Estimation of mixture distributions

Martin Zaefferer

July 20, 2021

Mixture distributions: why?

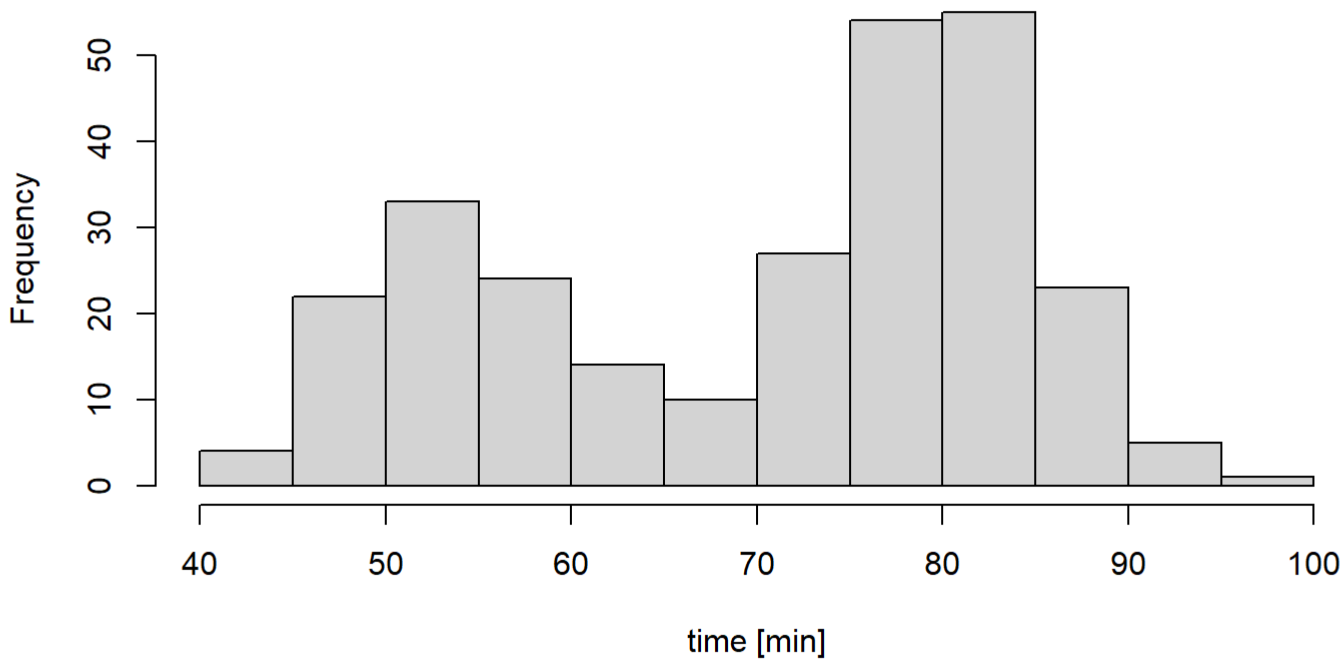
0:00 / 0:30

Old Faithful geyser

Yellowstone National Park (WY, USA), National Park Service, https://www.nps.gov/yell/learn/photosmultimedia/vl_00090mts.htm

Old Faithful: data is multimodal

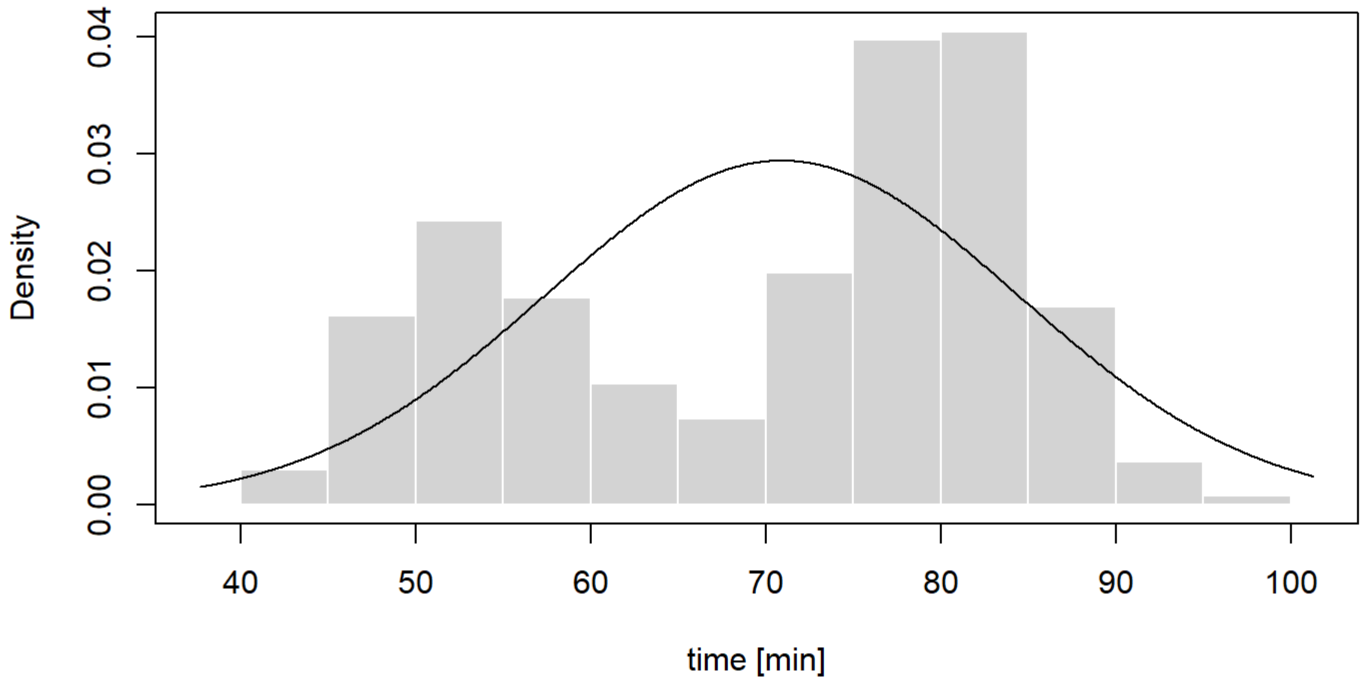
```
hist(multimode::geyser,main="",xlab="time [min]")
```



Time between start of eruptions.

Unimodal distribution (Gaussian): poor fit

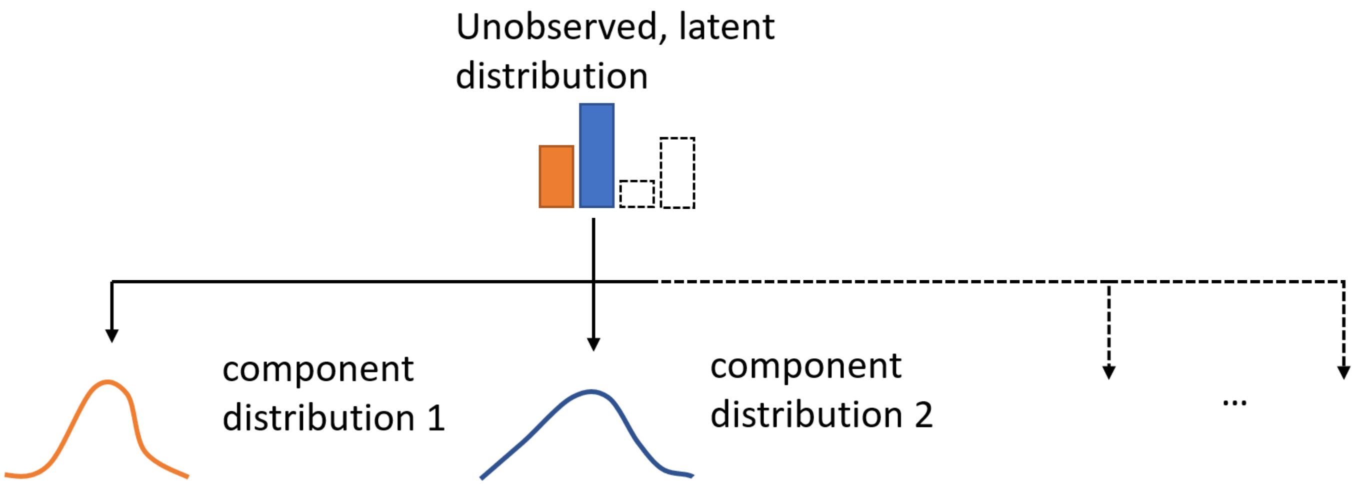
```
require(mclust)
fit <- densityMclust(multimode::geyser,G=1,model="V")
plot(fit, what="density", main="", xlab="time [min]",data=multimode::geyser)
```



Mixture distributions

Mixture distributions

- Mixture of individual component distributions
- Additional distribution (latent) which ‘selects’ a component distribution to be sampled from

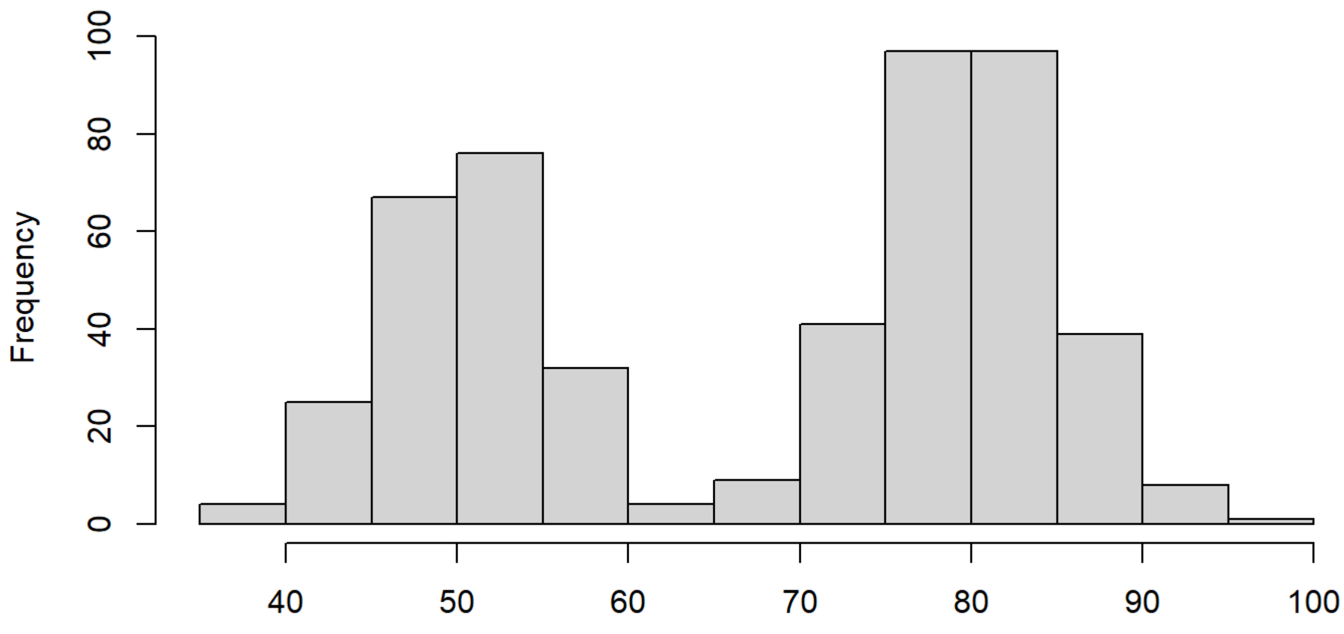


Example: Gaussian mixture

- Gaussian mixture with $k = 2$ components
- Latent distribution: $z \sim Multinomial(0.4, 0.6)$
    - Component 1 with probability $\pi_1 = 0.4$
    - Component 2 with probability $\pi_2 = 0.6$
- Component 1: $x|z = 1 \sim Gaussian(50, 5)$
- Component 2: $x|z = 2 \sim Gaussian(80, 5)$

Example: Gaussian mixture

```
z <- sample(c(1,2),500,prob=c(0.4,0.6),replace=T)
x1 <- rnorm(sum(z==1),50,5); x2 <- rnorm(sum(z==2),80,5)
h <- hist(c(x1,x2),main="",xlab="")
```



Estimation

Parameters

- Parameters of latent distribution
  - Probabilities π_i
- Parameters of each component distribution
  - Mean μ_i and standard deviation σ_i
- Parameters $\theta = [\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2]$ need to be estimated!

Estimation: Maximum Likelihood (ML)

- Idea: find parameters for which the data receives the largest likelihood according to model
- Analytical solution
  - Compute derivative of log-likelihoods $\ell(\theta)$
  - Set derivative to zero, solve for θ
- Turns out to be hard to do
  - E.g., because we do not know z.

What can we do about that?

- Assume that we know from which distribution each sample was drawn
  - I.e., assume to know $z^{(i)}$ for each sample $x^{(i)}$
- MLE becomes straight forward
- Log-likelihood is:
  - $\ell(\theta) = \sum_{i=1}^{m} log\, p(x^{(i)}, z^{(i)}; \theta)$
  - Compute derivatives, set to zero, solve.

MLE solutions (assuming $z^{(i)}$)

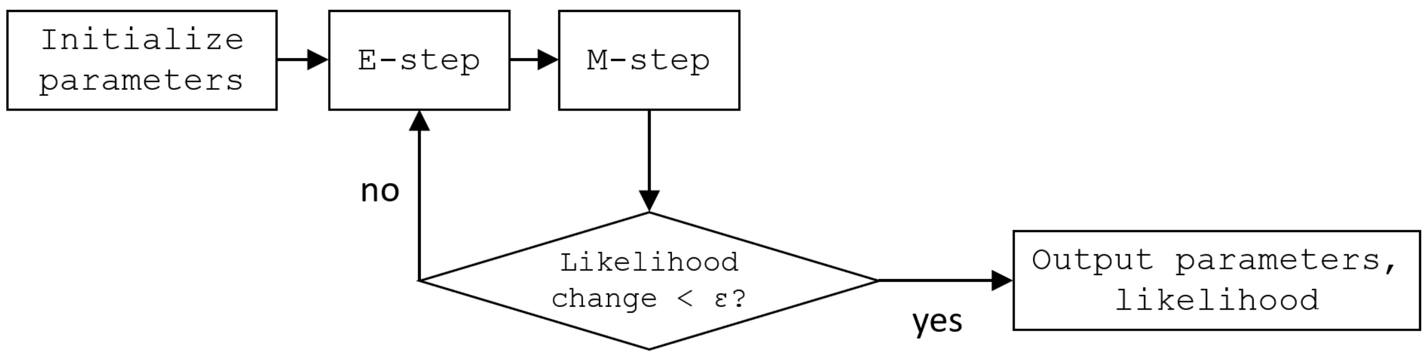
- Solution for π
  - $\pi_j = 1/m \sum_{i=1}^{m} I(z^{(i)} = j)$
  - Intuitively:
- Solution for μ
  - $\mu_j = \frac{\sum_{i=1}^{m} I(z^{(i)}=j)\, x^{(i)}}{\sum_{i=1}^{m} I(z^{(i)}=j)}$
  - Intuitively:
- Solution for σ
  - $\sigma_j^2 = \frac{\sum_{i=1}^{m} I(z^{(i)}=j)\, (x^{(i)}-\mu_j)^2}{\sum_{i=1}^{m} I(z^{(i)}=j)}$
  - Intuitively:

MLE solutions (assuming $z^{(i)}$)

- Solution for π
  - $\pi_j = 1/m \sum_{i=1}^m I(z^{(i)} = j)$
  - Intuitively: π_j fraction of samples from component j
- Solution for μ
  - $\mu_j = \frac{\sum_{i=1}^m I(z^{(i)}=j)\ x^{(i)}}{\sum_{i=1}^m I(z^{(i)}=j)}$
  - Intuitively: μ_j mean of samples from component j
- Solution for σ
  - $\sigma_j^2 = \frac{\sum_{i=1}^m I(z^{(i)}=j)\ (x^{(i)}-\mu_j)^2}{\sum_{i=1}^m I(z^{(i)}=j)}$
  - Intuitively: σ_j variance for component component j

Estimation: Expectation Maximization (EM)

- Attempts to approximate ML estimate iteratively
- Procedure
  - Initialize parameters to some value
  - Alternatingly performs two steps:
    - E-step: Guess $z^{(i)}$ (Expectation!)
    - M-step: Using $z^{(i)}$, maximize likelihoods w.r.t. θ (Maximization!)
  - Stop when change in likelihood below threshold



E-step

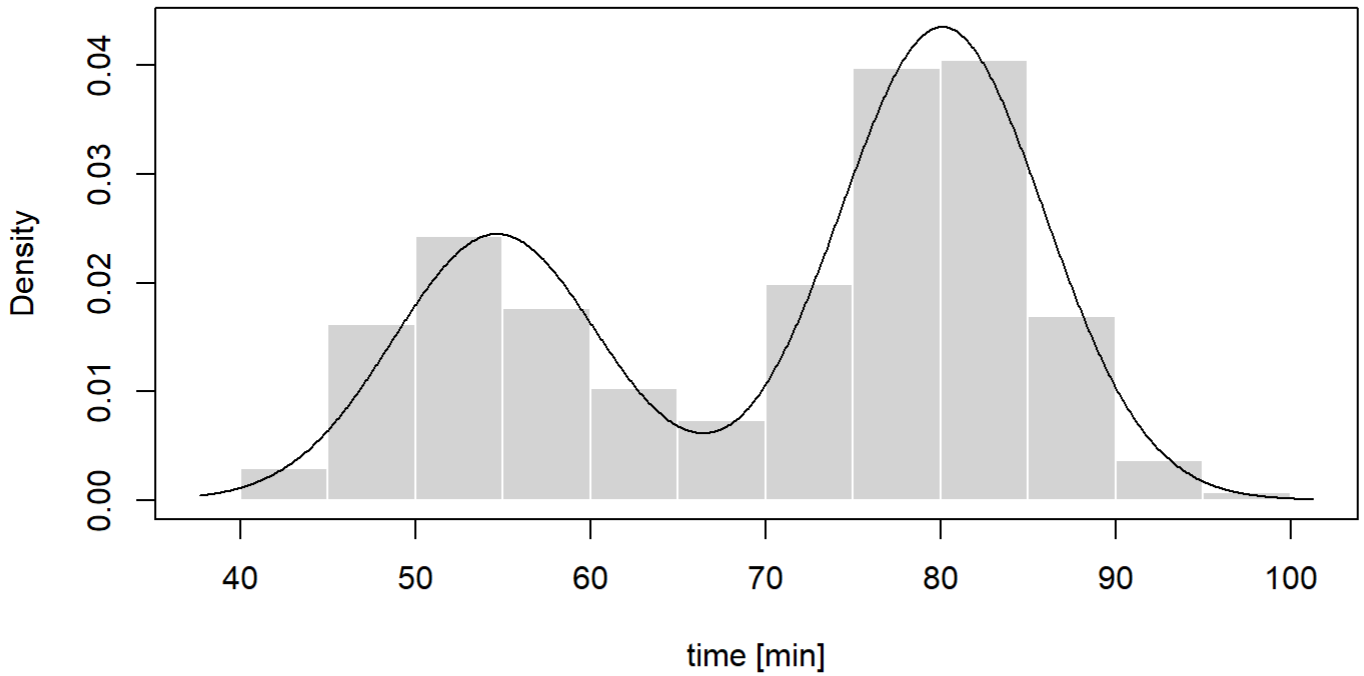
- Estimate $z^{(i)}$, or rather the probability of $z^{(i)} = j$
  - $w_j^{(i)} = p(z^{(i)} = j | x^{(i)}; \theta)$
  - Can be derived via Bayes theorem
  - $w_j^{(i)} = \frac{p(x^{(i)} | z^{(i)} = j)\, p(z^{(i)} = j)}{\sum_{l=1}^{k} p(x^{(i)} | z^{(i)} = l)\, p(z^{(i)} = l)}$

M-step

- With eqs. from slide 15:
  - Replace $I(z^{(i)} = j)$ with $w_j^{(i)}$
    - Because: $w_j^{(i)} = \mathbb{E}[I(z^{(i)} = j)]$
  - Compute θ

Back to Old Faithful: apply EM to data

```
require(mclust)
fit <- densityMclust(multimode::geyser,G=2,model="V")
plot(fit, what="density", main="", xlab="time [min]",data=multimode::geyser)
```

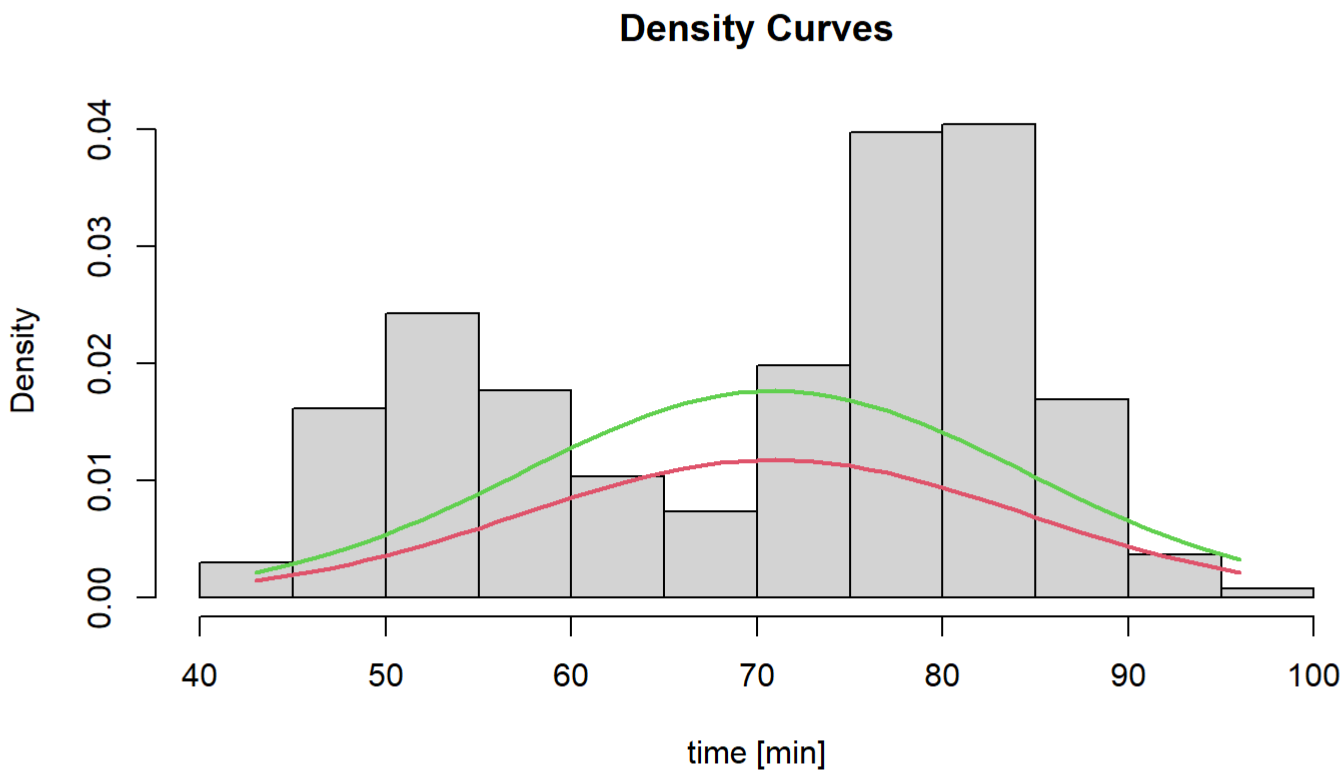


Gaussian mixture density for Old Faithful.

Remaining difficulties

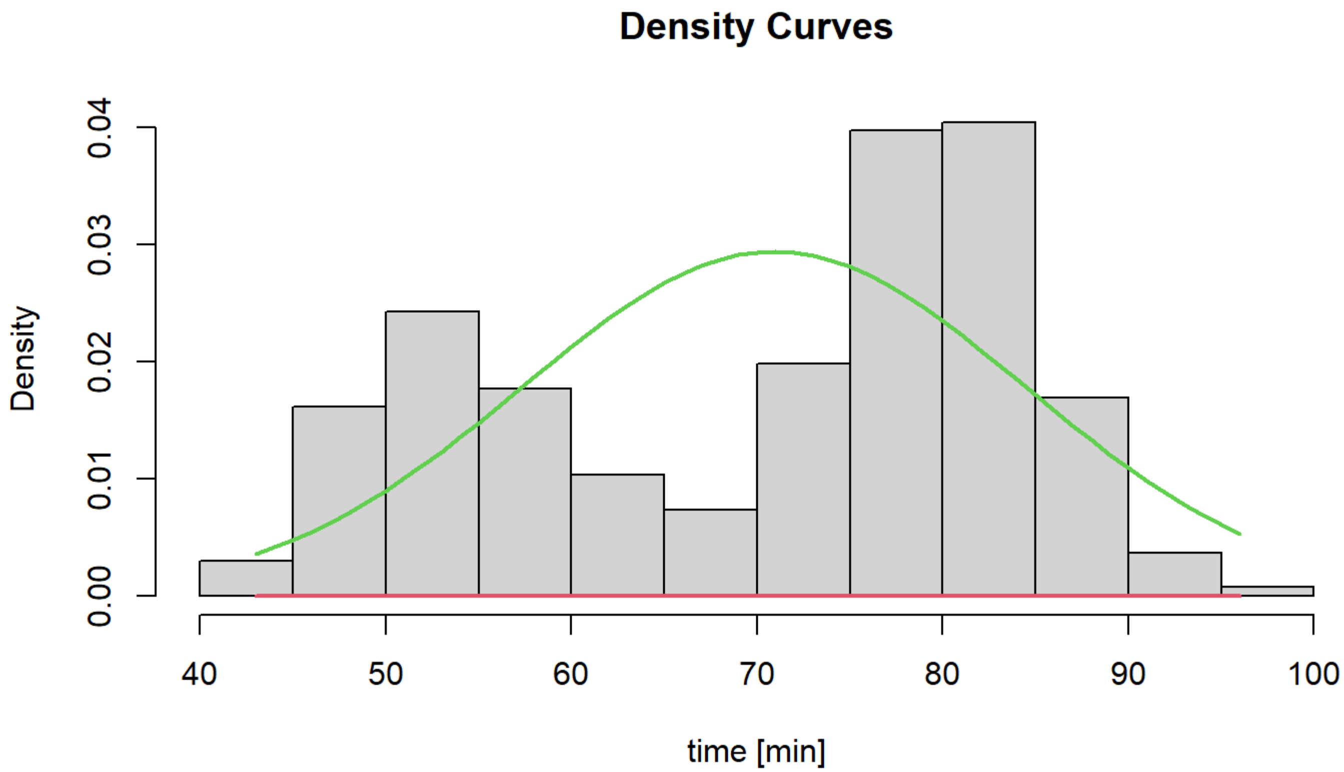
Issue: identical initial guess (μ, σ)

```
require(mixtools)
fit <- normalmixEM(multimode::geyser,k=2,mu=c(50,50),sigma=c(10,10),lambda=c(0.4,0.6))
plot(fit,which=2,xlab2="time [min]")
```



Issue: isolated initial guess

```
fit <- normalmixEM(multimode::geyser,k=2,mu=c(-100,50),sigma=c(10,10),lambda=c(0.5,0.5))
plot(fit,which=2,xlab2="time [min]")
```

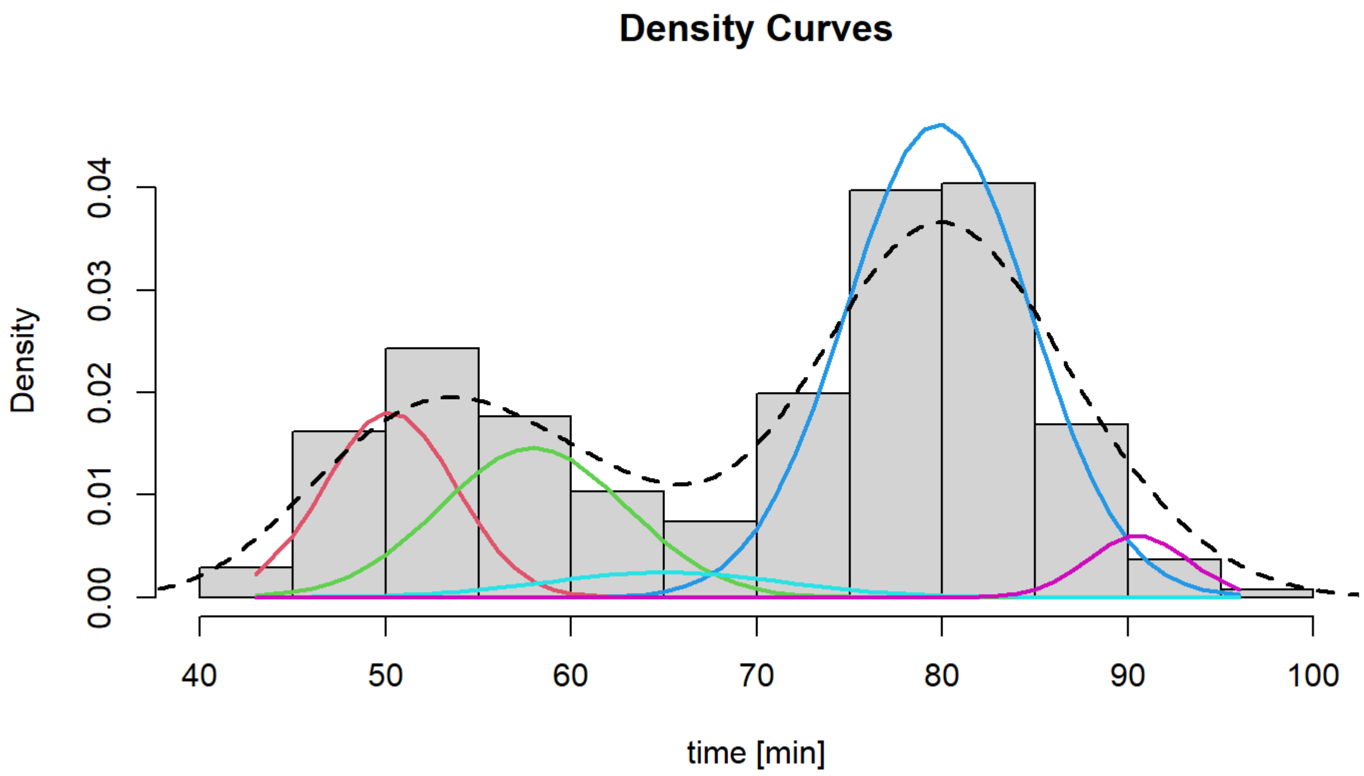


Initial guesses

- Reasonable choice:
  - Set each μ to a different, randomly selected sample value
  - Set all σ to global standard deviation (of all samples)
  - Set probabilities $\pi_i = 1/k$
- Alternative: guess parameters with other algorithm, e.g., k-means
- If results are unsatisfactory, do a restart

Issue: specify k

```
fit <- normalmixEM(multimode::geyser,k=5)
plot(fit,which=2,xlab2="time [min]")
lines(density(fit$x),lwd=2,lty=2)
```



Specify k

- Sometimes k may be known
  - From expert knowledge
  - Or visual inspection
  - Or user ‘needs’ a specific value k
- Else: treat as hyperparameter
  - Select k that optimizes a criterion, e.g., AIC

See: presentation by Andrew Ng, 2020 (stanfordonline)

https://youtu.be/rVfZHWTwXSA

Thanks for your attention. Questions?