Expectation Maximization Tool for Mixed-Gaussian distributions

Lorenzo Papa

2025-10-16

What is a Gaussian Mixture model?

This is an educational document that will cover the Expectation Maximization algorithm in the context of estimating latent variables for Gaussian mixture Models. We will mainly stay with the EM algorithm applied to mixtures of two components but EM works with K clusters of the same parametric family. So what is a Gaussian Mixture model? An observation X_i that comes from a Gaussian mixture is modeled by the marginal distribution

$$P(X_i = x) = \sum_{k=1}^{K} P(Z_i = k) f_{X|Z}(X_i = x|Z_i = k) = \sum_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k)$$

Where $Z \in \{1, ..., K\}$ is the latent variable (meaning an unobserved variable) representing which mixture component X_i belongs to, π_k is the probability that X_i belongs to component k. Each component in the mixture is assumed to be well modeled by a normal distribution where $X|Z = k \sim N(\mu_k, \sigma_k)$. Gaussian mixture models are applied to incomplete data where the membership variable Z_i is not observed for each X_i.

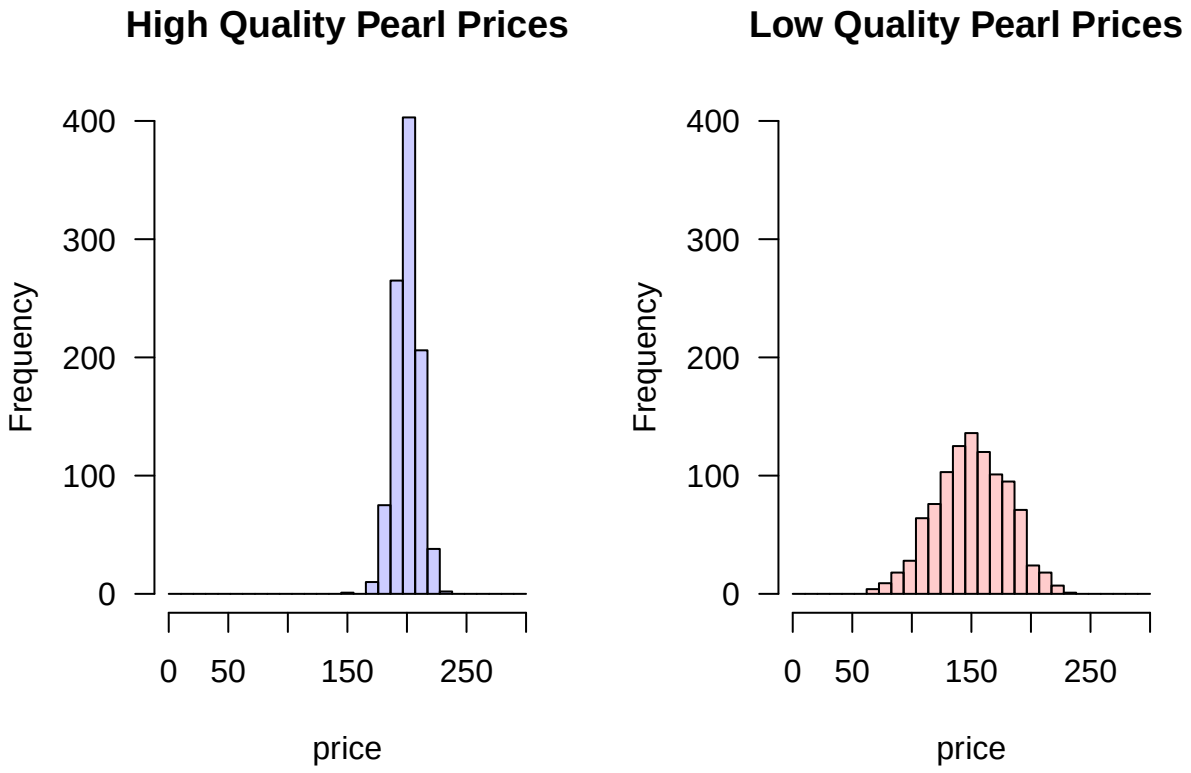
The EM algorithm aims to obtain the maximum likelihood estimates for $\theta = \{\pi_1, ..., \pi_k; \mu_1, ...\mu_k; \sigma_1, ..., \sigma_k\}$ by maximizing the expectation of the Gaussian Mixture's log-likelihood with respect to the posterior distribution of of our latent variable Z

A Pearly Example

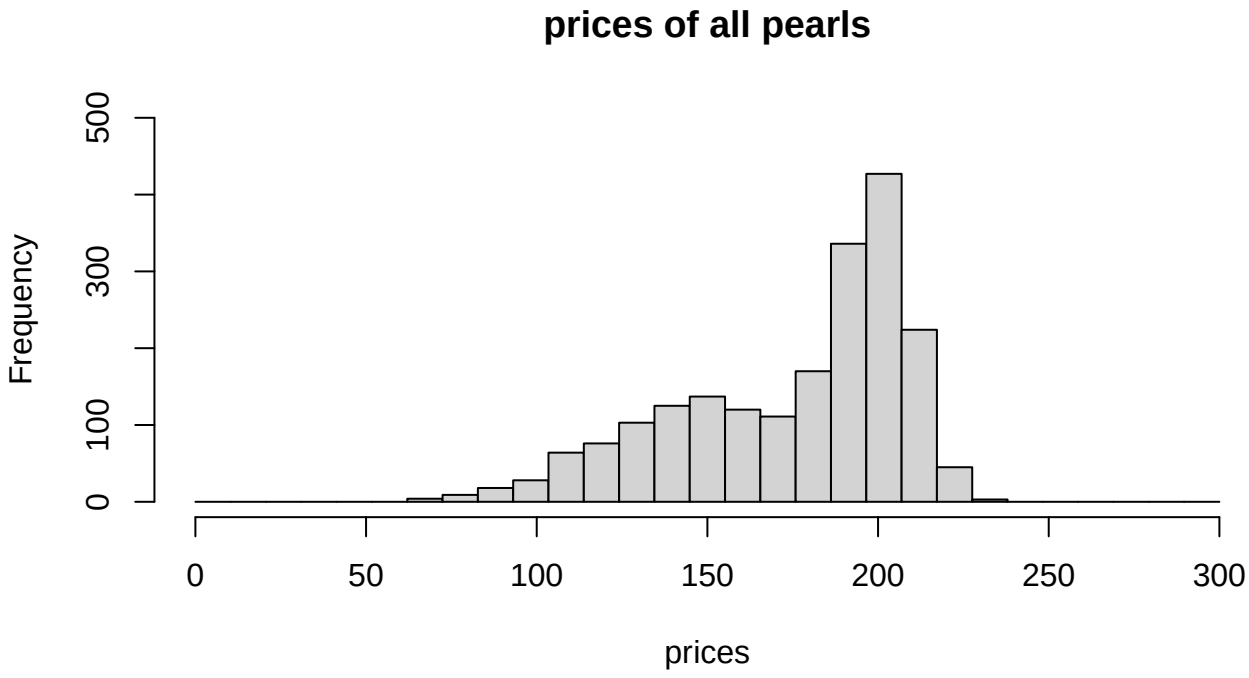
Suppose we are given price data for a year's harvest of farmed pearls. Assume also that it is industry standard to classify pearls into two categories: high and low quality. High quality pearls being more round

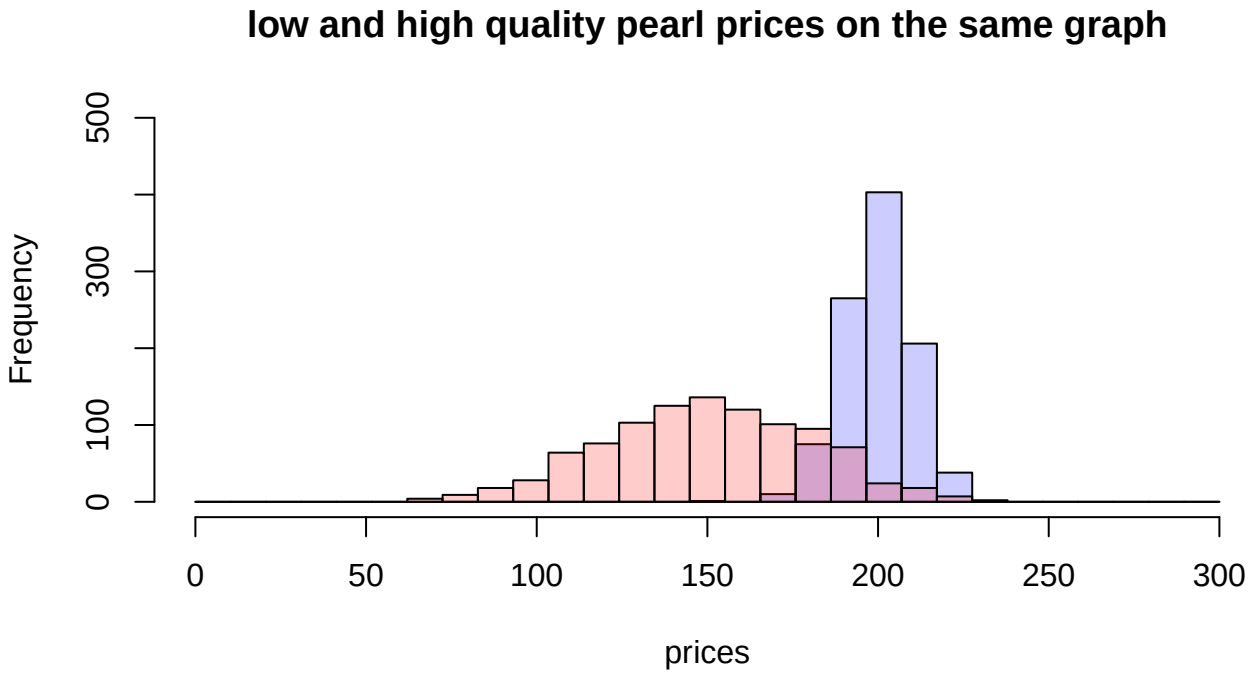
and smooth, low quality pearls being amorphous and/or having imperfections. To make things easier, let's also assume that prices of high quality and prices of low quality pearls follow a normal distribution. If the pearl quality and price data is collected, we might be able to graph prices by quality like so:

```
highqual <- rnorm(1000, 200,10)
lowqual <- rnorm(1000,150, 30)
```



life is easy and we can calculate point estimates and conduct all sorts of inference on high quality and low quality pearl prices. Where things are not so easy is when quality data is not collected. For example, all pearls are chucked into a basket and only at the point of sale is **price** data collected. Now if we plot our frequency graph of prices, it would look something like this:





Notice the similarity to low and high quality pearl prices on the same graph. So what now? We have price data but no distinction between high and low quality pearls. First notice that quality data is our latent variable that is unobserved. Letting X be the price of each pearl we can let Z represent the quality of each

pearl that was not observed. This is a perfect scenario to use the expectation maximization algorithm. Let's get into how it works.

MLE of the Gaussian

First, review maximum likelihood estimates of the univariate Gaussian distribution. For n observations $X_1, ...X_n$, of a Gaussian distribution with unknown mean μ and known σ, the maximum likelihood estimate for μ can be obtained by computing the derivative of the log likelihood with the respect to μ, setting to zero and solving for μ

$$
\begin{aligned}
L(\mu) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x_i - \mu)^2}{2\sigma^2} \\
\Rightarrow l(\mu) &= \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi\sigma^2}}) - \frac{(x_i - \mu)^2}{2\sigma^2} \\
\frac{\partial l(\mu)}{\partial \mu} &= \sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2}
\end{aligned}
$$

Setting this to zero and solving for μ gives the MLE $\frac{1}{n}\sum_{i=1}^{n} x_i$

Gaussian Mixture MLE

Suppose we have the complete data X, Z. Let Z_{ik} be the indicator for cluster membership. That is, $Z_{ik} = 1$ when X_i comes from the the k^{th} cluster and $Z_{ik} = 0$ otherwise. We have that the complete likelihood for a Gaussian Mixture with K clusters is:

$$
\begin{aligned}
L(\theta|X, Z) &= \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k)^{Z_{ik}} \\
\Rightarrow l(\theta) &= \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} log(\pi_k N(x_i|\mu_k, \sigma_k))
\end{aligned}
$$

Here we can easily compute MLEs for $\{\mu_k, \sigma_k, \pi_k\}$ as computing the derivative with respect to any parameter and setting to zero yields an analytic solution. Note that the inside sum contains only one term for each

observation i since all other terms are multiplied by $Z_{ik} = 0$ when X_i does not belong to that cluster/mixture component.

Things get difficult When Z_i is unknown. In this scenario the likelihood function is the product of the marginal distribution over all observations.

$$L(\theta|X) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k)$$
$$\Rightarrow l(\theta) = \sum_{i=1}^{n} log(\sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k))$$

This function is difficult to optimize due to the sum inside the log function. The EM algorithm works around this problem by substituting the membership variables Z_i in the complete data log-likelihood with its conditional expectation $E[Z_{ik}|X]$.

Expectation-Maximization

We've covered all the preliminary information needed to get into the weeds of the EM algorithm applied to the Gaussian Mixture with two mixture components. As the name suggests, the EM algorithm consists of an expectation step followed by a maximization step. In the expectation step, we must compute **the expectation of the log likelihood with respect to the conditional distribution of Z given the data X and current estimates of the model parameters** $\theta^{(t)}$. As stated above we need to replace unknown values of Z_{i1} and $Z_{i2} = 1 - Z_{i1}$ with their expected value. By Bayes' Theorem:

$$\begin{aligned} E[Z_{i1}|X_i] &= 1P(Z_{i1} = 1|X_i) + 0P(Z_{i1} = 0|X) \\ &= \frac{P(X_i|Z_{i1} = 1)P(Z_{i1} = 1)}{P(X_i)} \\ &= \frac{\pi_1 N(X_i|\mu_1, \sigma_1)}{\pi_1 N(X_i|\mu_1, \sigma_1) + \pi_2 N(X_i|\mu_2, \sigma_2)} \end{aligned}$$

Define $Q(\theta|\theta^{(t)})$ to be the expectation of the log-likelihood and let $\gamma_i = P(Z_{i1} = 1|X_i)$

$$
\begin{aligned}
Q(\theta|\theta^{(t)}) : &= E_{Z\sim P(\cdot|X,\theta^{(t)})}[\sum_{i=1}^{n}\sum_{k=1}^{K} Z_{ik} log(\pi_k N(x_i|\mu_k,\sigma_k))] \\
&= \sum_{i=1}^{n} \gamma_i log(\pi_1 N(X_i|\mu_1,\sigma_1)) + (1-\gamma_i) log(\pi_2 N(X_i|\mu_2,\sigma_2)) \\
&= \sum_{i=1}^{n} \gamma_i log(\pi_1) + (1-\gamma_i) log(\pi_2) + \sum_{i=1}^{n} \gamma_i log(N(X_i|\mu_1,\sigma_1)) + (1-\gamma_i) log(N(X_i|\mu_2,\sigma_2))
\end{aligned}
$$

Note that our data X and $\theta^{(t)} = \{\pi, \mu_1, \mu_2, \sigma_1, \sigma_2\}$ are fixed while Z is a random variable. $Q(\theta|\theta^{(t)})$ is the quantity we wish to maximize by updating $\{\pi, \mu_1, \mu_2, \sigma_1, \sigma_2\}$. Since these parameters appear in separate linear terms, they can be maximized independently like so:

$$
\begin{aligned}
\hat{\pi} &= \arg\max_{\pi} \sum_{i=1}^{n} \gamma_i log(\pi) + (1-\gamma_i) log(1-\pi) \\
\{\hat{\mu}_1, \hat{\sigma}_1\} &= \arg\max_{\mu_1,\sigma_1} \sum_{i=1}^{n} \gamma_i log(N(X_i|\mu_1,\sigma_1)) \\
\{\hat{\mu}_2, \hat{\sigma}_2\} &= \arg\max_{\mu_2,\sigma_2} \sum_{i=1}^{n} (1-\gamma_i) log(N(X_i|\mu_2,\sigma_2))
\end{aligned}
$$

and the arguments that maximize $Q(\theta|\theta^{(t)})$ are:

$$
\hat{\mu}_1 = \frac{\sum_{i=1}^{n} \gamma_i x_i}{\sum_{i=1}^{n} \gamma_i} \qquad \hat{\sigma}_1 = \frac{\sum_{i=1}^{n} \gamma_i (x_i - \hat{\mu})^2}{\sum_{i=1}^{n} \gamma_i}
$$

$$
\hat{\mu}_2 = \frac{\sum_{i=1}^{n} (1-\gamma_i) x_i}{\sum_{i=1}^{n} 1-\gamma_i} \qquad \hat{\sigma}_2 = \frac{\sum_{i=1}^{n} (1-\gamma_i)(x_i - \hat{\mu})^2}{\sum_{i=1}^{n} 1-\gamma_i}
$$

$$
\hat{\pi} = \frac{1}{n} \sum_{i=1}^{n} \gamma_i
$$

Now that $\theta^{(t+1)} = \{\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\pi}\}$ are computed, we plug into the expression for the expectation of the log likelihood to compute $Q(\theta|\theta^{(t+1)})$. The expectation and maximization steps are iterated until for some small $\epsilon > 0$ we get $|Q(\theta|\theta^{(t)}) - Q(\theta|\theta^{(t+1)})| < \epsilon$

EM R implementation for two mixture components

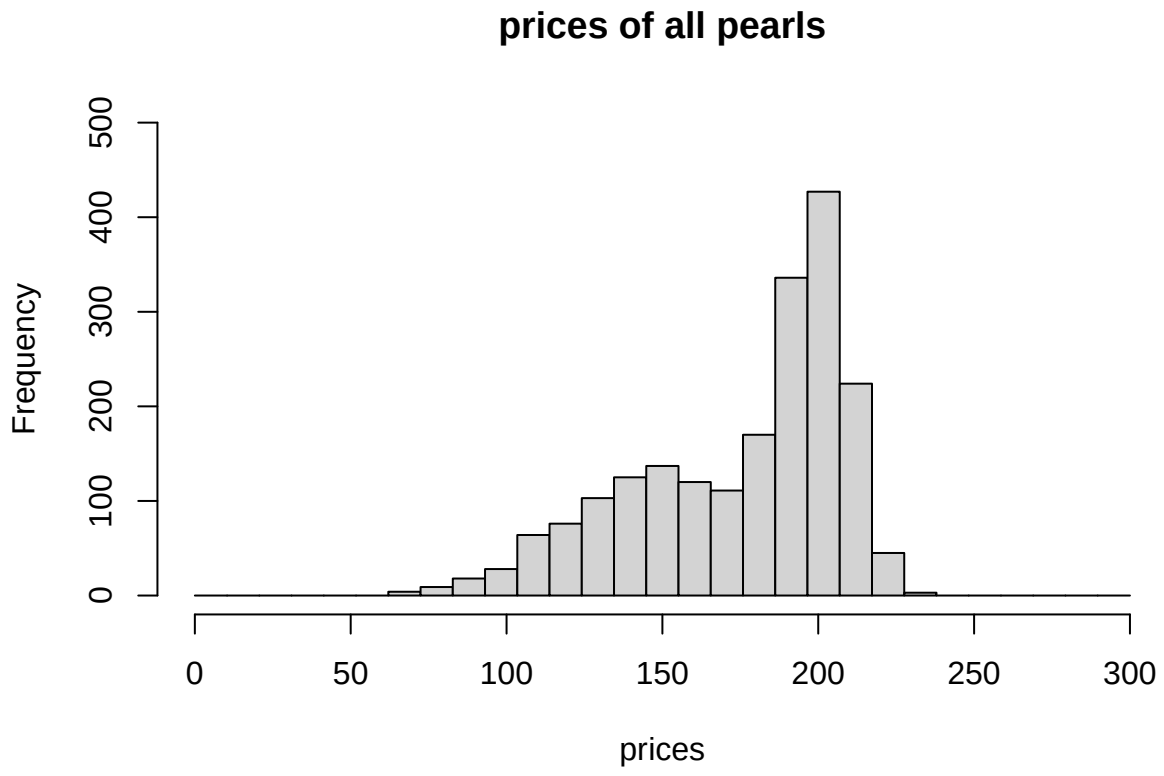
Here I write a function to obtain MLEs for the Gaussian mixture with two clusters then apply it to the pearl problem

```
GaussianMixtureEM <- function(data,theta){
e <- 1e-8
#initial guesses
mu1 <- theta[1]
mu2 <- theta[2]
v1 <- theta[3]
v2 <- theta[4]
p1 <- theta[5]
p2 <- theta[6]

Q_t <- 0
Q_t1 <-  sum(log(p1*((p1*dnorm(data,mu1,v1))) + (p2*(p2*dnorm(data,mu2,v2)))))

while(abs(Q_t1 - Q_t) >= e){
  ##----E-step----##
  #posterior distribution of the mixture parameters
  tau1 <- p1*dnorm(data,mu1,v1)/( p1*dnorm(data,mu1,v1) + p2*dnorm(data,mu2,v2) )
  tau2 <- p2*dnorm(data,mu2,v2)/( p1*dnorm(data,mu1,v1) + p2*dnorm(data,mu2,v2) )

  #Expectation of the log likelihood with current parameters
  Q_t <- sum(log(tau1*((p1*dnorm(data,mu1,v1))) + (tau2*(p2*dnorm(data,mu2,v2)))))

  ##----M-step----##
  #new parameters to maximize log likelihood
  p1<-sum(tau1)/length(data)
  p2<-sum(tau2)/length(data)
  mu1<-sum(tau1*data)/sum(tau1)
  mu2<-sum(tau2*data)/sum(tau2)
```

```
  v1<-sqrt(sum(tau1*(data-mu1)^2)/sum(tau1))
  v2<-sqrt(sum(tau2*(data-mu2)^2)/sum(tau2))


  #updated log likelihood
  Q_t1 <-  sum(log(tau1*((p1*dnorm(data,mu1,v1))) + (tau2*(p2*dnorm(data,mu2,v2)))))
  }
theta_estimate <- c(mu1,mu2,v1,v2,p1,p2)
return(theta_estimate)
}
```

\

Recall that the pearl data is generated by 1000 samples from $N(\mu = 200, \sigma = 10)$ for high quality pearls and 1000 samples from $N(\mu = 150, \sigma = 30)$ for low quality ones. These are the true parameters of our Gaussian mixture model. The first step is to generate a first guess for the parameters to feed to the EM algorithm



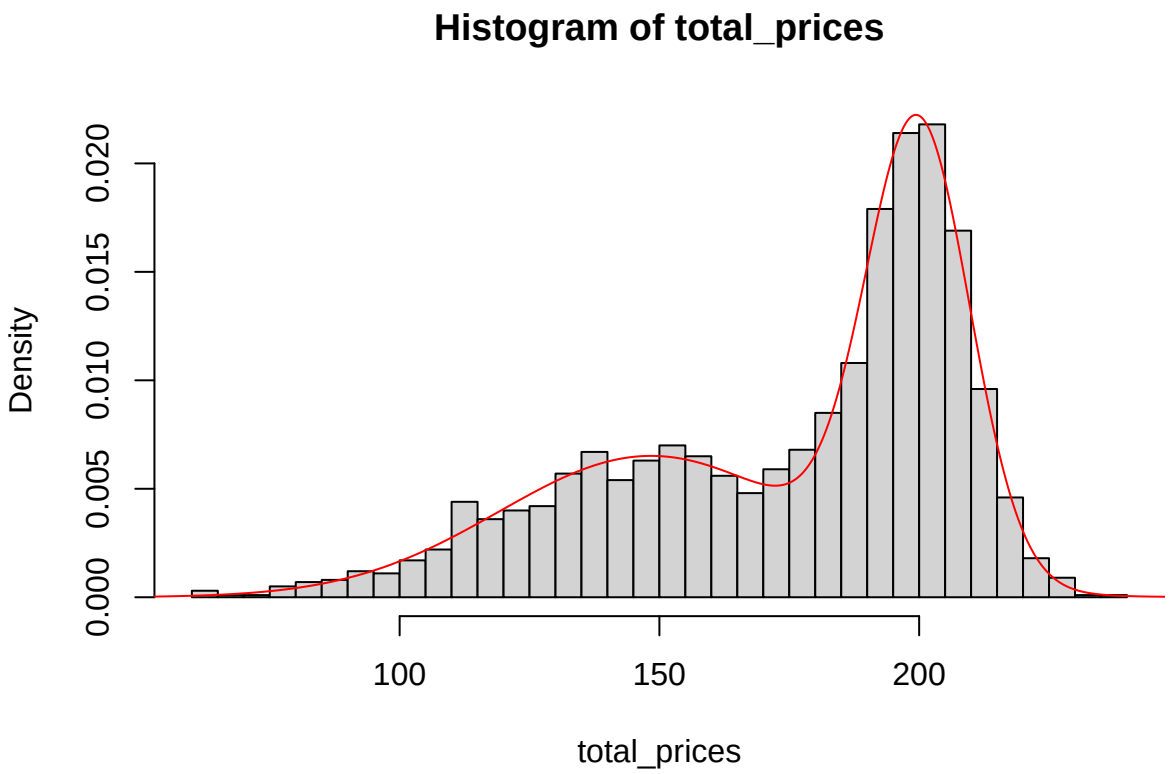
A good place to start for $\{\mu_1, \mu_2\}$ is by looking where the peaks appear to be. In practice, you can sample any two random data points as long as they are not too close or equal to each other. For the initial guess of

$\{\sigma_1, \sigma_2\}$ we can take the sample standard deviation. Finally for the mixture parameter set them to .5 each.

```
theta = c(50,100,sqrt(var(total_prices)),sqrt(var(total_prices)),.5,.5)


MLE <- GaussianMixtureEM(total_prices,theta)


MLE
```

```
## [1] 148.3982154 199.7934387  29.2514370  10.0027929   0.4777684   0.5222316
```

```
mu1 <- MLE[1]
mu2 <- MLE[2]
v1 <- MLE[3]
v2 <- MLE[4]
p1 <- MLE[5]
p2 <- MLE[6]


hist(total_prices,breaks = 50,freq = FALSE)
x_vals<-seq(0,300,0.01)
points( x_vals, p1*dnorm(x_vals,mu1,v1) + p2*dnorm(x_vals,mu2,v2),  type = "l" , col = "red")
```


Histogram of total_prices
Density
0.000
0.005
0.010
0.015
0.020
100
150
200
total_prices