1: Parallel Computing for EM Alogorithm (40%)

The EM algorithm in the question:

Given initial guess: $\pi_1^{(0)}, \pi_2^{(0)}, \mu_1^{(0)}, \mu_2^{(0)}, \mu_3^{(0)}, \sigma_1^{(0)}, \sigma_2^{(0)}, \sigma_3^{(0)}$, for $t \geq 0$ and $t \in \mathbb{Z}$:

E − step: Calculate $E(Z_i^{(t)}|\Theta^{(t)})$, where $\Theta^{(t)} = \pi_1^{(t)}, \pi_2^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \mu_3^{(t)}, \sigma_1^{2(t)}, \sigma_2^{2(t)}, \sigma_3^{2(t)}$.

$$\begin{aligned}\widehat{Z_{ik}}^{(t)} &= E(Z_i = k|\Theta^{(t)}) = E(Z_i^{(t)}|\pi_1^{(t)}, \pi_2^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \mu_3^{(t)}, \sigma_1^{2(t)}, \sigma_2^{2(t)}, \sigma_3^{2(t)}) \\ &= \frac{\pi_k^{(t)} \frac{1}{\sqrt{2\pi}\sigma_k{}^{(t)}} e^{-\frac{(y_i-\mu_k^{(t)})^2}{2\sigma_k^{2(t)}}}}{\pi_1^{(t)} \frac{1}{\sqrt{2\pi}\sigma_1{}^{(t)}} e^{-\frac{(y_i-\mu_1^{(t)})^2}{2\sigma_1^{2(t)}}} + \pi_2^{(t)} \frac{1}{\sqrt{2\pi}\sigma_2{}^{(t)}} e^{-\frac{(y_i-\mu_2^{(t)})^2}{2\sigma_2^{2(t)}}} + (1-\pi_1^{(t)}-\pi_2^{(t)}) \frac{1}{\sqrt{2\pi}\sigma_3{}^{(t)}} e^{-\frac{(y_i-\mu_3^{(t)})^2}{2\sigma_3^{2(t)}}}}\end{aligned}$$

M − step: Update $\Theta^{(t+1)}$ by equations (1) to (8).

Stopping criterion: $|L(\Theta^{(t)}|\mathbf{Y})) - L(\Theta^{(T+1)}|\mathbf{Y}))| <$ tolerance.

Iterative scheme:

$$\pi_1^{(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i1}}^{(t)}}{n} \tag{1}$$

$$\pi_2^{(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i2}}^{(t)}}{n} \tag{2}$$

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i1}}^{(t)} y_i}{\sum_{i=1}^n \widehat{Z_{i1}}^{(t)}} \tag{3}$$

$$\mu_2^{(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i2}}^{(t)} y_i}{\sum_{i=1}^n \widehat{Z_{i2}}^{(t)}} \tag{4}$$

$$\mu_3^{(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i3}}^{(t)} y_i}{\sum_{i=1}^n \widehat{Z_{i3}}^{(t)}} \tag{5}$$

$$\sigma_1^{2(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i1}}^{(t)} (y_i - \mu_1^{(t)})^2}{\sum_{i=1}^n \widehat{Z_{i1}}^{(t)}} \tag{6}$$

$$\sigma_2^{2(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i2}}^{(t)} (y_i - \mu_2^{(t)})^2}{\sum_{i=1}^n \widehat{Z_{i2}}^{(t)}} \tag{7}$$

$$\sigma_3^{2(t+1)} = \frac{\sum_{i=1}^n \widehat{Z_{i3}}^{(t)} (y_i - \mu_3^{(t)})^2}{\sum_{i=1}^n \widehat{Z_{i3}}^{(t)}} \tag{8}$$

In both E-step and M-step, the iterative schemes for each parameter and missing data are independent. Therefore, we can apply parallel computing in updating all the parameters and Z's.

The detailed process is as follows: given $\Theta^{(t)}$,

1. E-step: Compute conditional expectation values to be stored in an $n \times 3$ matrix, in which the computation tasks are distributed in rows in parallel.
2. M-step:
When computing the above matrix, to pre-compute some intermediate parameters such as $\widehat{Z_{i1}}^{(t)} y_i$ and $\widehat{Z_{i1}}^{(t)} y_i^2$, which are collected in other $n \times 3$ matrices.
The master gather all values computed before and then distribute tasks about updating 8 parameters in parallel.
The master gather the updated parameters $\Theta^{(t+1)}$ and then go to the next loop.

```
> # original version
> system.time(maximization(pi1_0, pi2_0, mu1_0, mu2_0, mu3_0, sigma1_
   user  system elapsed
  0.538   0.015   0.574
>
> # parallel version
> num_core = detectCores()
> cl = makeCluster(num_core, type = "FORK")
> system.time(maximization_l(pi1_0, pi2_0, mu1_0, mu2_0, mu3_0, sigma
   user  system elapsed
  0.186   0.082   0.692
> stopCluster(cl)
```

Figure 1: Original VS Parallel Computing Time

From the computation, we ca know that parallel computing is much faster than the original version.

2: Database Access from R (30%)

SQL in the pictures following highlighted in blue in the double quotes.

(a) The 'Book' Table:

```
> dbGetQuery(con,"SELECT * FROM Book;")
  BookNumber  Classification
1          1 Natural Science
2          2 Natural Science
3          3 Natural Science
4          4         History
5          5         History
6          6      Philosophy
7          7      Philosophy
8          8      Philosophy
9          9      Philosophy
```

Figure 2: 'Book' Tbale

(b)

```
> dbGetQuery(con, "SELECT Student.StudentID, EntryYear FROM Student, Record, Book
+     where Book.Classification = 'Natural Science'
+     And Record.BookNumber = Book.BookNumber
+     And Record.StudentID = Student.StudentID;")
  StudentID EntryYear
1         1      2018
2         3      2019
3         8      2018
```

Figure 3: Students who borrowed natural science books

(c)

```
> dbGetQuery(con, "SELECT Student.StudentID, Major FROM Student, Record
+            where Record.BookNumber = '8'
+            And Record.StudentID = Student.StudentID
+            And TIMESTAMPDIFF(day, BorrowingTime, ReturnTime)>30;")
  StudentID Major
1         6   Art
```

Figure 4: Students who borrowed book 8 for more than 30 days

3: Parse HTML (30%)

(a) The result is stored in variable 'comp' in R code.

(b)