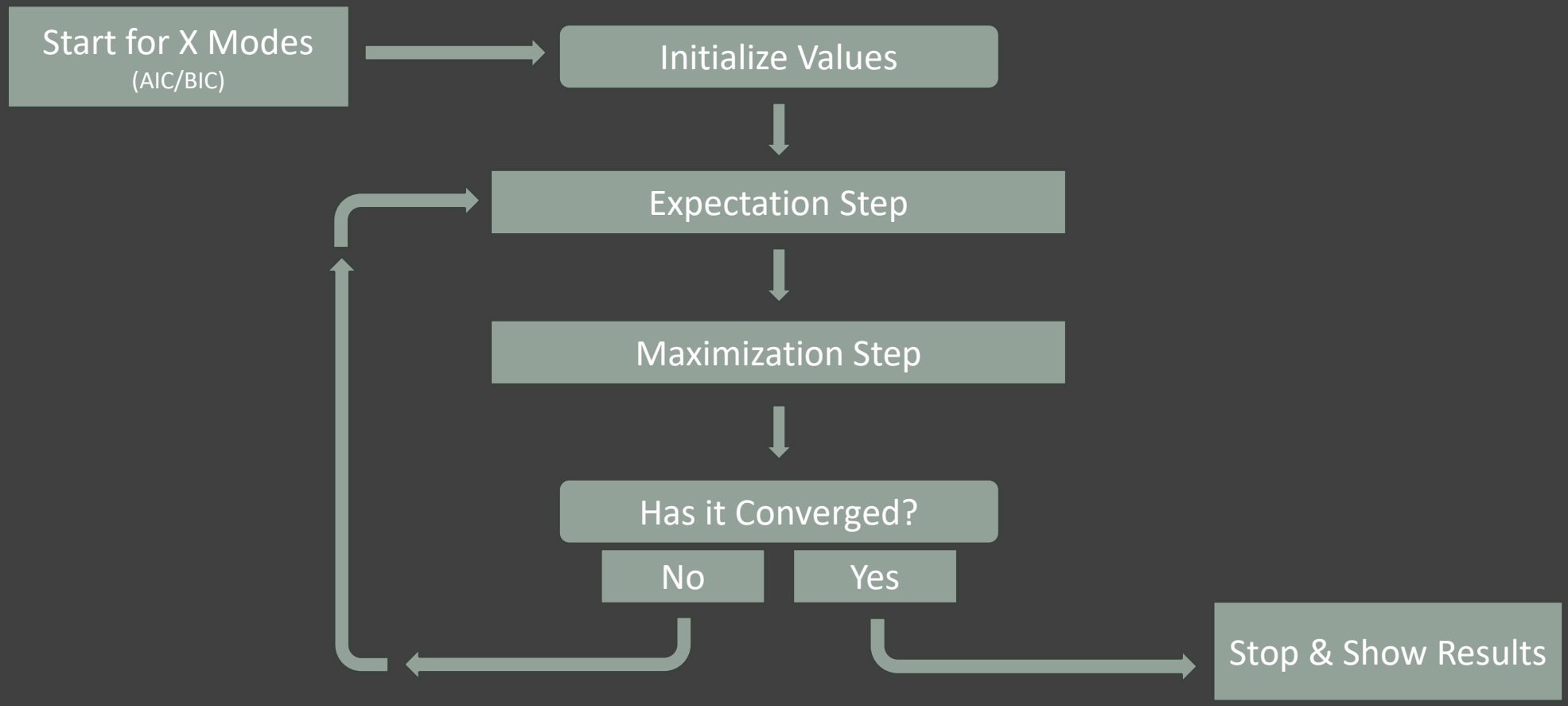
Group B – EM Algorithm

HENRI, JENNY, ROSS

Outline

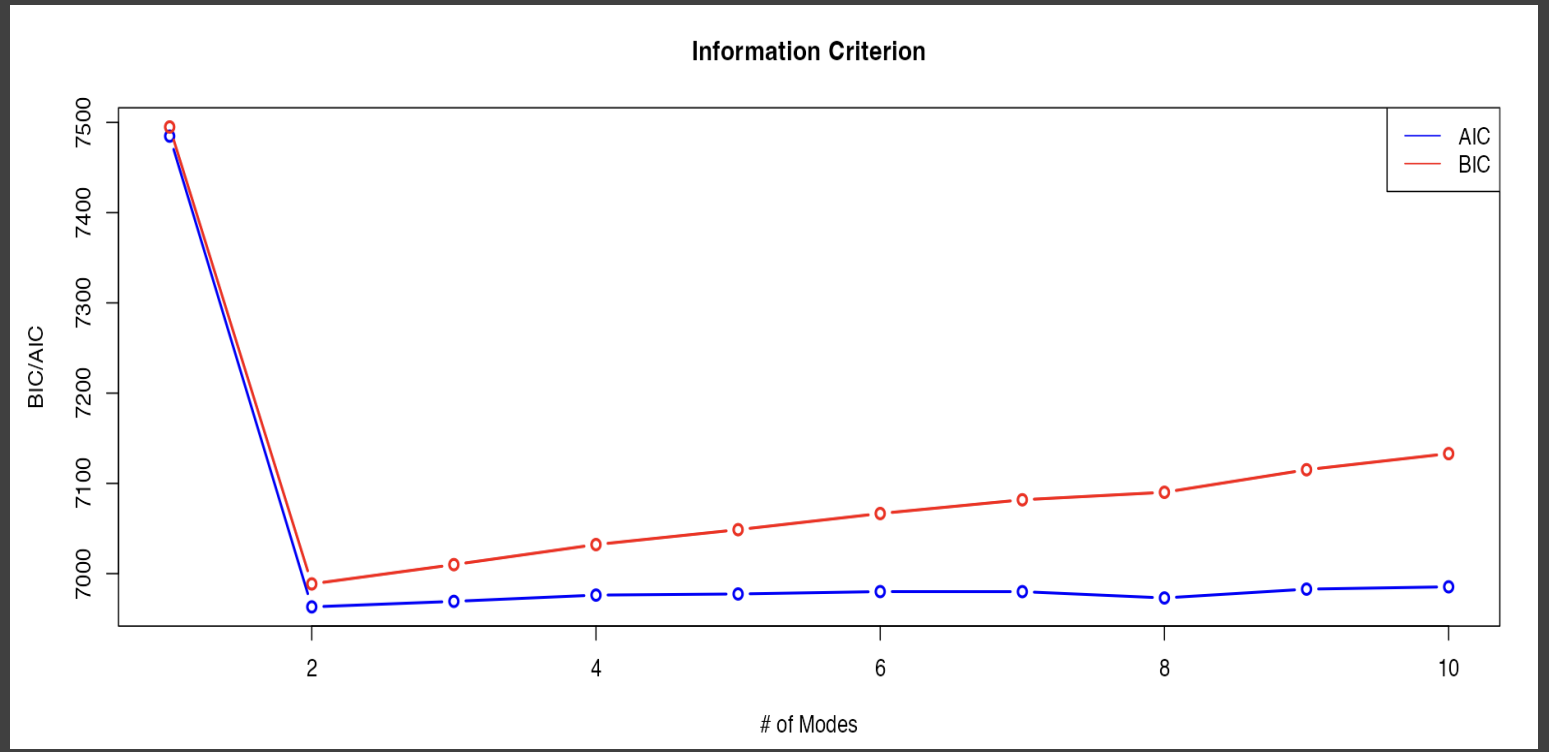
- Steps of EM Algorithm
- Demonstrate App on sim-data.csv
- Demonstrate on Sample Data Set

Steps of EM


Start for X Modes
(AIC/BIC)
Initialize Values
Expectation Step
Maximization Step
Has it Converged?
No
Yes
Stop & Show Results

Steps of EM – Start with X Modes

- Source
- Test 1 to 10 modes
  - Feed in results of EM algorithm
  - Run AIC/BIC
- Pick the minimum



Steps of EM – Initialization of Parameters

-Source

-Using K mean algorithm get x clusters for x modes

-Get mu and sigma for each cluster

```
# Initialize parameters
mem <- kmeans(data, modes)$cluster
mu <- c()
sigma <- c()

for(i in 1:modes)
{
  mu <- c(mu,mean(data[mem==i]))
  sigma <- c(sigma,sd(data[mem==i]))
}
```

Steps of EM – E Step

-Source

-E Step:

- Initialize variables in vector form
- For the number of modes
  - Compute likelihood
- Find Total Likelihood
- For the number of modes
  - Find probability of each likelihood

```r
# Function to compute the probability of each data point belonging to each component
compute_probabilities <- function(data, mu, sigma) {
  likelihood <- rep(0, length(mu))
  likelihood <- matrix(rep(likelihood, length(data)),ncol=length(mu))
  probability_component <- rep(0, length(mu))
  probability_component <- matrix(
    rep(probability_component, length(data)),ncol=length(mu))


  for(i in 1:length(mu)){
    likelihood[,i] <- dnorm(data, mean = mu[i], sd = sigma[i])
  }
  total_likelihood <- rowSums(likelihood)

  for(i in 1:length(mu)){
    probability_component[,i] <- likelihood[,i] / total_likelihood
  }
  return(probability_component)
}
```

Steps of EM – M Step

- M Step
  - Initialize variables in vector form
  - For the number of modes
    - Updated mu and sigma based off formula

```r
# Function to update the parameters using the computed probabilities
update_parameters <- function(data, probabilities) {
  sum_prob<-rep(0, dim(probabilities)[2])
  mu<-rep(0, dim(probabilities)[2])
  sigma<-rep(0, dim(probabilities)[2])

  for(i in 1:dim(probabilities)[2]){
    sum_prob[i] <- sum(probabilities[,i])
    mu[i] <- sum(probabilities[,i] * data) / sum_prob[i]
    sigma[i] <- sqrt(sum(probabilities[,i] * (data - mu[i])^2) / sum_prob[i])
  }
  return(list(mu = mu, sigma = sigma))
}
```

Source:

$$\mu_j^1 = \frac{\sum_i \hat{p}_{ij} y_i}{\sum_i \hat{p}_{ij}}$$

$$\sigma_j^{2(1)} = \frac{\sum_i \hat{p}_{ij} (y_i - \mu_j^1)^2}{\sum_i \hat{p}_{ij}}.$$

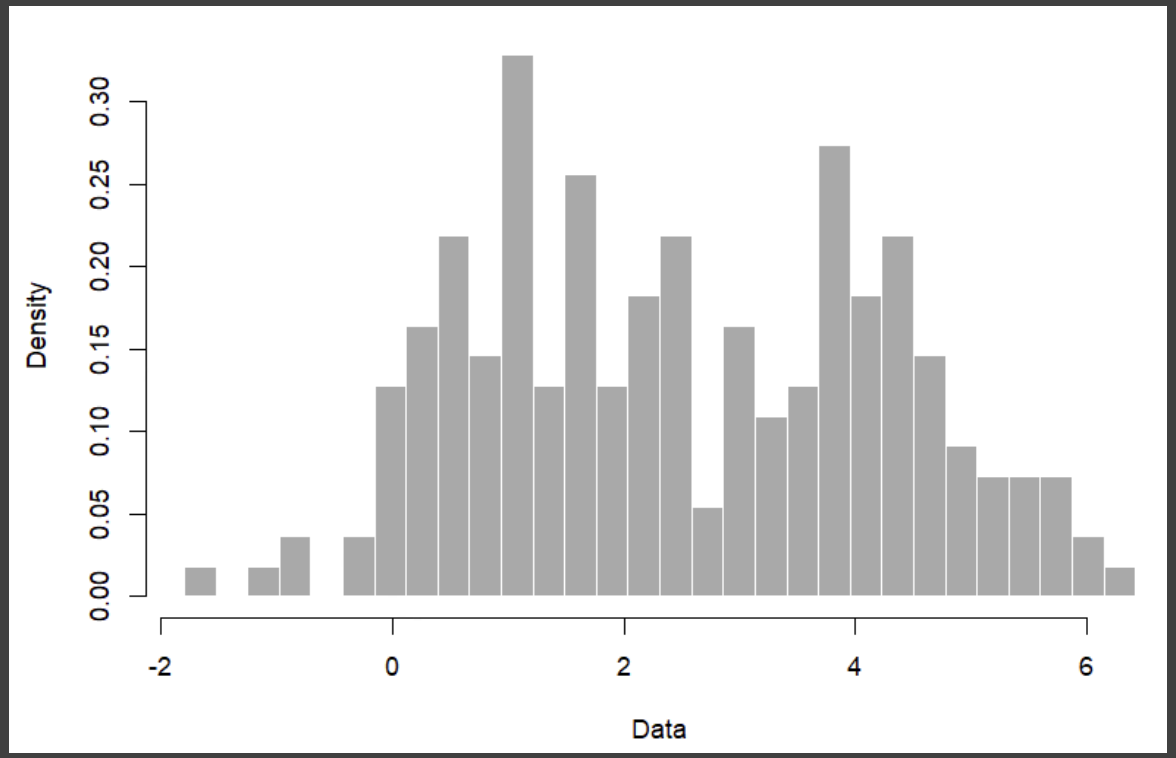
Steps of EM – Checking for Convergence

- Epsilon
  - We set very small (0.000001)
  - The decimal differences between previous and current iteration
- Convergence
  - 2 norm of mu + 2 norm of sigma
  - Or
  - Reach max iterations which is 1000

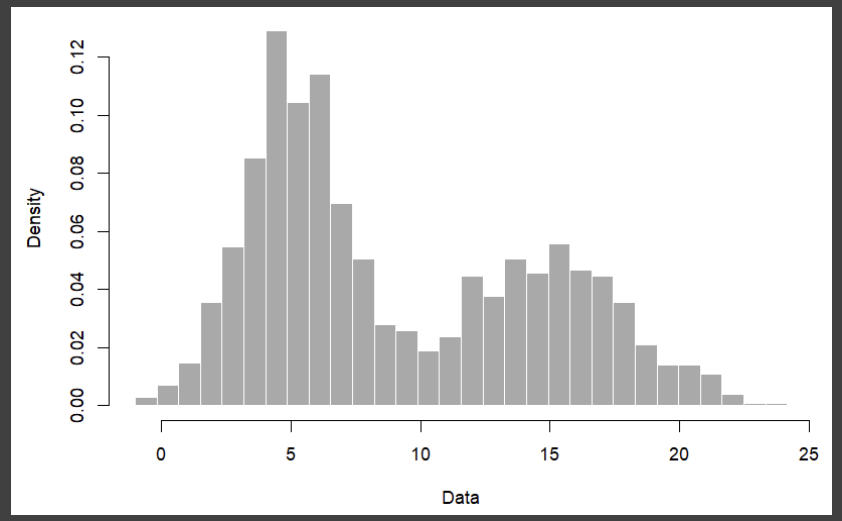
```
norm2 <- ((norm(t(mu),"2")+norm(t(sigma),"2")) - (norm(t(parameters$mu),"2")+norm(t(parameters$sigma),"2")))**2

if(norm2 <epsilon**2){
  globalValues$convergence <-iteration
  break
}
```

Demonstrate on sim-data.csv

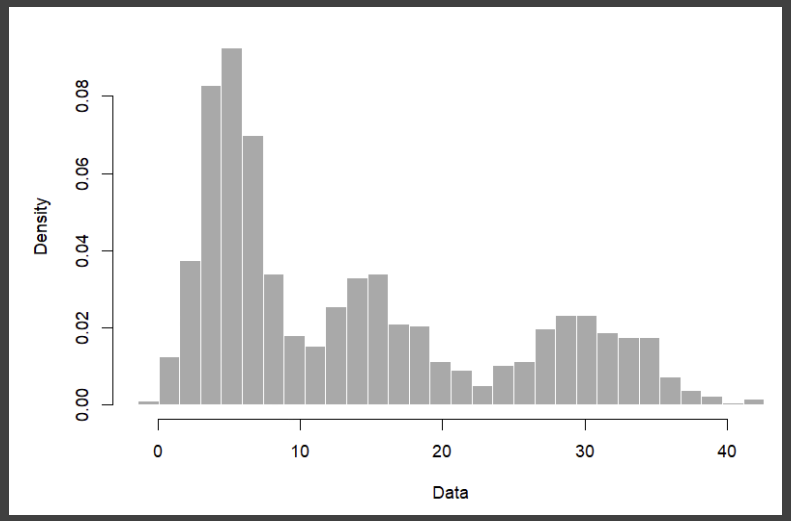


New Data Set



c(rnorm(500, mean = 5, sd = 2),

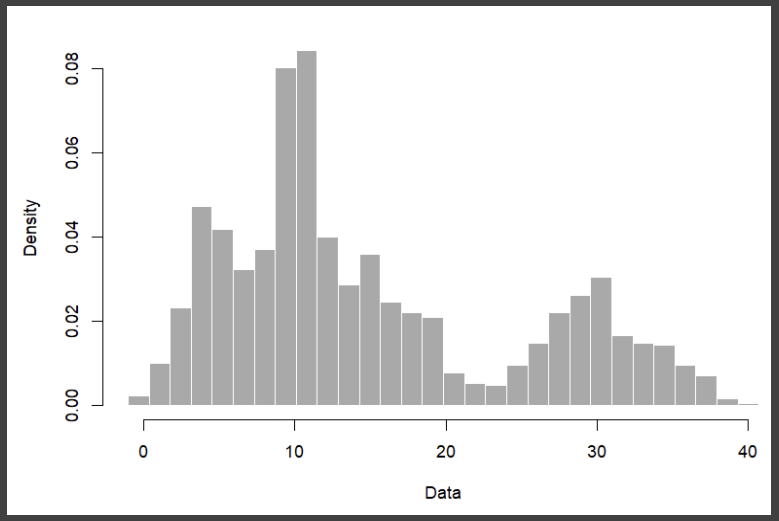
rnorm (500, mean = 15, sd = 3))



c(rnorm(300, mean = 5, sd = 2),

rnorm(300, mean = 15, sd = 3),

rnorm(300, mean = 30, sd = 4))



c(rnorm(300, mean = 5, sd = 2),

rnorm(300, mean = 15, sd = 3),

rnorm(300, mean = 30, sd = 4),

rnorm(300, mean = 10, sd = 1)

Questions?

Example Questions:

- What bugs were found in the original but not on ours?
- What was the hardest part of the assignment?