Question :
Mixture models form one of the most fundamental classes of generative models for clustered data. This will have a multimodal distribution.
Using mixture normal distribution (a bimodal distribution)

$$0.7\times N(\mu_1,\sigma_1^2)+0.3\times N(\mu_2,\sigma_2^2)$$

(a) Find the maximum likelihood estimates of the unknown parameters and their standard errors. Use any appropriate R package or otherwise.
(b) Draw a histogram of the data and superimpose the density of the above mixture normal distribution using maximum likelihood estimates of the unknown parameters.

Solution(a)：

Step 1: Input given data from the question named “data”

```
cpu <- scan(file="F:\\6313 statistic for DS\\project\\2\\cpu.txt", what="numeric")
```

After this execution, data of cpu.txt is uploaded in to R, but the data type is “character”. So I need to

```
cpu<-as.numeric(cpu)
```

Step 2: Using the given function from the question named “mnd”

Based on the function : $0.7\times N(\mu_1,\sigma_1^2)+0.3\times N(\mu_2,\sigma_2^2)$, I defined a function named “mnd” in R. From the function we know that there are 5 parameters.

```
mnd <- function(N,u1,d1,u2,d2){
result <- 0.7*dnorm(N, mean=u1, sd=d1,log=FALSE)+0.3*dnorm(N,mean=u2, sd=d2,
log=FALSE)
return(result)
}
```

Step 3:

#neg.loglik.fun is Negative of log-likelihood function assuming mixture normal distribution.

```
neg.loglik.fun <- function(par,N)
{
   result <- sum(log(mnf(N = N, u1=par[1], d1= d1<-par[2], u2 =par[3], d2=par[4]),10))
   return(-result)
}
```

Step 4:

Minimize -log (L), i.e., maximize log (L)

```
ml.est <- optim(par=c(100,2,30,2), fn=neg.loglik.fun, method = "L-BFGS-B",
lower=rep(0,2),hessian=TRUE, N=cpu)
```

Step 5: Show MLE answer

```
ml.est$par
```

Answer:

```
[1] 100.573570   2.736788  29.418334   2.479898
```

Step 6:

#find standard errors

```
sqrt(diag(solve(ml.est$hessian)))
```

The answer is :

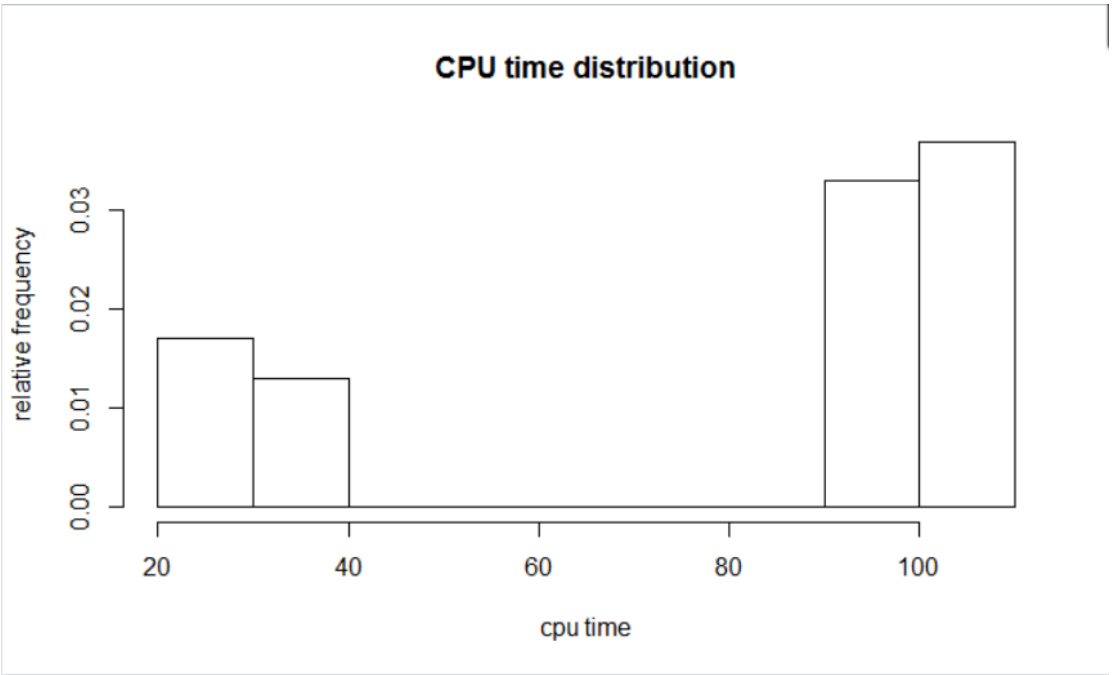
```
[1] 0.4963637 0.3509821 0.6870384 0.4858091
```

Solution(b)

Step 1:Build a basic histogram which is about cpu time distribution.

```
hist(cpu, freq=FALSE, xlab="cpu time", ylab="relative frequency", main="CPU time
distribution")
```

From the graph I can get two information:
(1) There is no data value between 40 to 90. So when I draw a histogram of density, it must be no distribution between 40 to 90.
(2) The relative frequency between 90 to 110 is higher than 20 to 40. So density between 90 to 110 is higher than the other.



Step 3: Define a function, when the function receive the number of cpu time(x-coordinate value), it will return mnf(N,u1,d1,u2,d2) which we defined before as y-coordinate value so that there will be a one-to-one relation and get a curve.

```
curvefun <-
function(N,u1=ml.est$par[1],d1=ml.est$par[2],u2=ml.est$par[3],d2=ml.est$par[4]){
   return(mnf(N,u1,d1,u2,d2))
}
```

Step 4: Making curves

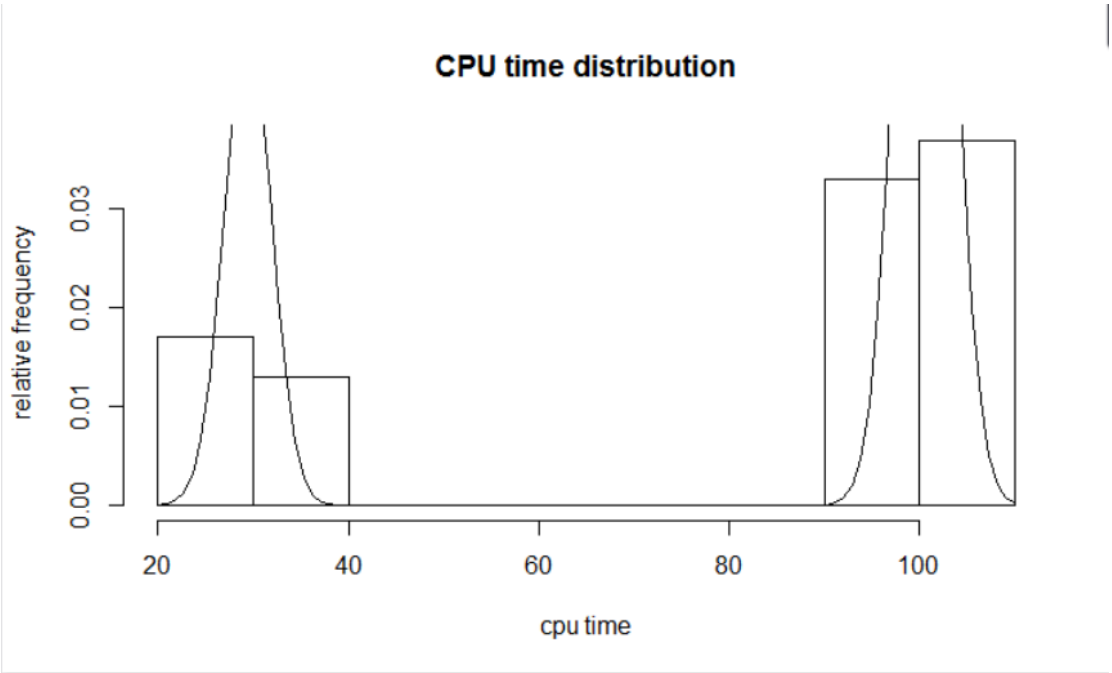
First, let x-coordinate value equals to value of cpu.txt by using the command below:

```
x<-cpu
```

And then, using the command below to make curves:

```
curve(curvefun(x),add = T)
```

Answer:



I can see that the result is same with I participate in Step 1.