question no. 3

AKASH

2022-11-15

Problem 3: Analysis of faithful datasets.

First we have to attach the “faithful” dataset and sort the waiting time in ascending order for model fitting.

```
attach(faithful)
data=faithful
waiting=sort(waiting)
```

MODEL 1>

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

First we calculate the negative log likelihood function for the given model and try to minimize the function . By minimizing the negative log likelihood basically we maximize the original likelihood function.

```
NegLogLikeMix=function(theta,data){
  shape=(theta[1])
  scale=(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,scale)+(1-p)*dnorm(data[i],mu,sigma2))
  }
  return(-l)
}
```

Now we have to find the initial value of parameters and optimize the function with respect to that for close estimation. from the graph we can say that for the left distribution mean is around 52 and s.d is around 6, so we can take gamma distribution with shape parameter=74 and scale parameter=0.7 similarly for the right distribution mean is around 80 and s.d is around 6, so the normal distribution with meu=80 and sigma=6 . Let us take initial value of p as 0.5

now we use optim function(since we have multiple parameters to estimate) to get the optimised parameters.

```
theta_initial=c(74,.7,80,6,0.5)

fit=optim(theta_initial,NegLogLikeMix,control=list(maxit=10000),data =data$waiting)
fit
```

```
## $par
## [1] 53.3846504  0.9604804 80.3986178  1.7395942 -0.4703063
##
## $value
## [1] 1035.548
##
## $counts
## function gradient
##      585       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Now the estimated parameters are as follows

```
theta_hat1=fit$par
shape_hat=(theta_hat1[1])
scale_hat=(theta_hat1[2])
mu2_hat=theta_hat1[3]
sigma2_hat=exp(theta_hat1[4])
p_hat_1=exp(theta_hat1[5])/(1+exp(theta_hat1[5]))

shape_hat
```

```
## [1] 53.38465
```

```
scale_hat
```

```
## [1] 0.9604804
```

```
mu2_hat
```

```
## [1] 80.39862
```

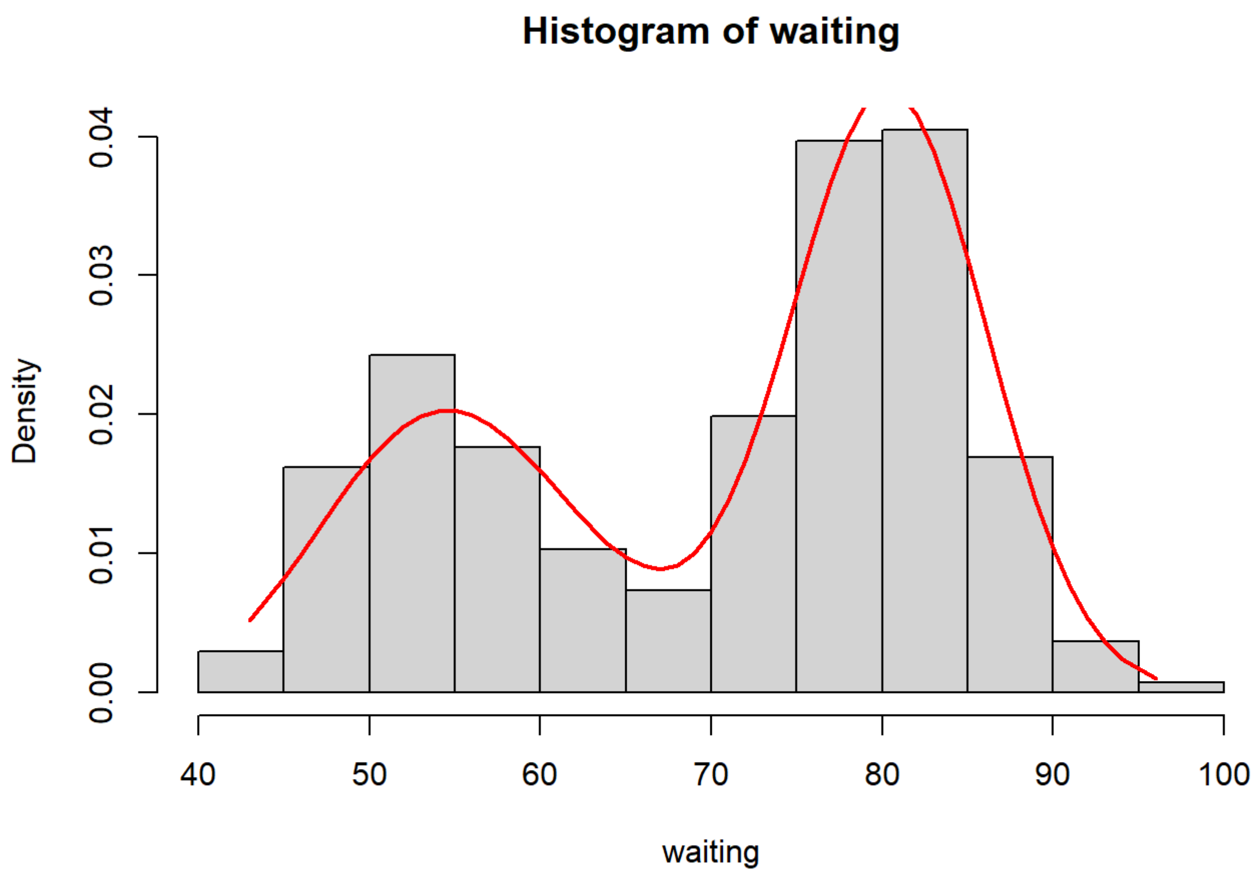
```
sigma2_hat
```

```
## [1] 5.695032
```

```
p_hat_1
```

```
## [1] 0.3845437
```

Let us fit a curve based on estimated parameters on the given distribution to visualize our fitted model 1.



MODEL 2>

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

First we calculate the negative log likelihood function for the given model and try to minimize the function . By minimizing the negative log likelihood basically we maximize the original likelihood function.

```
LogLikeMix2=function(theta,data){
  shape1=(theta[1])
  scale1=(theta[2])
  shape2=(theta[3])
  scale2=(theta[4])
  p=(theta[5])
  n=length(data)
  l=0
  for(i in 1:n){
    l=l+log(p*dgamma(data[i],shape1,scale1)+(1-p)*dgamma(data[i],shape2,scale2))
  }
  return(-l)
}
```

Now we have to find the initial value of parameters and optimize the function with respect to that for close estimation. from the graph we can say that for the left distribution mean is around 52 and s.d is around 6, so we can take gamma distribution with shape parameter=74 and scale parameter=0.7 similarly for the right distribution mean is around 80 and s.d is around 6, so the gamma distribution with shape parameter=88 and scale parameter=0.9 . Let us take initial value of p as 0.5

now we use optim function(since we have multiple parameters to estimate) to get the optimised parameters.

```
theta_initial1=c(74,.7,88,.9,0.5)
LogLikeMix2(theta_initial1,waiting)
```

```
## [1] 2482.297
```

```
fit1=optim(theta_initial1,LogLikeMix2,control=list(maxit=10000),data =waiting)
fit1
```

```
## $par
## [1] 199.8642599   2.4891755  79.6392069   1.4485848   0.6293747
##
## $value
## [1] 1033.058
##
## $counts
## function gradient
##     1136       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Now the estimated parameters are as follows

```
theta_hat2=fit1$par
shape_hat1=theta_hat2[1]
scale_hat1=(theta_hat2[2])
shape_hat2=theta_hat2[3]
scale_hat2=(theta_hat2[4])
p_hat_2=theta_hat2[5]

shape_hat1
```

```
## [1] 199.8643
```

```
scale_hat1
```

```
## [1] 2.489175
```

```
shape_hat2
```

```
## [1] 79.63921
```

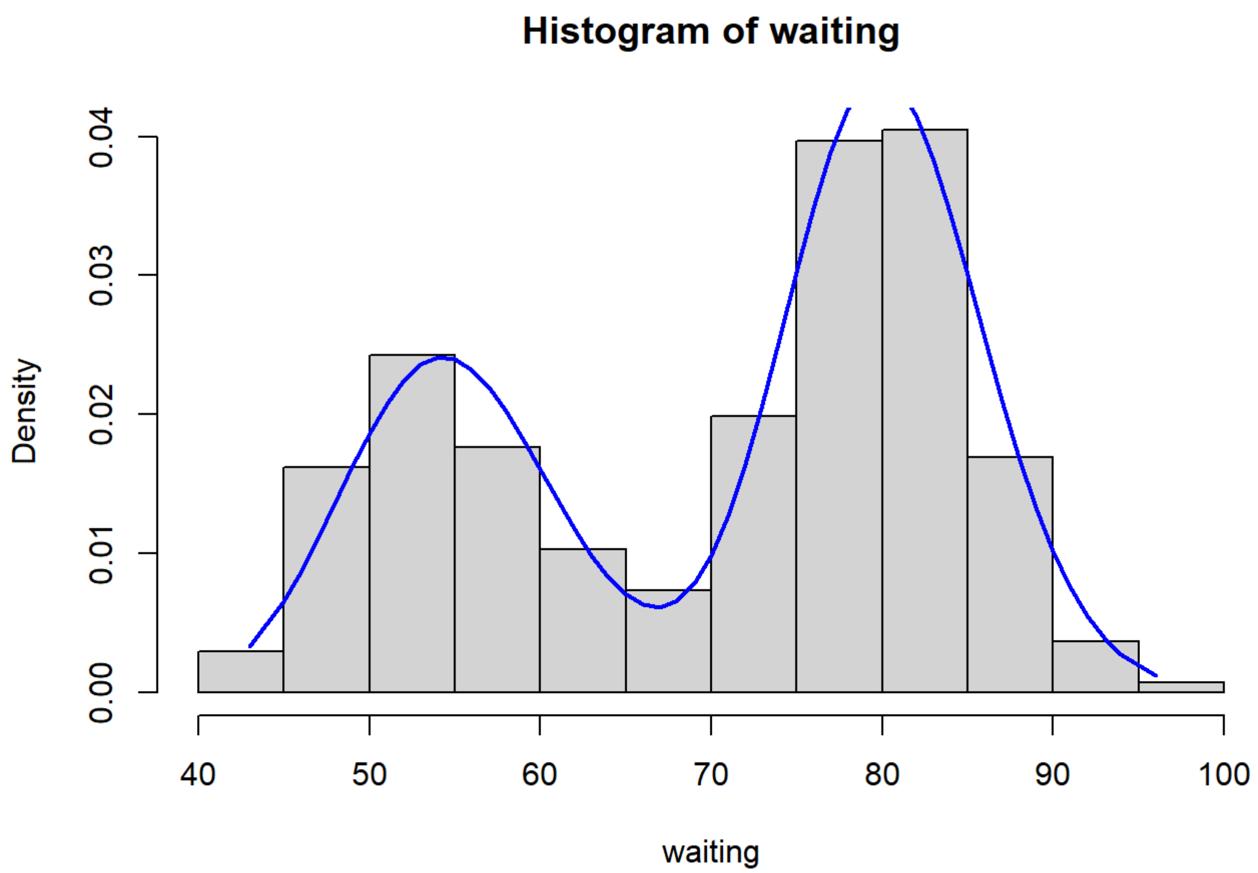
```
scale_hat2
```

```
## [1] 1.448585
```

```
p_hat_2
```

```
## [1] 0.6293747
```

Let us fit a curve based on estimated parameters on the given distribution to visualize our fitted model 2.



MODEL 3>

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

First we calculate the negative log likelihood function for the given model and try to minimize the function . By minimizing the negative log likelihood basically we maximize the original likelihood function.

```
ll3=function(data,theta)
{
  meu1=(theta[1])
  sig1=exp(theta[2])
  meu2=(theta[3])
  sig2=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],meu1,sig1)+(1-p)*dlnorm(data[i],meu2,sig2))
  }
  return(-l)
}
```

Now we have to find the initial value of parameters and optimize the function with respect to that for close estimation. from the graph we can say that for the left distribution mean is around 52 and s.d is around 6, so we can take Log normal distribution with meu1 =3.9 and sigma1=0.015 . similarly for the right distribution mean is around 80 and s.d is around 6, so the Log normal distribution with meu1 =4.4 and sigma1=0.012. Let us take initial value of p as 0.5

now we use optim function(since we have multiple parameters to estimate) to get the optimised parameters.

```
theta_initial=c(3.9,0.015,4.4,0.012,0.5)
ll3(waiting,theta_initial)
```

```
## [1] 1423.317
```

```
fit3=optim(theta_initial,ll3,control=list(maxit=10000),data =waiting)
fit3
```

```
## $par
## [1]  4.0040347 -2.1637959  4.3842957 -2.6633988 -0.5058565
##
## $value
## [1] 1032.71
##
## $counts
## function gradient
##      982       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Now the estimated parameters are as follows

```
theta_hat3=fit3$par
mu1_hat1=theta_hat3[1]
sig_hat1=exp(theta_hat3[2])
mu2_hat2=theta_hat3[3]
sig_hat2=exp(theta_hat3[4])
p_hat_3=exp(theta_hat3[5])/(1+exp(theta_hat3[5]))

mu1_hat1
```

```
## [1] 4.004035
```

```
sig_hat1
```

```
## [1] 0.1148882
```

```
mu2_hat2
```

```
## [1] 4.384296
```

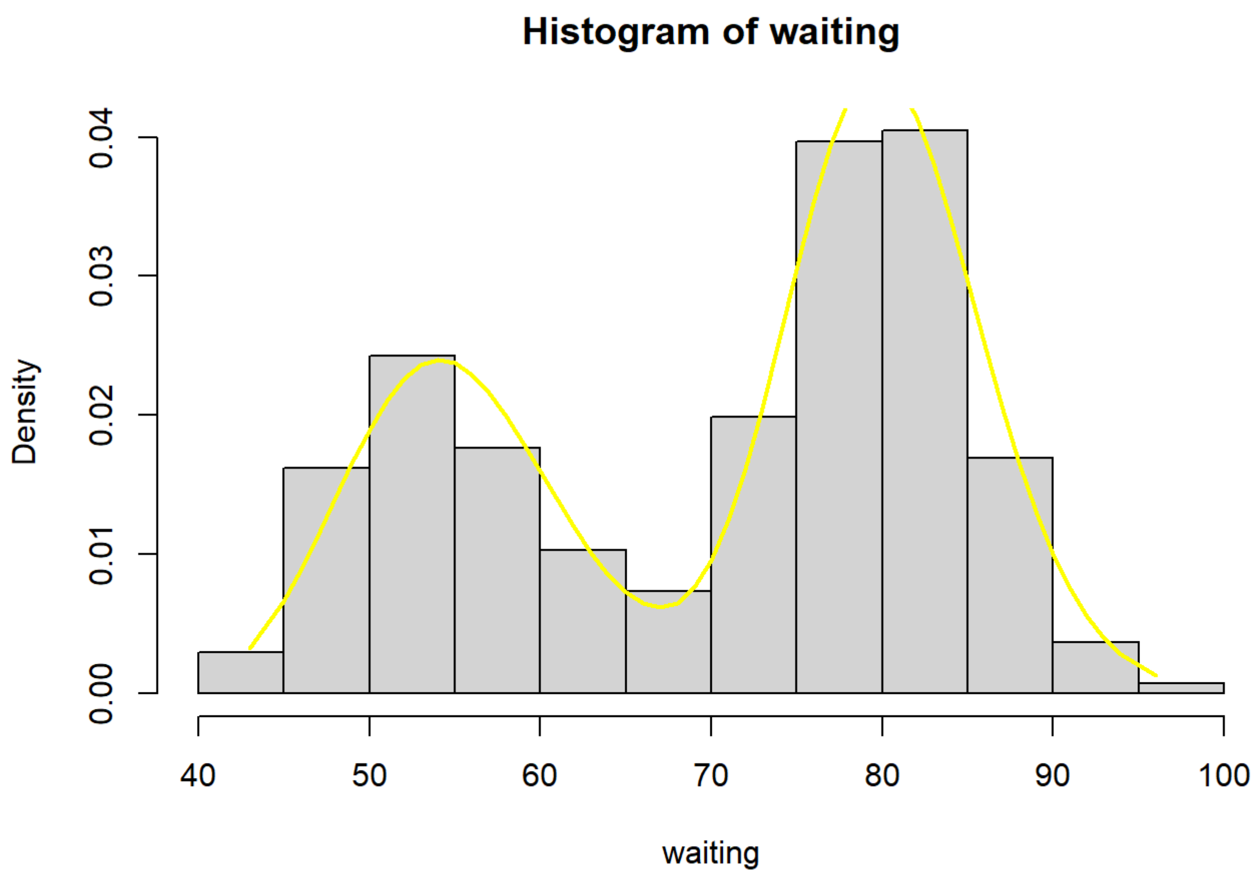
```
sig_hat2
```

```
## [1] 0.06971088
```

```
p_hat_3
```

```
## [1] 0.3761654
```

Let us fit a curve based on estimated parameters on the given distribution to visualize our fitted model 3.



AIC calculations for the above 3 models.

aic= (2 * number_of_parameter)- (2 * loglikelihhod_function(theta_hat,data))

since here no. of parameters =5 for every model,

aic=(2 * 5)- (2 * loglikelihhod_function(theta_hat,data))

```
l1=NegLogLikeMix(theta_hat1,waiting)*-1
l2=LogLikeMix2(theta_hat2,waiting)*-1
l3=ll3(waiting,theta_hat3)*-1
aic1=2*5-2*l1
aic2=2*5-2*l2
aic3=2*5-2*l3
aic_table= data.frame (
  Models = c("gamma+normal", "gamma+gamma", "lognormal+lognormal"),
  AIC = c(aic1,aic2,aic3))

aic_table
```

```
##                Models      AIC
## 1        gamma+normal 2081.096
## 2         gamma+gamma 2076.117
## 3 lognormal+lognormal 2075.420
```

Since AIC value of model 3 is lowest , hence its the best fitted model.

Now ased on the best model calculate thefollowing probability

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

so the required probability is :

```
dmix=function(x,theta){
  mu1=theta[1]
  sigma1=theta[2]
  mu2=theta[3]
  sigma2=theta[4]
  p=theta[5]
  f=p*dlnorm(x,mu1,sigma1)+(1-p)*dlnorm(x,mu2,sigma2)
  return(f)
}
P=integrate(dmix,60,70,c(mu1_hat1,sig_hat1,mu2_hat2,sig_hat2,p_hat_3))
P
```

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

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.