PBSR- Assignment 2

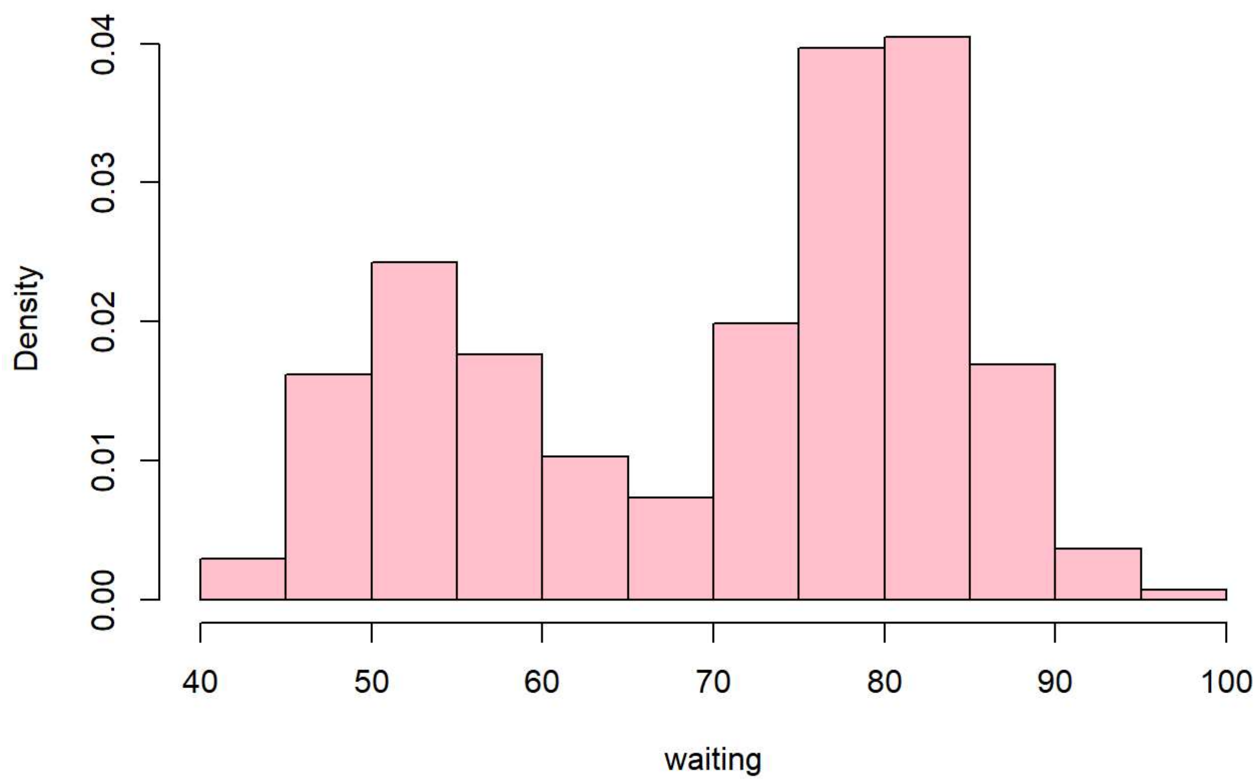
Anurag

2022-11-15

Problem 3: Analysis of `faithful` datasets.

Consider the `faithful` datasets:

```
attach(faithful)
hist(faithful$waiting,xlab = 'waiting',probability = T,col='pink',main='')
```

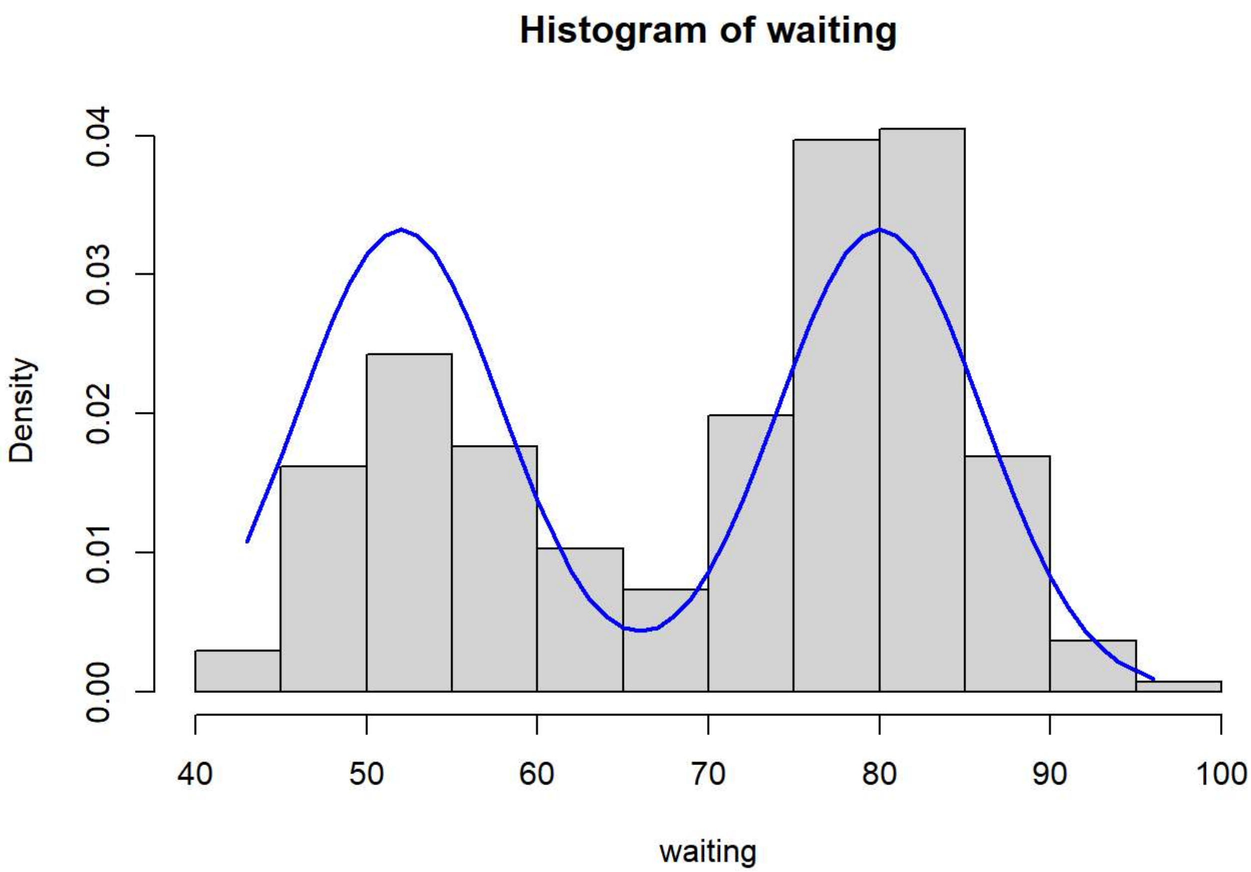


Fit following three models using MLE method and calculate **Akaike information criterion** (aka., AIC) for each fitted model. Based on AIC decides which model is the best model? Based on the best model calculate the following probability

$$\mathbb{P}(60 < \texttt{waiting} < 70)$$

Faithful$Waiting represents bimodal data. If we divide the data into two halves, First half approximately has a mean of 52 with a sd of 6 while the second half approximately has a mean of 80 with sd 6.

```
waiting=sort(waiting)
hist(waiting,probability = T)
p=0.5

d = p*dnorm(waiting,mean = 52,sd=6)+(1-p)*dnorm(waiting,mean=80,sd=6)
lines(waiting,d,lwd=2,col='blue')
```



i. **Model 1**:

$$f(x) = p * Gamma(x|\alpha, \sigma_1) + (1-p)N(x|\mu, \sigma_2^2), \quad 0 < p < 1$$

For a gamma distribution $Gamma(x|\alpha, \sigma_1)$ $E(X) = \frac{\alpha}{\sigma}$ $V(X) = \frac{\alpha}{\sigma^2}$

Equating $E(X)$ to 52 ans $V(X)$ equal to 36 we get the values of α and σ to be 75 and 0.7 respectively. For the normal distribution we keep μ as 80 and σ as 6.

```
NegLogLikeMix1 <- function(theta,data){
  alpha= theta[1]
  sigma1 = theta[2]
  mu = theta[3]
  sigma2 = exp(theta[4])
  p = exp(theta[5])/(1+exp(theta[5]))
  n = length(data)
  l=0
  for(i in 1:n){
    l = l + log(p*dgamma(data[i],shape=alpha,scale=sigma1)
                +(1-p)*dnorm(data[i],mean=mu,sd=sigma2))
  }
  return(-l)
}
theta_initial1=c(75,0.7,80,6,0.35)
NegLogLikeMix1(theta_initial1,waiting)
```

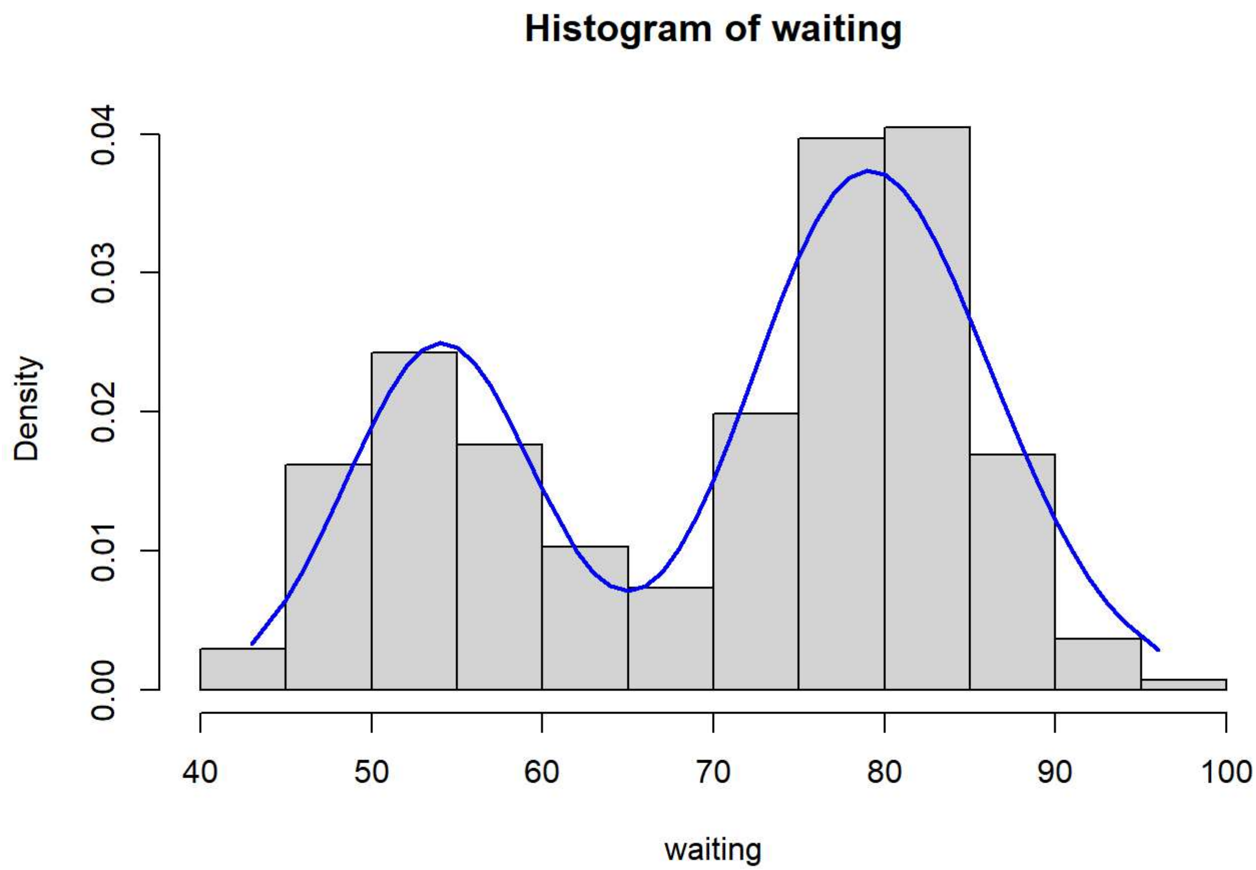
```
## [1] 1690.368
```

```
fit = optim(theta_initial1
            ,NegLogLikeMix1
            ,data=waiting
            ,control = list(maxit=1000))

theta_hat = fit$par
alpha_hat= theta_hat[1]
sigma1_hat = theta_hat[2]
mu_hat = theta_hat[3]
sigma2_hat = exp(theta_hat[4])
p_hat = exp(theta_hat[5])/(1+exp(theta_hat[5]))


theta_predict=c(alpha_hat,sigma1_hat,mu_hat,sigma2_hat,p_hat)
```

```
theta_predict
```

```
## [1] 129.6289769   0.6151857  54.0766822   5.5218204   0.6540931
```

```
d_mle1= p_hat*dgamma(waiting,shape=alpha_hat,scale=sigma1_hat)+(1-p_hat)*dnorm(waiting,mean=mu_h
at,sd=sigma2_hat)
hist(waiting,probability =TRUE)
lines(waiting,d_mle1,lwd=2,col='blue')
```



ii. **Model 2**:

$$f(x) = p * Gamma(x|\alpha_1, \sigma_1) + (1-p)Gamma(x|\alpha_2, \sigma_2), \ \ 0 < p < 1$$

For a gamma distribution $Gamma(x|\alpha, \sigma_1)$ $E(X) = \frac{\alpha}{\sigma}$ $V(X) = \frac{\alpha}{\sigma^2}$

Equating $E(X)$ to 52 ans $V(X)$ equal to 36 we get the values of α_1 and σ_1 to be 75 and 0.7 respectively. For the second distribution calculated α_2 and σ_2 to be 88 and 0.9 respectively

```
NegLogLikeMix2 <- function(theta,data){
  alpha_1= theta[1]
  sigma_1 =theta[2]
  alpha_2 =theta[3]
  sigma_2 =theta[4]
  p = theta[5]
  n = length(data)
  l=0
  for(i in 1:n){
    l = l + log(p*dgamma(data[i],shape=alpha_1,scale=sigma_1)
                +(1-p)*dgamma(data[i],shape=alpha_2,scale=sigma_2))
  }
  return(-l)
}

theta_initial2=c(75,0.7,87,0.9,0.5)
NegLogLikeMix1(theta_initial1,waiting)
```

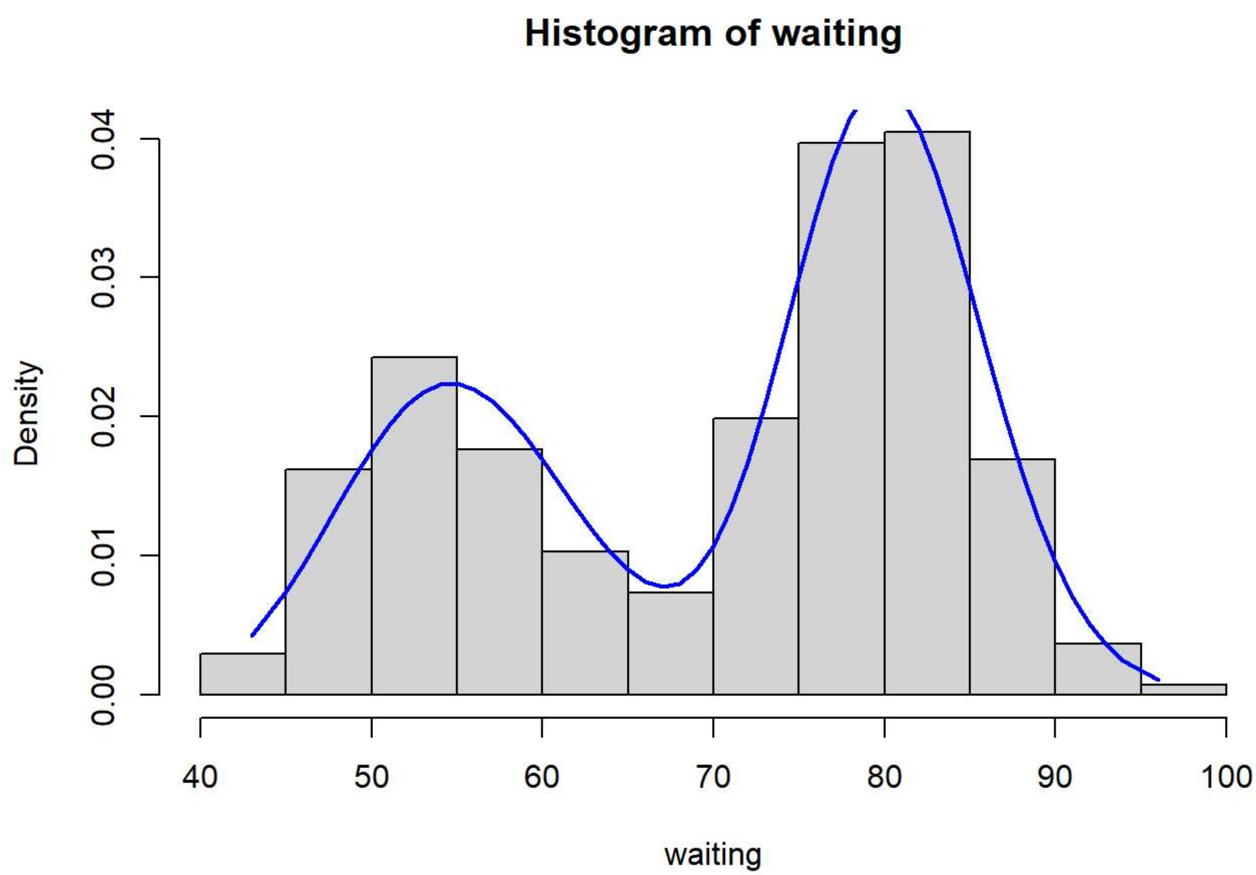
```
## [1] 1690.368
```

```
fit1 = optim(theta_initial2
            ,NegLogLikeMix2
            ,data=waiting,
            control = list(maxit=1500))
theta_hat = fit1$par
alpha1_hat= theta_hat[1]
sigma1_hat = theta_hat[2]
alpha2_hat = theta_hat[3]
sigma2_hat = theta_hat[4]
p_hat = theta_hat[5]

theta_predict1=c(alpha1_hat,sigma1_hat,alpha2_hat,sigma2_hat,p_hat)
```

```
theta_predict1
```

```
## [1]  63.6029510   0.8729447 205.2676178   0.3910166   0.3887341
```

```
d_mle2= p_hat*dgamma(waiting,shape=alpha1_hat,scale=sigma1_hat)+(1-p_hat)*dgamma(waiting,shape=a
lpha2_hat,scale=sigma2_hat)
hist(waiting,probability =TRUE)
lines(waiting,d_mle2,lwd=2,col='blue')
```



iii. **Model 3**:

For lognormal distribution $E(X) = \exp(\mu + \sigma^2/2)$ and $V(X) = \exp(2(\mu + \sigma^2) - \exp(2\mu + \sigma^2)$

For the first part calculated μ_1 and σ_1 is 3.9 and 0.015 respectively For the second part calculated μ_2 and σ_2 is 4.4 and 0.012 respectively

$$f(x) = p * logNormal(x|\mu_1, \sigma_1^2) + (1-p)logNormal(x|\mu_1, \sigma_1^2), \;\; 0 < p < 1$$

```
NegLogLikeMix3=function(data,theta)
{
  mu1=(theta[1])
  sigma1=exp(theta[2])
  mu2=(theta[3])
  sigma2=exp(theta[4])
  p=exp(theta[5])/(1+exp(theta[5]))
  n=length(data)
  l=0
  for(i in 1:n){
    l=l+log(p*dlnorm(data[i],mu1,sigma1)+(1-p)*dlnorm(data[i],mu2,sigma2))
  }
  return(-l)
}

theta_initial3=c(3.9,0.015,4.4,0.012,0.5)
NegLogLikeMix3(waiting,theta_initial2)
```

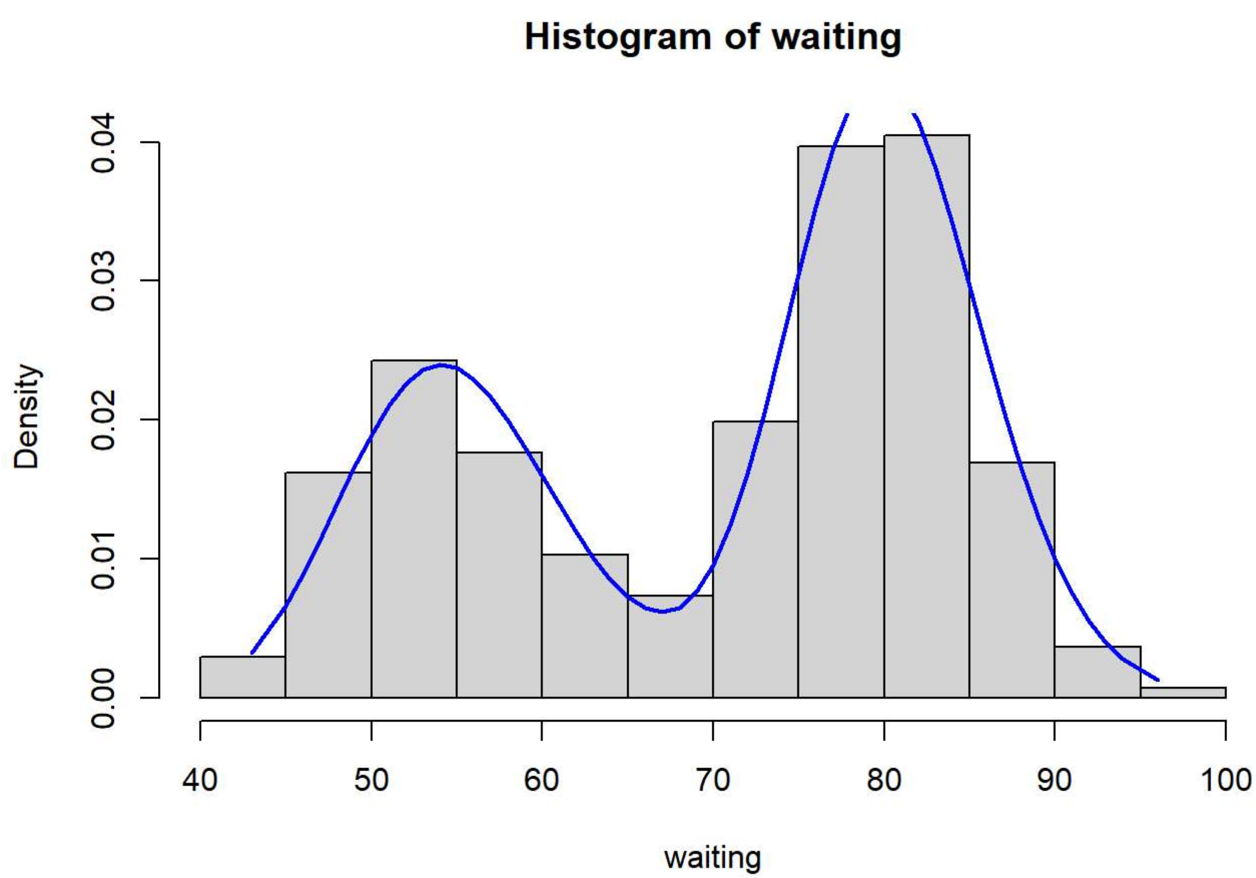
```
## [1] 155884.7
```

```
fit3=optim(theta_initial3,NegLogLikeMix3,control=list(maxit=10000),data =waiting)

theta_hat=fit3$par
mu1_hat=theta_hat[1]
sigma1_hat=exp(theta_hat[2])
mu2_hat=theta_hat[3]
sigma2_hat=exp(theta_hat[4])

p_hat=exp(theta_hat[5])/(1+exp(theta_hat[5]))

theta_predict2=c(mu1_hat,sigma1_hat,mu2_hat,sigma2_hat,p_hat)
```

```
theta_predict2
```

```
## [1] 4.00403469 0.11488819 4.38429573 0.06971088 0.37616536
```

```
d_mle3=p_hat*dlnorm(waiting,mu1_hat,sigma1_hat)+(1-p_hat)*dlnorm(waiting,mu2_hat,sigma2_hat)
hist(waiting,probability =TRUE)
lines(waiting,d_mle3,lwd=2,col='blue')
```



```
data=faithful
wt_dens=density(data$waiting,n=272)
wt_dens$y
```

```
##   [1] 8.951193e-06 1.136027e-05 1.434905e-05 1.799378e-05 2.254093e-05
##   [6] 2.810667e-05 3.480778e-05 4.304564e-05 5.299384e-05 6.481969e-05
##  [11] 7.914542e-05 9.621399e-05 1.162475e-04 1.401631e-04 1.682765e-04
##  [16] 2.008565e-04 2.391863e-04 2.836415e-04 3.345100e-04 3.934919e-04
##  [21] 4.609857e-04 5.372445e-04 6.243958e-04 7.227947e-04 8.325780e-04
##  [26] 9.562509e-04 1.094031e-03 1.245835e-03 1.414425e-03 1.599767e-03
##  [31] 1.801454e-03 2.022311e-03 2.261947e-03 2.519537e-03 2.797735e-03
##  [36] 3.095712e-03 3.412188e-03 3.749393e-03 4.106041e-03 4.480419e-03
##  [41] 4.874125e-03 5.285462e-03 5.712391e-03 6.155734e-03 6.613502e-03
##  [46] 7.083495e-03 7.565711e-03 8.058033e-03 8.558295e-03 9.065709e-03
##  [51] 9.578200e-03 1.009382e-02 1.061109e-02 1.112811e-02 1.164331e-02
##  [56] 1.215460e-02 1.266037e-02 1.315947e-02 1.364934e-02 1.412865e-02
##  [61] 1.459672e-02 1.505057e-02 1.548918e-02 1.591232e-02 1.631667e-02
##  [66] 1.670150e-02 1.706706e-02 1.740978e-02 1.772923e-02 1.802618e-02
##  [71] 1.829695e-02 1.854150e-02 1.876113e-02 1.895228e-02 1.911535e-02
##  [76] 1.925219e-02 1.935961e-02 1.943848e-02 1.949120e-02 1.951511e-02
##  [81] 1.951159e-02 1.948342e-02 1.942866e-02 1.934906e-02 1.924761e-02
##  [86] 1.912310e-02 1.897752e-02 1.881382e-02 1.863146e-02 1.843245e-02
##  [91] 1.821949e-02 1.799257e-02 1.775356e-02 1.750476e-02 1.724652e-02
##  [96] 1.698045e-02 1.670838e-02 1.643092e-02 1.614940e-02 1.586519e-02
## [101] 1.557911e-02 1.529226e-02 1.500564e-02 1.472028e-02 1.443719e-02
## [106] 1.415709e-02 1.388132e-02 1.361091e-02 1.334637e-02 1.308949e-02
## [111] 1.284141e-02 1.260250e-02 1.237511e-02 1.216052e-02 1.195898e-02
## [116] 1.177344e-02 1.160537e-02 1.145478e-02 1.132524e-02 1.121837e-02
## [121] 1.113380e-02 1.107566e-02 1.104556e-02 1.104265e-02 1.107142e-02
## [126] 1.113337e-02 1.122696e-02 1.135687e-02 1.152426e-02 1.172684e-02
## [131] 1.196917e-02 1.225192e-02 1.257200e-02 1.293353e-02 1.333657e-02
## [136] 1.377726e-02 1.425896e-02 1.478100e-02 1.533887e-02 1.593489e-02
## [141] 1.656757e-02 1.723194e-02 1.792904e-02 1.865654e-02 1.940925e-02
## [146] 2.018676e-02 2.098592e-02 2.180166e-02 2.263200e-02 2.347312e-02
## [151] 2.432039e-02 2.517035e-02 2.601863e-02 2.686147e-02 2.769404e-02
## [156] 2.851171e-02 2.931193e-02 3.008878e-02 3.083762e-02 3.155739e-02
## [161] 3.224143e-02 3.288536e-02 3.348975e-02 3.404761e-02 3.455503e-02
## [166] 3.501426e-02 3.541831e-02 3.576393e-02 3.605498e-02 3.628483e-02
## [171] 3.645102e-02 3.655885e-02 3.660238e-02 3.658007e-02 3.649842e-02
## [176] 3.635250e-02 3.614178e-02 3.587371e-02 3.554461e-02 3.515506e-02
## [181] 3.471307e-02 3.421648e-02 3.366693e-02 3.307258e-02 3.243281e-02
## [186] 3.175025e-02 3.103271e-02 3.028100e-02 2.949848e-02 2.869208e-02
## [191] 2.786379e-02 2.701732e-02 2.615827e-02 2.528939e-02 2.441438e-02
## [196] 2.353718e-02 2.266087e-02 2.178878e-02 2.092311e-02 2.006690e-02
## [201] 1.922289e-02 1.839167e-02 1.757597e-02 1.677788e-02 1.599669e-02
## [206] 1.523474e-02 1.449352e-02 1.377141e-02 1.307036e-02 1.239146e-02
## [211] 1.173250e-02 1.109517e-02 1.048029e-02 9.885405e-03 9.311987e-03
## [216] 8.760752e-03 8.229153e-03 7.718540e-03 7.229577e-03 6.759712e-03
## [221] 6.310199e-03 5.881661e-03 5.471567e-03 5.081057e-03 4.710696e-03
## [226] 4.357973e-03 4.023886e-03 3.708908e-03 3.410555e-03 3.129658e-03
## [231] 2.866570e-03 2.618862e-03 2.387178e-03 2.171747e-03 1.970230e-03
## [236] 1.783092e-03 1.610435e-03 1.450064e-03 1.302275e-03 1.167064e-03
## [241] 1.042421e-03 9.284984e-04 8.252088e-04 7.307619e-04 6.451961e-04
## [246] 5.683647e-04 4.987190e-04 4.362146e-04 3.806701e-04 3.307859e-04
## [251] 2.864650e-04 2.475129e-04 2.128743e-04 1.824257e-04 1.559783e-04
## [256] 1.327042e-04 1.124747e-04 9.512018e-05 8.001492e-05 6.703977e-05
```

```
## [261] 5.605243e-05 4.659827e-05 3.857658e-05 3.187520e-05 2.617719e-05
## [266] 2.140368e-05 1.747136e-05 1.416864e-05 1.143778e-05 9.220401e-06
## [271] 7.381377e-06 5.881023e-06
```

```
library(tidyverse)
```

```
## ── Attaching packages ──────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
data=data %>% arrange(waiting)

data$density=wt_dens$y
data$d1=d_mle1
data$d2=d_mle2
data$d3=d_mle3

m1 <- lm(eruptions ~ d1, data = data)
m2 <- lm(eruptions ~ d2, data = data)
m3 <- lm(eruptions ~ d3, data = data)
models=list(m1,m2,m3)

mod.names <- c('gamma,normal', 'gamma,gamma', 'lognormal+lognormal')
```

```
library(AICcmodavg)
```

```
## Warning: package 'AICcmodavg' was built under R version 4.2.2
```

```
#calculate AIC of each model
aictab(models,mod.names)
```

```
## 
## Model selection based on AICc:
## 
##                     K   AICc Delta_AICc AICcWt Cum.Wt      LL
## gamma,normal        3 757.57       0.00   0.56   0.56 -375.74
## gamma,gamma         3 758.23       0.66   0.41   0.97 -376.07
## lognormal+lognormal 3 763.48       5.90   0.03   1.00 -378.69
```

The model with with the best fit is the third model.

```
dmix=function(x,theta){
  mu1=theta[1]
  sigma1=theta[2]
  mu2=theta[3]
  sigma2=theta[4]
  p=theta[5]
  ans=p*dlnorm(x,mu1,sigma1)+(1-p)*dlnorm(x,mu2,sigma2)
  return(ans)
}
### parameters of ln + ln model :


#P(60 < waiting < 70)
P=integrate(dmix,60,70,c(theta_predict2))  ##p(70<waiting<100)
P
```

```
## 0.09097159 with absolute error < 1e-15
```