STAT5703 HW2 Exercise 2

Wen Fan(wf2255), Banruo Xie(bx2168), Hanjun Li(hl3339)

Exercise 2.

Question 1.

```
data <- read.table('scores.txt', header = TRUE)
```

```
# Complete case analysis.
c1 <- cov(data,use="complete")
c1
```

```
##        x1     x2     x3     x4     x5
## x1 216.30  -7.50  45.05  77.65  94.50
## x2  -7.50 221.50 117.50  77.00 226.75
## x3  45.05 117.50 157.30  85.90 242.00
## x4  77.65  77.00  85.90  75.20 132.25
## x5  94.50 226.75 242.00 132.25 422.00
```

```
# Available case analysis.
c2 <- cov(data,use="pairwise")
c2
```

```
##            x1         x2        x3        x4       x5
## x1 121.363636   4.563636  35.79091  42.12727  94.5000
## x2   4.563636 179.134199 112.26840 114.60173 172.5000
## x3  35.790909 112.268398 151.48918 125.96537 182.3727
## x4  42.127273 114.601732 125.96537 153.56061 142.8636
## x5  94.500000 172.500000 182.37273 142.86364 294.5636
```

```
# Mean imputation
data_mean=data
for(i in 1:ncol(data_mean)) {
  data_mean[ , i][is.na(data_mean[ , i])] <- mean(data_mean[ , i], na.rm = TRUE)
}
c3 <- cov(data_mean)
c3
```

```
##          x1        x2        x3        x4        x5
## x1 57.79221   2.17316  17.04329  20.06061  21.50138
## x2  2.17316 179.13420 112.26840 114.60173  82.14286
## x3 17.04329 112.26840 151.48918 125.96537  86.84416
## x4 20.06061 114.60173 125.96537 153.56061  68.03030
## x5 21.50138  82.14286  86.84416  68.03030 140.26840
```

```
# Mean imputation with bootstrap
cov<-matrix(rep(0,25),ncol=5)
for(i in 1:400){
```

```r
  sam<-sample(nrow(data),22,replace=TRUE)
  temp <- sapply(data[sam,], function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
  cov <- cov + cov(temp)
}
c4 <- cov/400
c4
```

```
##           x1         x2        x3        x4        x5
## x1 51.469485   1.847223  15.92604  18.04012  19.52150
## x2  1.847223 172.425795 109.26311 110.63484  77.04685
## x3 15.926040 109.263111 148.76830 123.05337  83.48849
## x4 18.040120 110.634843 123.05337 148.02043  65.23093
## x5 19.521499  77.046851  83.48849  65.23093 133.72559
```

```r
# The EM-algorithm
library(Amelia)
Completed_data <- amelia(data,m=1,p2s=0)
c5 <- cov(Completed_data$imputations$imp1)
c5
```

```
##           x1         x2        x3        x4         x5
## x1 1169.7394 -266.40457 -195.5106 -120.3611 -141.57374
## x2 -266.4046  179.13420  112.2684  114.6017   93.49108
## x3 -195.5106  112.26840  151.4892  125.9654  119.80004
## x4 -120.3611  114.60173  125.9654  153.5606  110.75720
## x5 -141.5737   93.49108  119.8000  110.7572  182.58553
```

Mean imputation and Mean imputation with the bootstrap have smaller covariance than others. Only EM-algorithm has a negative covariance of x1 and x2.

Question 2.

By delta method, we can get $\sqrt{n}(\hat{\lambda}_1 - \lambda_1) \to N(0, 2\lambda_1^2)$, therefore asymptotic normality of $\hat{\lambda}_1$ is :$\hat{\lambda}_1 \to N(\lambda_1, \frac{2\lambda_1^2}{n})$, the confidence interval of λ_1 is:

$$[\frac{\hat{\lambda}_1}{1 + z_{1-\alpha/2}\sqrt{\frac{2}{n}}}, \frac{\hat{\lambda}_1}{1 - z_{1-\alpha/2}\sqrt{\frac{2}{n}}}]$$

Because λ_1 is the largest eigenvalue of the population covariance matrix, we can get $\hat{\lambda}_1$ from each method and the intervals of λ_1:

```r
get_interval <- function(lambda) {
  n=nrow(data)
  print(paste0('[',lambda/(1+sqrt(2/n)*qnorm(0.975)),', ',lambda/(1-sqrt(2/n)*qnorm(0.975)),']'))}
get_interval(max(eigen(c1)$value))
```

```
## [1] "[482.301219299174, 1875.8596024619]"
```

```r
get_interval(max(eigen(c2)$value))
```

```
## [1] "[412.134651567631, 1602.95415544215]"
```

```r
get_interval(max(eigen(c3)$value))
```

```
## [1] "[288.024056247535, 1120.2391162043]"
```

```r
get_interval(max(eigen(c4)$value))
```

```
## [1] "[278.051793765915, 1081.45305557273]"
```

```r
get_interval(max(eigen(c5)$value))
```

```
## [1] "[840.407273828309, 3268.67524175127]"
```

Complete case analysis and available case analysis give us a higher covariance than Mean imputation but also have larger confidence intervals because our data only has few complete records. The EM-algorithm generates a smaller range of confidence interval than Complete case and available case but larger than mean imputation(with bootstrap or not) Therefore Mean imputation with the bootstrap might be a good method to handle missing data for this particular scores data.

Question 3.

```r
library(SMPracticals)
cov(mathmarks)
```

```
##            mechanics   vectors   algebra  analysis statistics
## mechanics   305.7680 127.22257 101.57941 106.27273  117.40491
## vectors     127.2226 172.84222  85.15726  94.67294   99.01202
## algebra     101.5794  85.15726 112.88597 112.11338  121.87056
## analysis    106.2727  94.67294 112.11338 220.38036  155.53553
## statistics  117.4049  99.01202 121.87056 155.53553  297.75536
```

```r
get_interval(max(eigen(cov(mathmarks))$value))
```

```
## [1] "[431.810689297739, 1679.48202399712]"
```

Using EM-algorithm generates a cloest confidence interval of λ_1 from the full data. Therefore the EM-algorithm might be the best method to fill in the missing data in this case which is not consistent with the result we thought at question2, because the data size in questions before is really small.

Question 4.

partially observed vectors:

$$X_i = \begin{bmatrix} X_{io} \\ X_{im} \end{bmatrix}$$

we have that,

$$\mu^{(k)} = \begin{bmatrix} \mu_{io}^{(k)} \\ \mu_{im}^{(k)} \end{bmatrix}, \Sigma^{(k)} = \begin{bmatrix} \Sigma_{ioo}^{(k)} & \Sigma_{iom}^{(k)} \\ \Sigma_{imo}^{(k)} & \Sigma_{imm}^{(k)} \end{bmatrix}$$

Then, for E-step: Because of

$$E(X_i|X_io) = \begin{bmatrix} X_{io} \\ E(X_{im}|X_{io}) \end{bmatrix}$$

$$E(X_iX_i^T|X_{io}) = \begin{bmatrix} X_{io}X_{io}^T & X_{io}E(X_{im}^T|X_{io}) \\ E(X_{im}|X_{io})X_{io}^T & E(X_{im}X_{im}^T|X_{io}) \end{bmatrix}$$

where from the propertites of multivariate normal distribution,

$$E(X_{im}|X_{io}) = \mu_{im}^{(k)} + \Sigma_{imo}^{(k)}(\Sigma_{ioo}^{(k)})^{-1}(X_{io} - \mu_{io}^{(k)})$$

$$E(X_{im}X_{im}^T|X_{io}) = Cov(X_{im}|X_{io}) + E(X_{im}|X_{io})E(X_{im}|X_{io})^T = (\Sigma_{imm}^{(k)} - \Sigma_{imo}^{(k)}(\Sigma_{ioo}^{(k)})^{-1}\Sigma_{iom}^{(k)}) + E(X_{im}|X_{io})E(X_{im}|X_{io})^T$$

Then, for M-step:

$$\mu^{(k+1)} : \frac{1}{n}\sum_{i=1}^{n} E(X_i|X_{io}) = 0,\ \Sigma^{(k+1)} : \frac{1}{n}\sum_{i=1}^{n} E(X_iX_i^T|X_{io}) - \mu^{(k+1)}\mu^{(k+1)^T} = 0$$

To simplify using the information above, we can get:

$$\mu^{(k+1)} : \sum_{i=1}^{n}(\hat{X}_i - \mu) = 0,\ \Sigma^{(k+1)} : \sum_{i=1}^{n}(\Sigma - (\hat{X}_i - \mu)(\hat{X}_i - \mu)^T - C_i^{(k)}) = 0$$