STAT5703 HW2 Ex1

Chao Huang (ch3474), Wancheng Chen (wc2687), Chengchao Jin (cj2628)

Exercise 2

Question 1

```
data <- read.table('scores.txt', header = TRUE, sep = "", dec = ".")
colnames(data) <- c('A','B','C','D','E')
```

```
# Complete case analysis.
cov_1 <- cov(data,use="complete.obs")
cov_1
```

```
##        A      B      C      D      E
## A 216.30  -7.50  45.05  77.65  94.50
## B  -7.50 221.50 117.50  77.00 226.75
## C  45.05 117.50 157.30  85.90 242.00
## D  77.65  77.00  85.90  75.20 132.25
## E  94.50 226.75 242.00 132.25 422.00
```

```
# Available case analysis.
cov_2 <- cov(data,use="pairwise.complete.obs")
cov_2
```

```
##            A          B         C         D        E
## A 121.363636   4.563636  35.79091  42.12727  94.5000
## B   4.563636 179.134199 112.26840 114.60173 172.5000
## C  35.790909 112.268398 151.48918 125.96537 182.3727
## D  42.127273 114.601732 125.96537 153.56061 142.8636
## E  94.500000 172.500000 182.37273 142.86364 294.5636
```

```
# Mean imputation
data_mean <- sapply(data, function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
cov_3 <- cov(data_mean,use="complete.obs")
cov_3
```

```
##          A         B         C         D         E
## A 57.79221   2.17316  17.04329  20.06061  21.50138
## B  2.17316 179.13420 112.26840 114.60173  82.14286
## C 17.04329 112.26840 151.48918 125.96537  86.84416
## D 20.06061 114.60173 125.96537 153.56061  68.03030
## E 21.50138  82.14286  86.84416  68.03030 140.26840
```

```
# Mean inputation with bootstrap
cov_4<-matrix(rep(0,25),ncol=5)
for(i in 1:200){
  ind<-sample(nrow(data),22,replace=TRUE)
  temp <- sapply(data[ind,], function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
  cov_4 <- cov_4 + cov(temp,use="complete.obs")
}
cov_4/200
```

```
##            A           B         C         D         E
## A 53.4993616   0.1705027  14.97522  18.41280  16.59634
## B  0.1705027 170.7175649 105.29768 106.47390  74.75521
```

```
## C 14.9752157 105.2976840 144.00782 116.89853  78.90237
## D 18.4127987 106.4738961 116.89853 139.86856  61.16055
## E 16.5963374  74.7552058  78.90237  61.16055 126.13911
```

```r
# The EM-algorithm
a.out <- amelia(data,m=1,boot.type='none')
```

```
## -- Imputation 1 --
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
##  61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
##  81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
##  101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##  121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##  141 142 143 144 145 146 147 148 149
```

```r
cov_5 <- cov(a.out$imputations$imp1,use='complete.obs')
cov_5
```

```
##           A         B         C         D        E
## A 220.01365  15.64543  78.02824  92.66501 144.8690
## B  15.64543 179.13420 112.26840 114.60173 144.1380
## C  78.02824 112.26840 151.48918 125.96537 187.2759
## D  92.66501 114.60173 125.96537 153.56061 112.0304
## E 144.86901 144.13798 187.27594 112.03036 337.7459
```

- Using mean imputation or mean imputation with bootstrap to fill the missing data has smaller covariance matrix value comparing to the other three methods.
- Complete case analysis and em-algorithm show the negative correlation between variable A & B while the other methods do not.
- The results of covariance matrix using methods of mean inputation and mean inputation with bootstrap are quite close.

Question 2

Because $\sqrt{n}(\hat{\lambda}_1 - \lambda_1) \to N(0, 2\lambda^2)$, we can get that $\hat{\lambda}_1 \sim N(\lambda_1, \frac{2\lambda^2}{n})$.

$$P[-Z_{1-\frac{\alpha}{2}} \le \frac{\hat{\lambda}_1 - \lambda_1}{\sqrt{\frac{2\lambda_1^2}{n}}} \le Z_{1-\frac{\alpha}{2}}] = 1 - \alpha$$

Try to calculate the lower bound for λ_1

$$\hat{\lambda}_1 - \lambda_1 \le \frac{Z_{1-\frac{\alpha}{2}}\sqrt{2}\lambda_1}{\sqrt{n}}$$

$$\lambda_1 \ge \frac{\hat{\lambda}_1}{1 + \sqrt{\frac{2}{n}}Z_{1-\frac{\alpha}{2}}}$$

In the same way, we can calculate the upper bound for λ_1

$$\lambda_1 \le \frac{\hat{\lambda}_1}{1 - \sqrt{\frac{2}{n}}Z_{1-\frac{\alpha}{2}}}$$

So the confidence interval for λ_1 is:

$$\lambda_1 \in [\frac{\hat{\lambda}_1}{1+\sqrt{\frac{2}{n}}Z_{1-\frac{\alpha}{2}}}, \frac{\hat{\lambda}_1}{1-\sqrt{\frac{2}{n}}Z_{1-\frac{\alpha}{2}}}]$$

```
get_interval <- function(lambda) {
  mu = lambda
  sd <- sqrt(2*lambda*lambda)
  print(paste0('Left: ',mu/(1+sqrt(2/nrow(data))*qnorm(0.975)),
             ' right: ',mu/(1-sqrt(2/nrow(data))*qnorm(0.975))))
}
get_interval(max(eigen(cov_1)$value))
```

```
## [1] "Left: 482.301219299174 right: 1875.8596024619"
```

```
get_interval(max(eigen(cov_2)$value))
```

```
## [1] "Left: 412.134651567631 right: 1602.95415544215"
```

```
get_interval(max(eigen(cov_3)$value))
```

```
## [1] "Left: 288.024056247535 right: 1120.2391162043"
```

```
get_interval(max(eigen(cov_4)$value))
```

```
## [1] "Left: 53352.3819269223 right: 207508.448967347"
```

```
get_interval(max(eigen(cov_5)$value))
```

```
## [1] "Left: 435.757689360586 right: 1694.83346345557"
```

- Using mean imputation with bootstrap to fill the missing data will generate much greater confidence interval range than the other four methods, Meanwhile, the value of lower bound and upper bound becomes much different than the others. Thus, we may not use mean imputation with bootstrape to fill the missing data.
- Using em-algorithm to fill in the missing data, it can generate smaller range of confidence interval than using complete case analysis or available case analysis methods. It seems like a good method to fill in the missing data.

Question 3

```
pvar<-cov(mathmarks)
pvar
```

```
##            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
```

```
get_interval(max(eigen(pvar)$value))
```

```
## [1] "Left: 431.810689297739 right: 1679.48202399712"
```

- Using available case analysis method to fill in the missing data can generate more close sample covariance matrix and confidence interval of λ_1 to the results of true complete data comparing to other methods.
- Imputation methods for the missing data, especially em-algorithm method, may be affected due to the insufficient data as the input data has only 22 rows.

Question 4

For Missing data, we can construct:

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

Let,

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

Then, for E-step:

$$E(X_i|X_io) = \begin{bmatrix} X_{io} \\ E(X_{im}|X_{io}) \end{bmatrix}, E(X_iX_i'|X_{io}) = \begin{bmatrix} X_{io}X_{io}' & X_{io}E(X_{im}'|X_{io}) \\ E(X_{im}|X_{io})X_{io}' & E(X_{im}X_{im}'|X_{io}) \end{bmatrix} E(X_{im}|X_{io}) = \mu_{im}^{(k)} + \Sigma_{imo}^{(k)}(\Sigma_{ioo}^{(k)})^{-1}(X_{io} -$$

Then, for M-step:

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

To simplify using the information above, we can get:

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