The correct formula, called the Law of Total Variance, is

$$\text{Var}(Y) = E[\text{Var}(Y|W)] + \text{Var}[E(Y|W)]$$  \hspace{1cm} (10.33)

Deriving this formula is easy, by simply evaluating both sides of (10.33), and using the relation

$$\text{Var}(X) = E(X^2) - (EX)^2.$$ 

This exercise is left to the reader. See also Section 10.9.2.

10.5 The EM Algorithm

Now returning to our main topic of mixture models, let’s discuss a very common tool for fitting such models.

Many mixture applications make use of the Expection/Maximization (EM) algorithm. The derivation and ultimate formulas can get quite complex. Fortunately, R libraries exist, such as mixtools, so you can avoid knowing all the details, as long as you understand the basic notion of a mixture model.

10.5.1 Overall Idea

In something like (10.2), for instance, one would make initial guesses for the $p_M(i)$ and then estimate the parameters of the $g_i$. In the next step, we’d do the opposite—take our current guesses for the latter parameters as known, and estimate the $p_M(i)$. Keep going until convergence.

To make things concrete, recall the trick coin example Section 10.1. But change it a little, so that the probabilities of heads for the two coins are unknown; call them $p_0$ (heads-light coin) and $p_1$ (heads-heavy coin). And also suppose that the two coins are not equally likely to be chosen, so that $p_M()$ is not known; denote $P(M = 1)$ by $q$.

Suppose we have sample data, consisting of doing this experiment multiple times, say by reaching into the box n times and then doing r flips each time. We then wish to estimate 3 quantities—$q$ and the two $p_i$—using our sample data.

We do so using the following iterative process. We set up initial guesses, and iterate until convergence:

- **E step:** Update guess for $q$ (complicated Bayes Rule equations).
- **M step:** Using the new guess for $q$, update the guesses for the two $p_i$. 

The details are beyond the scope of this book.

10.5.2 The mixtools Library

R’s CRAN repository of contributed software includes the mixtools library. Let’s see how to use one of the functions in that library, normalmixEM(), which assumes that the densities in (10.3) are from the normal distribution family.

Let’s suppose we are modeling the data as a mixture of 2 normals. So, with our observed data set, our goal is to estimate 5 parameters from our data:

- $\mu_1$, the mean of the first normal distribution
- $\mu_2$, the mean of the second normal distribution
- $\sigma_1$, the mean of the second normal distribution
- $\sigma_2$, the mean of the second normal distribution
- $q = P(M = 1) = 1 - P(M = 2)$

The basic form of the call is

```
normalmixEM ( x , lambda=NULL, mu=NULL, sigma=NULL, k )
```

where

- `x` is the data, a vector
- `k` is the number of values $M$ can take on, e.g. 2 for a mixture of 2 normals
- `lambda` is a vector of our initial guesses for the quantities $P(M = i)$, of length `k` (recycling will be used if it is shorter); note that these must be probabilities summing to 1
- `mu` is a vector of our initial guesses for the $\mu_i$
- `sigma` is a vector of our initial guesses for the $\sigma_i$

One can leave the initial guesses as the default NULL values, but reasonable guesses may help convergence.

The return value will be an R list, whose components include `lambda`, `mu` and `sigma`, the final estimates of those values.

---

5 “M” in the M step refers to the Maximum Likelihood method, a special case of the material in Section 20.1.3.
10.5.3 Example: Old Faithful Geyser

This example is taken from the online documentation in **mixtools**. The data concern the Old Faithful Geyser, a built-in data set in R.

The data here consist of waiting times between eruptions. A histogram, obtained via
\[
\text{hist}(\text{faithful}$\text{waiting})
\]
and shown in Figure [10.1] seems to suggest that the waiting time distribution is a mixture of 2 normals, and the proportion of eruptions caused by the first type of process.

I tried initial guesses of means and standard deviations from the appearance of the histogram, and used equal weights for my initial guesses in that aspect:
\[
\text{mixout <- normalmixEM(faithful$waiting , lambda=0.5, mu=c(55,80), sigma=10, k=2)}
\]
number of iterations= 9
\[
\text{str(mixout)}
\]
List of 9
\[
\begin{align*}
\$ x & : \text{num [1:272]} 79 54 74 62 85 55 88 85 51 85 \ldots \\
\$ lambda & : \text{num [1:2]} 0.361 0.639 \\
\$ mu & : \text{num [1:2]} 54.6 80.1 \\
\$ sigma & : \text{num [1:2]} 5.87 5.87 \\
\$ loglik & : \text{num } -1034
\end{align*}
\]
10.6. MEAN AND VARIANCE OF RANDOM VARIABLES HAVING MIXTURE DISTRIBUTIONS

\[ p_{\text{posterior}} : \text{num } [1:272, 1:2] 1.02e-04 1.00 4.12e-03 9.67e-01 1.21e-06 \ldots \]

\[ \text{attr}(*, "\text{dimnames"})=\text{List of 2} \]

\[ \text{all.loglik} : \text{num } [1:10] -1085 -1051 -1037 -1034 -1034 \ldots \]

\[ \text{restarts} : \text{num 0} \]

\[ \text{ft} : \text{chr } "\text{normalmixEM}" \]

\[ \text{attr}(*, "\text{class"})=\text{chr } "\text{mixEM}" \]

So, the estimate from EM is that about 36% of the eruptions are of Type 1, etc. Interesting, when I tried it without my own initial guesses,

\[
\text{mixout} <- \text{normalmixEM(faithful$waiting, k=2)}
\]

the results were the same, so it seems to be a fairly stable model here.

By the way, since we are working with a hidden variable M here—in fact, we are merely postulating that it exists—how do we check this assumption? We’ll return to this general idea of model fitting in Chapter 22.

10.6 Mean and Variance of Random Variables Having Mixture Distributions

Think of the random variables M and Y in the discussion following (10.3). Then \( E(Y) \) is easy to find using the Law of Total Expectation:

\[
EY = E[E(Y|M)]
\]  

(10.34)

Of course, evaluating this would require being able to compute \( E(Y \mid M) \), which is easy in some cases, not so easy in others.

Also, using the Law of Total Variance, we have that

\[
EY = E[\text{Var}(Y|M)] + \text{Var}[E(Y|M)]
\]

(10.35)