時間序列 HW1

0853411 劉書維

1. Simulate X(t) :N(0,1) for t=1 to 100 and N(1,1) for t=101 to 200：

我是運用 R 語言來生成資料。程式碼如下：

```
x1 <- ts(rnorm( 100,mean= 0,sd= 1 ),1,100)
x2 <- ts(rnorm( 100,mean= 1,sd= 1 ),101,200)
```

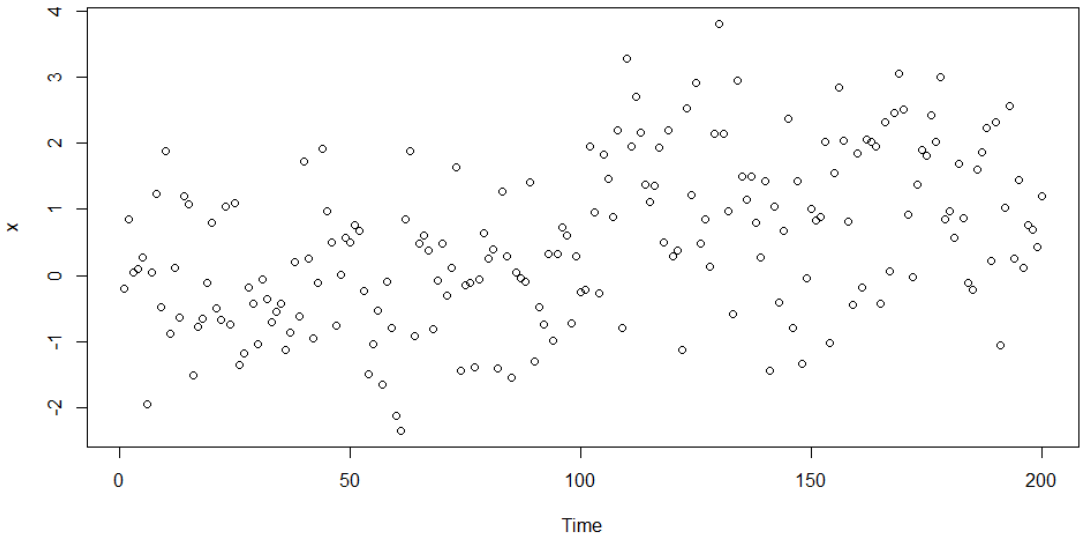
＃分別生成 x1 和 x2 不同 mean 的時間序列

```
x <- ts(c(x1, x2),start = start(x1),frequency = frequency(x1))
```

＃再將 x1, x2 合併成 x

```
plot(x,type="p")
```

#圖形如下：



2. #Fit X with a single normal distribution.

利用套件 MASS 完成分布分析，並且畫出分布圖。程式碼如下：

```
library(MASS)
```

＃引入套件

```
fit <- fitdistr(x, "normal")
```

＃將要 fit 的資料放入，並輸入是 normal distribution

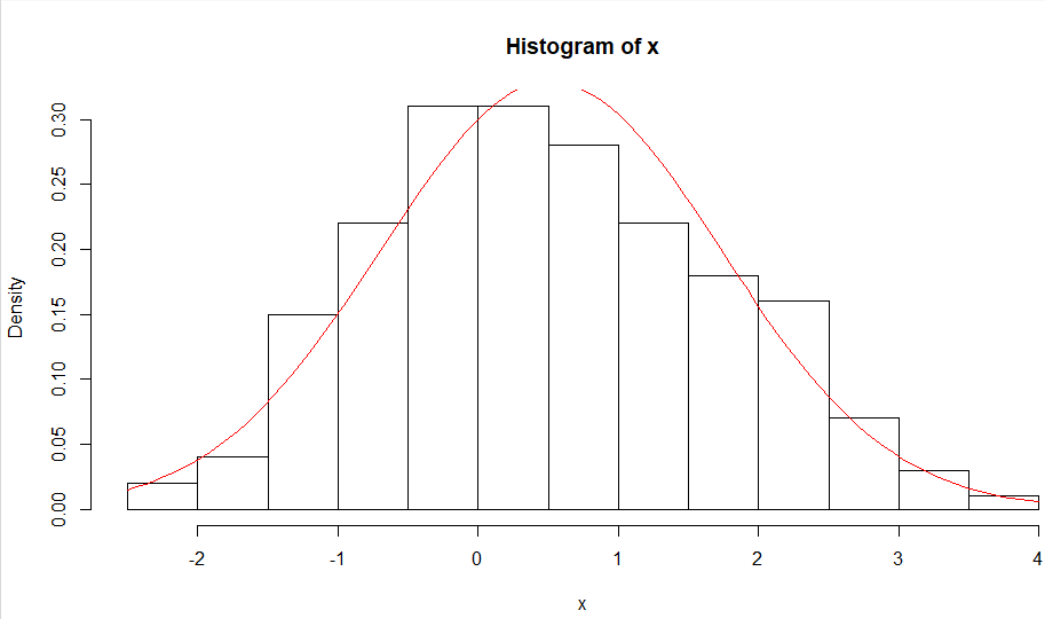
```
para <- fit$estimate
print(para)
```

#印出參數如下圖：

```
> print(para)
      mean         sd
0.5179336 1.2126351
```

```
hist(x, prob = TRUE)
curve(dnorm(x, para[1], para[2]), col = 2, add = TRUE)
```

＃分布曲線如下圖：



3. **Fix X with a mixture of normal when you know there are two mixture components：**

在已知是兩個組合成分下可以使用 mclust 套件，可以幫忙分析出資料內的 normal distribution。程式碼如下：

```
library(mclust, quietly=TRUE)
```

＃引入套件

```
mixmdl = Mclust(x, G=2, model="V")
```

＃輸入 x 並且輸入 components 的數量

```
summary(mixmdl)
```

＃結果如下：

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust V (univariate, unequal variance) model with 2 components:

 log-likelihood   n df       BIC       ICL
      -319.1456 200  5 -664.7827 -738.8949

Clustering table:
  1   2
106  94
```

4. **What would you do if you do not know the number of mixture components：**

在不知有多少 components 的情況下，可以使用 mixtools 套件協助，並且可以畫出 components 的分布曲線。程式碼如下：

```
library(mixtools)
```

＃引入套件

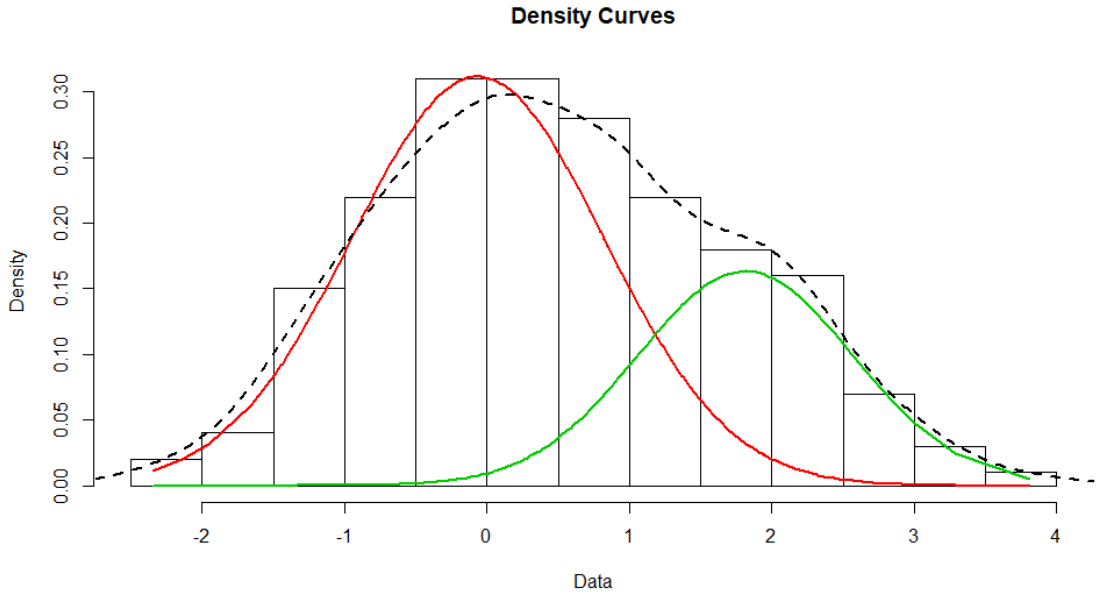
```
mixmd2 = normalmixEM(x)
summary(mixmd2)
```

＃結果如下：

```
summary of normalmixEM object:
           comp 1    comp 2
lambda  0.6900862 0.309914
mu     -0.0642459 1.814266
sigma   0.8831259 0.756967
loglik at estimate:  -318.9632
```

```
plot(mixmd2,which=2)
lines(density(x), lty=2, lwd=2)
```

＃兩個 components 分布曲線如下：



5. **suppose you know that the data actually comes from two different model, but you don’ t know the cutting point 100. How can you fit the model?**

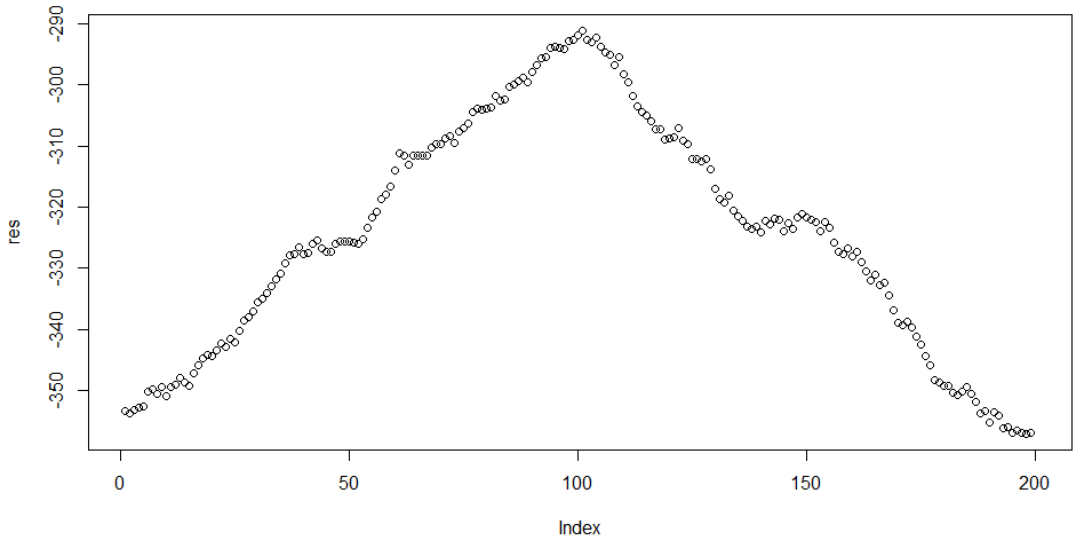
藉由寫一個 cutpoint function 來將 x 分為兩部分，並界定兩者的分布，來找到合適的 cut point。程式碼如下：

```
log_lik = function(cutpoint) {
   part1 = x[1:cutpoint]
   part2 = x[(cutpoint + 1):length(x)]
   sum(dnorm(part1, mean = 0, log = TRUE)) +
      sum(dnorm(part2, mean = 1, log = TRUE))
}
```

＃寫一個 function，並將 part1 和 part2 的分布界定好，來找出合適的切點。

```
res = sapply(1:length(x), log_lik)
plot(res)
```

＃輸出的 index 如下圖



which.max(res)

#建議切點（找上圖中的最高點）:

```
> which.max(res)
[1] 101
```

6. What if you don’t know how many cutting points are there?

可以使用套件 strucchange，來找到適當的切點個數與位置。程式碼如下：

```
library(strucchange)
# 引入套件
bp <- breakpoints(x ~ 1, breaks = NULL)
bp$breakpoints
```

使用 function：breakpoints 來找出適合的 cut points，結果如下圖。

```
> bp$breakpoints
[1] 101
```