Q2

```
library(psych)
A=matrix(c(20,16,5,9,16,6,15,5,14),3,3)
B=matrix(c(7,19,15,15,20,13,8,15,14),3,3)
C=matrix(c(1,11,16,6,12,9,20,20,15),3,3)
A%*%B%*%C # ABC
```

```
##       [,1]  [,2]  [,3]
## [1,] 16041 15861 31795
## [2,] 14374 14388 28890
## [3,]  9722  9612 19610
```

```
tr(A%*%B%*%C) # Trace of ABC
```

```
## [1] 50039
```

```
B%*%C%*%A # BCA
```

```
##       [,1]  [,2]  [,3]
## [1,] 13504 10764 13810
## [2,] 22429 18165 23700
## [3,] 17442 14010 18370
```

```
tr(B%*%C%*%A) # Trace of BCA
```

```
## [1] 50039
```

```
C%*%A%*%B # CAB
```

```
##       [,1]  [,2]  [,3]
## [1,] 10662 11965  9653
## [2,] 18968 22465 17331
## [3,] 18380 22080 16912
```

```
tr(C%*%A%*%B) # Trace of CAB
```

```
## [1] 50039
```

```
A%*%C%*%B # ACB
```

```
##       [,1]  [,2]  [,3]
## [1,] 21485 23110 19587
## [2,] 18956 20035 17181
## [3,] 12847 14575 11800
```

```
tr(A%*%C%*%B) # Trace of ACB
```

```
## [1] 53320
```

tr(ABC)=tr(BCA)=tr(CAB), but tr(ACB)!=tr(ABC).

Q3

(a)

```
p1 = read.table("p1_4.txt", header=FALSE)
x_bar = colMeans(p1)
```

```
x_bar = as.matrix(x_bar)
x_bar # Sample mean
```

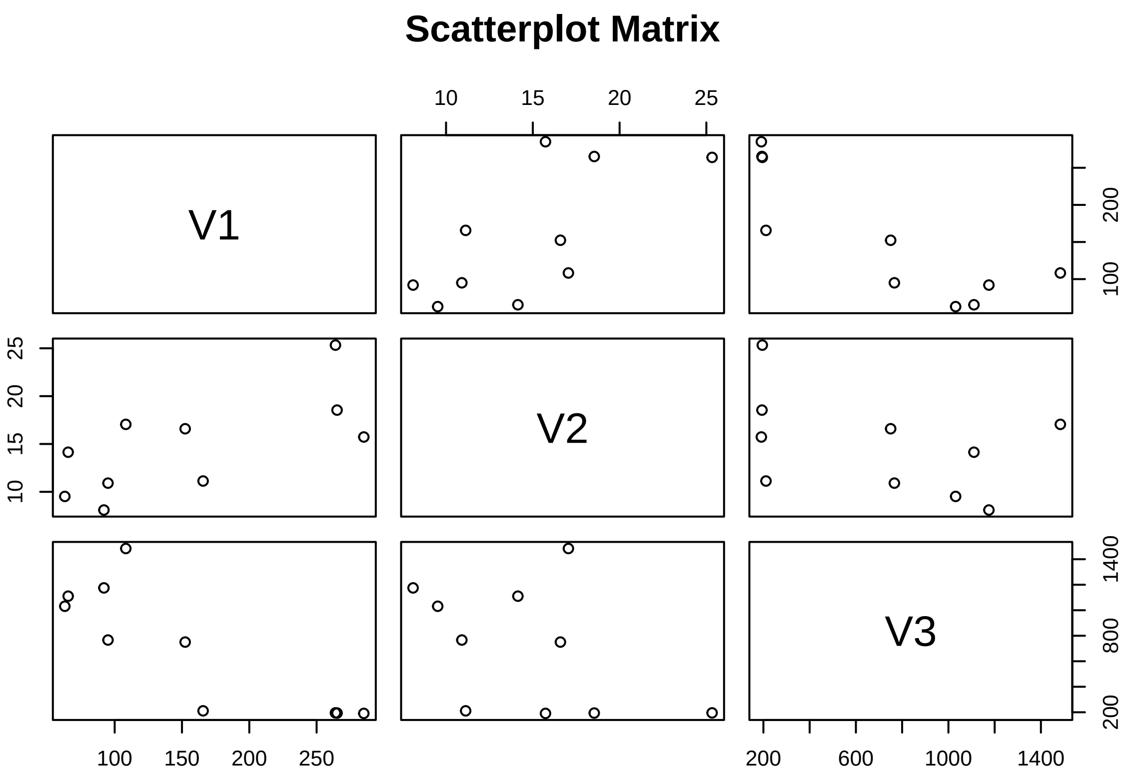
```
##        [,1]
## V1 155.603
## V2  14.704
## V3 710.911
```

```
n = nrow(p1)
Sn=((n-1)/n)*cov(p1)
Sn # Sn
```

```
##             V1          V2          V3
## V1   6728.8079  273.25676 -32018.3636
## V2    273.2568   23.57128   -948.4447
## V3 -32018.3636 -948.44465 213348.8428
```

(b)

```
plot(p1, main="Scatterplot Matrix") # Scatterplot
```




```
R = cor(p1)
R # Correlation matrix
```

```
##            V1         V2         V3
## V1  1.0000000  0.6861360 -0.8450549
## V2  0.6861360  1.0000000 -0.4229366
## V3 -0.8450549 -0.4229366  1.0000000
```

All variables are correlated, V1 and V2 has positive correlation; V1 and V3, V2 and V3 have negative correlation. It seems the data can be separated in two groups. There seems exist a outlier, but since the sample size is too small, we can't conclude that there is a outlier.

(c)

```
I = diag(n) # Identity matrix
one = as.matrix(c(rep(1,n))) # vector of 1
J = one%*%t(one) # matrix of 1
Cn=I-(1/n)*J # Cn matrix
Cn
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]  0.9 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  -0.1
##  [2,] -0.1  0.9 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  -0.1
##  [3,] -0.1 -0.1  0.9 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  -0.1
##  [4,] -0.1 -0.1 -0.1  0.9 -0.1 -0.1 -0.1 -0.1 -0.1  -0.1
##  [5,] -0.1 -0.1 -0.1 -0.1  0.9 -0.1 -0.1 -0.1 -0.1  -0.1
##  [6,] -0.1 -0.1 -0.1 -0.1 -0.1  0.9 -0.1 -0.1 -0.1  -0.1
##  [7,] -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  0.9 -0.1 -0.1  -0.1
##  [8,] -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  0.9 -0.1  -0.1
##  [9,] -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1  0.9  -0.1
## [10,] -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1   0.9
```

```
X = as.matrix(p1)
Xc = Cn%*%X # Xc matrix
Xc
```

```
##            V1     V2       V3
##  [1,] -47.323  2.346  773.189
##  [2,]  -3.243  1.886   39.419
##  [3,] -60.563 -3.794   55.509
##  [4,] -90.153 -0.564  399.549
##  [5,] -92.633 -5.184  320.379
##  [6,] 108.387 10.626 -515.651
##  [7,] 109.587  3.836 -517.081
##  [8,] 129.457  1.026 -519.801
##  [9,] -63.593 -6.604  464.249
## [10,]  10.077 -3.574 -499.761
```

Q6

```
A = matrix(c(404.6, 282.4, 118.3, 109.4, 282.4, 284.4, -13.6,
             45.4,118.3, -13.6, 599.7, 488.3, 109.4, 45.4,
             488.3, 447.4), 4, 4)
ev = eigen(A)
P = ev$vectors # P matrix
P
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.2841936  0.7077960 -0.5510890  0.3384668
## [2,] -0.1260140  0.6385478  0.5972527 -0.4686859
## [3,] -0.7174597 -0.2624042 -0.3134530 -0.5640415
## [4,] -0.6233827 -0.1497517  0.4912610  0.5896024
```

```
Lambda = diag(ev$values) # Lambda matrix
Lambda
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 1068.443   0.0000  0.00000 0.000000
## [2,]    0.000 592.3669  0.00000 0.000000
## [3,]    0.000   0.0000 68.30835 0.000000
## [4,]    0.000   0.0000  0.00000 6.981935
```

```
P%*%Lambda%*%t(P) # PLambdaP' = A
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] 404.6 282.4 118.3 109.4
## [2,] 282.4 284.4 -13.6  45.4
## [3,] 118.3 -13.6 599.7 488.3
## [4,] 109.4  45.4 488.3 447.4
```

```
rootLambda = diag(sqrt(ev$values)) # Lambda^1/2
sqrtA = P%*%rootLambda%*%t(P) # A^1/2
sqrtA
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 17.645774  9.031242  3.067667  1.500912
## [2,]  9.031242 13.971557 -1.971632  1.935183
## [3,]  3.067667 -1.971632 20.154149 13.424319
## [4,]  1.500912  1.935183 13.424319 16.161373
```

Q7(b)

(i)

```
library(MASS)
A = matrix(c(-6, 2, -2, -3, 3, -1, 5, 2, -3, 1, 3, -1),3,4,byrow=TRUE)
generalized_inverseA = ginv(A) # Generalized Inverse of A
generalized_inverseA
```

```
##              [,1]         [,2]        [,3]
## [1,] -0.084306096  0.005188067 -0.07911803
## [2,]  0.028102032 -0.001729356  0.02637268
## [3,]  0.004755728  0.122784263  0.12753999
## [4,] -0.038045828  0.017725897 -0.02031993
```

(ii)

```
A%*%generalized_inverseA%*%A # A*(Generalized Inverse of A)*A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -6    2   -2   -3
## [2,]    3   -1    5    2
## [3,]   -3    1    3   -1
```

Q8

(a)

```
# Miu Matrices
miu1 = matrix(c(rep(0,2))) # miu of X_1
```

```r
miu1
```

```
##      [,1]
## [1,]    0
## [2,]    0
```

```r
miu2 = matrix(c(rep(0,3))) # miu of X_2
miu2
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
```

```r
# Sigma matrix
sigma = matrix(c(4.2, -0.01, -0.08, -0.32, -0.89, -0.01,
                 3.56, -0.25, 0.3, -1.28, -0.08, -0.25,
                 2.16, 0.55, 0.07, -0.32, 0.3, 0.55, 2.63,
                 0.74, -0.89, -1.28, 0.07, 0.74, 2.45),5,5,byrow=TRUE)
sigma
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,]  4.20 -0.01 -0.08 -0.32 -0.89
## [2,] -0.01  3.56 -0.25  0.30 -1.28
## [3,] -0.08 -0.25  2.16  0.55  0.07
## [4,] -0.32  0.30  0.55  2.63  0.74
## [5,] -0.89 -1.28  0.07  0.74  2.45
```

```r
# Manually partition Sigma matrix into sigma11, sigma12, sigma21, sigma22
sigma11 = matrix(c(4.2, -0.01, -0.01,
                 3.56),2,2,byrow=TRUE)
sigma11
```

```
##       [,1]  [,2]
## [1,]  4.20 -0.01
## [2,] -0.01  3.56
```

```r
sigma12 = matrix(c(-0.08, -0.32, -0.89,-0.25, 0.3, -1.28),2,3,byrow=TRUE)
sigma12
```

```
##       [,1]  [,2]  [,3]
## [1,] -0.08 -0.32 -0.89
## [2,] -0.25  0.30 -1.28
```

```r
sigma21 = t(sigma12) # Sigma21 = (Sigma12)'
sigma21
```

```
##       [,1]  [,2]
## [1,] -0.08 -0.25
## [2,] -0.32  0.30
## [3,] -0.89 -1.28
```

```r
sigma22 = matrix(c(2.16, 0.55, 0.07, 0.55, 2.63,
                 0.74, 0.07, 0.74, 2.45),3,3,byrow=TRUE)
sigma22
```

```
##      [,1] [,2] [,3]
## [1,] 2.16 0.55 0.07
## [2,] 0.55 2.63 0.74
```

```
## [3,] 0.07 0.74 2.45
```

```r
# Calculate miuY
X_2 = matrix(c(1,1,0),3,1,byrow=TRUE)
X_2
```

```
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    0
```

```r
miuY = miu1 + sigma12%*%solve(sigma22)%*%(X_2-miu2)
miuY
```

```
##             [,1]
## [1,] -0.03782437
## [2,]  0.14609683
```

(b)

```r
# sigmaY and |sigmaY|
sigmaY = sigma11 - sigma12%*%solve(sigma22)%*%sigma21
sigmaY
```

```
##            [,1]       [,2]
## [1,]  3.8746860 -0.4681063
## [2,] -0.4681063  2.6303564
```

```r
det_sigmaY = det(sigmaY)
det_sigmaY
```

```
## [1] 9.972682
```

```r
# |sigma22|
det_sigma22 = det(sigma22)
det_sigma22
```

```
## [1] 12.03811
```

```r
# |sigmaY|*|sigma22|
product_sigma = det_sigmaY*det_sigma22
product_sigma
```

```
## [1] 120.0523
```

```r
# |sigma|
det_sigma = det(sigma)
det_sigma
```

```
## [1] 120.0523
```

|sigma| = |sigmaY|*|sigma22|, this remainds me that P(x1|x2) = P(x1, x2)/P(x2)

Q10

(a)

```
d = read.table("data_chisqplot.txt", header=TRUE)
p = ncol(d)
x_bar = colMeans(d)
x_bar = as.matrix(x_bar) # sample mean
x_bar
```

```
##       [,1]
## x1 48.7548
## x2 60.0272
```

```
S = cov(d) # variance-covariance matrix
S
```

```
##          x1       x2
## x1 27.82135 3.812360
## x2  3.81236 8.431554
```

```
dsq = mahalanobis(d, x_bar, S) # Mahalanobis squared distances
dsq
```

```
##  [1]  7.38083045  0.74983035  0.59872191  0.77284313  0.35881941  0.28609181
##  [7]  0.49599731  0.52330734 18.82971297  1.80897455  0.56863484  0.03479158
## [13]  2.00162039  1.04114638  0.69331747  0.77917700  0.57515734  0.69594878
## [19]  0.69769845  3.99584884  0.73682060  2.42647391  0.63767460  0.67680614
## [25]  0.63375447
```

```
ro = qchisq(0.5,df=p) # chisq at 50% quantile with df = 2
ro
```

```
## [1] 1.386294
```

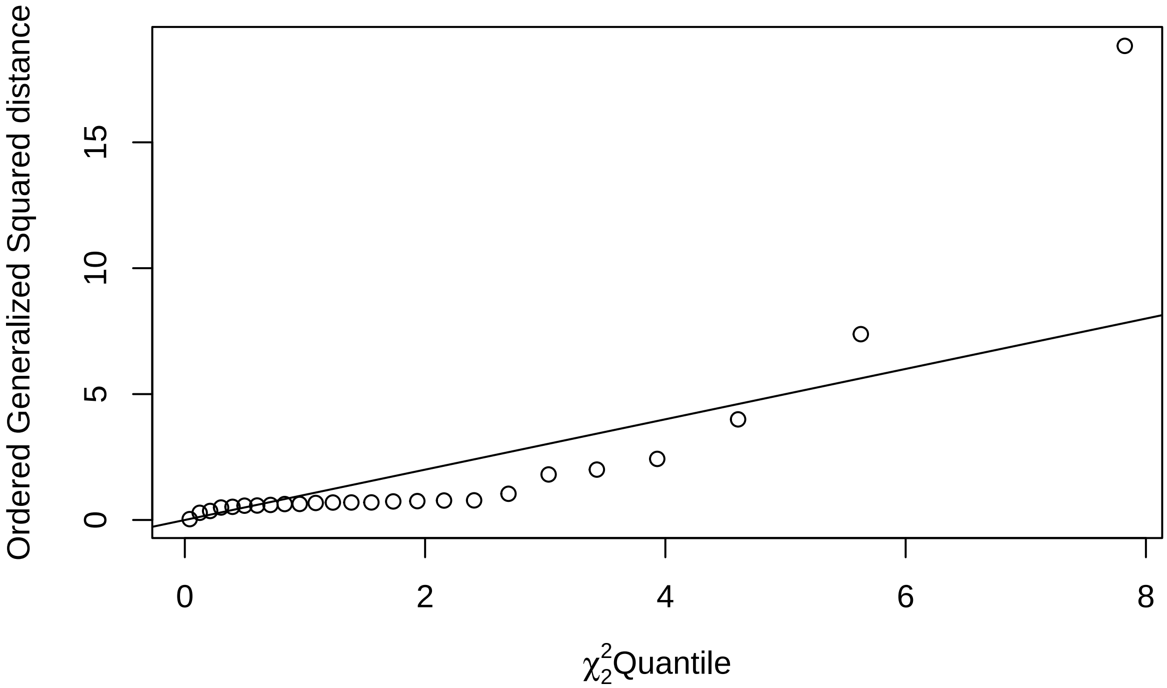
```
length(which(dsq<ro)) # Number of distances that < ro
```

```
## [1] 19
```

Since out of 25 sample, 19 of them have Mahalanobis squared distances less than chisq at 50% quantile with degree of freedom equals to 2, the percentage of Mahalanobis squared distances less than chisq at 50% quantile with degree of freedom equals to 2 is much bigger than 50%, so univariate Normality assumption violated.

(b)

```
n = nrow(d)
plot(qchisq((1:n-1/2)/n,df=p), sort(dsq),xlab=expression(
  paste(chi[2]^2,"Quantile")),
  ylab="Ordered Generalized Squared distance")
abline(a=0,b=1) # The straight line origin having slope 1
```



The graph is not close to the straight line origin having slope 1, so the univariate Normality assumption violated.