Report

Qi Qi

November 27, 2018

Distribution of Experimental Data

Test the normality of experimental data set:

```
test <- read.csv("Data for Root Cause Determination - Test Data.csv")
dat <- read.csv("Data for Root Cause Determination.csv")
mean(test$Response[test$Group=="Test Group - Root Cause"])
```

```
## [1] 108.9863
```

```
sd(test$Response[test$Group=="Test Group - Root Cause"])
```

```
## [1] 5.290166
```

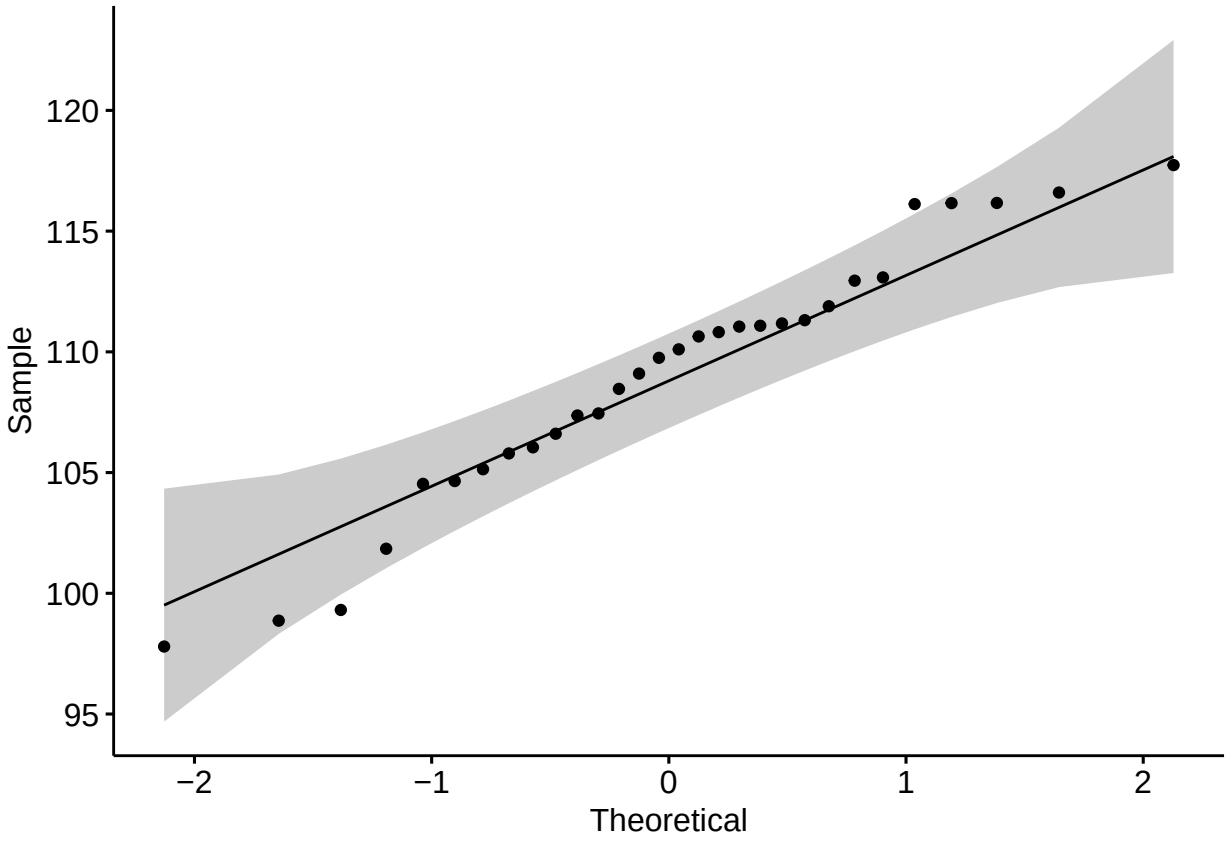
```
mean(test$Response[test$Group=="Test Group - No Root Cause"])
```

```
## [1] 100.7146
```

```
sd(test$Response[test$Group=="Test Group - No Root Cause"])
```

```
## [1] 5.821144
```

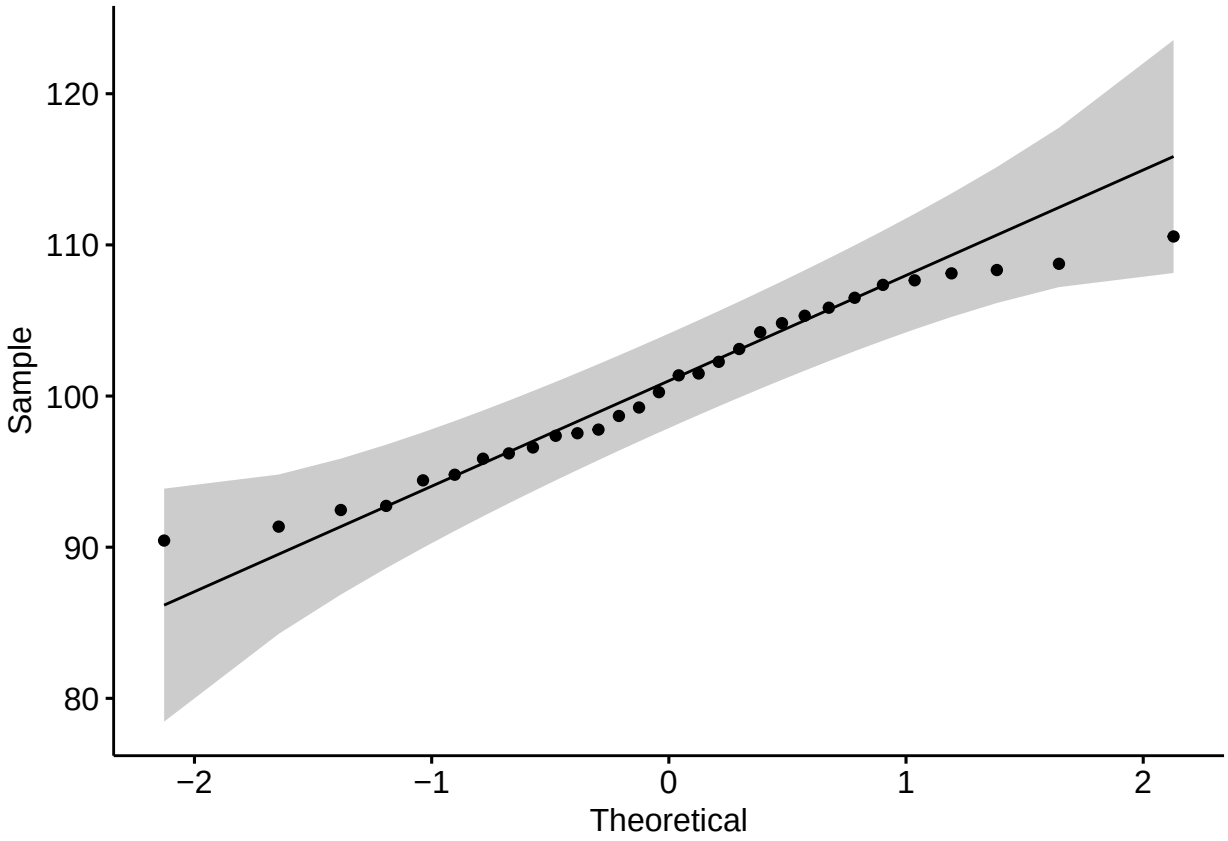
```
ggqqplot(test[test$Group == "Test Group - Root Cause",]$Response)
```




```
shapiro.test(test[test$Group == "Test Group - Root Cause",]$Response)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  test[test$Group == "Test Group - Root Cause", ]$Response
## W = 0.9609, p-value = 0.3266
```

```
ggqqplot(test[test$Group == "Test Group - No Root Cause",]$Response)
```




```
shapiro.test(test[test$Group == "Test Group - No Root Cause",]$Response)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  test[test$Group == "Test Group - No Root Cause", ]$Response
## W = 0.95875, p-value = 0.2877
```

From above result, we know for each group the data is normally distributed. Since sample size is 30 in each group, we do not have much power. Then I also test goodness of fit of Gamma distribution:

```
gamma_test(test[test$Group == "Test Group - Root Cause",]$Response)
```

```
##
##  Test of fit for the Gamma distribution
##
## data:  test[test$Group == "Test Group - Root Cause", ]$Response
## V = -1.2239, p-value = 0.3868
```

```
gamma_test(test[test$Group == "Test Group - No Root Cause",]$Response)
```

```
##
##  Test of fit for the Gamma distribution
```

```
##
## data:  test[test$Group == "Test Group - No Root Cause", ]$Response
## V = -0.33574, p-value = 0.8123
```

Above results also show gamma distribution fits the data set.

Then I propose mixed normal distribution and mixed gamma distribution of original data set.

Mixture of Normal Distributions

Let z_i be the index of the Gaussian distribution from which x_i is sampled. The parameters to be estimated is $(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \xi_1, \xi_2)$, where $\xi_1 + \xi_2 = 1$. Let $\theta_t = (\mu_{t1}, \mu_{t2}, \sigma_{t1}^2, \sigma_{t2}^2, \xi_{t1}, \xi_{t2})$

$$Q(\theta|\theta_t) = \sum_z p(z|x, \theta_t) \ln p(x, z|\theta) = \sum_{i=1}^{n} \sum_{k=1}^{2} p(z_i = k|x_i, \theta_t) \ln p(x_i, z_i = k|\theta)$$

Let $w_{ik} = p(z_i = k|x_i, \theta_t)$, then

$$w_{ik} = \frac{p(z_i = k, x_i, |\theta_t)}{\sum_{k=1}^{2} p(z_i = k, x_i|\theta_t)} = \frac{\xi_{tk}\phi(x_i|\mu_{tk}, \sigma_{tk}^2)}{\sum_{k=1}^{2} \xi_{tk}\phi(x_i|\mu_{tk}, \sigma_{tk}^2)}$$

$$Q(\theta|\theta_t) = \sum_{k=1}^{2} \sum_{i=1}^{n} w_{ik} \ln(\xi_k/\sqrt{2\pi}) - \frac{1}{2} \sum_{k=1}^{2} \sum_{i=1}^{n} w_{ik} \ln \sigma_k^2 - \frac{1}{2} \sum_{k=1}^{2} \sum_{i=1}^{n} w_{ik} \frac{(x_i - \mu_k)^2}{\sigma_k^2}$$

$$\frac{\partial Q(\theta|\theta_t)}{\partial \mu_k} = 0 \Rightarrow \mu_k = \frac{\sum_{i=1}^{n} w_{ik} x_i}{\sum_{i=1}^{n} w_{ik}}$$

$$\frac{\partial Q(\theta|\theta_t)}{\partial \sigma_k^2} = 0 \Rightarrow \sigma_k^2 = \frac{\sum_{i=1}^{n} w_{ik}(x_i - \mu_k)^2}{\sum_{i=1}^{n} w_{ik}}$$

$$\xi_k = \frac{1}{n} \sum_{i=1}^{n} w_{ik}$$

Implement in R

```
mynormalmixEM <- function(x, xi1, xi2, mu1, mu2, sigma1, sigma2, maxit, tol){
  n <- length(x)
  w1 <- double(n)
  w2 <- double(n)
  for (i in 1:maxit){
     for (j in 1:n){
      w1[j] <- xi1 * dnorm(x[j], mu1, sigma1) / (xi1 * dnorm(x[j], mu1, sigma1) + xi2 * dnorm(x[j], mu2
      w2[j] <- 1 - w1[j]
    }
    xi1new <- mean(w1)
    xi2new <- mean(w2)
    mu1new <- sum(w1 * x) / sum(w1)
    mu2new <- sum(w2 * x) / sum(w2)
    sigma1new <- sqrt(sum(w1 * (x - mu1new)^2) / sum(w1))
    sigma2new <- sqrt(sum(w2 * (x - mu2new)^2) / sum(w2))
    if (max(abs(xi1new - xi1), abs(xi2new - xi2), abs(mu1new - mu1), abs(mu2new - mu2), abs(sigma1new -
      xi <- c(xi1new, xi2new)
```

```
      mu <- c(mu1new, mu2new)
      sigma <- c(sigma1new, sigma2new)
      iter <- i
      return(list(xi, mu, sigma, iter))
    }
    xi1 <- xi1new
    xi2 <- xi2new
    mu1 <- mu1new
    mu2 <- mu2new
    sigma1 <- sigma1new
    sigma2 <- sigma2new
  }
}

mixnormal <- function(n, xi, mu1, mu2, sigma1, sigma2){
  x <- double(n)
  for (i in 1:n){
    u <- runif(1)
    x[i] <- rnorm(1, ifelse(u < xi, mu1, mu2), ifelse(u < xi, sigma1, sigma2))
  }
  x
}
## simulated data
data <- mixnormal(1000, .6, 3, 8, 1, 1)
mynormalmixEM(data, .5, .5, 6, 7, 1, 4, 1e5, 1e-5)
```

```
## [[1]]
## [1] 0.4292453 0.5707547
##
## [[2]]
## [1] 7.956547 2.987372
##
## [[3]]
## [1] 0.9931318 1.0011016
##
## [[4]]
## [1] 28
```

```
out.1 <- normalmixEM(data, arbvar = FALSE, epsilon = 1e-03, fast=TRUE)
```

```
## number of iterations= 15
```

```
summary(out.1)
```

```
## summary of normalmixEM object:
##          comp 1   comp 2
## lambda 0.570537 0.429463
## mu     2.986420 7.955297
## sigma  0.997673 0.997673
## loglik at estimate:  -2082.5
```

```r
fit.1 <- mixfit(data, ncomp = 2, family = "normal")
fit.1
```

```
## Normal mixture model with 2 components
##        comp1     comp2
## pi 0.5707521 0.4292479
## mu 2.9873607 7.9565333
## sd 1.0010887 0.9931468
##
## EM iterations: 9 AIC: 4174.9791661 BIC: 4199.5179425 log-likelihood: -2082.489583
```

```r
## real data
mynormalmixEM(test$Response, .5, .5, 100, 110, 3, 5, 1e5, 1e-5)
```

```
## [[1]]
## [1] 0.3006257 0.6993743
##
## [[2]]
## [1]  96.59844 108.39757
##
## [[3]]
## [1] 3.344417 4.533871
##
## [[4]]
## [1] 104
```

```r
mynormalmixEM(dat$Response, .5, .5, 100, 110, 3, 5, 1e5, 1e-5)
```

```
## [[1]]
## [1] 0.93670027 0.06329973
##
## [[2]]
## [1] 100.7110 113.6586
##
## [[3]]
## [1] 5.139338 3.633500
##
## [[4]]
## [1] 2047
```

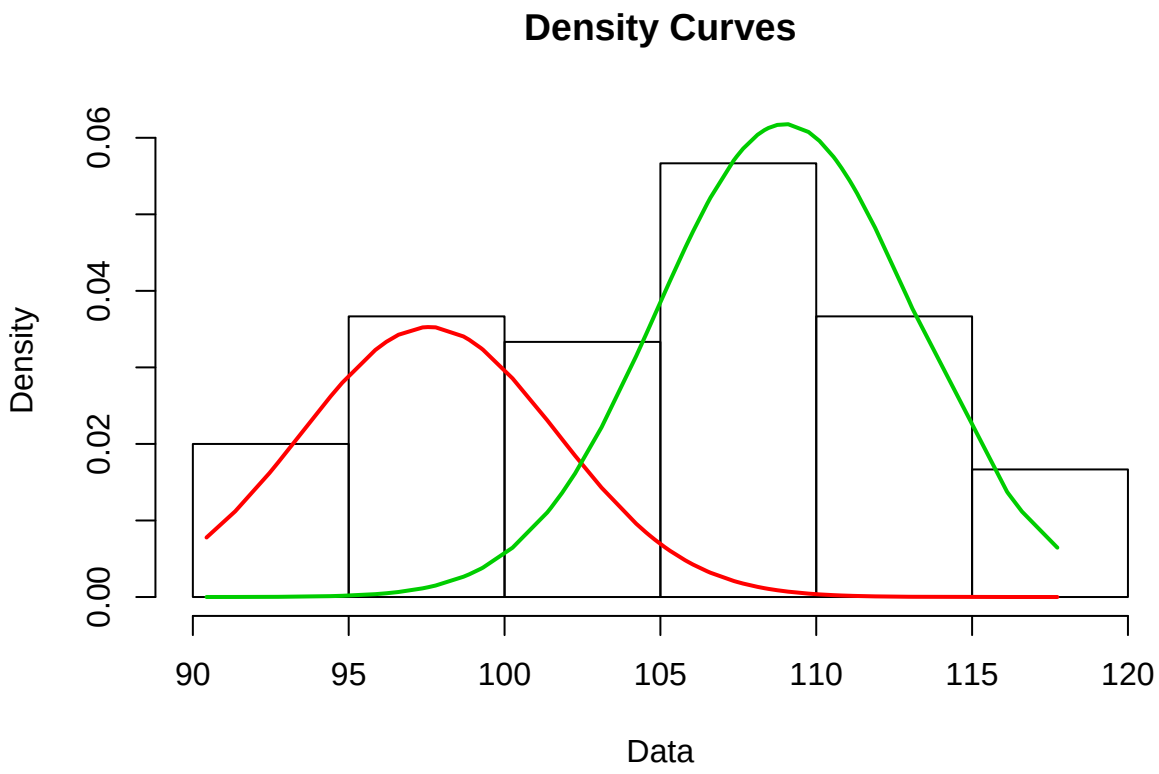
```r
## Using "mixtools" package
out.1 <- normalmixEM(test$Response, arbvar = FALSE, epsilon = 1e-03, fast=TRUE)
```

```
## number of iterations= 7
```

```r
summary(out.1)
```

```
## summary of normalmixEM object:
##           comp 1     comp 2
## lambda  0.363386   0.636614
## mu     97.586722 108.996657
## sigma   4.110096   4.110096
## loglik at estimate:  -197.9952
```

```
plot(out.1, density = TRUE, w = 1.1)
```



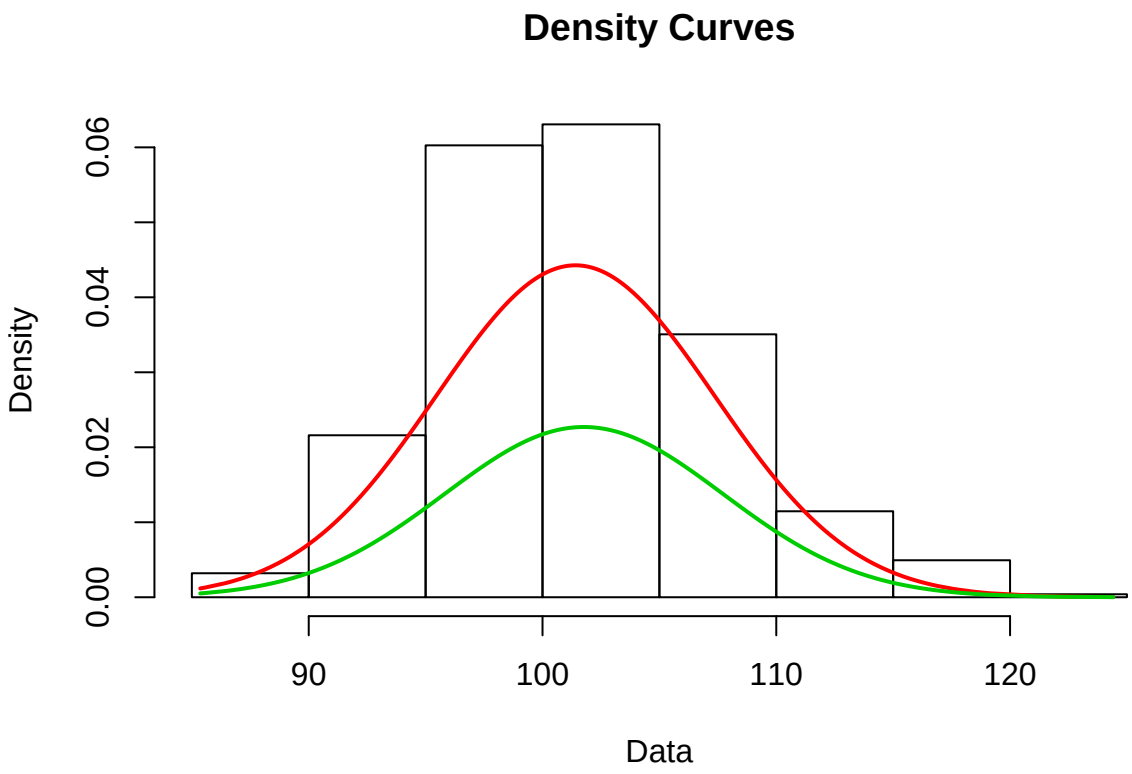

```
out.2 <- normalmixEM(dat$Response, arbvar = FALSE, epsilon = 1e-03, fast=TRUE)
```

```
## number of iterations= 4
```

```
summary(out.2)
```

```
## summary of normalmixEM object:
##            comp 1     comp 2
## lambda   0.660999   0.339001
## mu     101.409448 101.766750
## sigma    5.957172   5.957172
## loglik at estimate:  -4805.904
```

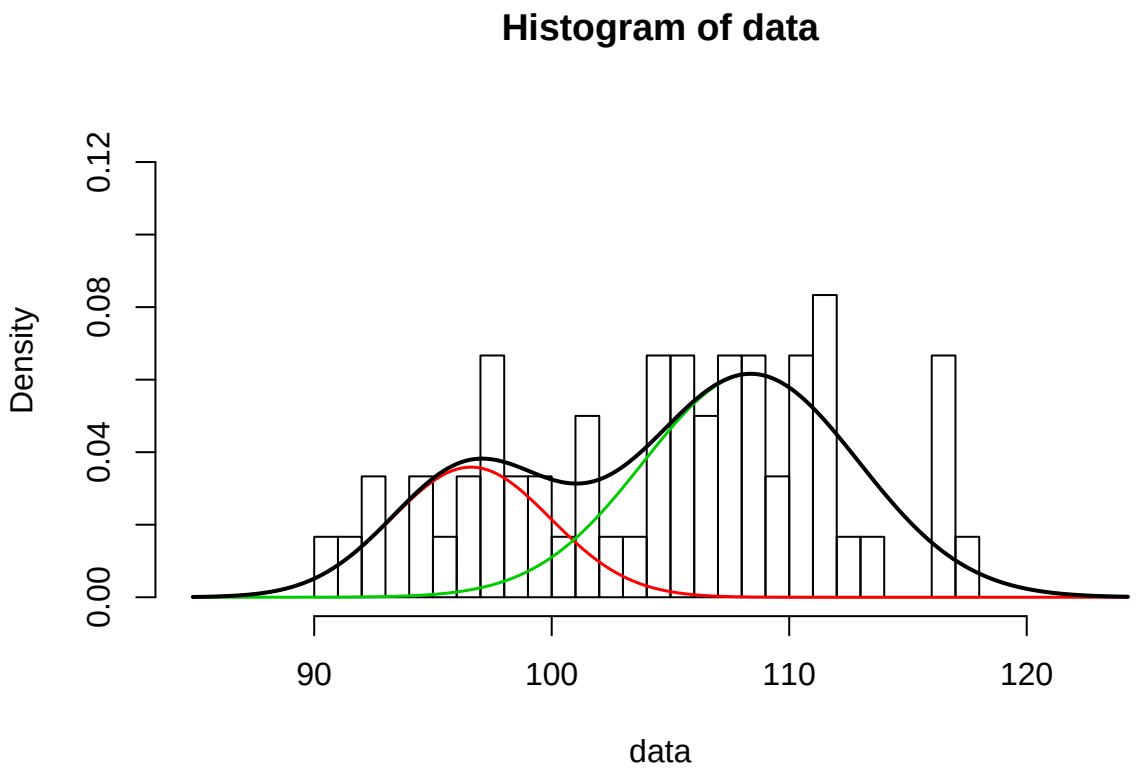
```
plot(out.2, density = TRUE, w = 1.1)
```




```
## Using "mixR" package
fit.1 <- mixfit(test$Response, ncomp = 2, family = "normal")
fit.1
```

```
## Normal mixture model with 2 components
##        comp1      comp2
## pi  0.300867   0.699133
## mu 96.601773 108.400210
## sd  3.346144   4.532115
##
## EM iterations: 123 AIC: 405.4654584 BIC: 415.9371812 log-likelihood: -197.7327292
```

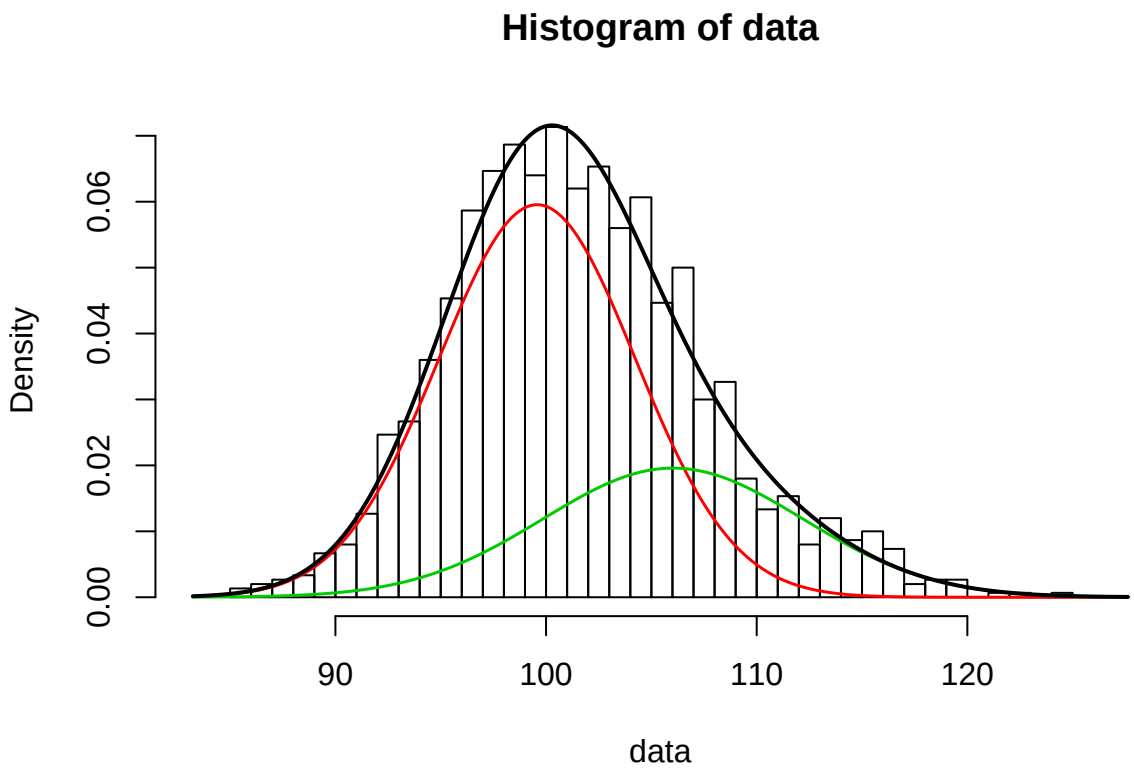
```
plot(fit.1)
```




```
fit.2 <- mixfit(dat$Response, ncomp = 2, family = "normal")
fit.2
```

```
## Normal mixture model with 2 components
##         comp1       comp2
## pi  0.6970251   0.3029749
## mu 99.5776316 106.0235184
## sd  4.6701012   6.1716343
##
## EM iterations: 500 AIC: 9574.9300337 BIC: 9601.4961356 log-likelihood: -4782.4650168
```

```
plot(fit.2)
```



Mixture of Gamma Distributions

Let z_i be the index of the Gaussian distribution from which x_i is sampled. The parameters to be estimated is $(\alpha_1, \alpha_2, \beta_1, \beta_2, \xi_1, \xi_2)$, where $\xi_1 + \xi_2 = 1$. Let $\theta_t = (\alpha_{t1}, \alpha_{t2}, \beta_{t1}, \beta_{t2}, \xi_{t1}, \xi_{t2})$

$$Q(\theta|\theta_t) = \sum_z p(z|x, \theta_t) \ln p(x, z|\theta) = \sum_{i=1}^{n} \sum_{k=1}^{2} p(z_i = k|x_i, \theta_t) \ln p(x_i, z_i = k|\theta)$$

Let $w_{ik} = p(z_i = k|x_i, \theta_t)$, then

$$w_{ik} = \frac{p(z_i = k, x_i, |\theta_t)}{\sum_{k=1}^{2} p(z_i = k, x_i|\theta_t)} = \frac{\xi_{tk}\Gamma(x_i|\alpha_{tk}, \beta_{tk})}{\sum_{k=1}^{2} \xi_{tk}\Gamma(x_i|\alpha_{tk}, \beta_{tk})}$$

$$Q(\theta|\theta_t) = \sum_{k=1}^{2}\sum_{i=1}^{n} w_{ik} \ln \xi_k - \sum_{k=1}^{2}\sum_{i=1}^{n} w_{ik} \ln \Gamma(\alpha_k) - \sum_{k=1}^{2}\sum_{i=1}^{n} w_{ik}\alpha_k \ln \beta_k + \sum_{k=1}^{2}\sum_{i=1}^{n} w_{ik}(\alpha_k - 1) \ln x_i - \sum_{k=1}^{2}\sum_{i=1}^{n} \frac{w_{ik}x_i}{\beta_k}$$

$$\frac{\partial Q(\theta|\theta_t)}{\partial \beta_k} = 0 \Rightarrow \beta_k = \frac{\sum_{i=1}^{n} w_{ik}x_i}{\alpha_k \sum_{i=1}^{n} w_{ik}}$$

$$\xi_k = \frac{1}{n}\sum_{i=1}^{n} w_{ik}$$

Implement in R

```r
## Using "mixtools" package
out.1 <- gammamixEM(test$Response)
```

```
## number of iterations= 60
```

```r
out.1$lambda
```

```
## [1] 0.6933612 0.3066388
```

```r
out.1$gamma.pars
```

```
##            comp.1       comp.2
## alpha 357.0949481 1.793120e+03
## beta    0.2853507 6.219911e-02
```

```r
out.2 <- gammamixEM(dat$Response)
```

```
## number of iterations= 89
```

```r
out.2$lambda
```

```
## [1] 0.1555763 0.8444237
```

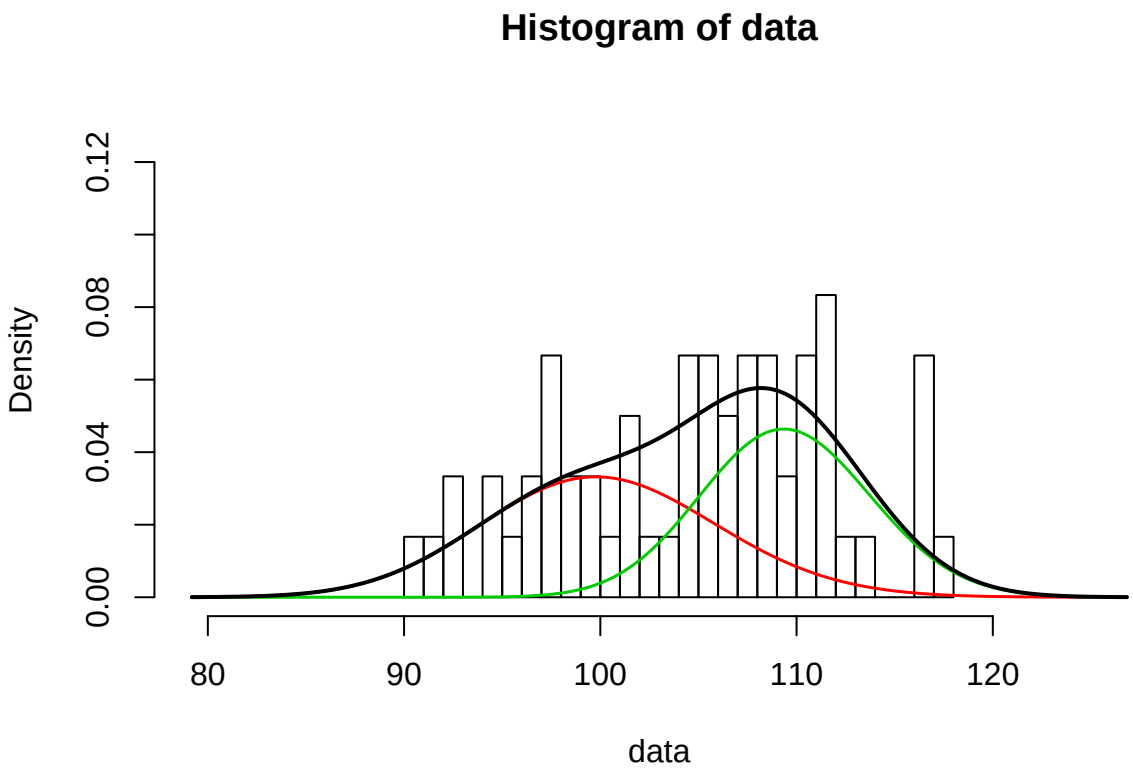
```r
out.2$gamma.pars
```

```
##             comp.1      comp.2
## alpha 1.720604e+03 428.3528487
## beta  5.527022e-02   0.2397912
```

```r
## Using "mixR" package
fit.1 <- mixfit(test$Response, ncomp = 2, family = "gamma")
fit.1
```

```
## Gamma mixture model with 2 components
##             comp1       comp2
## pi      0.4971635   0.5028365
## mu    100.1284380 109.5191880
## sd      5.9838166   4.3291254
## shape 280.0001953 640.0001953
## rate    2.7964103   5.8437266
##
## EM iterations: 63 AIC: 407.5022053 BIC: 417.9739282 log-likelihood: -198.7511027
```

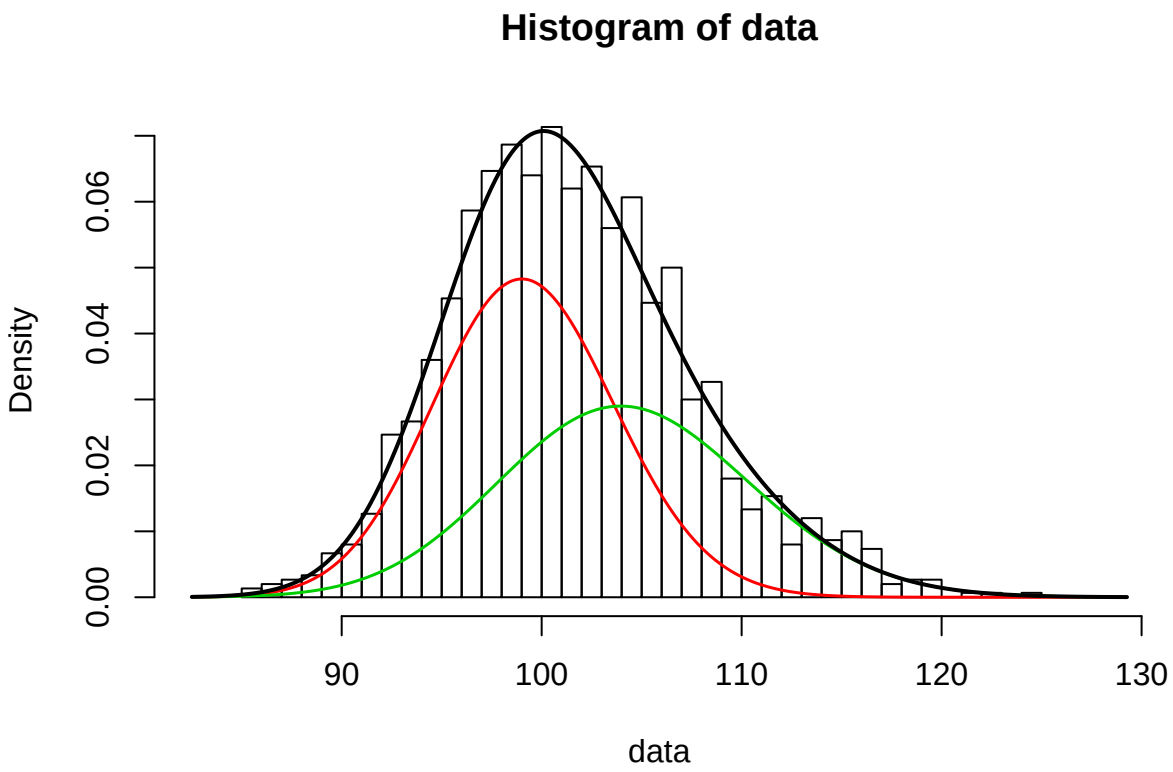
```r
plot(fit.1)
```




```
fit.2 <- mixfit(dat$Response, ncomp = 2, family = "gamma")
fit.2
```

```
## Gamma mixture model with 2 components
##              comp1       comp2
## pi       0.5475391   0.4524609
## mu      99.2184792 104.3285214
## sd       4.5286823   6.2348195
## shape  480.0001953 280.0001953
## rate     4.8378104   2.6838317
##
## EM iterations: 340 AIC: 9576.0562321 BIC: 9602.6223341 log-likelihood: -4783.0281161
```

```
plot(fit.2)
```


Histogram of data
Density
0.00
0.02
0.04
0.06
90
100
110
120
130
data