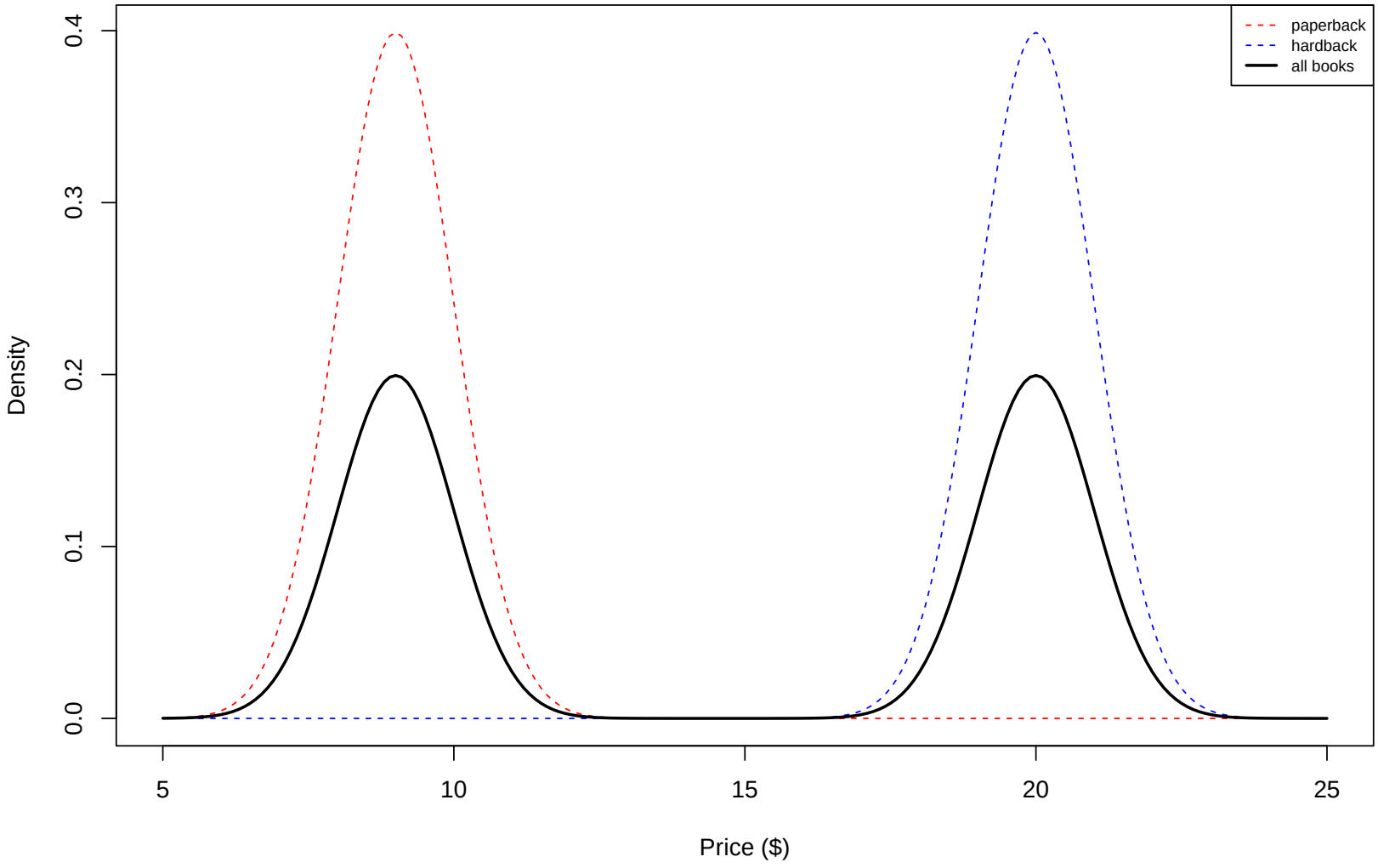
Motivation


Histogram of prices
Frequency
0
200
400
600
800
5
10
15
20
prices

Motivation

- Are the prices of books unimodal or bimodal?
- Suppose you would want to predict the price of a book. Would its distribution be Normal or something else based on the the histogram.

Motivation


paperback
hardback
all books
Density
0.0
0.1
0.2
0.3
0.4
5
10
15
20
25
Price ($)

Motivation

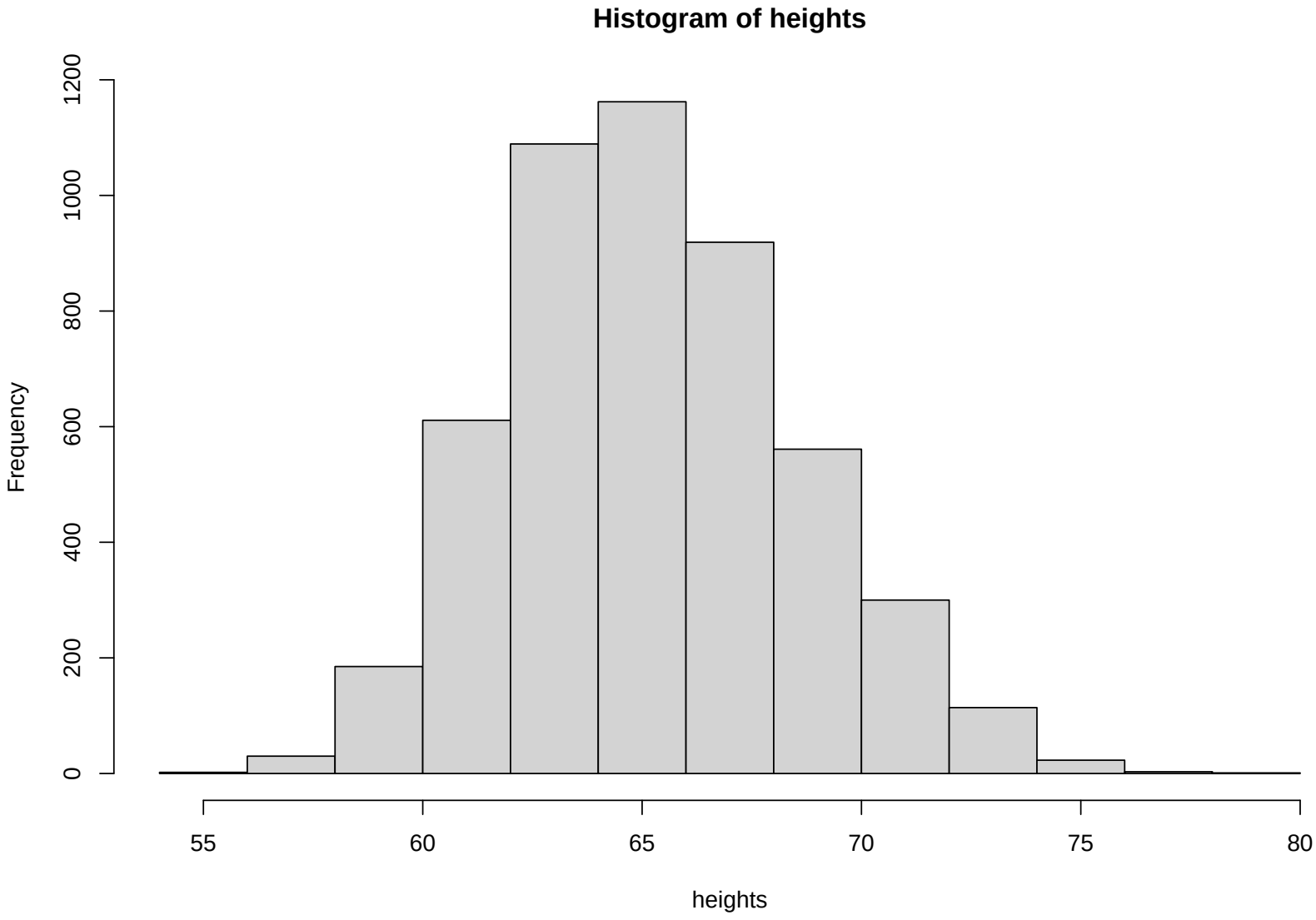
Now assume our data are the heights of students at university.

Male height: $N(69, 2.5^2)$, with units in inches.

Female height: $N(64, 2.5^2)$.

Assume that 75% of the population is female and 25% is male.

Motivation

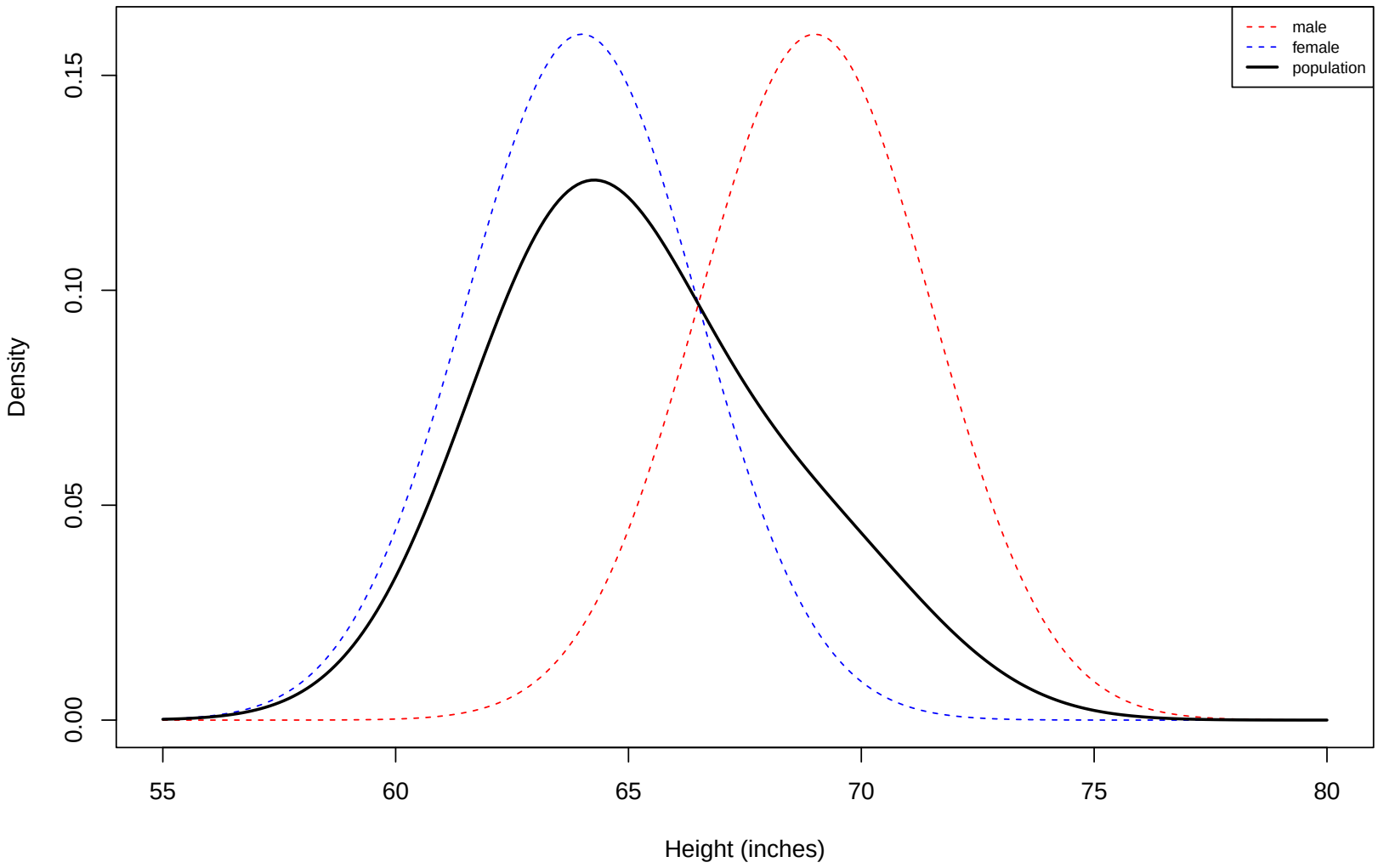

Histogram of heights
Frequency
0
200
400
600
800
1000
1200
55
60
65
70
75
80
heights

Motivation

The histogram is now unimodal.

Are heights normally distributed (assuming this model)? Let's invesitgate!

Motivation


male
female
population
Density
0.00
0.05
0.10
0.15
55
60
65
70
75
80
Height (inches)

Motivation

The Gaussian mixture model is unimodal because there is so much overlap between the two densities.

In this example, observe that the population density is not symmetric, and therefore not normally distributed.

Goal

The goal of this module is to introduce **mixture models**, which are commonly used in applications in classical and modern machine learning.

Mixture models can be viewed as probabilistic clustering

- Mixture models put similar data points into "clusters".
- This is appealing as we can potentially compare different probabilistic clustering methods by how well they predict (under cross-validation). We will not explore this in this particular lecture.
- This contrasts other methods such as k-means and hierarchical clustering as they produce clusters (and not predictions), so it's difficult to test if they are correct/incorrect.[1]

[1] Explore looking at these on your own and see if you can determine their limitations practically, compared to other machine learning models.

Two-component mixture model

Assume that both mixture components have the same precision, $\lambda = 1/\sigma^2$, which is fixed and known.

Let π be the mixture proportion for the first component.

Then the two-component Normal mixture model is:

$$X_1, \ldots, X_n \mid \mu, \pi \ \sim F(\mu, \pi) \tag{1}$$

where $F(\mu, \pi)$ is the distribution with p.d.f.

$$f(x|\mu, \pi) = (1 - \pi)\mathcal{N}(x \mid \mu_0, \lambda^{-1}) + \pi\mathcal{N}(x \mid \mu_1, \lambda^{-1}).$$

Likelihood

The likelihood is

$$p(x_{1:n}|\mu,\pi) = \prod_{i=1}^{n} f(x_i|\mu,\pi)$$
$$= \prod_{i=1}^{n} \Big[(1-\pi)\mathcal{N}(x_i \mid \mu_0, \lambda^{-1}) + \pi\mathcal{N}(x_i \mid \mu_1, \lambda^{-1})\Big].$$

Likelihood

What do you notice about the likelihood function?

$$
\begin{aligned}
p(x_{1:n}|\mu,\pi) &= \prod_{i=1}^{n} f(x_i|\mu,\pi) \\
&= \prod_{i=1}^{n} \Big[(1-\pi)\mathcal{N}(x_i \mid \mu_0, \lambda^{-1}) + \pi\mathcal{N}(x_i \mid \mu_1, \lambda^{-1})\Big].
\end{aligned}
$$

Likelihood

The **likelihood** is very complicated function of μ and π.

This makes working with it directly to find the MLE (or other estimates) difficult.

Thus, we will rewrite the likelihood using a two-stage approach.

Two-stage approach

Let Z_i indicate whether subject i is from component 1 or 2.

$$Z_1, \ldots, Z_n \overset{iid}{\sim} \text{Bernoulli}(\pi) \tag{2}$$

$$X_i \mid Z \sim \mathcal{N}(\mu_{Z_i}, \lambda^{-1}) \quad i = 1, \ldots, n. \tag{3}$$

Checking for understanding

Then the two-component Normal mixture model is:

$$X_1, \ldots, X_n \mid \mu, \pi \ \sim F(\mu, \pi) \tag{4}$$

where $F(\mu, \pi)$ is the distribution with p.d.f.

$$f(x|\mu, \pi) = (1 - \pi)\mathcal{N}(x \mid \mu_0, \lambda^{-1}) + \pi\mathcal{N}(x \mid \mu_1, \lambda^{-1}).$$

Written as a two-stage process:

$$Z_1, \ldots, Z_n \mid \mu, \pi \overset{iid}{\sim} \text{Bernoulli}(\pi) \tag{5}$$

$$X_i \mid \mu, Z \sim \mathcal{N}(\mu_{Z_i}, \lambda^{-1}) i = 1, \ldots, n. \tag{6}$$

Checking for understanding

Given the two equivalent models above, how would you simulate data from a two component mixture model?

Extension to k-components

Assume we observe $X_1, \ldots, X_n$ and that each X_i is sampled from one of K **mixture components**.

Associated with each random variable X_i is a label called $Z_i \in \{1, \ldots, K\}$ which indicates which component X_i came from.

Notation

Let π_k be called **mixture proportions** or **mixture weights**, which represent the probability that X_i belongs to the k-th mixture component.

The mixture proportions are non-negative and they sum to one, $\sum_{k=1}^{K} \pi_k = 1$.

Observe that $P(X_i \mid Z_i = k)$ represents the distribution of X_i assuming it came from component k.

Extension

Then the two-component Normal mixture model is:

$$X_1, \ldots, X_n \mid \mu, \pi \ \sim F(\mu, \pi) \tag{7}$$

where $F(\mu, \pi)$ is the distribution with p.d.f.

$$f(x|\mu, \pi) = \sum_{k=1}^{K} \pi_k N(\mu_k, \lambda^{-1}).$$

Written as a two-stage process: for $i = 1, \ldots, n$:

$$P(Z_i = k) = \pi_k \tag{8}$$

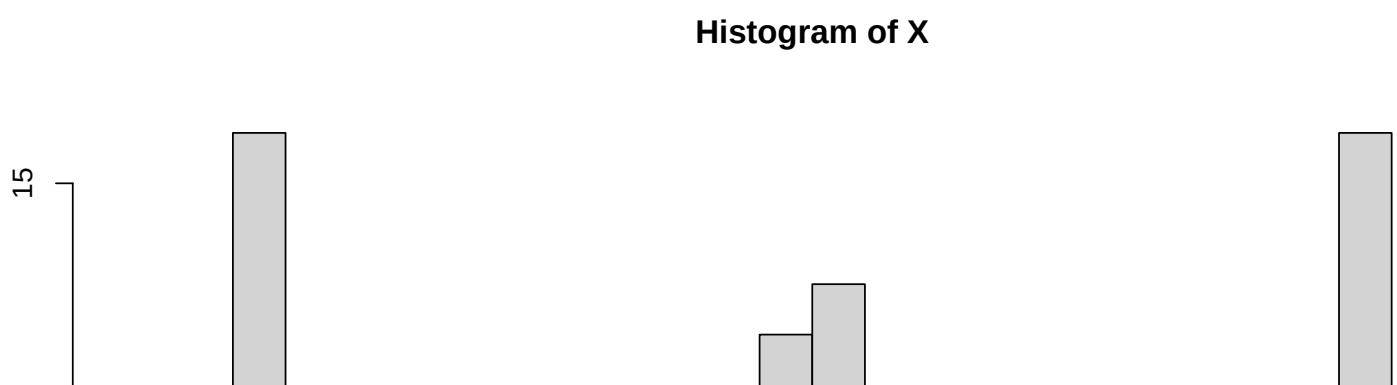
$$X_i \mid \mu, Z_i \sim \mathcal{N}(\mu_{Z_i}, \lambda^{-1}) \tag{9}$$

Example

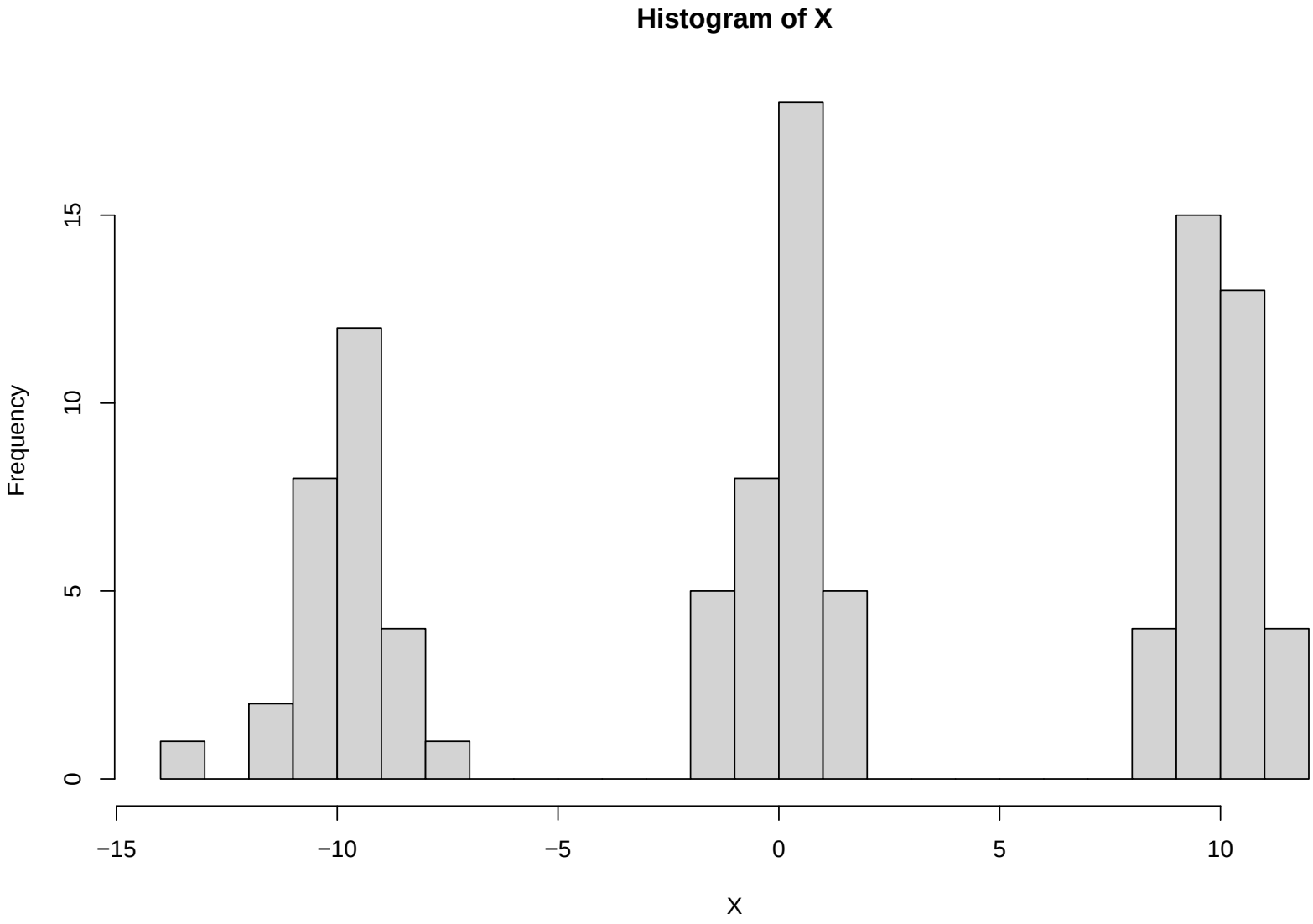
Let's look at a three component mixture model.

Suppose we assume that $\mu = (-10, 0, 10)$ and $\sigma^2 = 1$. Assume each mixture weight is equally likely.

```
set.seed(1234)
n <- 100
mu <- c(-10, 0, 10)
# sample Z first
Z <- sample(1:3, size=n, replace=TRUE)
# conditional on Z, sample the normal update
X <- rnorm(n, mean=mu[Z], sd=1)
hist(X, breaks=20)
```



Example



EM Algorithm

General way to deal with hidden class labels or clusters. (Can also be used for missing data or latent variables).

E-step: Fill in the hidden class labels.

M-step: Apply standard MLE (or other approaches) to complete the data.

The algorithm always converges to a local optima of the likelihood.

Simple EM Algorithm

Notation and Setup

We know the following:

- Observations $x_{1:n}$.
- K total classes
- $P(Z_i = k) = \pi_k$ (for $i = 1, \ldots, K$)
- Common variance σ^2.

We do not know $\mu_1, \ldots, \mu_K$ and want to learn these.

This is a very unrealistic setting, however, it hopefully provides intuition regarding the algorithm itself (and the math is simplified).

EM Algorithm

$\propto$ will drop any constants (and I will make sure to include them back in later). Common trick in Bayesian statistics.

$$p(x_1, \ldots, x_n \mid \mu_1, \ldots, \mu_K) \tag{10}$$

$$= \prod_{i=1}^{n} p(x_i \mid \mu_1, \ldots, \mu_K) \text{ independent data} \tag{11}$$

$$= \prod_{i=1}^{n} \sum_{k=1}^{K} p(x_i, z_i = k \mid \mu_1, \ldots, \mu_K) \text{ marg. over labels} \tag{12}$$

$$= \prod_{i=1}^{n} \sum_{k=1}^{K} p(x_i \mid z_i = k, \mu_1, \ldots, \mu_K) p(z_i = k) \tag{13}$$

$$\propto \prod_{i=1}^{n} \sum_{k=1}^{K} \exp(-\frac{1}{2\sigma^2}(x_i - \mu_k)^2)\pi_k \tag{14}$$

EM Algorithm

Let $\theta^{(t)} = (\mu_1^{(t)}, \ldots, \mu_k^{(t)})$ at some iteration t.

At iteration t consider the function:

$$Q(\theta^{(t)} \mid \theta^{(t-1)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} P(z_i = k \mid x_i, \theta^{(t-1)}) \tag{15}$$

$$\times \log P(x_i, z_i = k \mid \theta^{(t-1)}) \tag{16}$$

E-step

$$P(z_i = k \mid x_i, \theta^{t-1}) \tag{17}$$

$$= P(z_i = k \mid x_i, \mu_1^{(t-1)}, \ldots, \mu_K^{(t-1)}) \tag{18}$$

$$\propto P(x_i \mid z_i = k, \mu_1^{(t-1)}, \ldots, \mu_K^{(t-1)}) P(z_i = k) \tag{19}$$

$$\propto \exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2)\pi_k \tag{20}$$

$$= \frac{\exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2)\pi_k}{\sum_{k=1}^{K} \exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2)\pi_k} \tag{21}$$

This is equivalent to assigning clusters to each data point in a soft-way (clusters can overlap).

M-step

$$Q(\theta^{(t)} \mid \theta^{(t-1)}) \tag{22}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} P(z_i = k \mid x_i, \theta^{(t-1)}) \times \log P(x_i, z_i = k \mid \theta^{(t-1)}) \tag{23}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} P(z_i = k \mid x_i, \theta^{(t-1)}) \tag{24}$$

$$\times [\log P(x_i \mid z_i = k, \theta^{(t)}) + \log P(z_i = k \mid \theta^{(t)})] \tag{25}$$

$$\tag{26}$$

Recall that in the E-step, we calculated
$R_{ik}^{(t-1)} = P(z_i = k \mid x_i, \theta^{(t-1)})$

M-step

At each iteration t, maximize Q in term of $\theta^{(t)}$.

$$Q(\mu_k^{(t)} \mid \theta^{(t-1)}) \propto \sum_{i=1}^{n} R_{ik}^{(t-1)} \exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2), \implies \tag{27}$$

$$\frac{\partial Q(\mu_k^{(t)} \mid \theta^{(t-1)})}{\partial \mu_k^{(t)}} = \sum_{i=1}^{n} R_{ik}^{(t-1)}(x_i - \mu_k^{(t)})) = 0 \implies \tag{28}$$

$$\mu_k^{(t)} = \sum_{i=1}^{n} w_i x_i \quad \text{where}$$

$$w_i = \frac{R_{ik}^{t-1}}{\sum_{i=1}^{n} R_{ik}^{t-1}} = \frac{P(z_i = k \mid x_i, \theta^{(t-1)})}{\sum_{i=1}^{n} P(z_i = k \mid x_i, \theta^{(t-1)})}$$

This is equivalent to updating the cluster centers.

Summarize EM Algorithm

1. E-step

Compute the expected classes of all data points for each class:

$$P(z_i = k \mid x_i, \theta^{(t-1)}) = \frac{\exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2)\pi_k^{(t-1)}}{\sum_{k=1}^{K} \exp(-\frac{1}{2\sigma^2}(x_i - \mu_k^{(t-1)})^2)\pi_k^{(t-1)}}$$

2. M-step

Then compute the maximum value given our data's class membership.

$$\mu_i^{(t)} = \sum_{i=1}^{n} w_i x_i.$$

In this case, it's the MLE but with weighted data.

Exercise

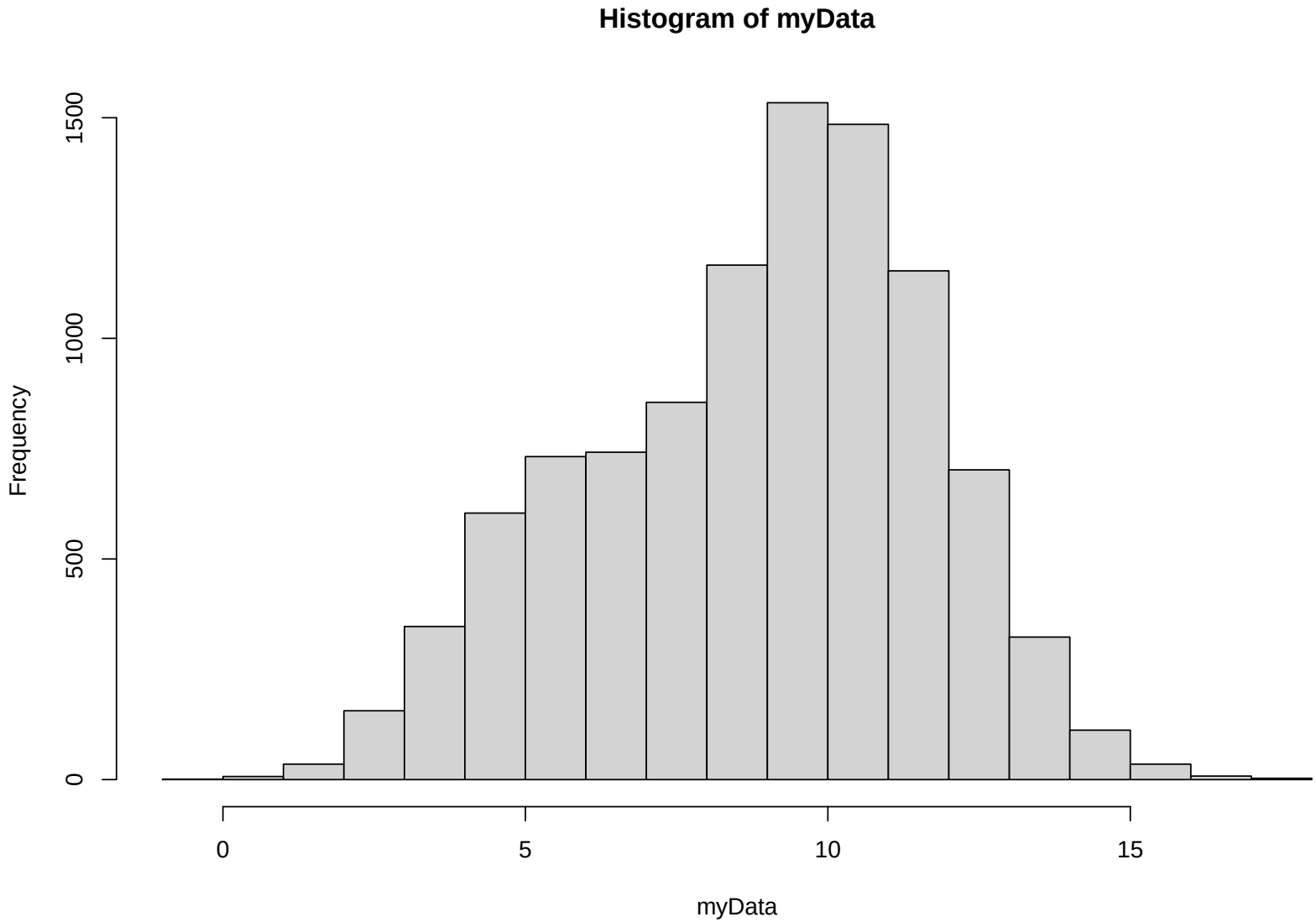
Assume our mixture components are fully specified Gaussian distributions with $K = 2$ components:

$$X_i \mid Z_i = 0 \sim N(5, 1.5) \tag{29}$$
$$X_i \mid Z_i = 1 \sim N(10, 2) \tag{30}$$

Let the true mixture proportions be $P(Z_i = 0) = 0.25$ and $P(Z_i = 1) = 0.75$, respectively.

Exercise

Simulate data from the mixture model on the previous slide, which should produce the following histogram.



Exercise

```
library(mixtools)
```

```
## mixtools package, version 2.0.0, Released 2022-12-04
## This package is based upon work supported by the Nationa
```

```
runEM <- normalmixEM(myData,  lambda = 0.5,
                     mu = c(10, 20), sigma = c(2,2))
```

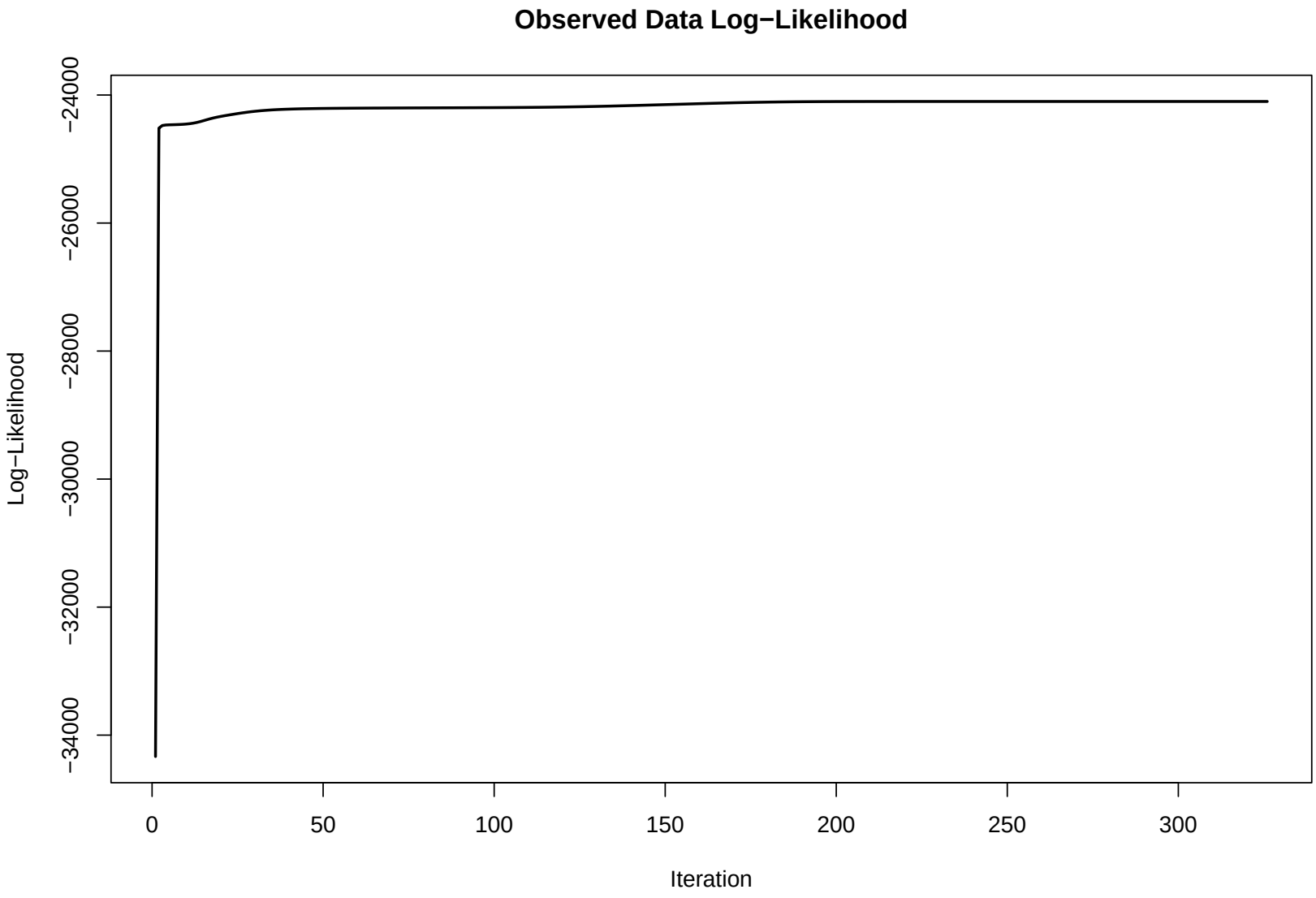
```
## number of iterations= 325
```

```
summary(runEM)
```

```
## summary of normalmixEM object:
##           comp 1    comp 2
## lambda 0.242555  0.757445
## mu     5.158150 10.034057
## sigma  1.504165  1.954039
## loglik at estimate:  -24100.72
```

Plot

```
plot(runEM)
```



R packages for mixture models

- The `mclust` package (http://www.stat.washington.edu/mclust/) is standard for Gaussian mixtures.
- The `mixtools` considers classic parametric densities, mixtures of regressions, and some non-parametric mixtures.

Exercise

Suppose that $X \sim N(\mu_1, \sigma_1^2)$ and $Y \sim N(\mu_2, \sigma_2^2)$ independently.

1. What is the distribution of $aX + bY$?

Solution: Due to independence,

$$Z \sim N(a\mu_1, +b\mu_2, a^2\sigma_1^2 + b^2\sigma_2^2).$$

2. Suppose that $X \sim N(\mu_1, \sigma_1^2)$ and $Y \sim N(\mu_2, \sigma_2^2)$ (and the observations are dependent).

Is the distribution of $aX + bY$ still Normal? No, not necessarily due to the dependence of the random variables.[2]

[2] In the case of a Gaussian mixture model, a random variable sampled from a Gaussian mixture model can be thought of as a two stage process. First, randomly sample a component (e.g., male or female). Second, then we sample our observation from the normal distribution that corresponds to the component sampled in step one.

Todo

1. I would like to write down the general case if possible. This might be another module as the notation needs to be multivatiate.
2. It would also be nice to highlight and illustrate the connection with k-means. This might be nicer as we assume we know everything.
3. Can go beyond Gaussians, but this might be too much.