MVA HW#3 Attachment

Chen Yue (cy179)

Before doing the questions, let's define the following 4 functions that can greatly help the computation:

```
> multi.mean_and_cov = function(data, na.action=F){
+   setClass('mean_and_cov', representation (mean = 'numeri
c', cov = 'matrix'))
+   returnValue(new('mean_and_cov', mean = apply(data, 2, m
ean), cov = cov(data)))
+ }

> test.TSquare = function(mu=rep(0,ncol(data)), data){
+   n = nrow(data)
+   p = ncol(data)
+   mean = apply(data, 2, mean)
+   diff = mean - mu
+   T.sq = n * t(diff) %*% solve(cov(data)) %*% (diff)
+   F = (n - p)/((n - 1) * p) * T.sq;
+   cat('T-square Test\n---------\n','T-square Value: ', T.
sq, '\nP-value: ', 1 - pf(F, p, n - p),'\n', sep = '')
+ }


> TF.convert=function(T.sq = 0, F=0, n, p, T_to_F = TRUE){
+   if(T_to_F){
+     F = (n - p)/((n - 1) * p) * T.sq
+     df1 = p
+     df2 = n-p
+     r = data.frame(F,df1,df2)
+     returnValue(r)
+   }else{
+     T.sq = F*(n-1)*p/(n-p)
+     df1 = p
+     df2 = n-1
+     returnValue(data.frame(T.sq,df1,df2))
+   }
+ }


# this function computes the confident region and simultane
ous and Bonferroni confident interval at the specified conf
ident level
> T.ellipse = function(data, confident.level=0.95){
+   n = nrow(data)
+   p = ncol(data)
+   cat('\n\nT.sq confidence ellipse')
+   center = colMeans(data)
+   cat('\ncenters\n')
+   print(center)
+   T.critical = TF.convert(F=qf(confident.level,p,(n-p)),
n=n, p=p,T_to_F = FALSE)
+   cat('\nThe critical T value\n')
+   print(T.critical)
+   S = cov(data)
+   cat('\ncovariance matrix\n')
+   print(S)
+   ei = eigen(S)
+   cat('\nwith eigenvalue and vectors\n')
+   print(ei)
+   length = 2*sqrt(ei$values) * sqrt(1/nrow(data)*T.critic
al$T.sq)
+   cat('\nlength\n')
+   print(length)
+   axes = NULL
+   for(i in c(1:length(length))){
+     axes = cbind (axes, length[i]/2 * ei$vectors[,i])
+   }
+   cat('\naxes\n')
+   print(axes)
+   distance.simontaneous = sqrt(T.critical$T.sq/nrow(dat
a))*sqrt(diag(S))
+   CI.simontaneous = rbind (center - distance.simontaneou
s, center + distance.simontaneous)
+   rownames(CI.simontaneous) = c('low','high')
+   cat('\nsimontaneous confident interval:\n')
+   print(t(CI.simontaneous))
+   t.Bonferroni = (qt(df=nrow(data)-1,p=(1-confident.leve
l)/(2*ncol(data)),lower.tail=F))
+   cat('\nt value for Bonferroni: \n')
+   print(t.Bonferroni)
+   distance.Bonferroni = t.Bonferroni*sqrt(diag(S)/nrow(da
ta))
+   CI.Bonferroni = rbind(center-distance.Bonferroni, cente
r+distance.Bonferroni)
+   rownames(CI.Bonferroni) = c('low','high')
+   cat('\nBonferroni confident interval:\n')
+   print(t(CI.Bonferroni))
+ }
```

Problem #5.1

```
> mu =  matrix(c(7,11))
> X = matrix(c(2,8,6,8,12,9,9,10),ncol = 2)
> summary = multi.mean_and_cov(X)
> print(summary)
An object of class "mean_and_cov"
Slot "mean":
[1]  6 10

Slot "cov":
          [,1]      [,2]
[1,]  8.000000 -3.333333
[2,] -3.333333  2.000000

> test.TSquare(mu=c(7,11),X)
T-square Test
---------
T-square Value: 13.63636
P-value: 0.1803279
```

Problem #5.3

```
> cov.xbar = (nrow(X)-1)/nrow(X) * summary@cov
> cov.mu = cov.xbar + (summary@mean - mu) %*% t(summary@mea
n - mu)
> T.sq = (nrow(X)-1) * (det(cov.mu)/det(cov.xbar)-1)
> cov.xbar
     [,1] [,2]
[1,]  6.0 -2.5
[2,] -2.5  1.5
> cov.mu
     [,1] [,2]
[1,]  7.0 -1.5
[2,] -1.5  2.5
> T.sq
[1] 13.63636
```

Problem #5.4

```
> x1 = c(3.7,5.7,3.8,3.2,3.1,4.6,2.4,7.2,6.7,5.4,3.9,4.5,3.
5,4.5,1.5,8.5,4.5,6.5,4.1,5.5)
> x2 = c(48.5,65.1,47.2,53.2,55.5,36.1,24.8,33.1,47.4,54.1,
36.9,58.8,27.8,40.2,13.5,56.4,71.6,52.8,44.1,40.9)
> x3 = c(9.3,8.0,10.9,12.0,9.7,7.9,14,7.6,8.5,11.3,12.7,12.
3,9.8,8.4,10.1,7.1,8.2,10.9,11.2,9.4)
> x = cbind(x1,x2,x3)
> T.ellipse(x,0.9)

T.sq confidence ellipse
centers
    x1     x2     x3
 4.640 45.400  9.965

The critical T value
      T.sq df1 df2
1 8.172573   3  19

covariance matrix
          x1       x2        x3
x1  2.879368  10.0100 -1.809053
x2 10.010000 199.7884 -5.640000
x3 -1.809053  -5.6400  3.627658

with eigenvalue and vectors
eigen() decomposition
$values
[1] 200.462464   4.531591   1.301392

$vectors
            [,1]        [,2]        [,3]
[1,] -0.05084144 -0.57370364  0.81748351
[2,] -0.99828352  0.05302042 -0.02487655
[3,]  0.02907156  0.81734508  0.57541452


length
[1] 18.101348  2.721571  1.458473

axes
           [,1]        [,2]        [,3]
[1,] -0.4601493 -0.78068768  0.59613898
[2,] -9.0351388  0.07214943 -0.01814089
[3,]  0.2631172  1.11223146  0.41961339

simultaneous confident interval:
         low      high
x1  3.555292  5.724708
x2 36.364555 54.435445
x3  8.747476 11.182524

t value for Bonferroni:
```

```
[1] 2.294392

Bonferroni confident interval:
          low      high
x1  3.769435  5.510565
x2 38.148336 52.651664
x3  8.987840 10.942160
```

Problem #5.5

```
> T.ellipse(x)


T.sq confidence ellipse
centers
    x1     x2     x3
 4.640 45.400  9.965

The Critical T Value
     T.sq df1 df2
1 10.7186   3  19

covariance matrix
          x1       x2        x3
x1  2.879368  10.0100 -1.809053
x2 10.010000 199.7884 -5.640000
x3 -1.809053  -5.6400  3.627658

with eigenvalue and vectors
eigen() decomposition
$values
[1] 200.462464   4.531591   1.301392

$vectors
            [,1]        [,2]        [,3]
[1,] -0.05084144 -0.57370364  0.81748351
[2,] -0.99828352  0.05302042 -0.02487655
[3,]  0.02907156  0.81734508  0.57541452


length
[1] 20.730065  3.116804  1.670276

axes
            [,1]        [,2]        [,3]
[1,]  -0.5269732 -0.89406082  0.68271156
[2,] -10.3472410  0.08262713 -0.02077535
[3,]   0.3013276  1.27375209  0.48055055

simultaneous confident interval:
          low      high
x1  3.397768  5.882232
x2 35.052408 55.747592
x3  8.570664 11.359336

t value for Bonferroni:
[1] 2.625106

Bonferroni confident interval:
          low      high
x1  3.643952  5.636048
x2 37.103078 53.696922
x3  8.846992 11.083008
```

Problem #5.18

```
> x1 = c(468,428,514,547,614,501,421,527,527,620,587,541,56
1,468,614,527,507,580,507,521,574,587,488,488,587,421,481,4
28,640,
+        574,547,580,494,554,647,507,454,427,521,468,587,50
7,574,507,494,541,362,408,594,501,687,633,647,647,614,633,4
48,408,
+        441,435,501,507,620,415,554,348,468,507,527,527,43
5,660,733,507,527,428,481,507,527,488,607,561,614,527,474,4
41,607)
> x2 = c(41,39,53,67,61,67,46,50,55,72,63,59,53,62,65,48,3
2,64,59,54,52,64,51,62,56,38,52,40,65,61,64,64,53,51,58,65,
52,57,
+        66,57,55,61,54,53,41,47,36,28,68,25,75,52,67,65,5
9,65,55,51,36,60,54,42,71,52,69,28,49,54,47,47,50,70,73,45,
62,37,
+        48,61,66,41,69,59,70,49,41,47,67)
> x3 = c(26,26,21,33,27,29,22,23,19,32,31,19,26,20,28,21,2
7,21,21,23,25,31,27,18,26,16,26,19,25,28,27,28,26,21,23,23,
28,21,
+        26,14,30,31,31,23,24,25,17,17,23,26,33,31,29,34,2
5,28,24,19,22,20,21,24,36,20,30,18,25,26,31,26,28,25,33,28,
29,19,
+        23,19,23,28,28,34,23,30,16,26,32)
> X = cbind(x1,x2,x3)
> mu = c(500,50,30)
> multi.mean_and_cov(X)
An object of class "mean_and_cov"
Slot "mean":
       x1        x2        x3
526.58621  54.70115  25.12644

Slot "cov":
          x1        x2        x3
x1 5808.0593 596.84002 222.02967
x2  596.8400 125.60732  23.35218
x3  222.0297  23.35218  23.11173

> test.TSquare(mu=mu,X)
T-square Test
---------
T-square Value: 223.4327
P-value: 0
> T.ellipse(X)

T.sq confidence ellipse
centers
       x1        x2        x3
526.58621  54.70115  25.12644

The Critical T Value
      T.sq df1 df2
1 8.333483   3  86

covariance matrix
          x1        x2        x3
x1 5808.0593 596.84002 222.02967
x2  596.8400 125.60732  23.35218
x3  222.0297  23.35218  23.11173

with eigenvalue and vectors
eigen() decomposition
$values
[1] 5878.58213   63.59827   14.59800

$vectors
           [,1]         [,2]         [,3]
[1,] 0.99392352  0.103560215 -0.037300504
[2,] 0.10326884 -0.994606566 -0.009660483
[3,] 0.03809977 -0.005749802  0.999257398

length
[1] 47.459149  4.936354  2.364996

axes
           [,1]        [,2]        [,3]
[1,] 23.5853823  0.25560495 -0.04410776
[2,]  2.4505257 -2.45486515 -0.01142350
[3,]  0.9040913 -0.01419153  1.18161969

simultaneous confident interval:
         low      high
x1 502.99940 550.17302
x2  51.23249  58.16980
x3  23.63855  26.61432

t value for Bonferroni:
[1] 2.441709

Bonferroni confident interval:
         low      high
x1 506.63589 546.53652
x2  51.76727  57.63503
x3  23.86794  26.38493
```

Problem #5.21

```
> x1 = c(1.103,0.842,0.925,0.857,0.795,0.787,0.933,0.799,0.
945,0.921,0.792,0.815,0.755,0.88,0.9,0.764,0.733,0.932,0.85
6,0.89,0.688,0.94,0.493,0.835,0.915)
> x2 = c(1.052,0.859,0.873,0.744,0.809,0.779,0.88,0.851,0.8
76,0.906,0.825,0.751,0.724,0.866,0.838,0.757,0.748,0.898,0.
786,0.95,0.532,0.85,0.616,0.752,0.936)
> x3 = c(2.139,1.873,1.887,1.739,1.734,1.509,1.695,1.74,1.8
11,1.954,1.624,2.204,1.508,1.784,1.902,1.743,1.863,2.028,1.
39,2.187,1.65,2.334,1.037,1.509,1.971)
> x4 = c(2.238,1.741,1.809,1.547,1.715,1.474,1.656,1.777,1.
759,2.009,1.657,1.846,1.458,1.811,1.606,1.794,1.869,2.032,
1.324,2.087,1.378,2.225,1.268,1.422,1.869)
> x5 = c(0.873,0.59,0.767,0.706,0.549,0.782,0.737,0.618,0.8
53,0.823,0.686,0.678,0.662,0.81,0.723,0.586,0.672,0.836,0.5
78,0.758,0.533,0.757,0.546,0.618,0.869)
> x6 = c(0.872,0.744,0.713,0.674,0.654,0.571,0.803,0.682,0.
777,0.765,0.668,0.546,0.595,0.819,0.677,0.541,0.752,0.805,
0.61,0.718,0.482,0.731,0.615,0.664,0.868)
> X = cbind(x1,x2,x3,x4,x5,x6)
> T.ellipse(X)


T.sq confidence ellipse
centers
     x1      x2      x3      x4      x5      x6
0.84380 0.81832 1.79260 1.73484 0.70440 0.69384

The Critical T Value
      T.sq df1 df2
1 19.91988   6  24

covariance matrix
            x1          x2         x3         x4          x5          x6
x1 0.013001583 0.010378442 0.02234696 0.02008568 0.009120708 0.007957842
x2 0.010378442 0.011417893 0.01853122 0.02109951 0.008529783 0.008908512
x3 0.022346958 0.018531217 0.08035850 0.06676985 0.016828125 0.012836600
x4 0.020085675 0.021099512 0.06676985 0.06948447 0.017735483 0.016793598
x5 0.009120708 0.008529783 0.01682812 0.01773548 0.011568417 0.008071150
x6 0.007957842 0.008908512 0.01283660 0.01679360 0.008071150 0.010599140

with eigenvalue and vectors
eigen() decomposition
$values
[1] 0.1635333121 0.0183637399 0.0080694108 0.0033556694 0.0022219234 0.0008859511

$vectors
```

```
           [,1]        [,2]        [,3]         [,4]       [,5]        [,6]
[1,] -0.2177374  0.33726261  0.51708011  0.400776794 -0.2725153  0.58016025
[2,] -0.2040296  0.43154108  0.03119159  0.425371500 -0.1982959 -0.74223251
[3,] -0.6729085 -0.50008903  0.43396201 -0.051478287  0.2659833 -0.18810889
[4,] -0.6325336  0.02442084 -0.69021068  0.009887147 -0.2605705  0.23434869
[5,] -0.1812878  0.43022767  0.23536179 -0.809715391 -0.2519046 -0.08688367
[6,] -0.1587212  0.51404763 -0.10745352  0.007089769  0.8272328  0.12104525


length
[1] 0.72194899 0.24192679 0.16037043 0.10341725 0.08415268
0.05313837

axes
            [,1]        [,2]         [,3]          [,4]         [,5]         [,6]
[1,] -0.07859764  0.040796431  0.041462181  0.0207236167 -0.011466444  0.015414386
[2,] -0.07364950  0.052200673  0.002501104  0.0219953750 -0.008343567 -0.019720515
[3,] -0.24290280 -0.060492467  0.034797338 -0.0026618714  0.011191605 -0.004997900
[4,] -0.22832851  0.002954028 -0.055344694  0.0005112508 -0.010963851  0.006226454
[5,] -0.06544029  0.052041799  0.018872536 -0.0418692689 -0.010599222 -0.002308428
[6,] -0.05729430  0.062180946 -0.008616184  0.0003666022  0.034806928  0.003216074

simultaneous confident interval:
         low      high
x1 0.7420179 0.9455821
x2 0.7229380 0.9137020
x3 1.5395599 2.0456401
x4 1.4995425 1.9701375
x5 0.6083914 0.8004086
x6 0.6019414 0.7857386

t value for Bonferroni:
[1] 2.875094

Bonferroni confident interval:
         low      high
x1 0.7782338 0.9093662
x2 0.7568766 0.8797634
x3 1.6295961 1.9556039
x4 1.5832656 1.8864144
x5 0.6425529 0.7662471
x6 0.6346406 0.7530394
```