Chapter 3 and 4 Homework

Shuowen Wei

1.

(a).

> cov(M)

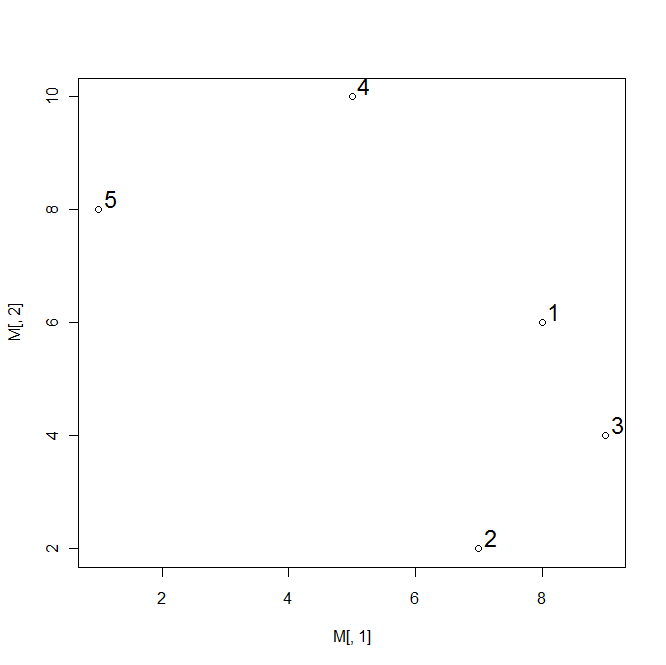
col1 col2

col1 10 -6

col2 -6 10

(b).

plot(M[,1],M[,2]);text(M[,1]+0.2,M[,2]+0.2,1:5,cex=1.5)



> E<-eigen(cov(M))

> scores<-M%\*%(E$vectors)

> scores

[,1] [,2]

[1,] -1.414214 -9.899495

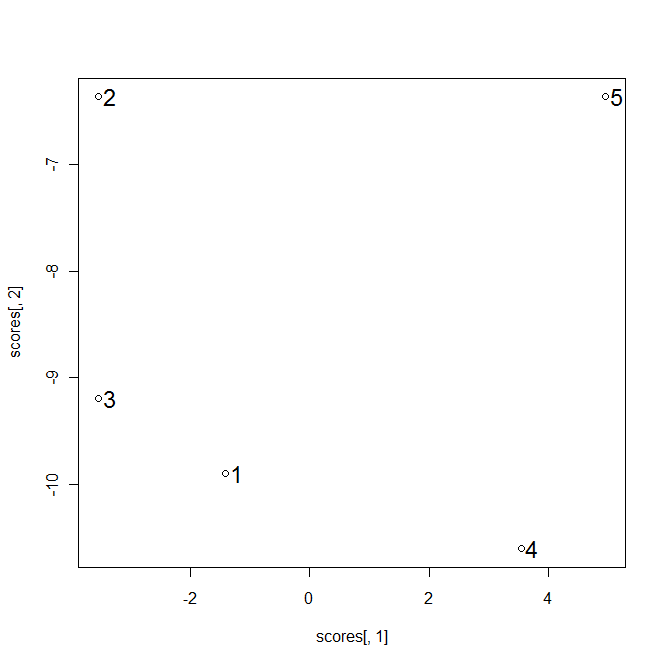
[2,] -3.535534 -6.363961

[3,] -3.535534 -9.192388

[4,] 3.535534 -10.606602

[5,] 4.949747 -6.363961

> plot(scores[,1],scores[,2]);text(scores[,1]+0.2,scores[,2],1:5,cex=1.5)



> dist(M)

1 2 3 4

2 4.123106

3 2.236068 2.828427

4 5.000000 8.246211 7.211103

5 7.280110 8.485281 8.944272 4.472136

> dist(M,"manhattan")

1 2 3 4

2 5

3 3 4

4 7 10 10

5 9 12 12 6

2.

> M<-matrix(0,5,5)

> fix(M)

> M

col1 col2 var3 var4 var5

[1,] 0 1 10 4 10

[2,] 1 0 5 9 5

[3,] 10 5 0 26 4

[4,] 4 9 26 0 26

[5,] 10 5 4 26 0

> meanrc<-matrix(0,5,5)

> fix(meanrc)

> meanrc

col1 col2 col3 col4 col5

[1,] 5 4 9 13 9

[2,] 5 4 9 13 9

[3,] 5 4 9 13 9

[4,] 5 4 9 13 9

[5,] 5 4 9 13 9

> t(meanrc)

[,1] [,2] [,3] [,4] [,5]

col1 5 5 5 5 5

col2 4 4 4 4 4

col3 9 9 9 9 9

col4 13 13 13 13 13

col5 9 9 9 9 9

> meanM<-matrix(8,5,5)

> meanM

[,1] [,2] [,3] [,4] [,5]

[1,] 8 8 8 8 8

[2,] 8 8 8 8 8

[3,] 8 8 8 8 8

[4,] 8 8 8 8 8

[5,] 8 8 8 8 8

> M-meanrc-t(meanrc)+meanM

col1 col2 var3 var4 var5

[1,] -2 0 4 -6 4

[2,] 0 0 0 0 0

[3,] 4 0 -10 12 -6

[4,] -6 0 12 -18 12

[5,] 4 0 -6 12 -10

> B<-(-1/2)\*(M-meanrc-t(meanrc)+meanM)

**> B**

**col1 col2 var3 var4 var5**

**[1,] 1 0 -2 3 -2**

**[2,] 0 0 0 0 0**

**[3,] -2 0 5 -6 3**

**[4,] 3 0 -6 9 -6**

**[5,] -2 0 3 -6 5**

> E<-eigen(B)

> E

$values

[1] 1.800000e+01 2.000000e+00 3.552714e-15 8.881784e-16 2.220446e-16

$vectors

[,1] [,2] [,3] [,4] [,5]

[1,] 0.2357023 0.000000e+00 0.9718253 0.0000000 0.0000000

[2,] 0.0000000 -1.603806e-16 0.0000000 -0.9553303 -0.2955402

[3,] -0.4714045 -7.071068e-01 0.1143324 0.1520542 -0.4915134

[4,] 0.7071068 1.182932e-16 -0.1714986 0.2027389 -0.6553512

[5,] -0.4714045 7.071068e-01 0.1143324 0.1520542 -0.4915134

> round(E$vectors%\*%diag(sqrt(E$values)))

[,1] [,2] [,3] [,4] [,5]

[1,] **1 0** 0 0 0

[2,] **0 0** 0 0 0

[3,] **-2 -1** 0 0 0

[4,]  **3 0** 0 0 0

[5,] **-2 1** 0 0 0

3.

(a).

**> M**

**col1 col2**

**[1,] 2 4**

**[2,] 4 2**

**[3,] 0 0**

**[4,] 6 6**

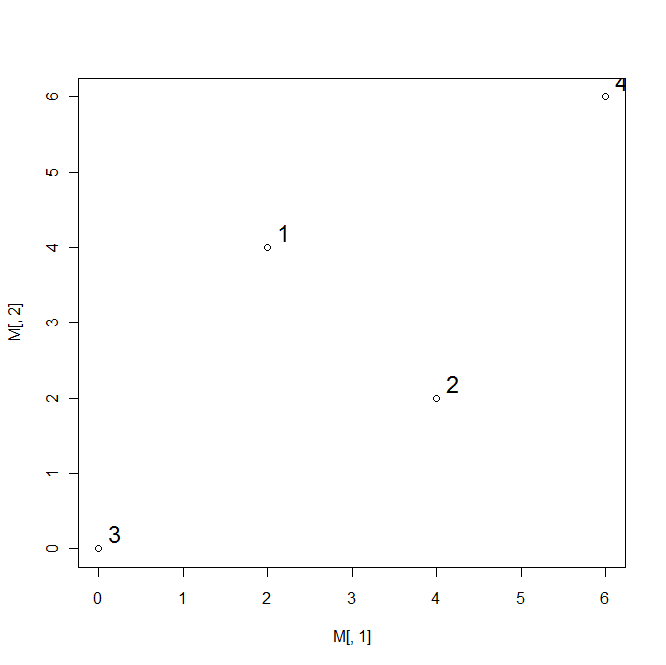
> dist(M,"manhattan")

1 2 3

2 **4**

3 **6 6**

4 **6 6 12**



(b).

> M

col1 col2 col3 col4

[1,] 0 4 6 6

[2,] 4 0 6 6

[3,] 6 6 0 12

[4,] 6 6 12 0

M<-M^2

> M

col1 col2 var3 var4

[1,] 0 16 36 36

[2,] 16 0 36 36

[3,] 36 36 0 144

[4,] 36 36 144 0

>meanrc<-matrix(0,4,4)

>meanrc[,1]<-mean(M[,1])

>meanrc[,2]<-mean(M[,2])

>meanrc[,3]<-mean(M[,3])

>meanrc[,4]<-mean(M[,4])

>meanM<-matrix(mean(M),4,4)

>B<-(-1/2)\*(M-meanrc-t(meanrc)+meanM)

> B

col1 col2 var3 var4

[1,] 3 -5 1 1

[2,] -5 3 1 1

[3,] 1 1 35 -37

[4,] 1 1 -37 35

E<-eigen(B)

round(E$vectors%\*%diag(sqrt(E$values)))

> E<-eigen(B)

> orgindata=round(E$vectors%\*%diag(sqrt(E$values)))

> orgindata

[,1] [,2] [,3] [,4]

[1,] 0 2 0 NaN

[2,] 0 -2 0 NaN

[3,] -6 0 0 NaN

[4,] 6 0 0 NaN

> orgindata<-orgindata[,1:2]

**> orgindata**

**[,1] [,2]**

**[1,] 0 2**

**[2,] 0 -2**

**[3,] -6 0**

**[4,] 6 0**

> newdist<-round(dist(orgindata),2)

> newdist

1 2 3

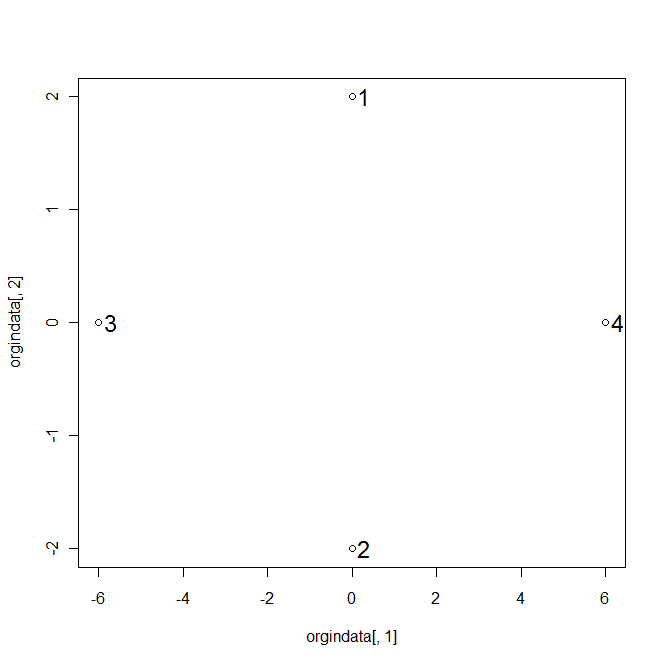
2 **4.00**

3 **6.32 6.32**

4 **6.32 6.32 12.00**

(c).

plot(orgindata[,1],orgindata[,2]);text(orgindata[,1]+0.3,orgindata[,2],1:4,cex=1.5)



(d).

> onedim<-orgindata[,1]

> onedim

**[1] 0 0 -6 6**

> dist(onedim)

1 2 3

2 **0**

3 **6 6**

4 **6 6 12**

(e).

> eigenvalue<-round(E$value,2)

> eigenvalue

[1] 72 8 0 -4

> anw<-eigenvalue^2

> anw

[1] 5184 64 0 16

> doimoni<-sum(eigenvalue^2)

> doimoni

[1] 5264

P1=

P2=

P3=

P4=

P\*1=

P\*2=

P\*3=

P\*4=

4.

5.

(a).

> class

col1 col2 col3 col4 col5 col6

[1,] **0 2 2** 2 4 **2**

[2,] 2 0 4 **2 4** 2

[3,] **2** 4 0 2 4 **4**

[4,] **2 2** 2 0 **2 2**

[5,] 4 **4** 4 **2** 0 2

[6,] 2 2 **4 2** 2 0

(b).

Load MVA

> course<-isoMDS(class,k=1)

initial value 29.026739

iter 5 value 22.428024

iter 5 value 22.414405

iter 5 value 22.407235

final value 22.407235

converged

> course

$points

[,1]

[1,] -0.5706295

[2,] -0.5402696

[3,] -2.0972740

[4,] 0.1657956

[5,] 1.9361563

[6,] 1.1062212

$stress

[1] 22.40724

> round(as.matrix(dist(course$points)),2)

1 2 3 4 5 6

1 0.00 0.03 1.53 0.74 2.51 1.68

2 0.03 0.00 **1.56** 0.71 2.48 1.65

3 1.53 1.56 0.00 2.26 4.03 3.20

4 0.74 0.71 **2.26** 0.00 1.77 0.94

5 2.51 2.48 4.03 1.77 0.00 0.83

**x=3; y=4; z=2**

**thus in the original data,** **dist(x,y)=2<dist(x,z)=4, while in the new one-dimensional coordinates, dist(x,z)=1.56<dist(x,y)=2.26.**

6.

(a).

> score

col1 col2 col3 col4 var5

[1,] 2 6 5 1 5

[2,] 10 6 9 8 5

[3,] 9 8 9 9 8

[4,] 6 6 6 4 5

[5,] 5 10 6 5 9

[6,] 5 5 4 4 4

[7,] 6 6 6 4 4

[8,] 4 4 4 4 4

[9,] 4 2 5 5 5

[10,] 5 5 5 5 5

[11,] 4 5 5 5 5

[12,] 4 4 6 9 10

[13,] 1 2 4 4 6

[14,] 8 5 4 5 1

[15,] 6 6 8 6 8

> S<-cov(score)

> S

col1 col2 col3 col4 var5

col1 5.780952 2.3333333 2.790476 3.0142857 -0.600000

col2 2.333333 4.0952381 1.809524 0.6428571 1.500000

col3 2.790476 1.8095238 2.923810 2.4142857 1.957143

col4 3.014286 0.6428571 2.414286 4.4571429 2.514286

var5 -0.600000 1.5000000 1.957143 2.5142857 5.257143

**> eigen(S)**

$values

[1] 12.05632688 6.05734880 3.58021775 0.75144699 0.06894529

$vectors

[,1] [,2] [,3] [,4] [,5]

[1,] -0.5425295 0.60023233 -0.08437479 0.041391824 0.5801297

[2,] -0.3662533 0.05097956 0.81418495 0.331742763 -0.3005149

[3,] -0.4455675 -0.02192816 0.01682708 -0.830842954 -0.3322734

[4,] -0.5025580 -0.12110741 -0.56922914 0.444829797 -0.4592092

[5,] -0.3470234 -0.78865352 0.07535780 0.007739115 0.5018582

(b).

> prcomp(score)$x

PC1 PC2 PC3 PC4 PC5

[1,] -4.17380146 -0.92884869 3.15162211 -1.17769523 0.22420277

[2,] 5.46661092 2.93754546 -1.44067183 -1.05612388 0.12672306

[3,] 7.20021628 -0.04779574 -0.07108287 0.03401697 0.26151695

[4,] -0.05044177 1.08683025 1.12326263 -0.50848150 -0.38641505

[5,] 2.76269367 -2.58520535 4.19657929 1.25288398 -0.15244968

[6,] -2.19738304 1.26812820 0.28444050 0.77233071 -0.26948874

[7,] -0.39746517 1.87548378 1.04790483 -0.51622062 0.11544318

[8,] -3.10616592 0.61691631 -0.44536966 0.39919612 0.01012610

[9,] -2.54352369 -0.41673191 -2.55078382 -0.64256345 -0.30127929

[10,] -0.90223413 0.33643910 -0.19260375 0.39405666 0.02013559

[11,] -1.44476367 -0.26379323 -0.10822897 0.35266484 0.60026528

[12,] 2.37989955 -4.76439822 -2.80571438 1.00809389 -0.04043061

[13,] -4.77221441 -2.86304686 -1.66989960 -0.37298665 0.14576899

[14,] -1.10830662 5.31367836 -0.76398640 1.31811863 -0.04509393

[15,] 2.88687945 -1.56520146 0.24453192 -1.25729047 -0.30902463

> round(cov(prcomp(score)$x),2)

PC1 PC2 PC3 PC4 PC5

PC1 **12.06** 0.00 0.00 0.00 0.00

PC2 0.00 **6.06** 0.00 0.00 0.00

PC3 0.00 0.00 **3.58** 0.00 0.00

PC4 0.00 0.00 0.00 **0.75** 0.00

PC5 0.00 0.00 0.00 0.00 **0.07**

(d).

> prcomp(score)

Standard deviations:

[1] 3.4722222 2.4611682 1.8921463 0.8668604 0.2625744

Rotation:

PC1 PC2 PC3 PC4 PC5

col1 0.5425295 0.60023233 -0.08437479 0.041391824 -0.5801297

col2 0.3662533 0.05097956 0.81418495 0.331742763 0.3005149

col3 0.4455675 -0.02192816 0.01682708 -0.830842954 0.3322734

col4 0.5025580 -0.12110741 -0.56922914 0.444829797 0.4592092

var5 0.3470234 -0.78865352 0.07535780 0.007739115 -0.5018582

> prcomp(score)$x

PC1 PC2 PC3 PC4 PC5

[1,] -4.17380146 -0.92884869 3.15162211 -1.17769523 0.22420277

[2,] 5.46661092 2.93754546 -1.44067183 -1.05612388 0.12672306

[3,] **7.20021628** -0.04779574 -0.07108287 0.03401697 0.26151695

[4,] -0.05044177 1.08683025 1.12326263 -0.50848150 -0.38641505

[5,] 2.76269367 -2.58520535 **4.19657929** 1.25288398 -0.15244968

[6,] -2.19738304 1.26812820 0.28444050 0.77233071 -0.26948874

[7,] -0.39746517 1.87548378 1.04790483 -0.51622062 0.11544318

[8,] -3.10616592 0.61691631 -0.44536966 0.39919612 0.01012610

[9,] -2.54352369 -0.41673191 -2.55078382 -0.64256345 -0.30127929

[10,] -0.90223413 0.33643910 -0.19260375 0.39405666 0.02013559

[11,] -1.44476367 -0.26379323 -0.10822897 0.35266484 0.60026528

[12,] 2.37989955 **-4.76439822** **-2.80571438** 1.00809389 -0.04043061

[13,] **-4.77221441** -2.86304686 -1.66989960 -0.37298665 0.14576899

[14,] -1.10830662 **5.31367836** -0.76398640 1.31811863 -0.04509393

[15,] 2.88687945 -1.56520146 0.24453192 -1.25729047 -0.30902463

**For the 1st column of the principal components, NO.3=7.2 is the highest and NO.13=-4.77 is the lowest, which means that the mean score of NO.3 student’s all tests scores is the best and the mean score of NO.13 student’s all tests scores is the worst.**

**For the 2nd column of the principal components, NO.14=5.31 is the highest and NO.12=-4.76 is the lowest, which means that the NO.14 student did the best in test C1 and worst in C5 while best while the NO.12 student did the worst in test C1 and the best in C5.**

**For the 3rd column of the principal components, NO.5=4.20 is the highest and NO.12=-2.81 is the lowest, which means that the NO.5 student did the best in test C2 and worst in C4 while best while the NO.12 student did the worst in test C2 and the best in C4.**

(e).

> cov(score)

col1 col2 col3 col4 var5

col1 5.780952 2.3333333 2.790476 3.0142857 -0.600000

col2 2.333333 4.0952381 1.809524 0.6428571 1.500000

col3 2.790476 1.8095238 2.923810 2.4142857 1.957143

col4 3.014286 0.6428571 2.414286 4.4571429 2.514286

var5 -0.600000 1.5000000 1.957143 2.5142857 5.257143

> cor(score)

col1 col2 col3 col4 var5

col1 1.0000000 0.4795540 0.6787410 0.5938222 -0.1088370

col2 0.4795540 1.0000000 0.5229380 0.1504689 0.3232785

col3 0.6787410 0.5229380 1.0000000 0.6687840 0.4991983

col4 0.5938222 0.1504689 0.6687840 1.0000000 0.5194114

var5 -0.1088370 0.3232785 0.4991983 0.5194114 1.0000000

**> mahalanobis(score,center=colMeans(score,),cov=cov(score))**

**[1] 6.9365131 6.2002404 5.2953659 3.0574278 9.0814751 2.5357343 1.4484349**

**[8] 1.1320539 4.2486212 0.3090895 5.5795527 7.7920808 4.5144189 7.2678364**

**[15] 4.6011552**

7.

(a).

> dist(score)

1 2 3 4 5 6 7

2 11.357817

3 11.916375 3.872983

4 5.099020 6.403124 7.483315

5 7.615773 8.660254 6.782330 5.830952

6 4.582576 8.246211 9.539392 2.645751 7.416198

7 5.196152 6.480741 7.937254 1.000000 6.557439 2.449490

8 4.358899 9.055385 10.344080 3.605551 8.185353 1.414214 3.464102

9 6.000000 8.774964 10.099505 4.690416 9.055385 3.605551 4.795832

10 5.099020 7.141428 8.124038 2.000000 6.480741 1.732051 2.236068

11 4.582576 7.874008 8.660254 2.645751 6.557439 2.000000 2.828427

12 9.899495 8.660254 7.348469 7.615773 7.348469 8.185353 8.306624

13 5.291503 11.789826 12.409674 6.782330 9.695360 5.385165 7.000000

14 8.366600 7.416198 10.000000 5.099020 10.099505 4.358899 4.358899

15 7.681146 5.477226 4.795832 4.123106 4.795832 6.164414 4.898979

8 9 10 11 12 13 14

2

3

4

5

6

7

8

9 2.645751

10 2.236068 3.162278

11 2.000000 3.000000 1.000000

12 8.062258 6.782330 6.633250 6.557439

13 4.123106 3.464102 5.291503 4.582576 7.615773

14 5.196152 6.480741 5.099020 5.744563 10.862780 9.165151

15 6.633250 6.244998 4.582576 4.898979 5.000000 8.062258 8.426150

(b).

> isoMDS(dist(score),k=2)

initial value 13.368977

iter 5 value 9.733347

final value 9.647658

converged

**$points**

**[,1] [,2]**

**[1,] 5.31706574 -0.6819479**

**[2,] -5.18580593 -3.3061428**

**[3,] -6.98539895 -0.2610055**

**[4,] 0.02268883 -0.9775235**

**[5,] -3.97848827 3.0198749**

**[6,] 1.82197053 -1.1466689**

**[7,] 0.37686275 -1.4924126**

**[8,] 2.69851897 -0.3040488**

**[9,] 2.66479200 1.4149747**

**[10,] 0.70803835 -0.2666745**

**[11,] 1.50235964 0.1157272**

**[12,] -1.94544449 5.5619799**

**[13,] 4.57917468 2.9865069**

**[14,] 1.22928370 -5.2207227**

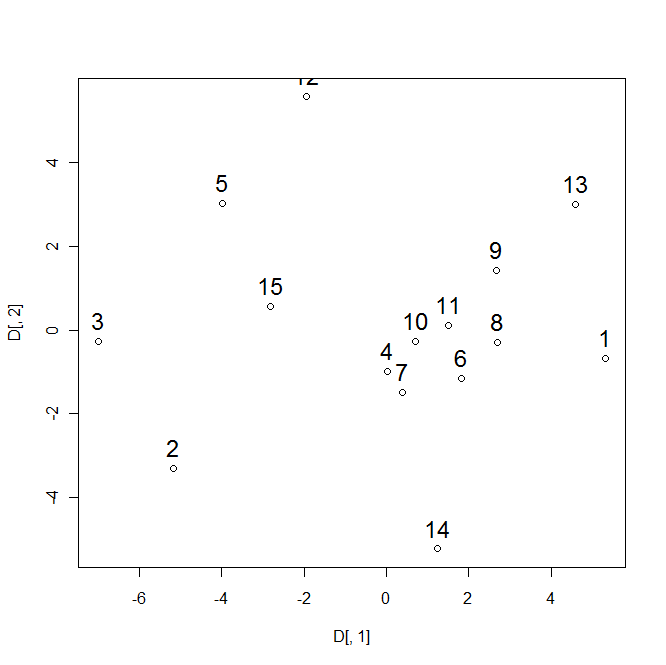
**[15,] -2.82561757 0.5580837**

$stress

[1] 9.647658

> D<-D$points

>plot(D);text(D,labels=1:15,cex=1.5)> plot(D[,1],D[,2]);text(D[,1],D[,2]+0.5,1:15,cex=1.5)



(c).

> S<-Shepard(as.dist(dist(score)),D)

> S

$x

[1] 1.000000 1.000000 1.414214 1.732051 2.000000 2.000000 2.000000

[8] 2.236068 2.236068 2.449490 2.645751 2.645751 2.645751 2.828427

[15] 3.000000 3.162278 3.464102 3.464102 3.605551 3.605551 3.872983

[22] 4.123106 4.123106 4.358899 4.358899 4.358899 4.582576 4.582576

[29] 4.582576 4.582576 4.690416 4.795832 4.795832 4.795832 4.898979

[36] 4.898979 5.000000 5.099020 5.099020 5.099020 5.099020 5.196152

[43] 5.196152 5.291503 5.291503 5.385165 5.477226 5.744563 5.830952

[50] 6.000000 6.164414 6.244998 6.403124 6.480741 6.480741 6.480741

[57] 6.557439 6.557439 6.557439 6.633250 6.633250 6.782330 6.782330

[64] 6.782330 7.000000 7.141428 7.348469 7.348469 7.416198 7.416198

[71] 7.483315 7.615773 7.615773 7.615773 7.681146 7.874008 7.937254

[78] 8.062258 8.062258 8.124038 8.185353 8.185353 8.246211 8.306624

[85] 8.366600 8.426150 8.660254 8.660254 8.660254 8.774964 9.055385

[92] 9.055385 9.165151 9.539392 9.695360 9.899495 10.000000 10.099505

[99] 10.099505 10.344080 10.862780 11.357817 11.789826 11.916375 12.409674

$y

[1] 0.6249399 0.8815766 1.2158725 1.4195898 0.9874261 1.3022270

[7] 1.2676786 1.2696894 1.9908315 1.4858920 1.8072147 1.8397345

[13] 1.7193543 1.9628696 1.7433568 2.5800831 2.6081212 2.4768073

[19] 2.7592816 2.6967325 3.5371452 3.2358830 3.7900689 2.6456748

[25] 4.1169396 3.8245154 3.5258554 3.8972129 3.6286292 4.2081072

[31] 3.5643733 4.2396566 2.7183684 3.6996650 3.8026852 4.3505248

[37] 5.0807167 5.3026212 4.6276977 4.4114182 4.9813944 5.0062420

[43] 5.1315041 3.7419305 5.0565685 4.9684320 4.5279946 5.3434323

[49] 5.6558476 3.3810709 4.9503794 5.5568750 5.7053383 5.8508888

[55] 5.7240666 6.7891946 6.2713492 6.2027226 6.4458531 5.5910068

[61] 6.4042317 4.4503582 6.0394619 6.2009623 6.1416728 6.6314228

[67] 7.7011882 3.2550829 6.6946987 7.1418072 7.0446215 10.0055393

[73] 6.8292499 7.0145361 8.2365630 7.5127061 7.4645335 7.4817571

[79] 7.7928292 7.6934394 7.4586121 7.6941137 7.3329570 7.4268139

[85] 6.1082271 7.0595203 6.4401953 9.4415858 8.4961152 9.1608317

[91] 8.4365365 6.8343893 8.8645579 8.8517884 8.5577280 9.5776140

[97] 9.5958223 9.7946462 9.7482479 9.6840136 11.2403547 10.8257430

[103] 11.6168966 12.3096641 12.0118983

$yf

[1] 0.6249399 0.8815766 1.2076295 1.2076295 1.2076295 1.2798650

[7] 1.2798650 1.2798650 1.7383618 1.7383618 1.7887678 1.7887678

[13] 1.7887678 1.8531132 1.8531132 2.5550039 2.5550039 2.5550039

[19] 2.7280071 2.7280071 3.3021930 3.3021930 3.3021930 3.3021930

[25] 3.7423323 3.7423323 3.7423323 3.7423323 3.7423323 3.7423323

[31] 3.7423323 3.7423323 3.7423323 3.7423323 3.8026852 4.3505248

[37] 4.7854406 4.7854406 4.7854406 4.7854406 4.7854406 4.7854406

[43] 4.7854406 4.7854406 4.8222243 4.8222243 4.8222243 4.8222243

[49] 4.8222243 4.8222243 4.9503794 5.5568750 5.7053383 5.7874777

[55] 5.7874777 6.0095775 6.0095775 6.0095775 6.0095775 6.0095775

[61] 6.0095775 6.0095775 6.0095775 6.0095775 6.0095775 6.0095775

[67] 6.0095775 6.0095775 6.6946987 7.0932144 7.0932144 7.4719746

[73] 7.4719746 7.4719746 7.4719746 7.4719746 7.4719746 7.4719746

[79] 7.4719746 7.4719746 7.4719746 7.4719746 7.4719746 7.4719746

[85] 7.4719746 7.4719746 7.4719746 8.4738917 8.4738917 8.4738917

[91] 8.4738917 8.4738917 8.7580248 8.7580248 8.7580248 9.5776140

[97] 9.5958223 9.7423026 9.7423026 9.7423026 11.0330489 11.0330489

[103] 11.6168966 12.1607812 12.1607812

> stress<-sqrt((sum((S$y-S$yf)^2)/(sum(S$y^2))))

**> stress**

**[1] 0.09647658**

(d).

> drawst

function(D,Mst)

{

st<-mst(D);

plot(Mst,type="n",xlab="x",ylab="y")

text(Mst,labels=1:dim(Mst)[1])

for(i in 1:nrow(D)){

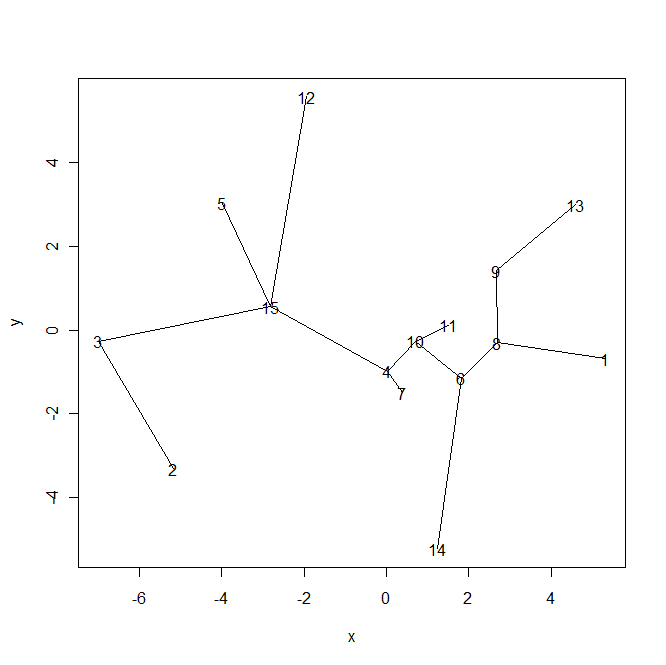
w1<-which(st[i,]==1)

segments(Mst[i,1],Mst[i,2],Mst[w1,1],Mst[w1,2])

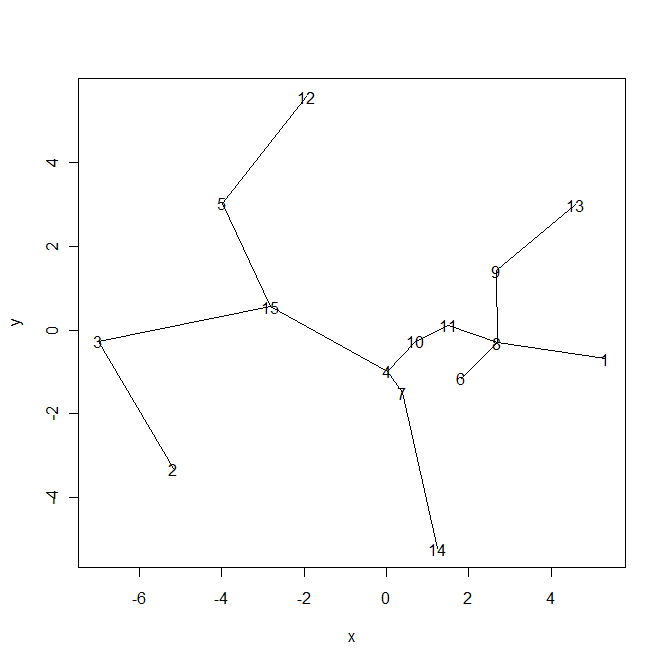
}

}

> drawst(as.matrix(dist(score)),D)



> drawst(as.matrix(dist(D)),D)



(f).

Differences: In the first figure, point 12 and point 15 are connected and point 10 and point 6 are connected, while in the second figure, point 12 and point 5 are connected and point 11 and point 8 are connected, this is because in the 2nd figure, we use the correct distance for the new coordinates.

8.

(a).

> scoremds<-cmdscale(as.matrix(dist(score)),k=2,eig=TRUE)

> scoremds

**$points**

**[,1] [,2]**

**1 4.17380146 0.92884869**

**2 -5.46661092 -2.93754546**

**3 -7.20021628 0.04779574**

**4 0.05044177 -1.08683025**

**5 -2.76269367 2.58520535**

**6 2.19738304 -1.26812820**

**7 0.39746517 -1.87548378**

**8 3.10616592 -0.61691631**

**9 2.54352369 0.41673191**

**10 0.90223413 -0.33643910**

**11 1.44476367 0.26379323**

**12 -2.37989955 4.76439822**

**13 4.77221441 2.86304686**

**14 1.10830662 -5.31367836**

**15 -2.88687945 1.56520146**

$eig

[1] 1.687886e+02 8.480288e+01 5.012305e+01 1.052026e+01 9.652341e-01

[6] 2.981874e-15 2.614533e-15 2.528529e-16 -9.183917e-17 -5.120270e-16

[11] -7.412470e-16 -2.120389e-15 -2.888758e-15 -5.625694e-15 -6.470994e-15

$x

NULL

$ac

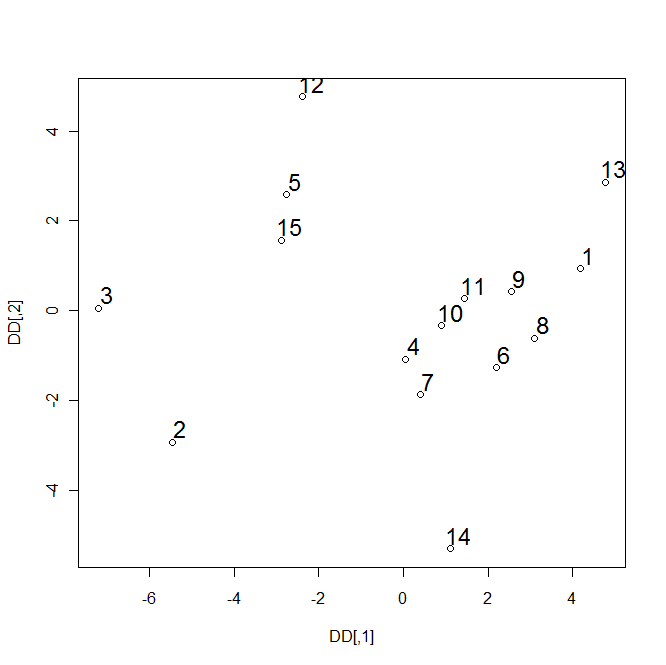
[1] 0

$GOF

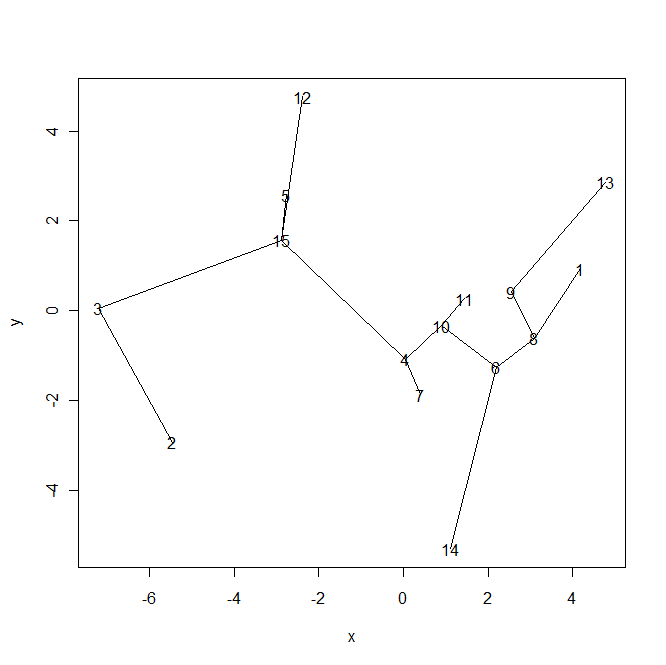
[1] 0.8045414 0.8045414

(b).

> plot(DD);text(DD[,1]+0.2,DD[,2]+0.3,1:15,cex=1.5)

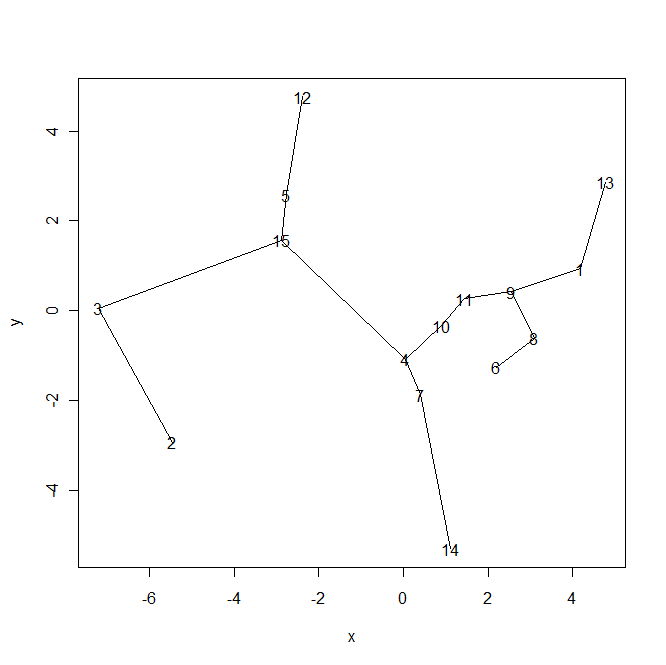


> drawst(as.matrix(dist(score)),DD)



(c).

> drawst(as.matrix(dist(DD)),DD)

Differences: In the first figure, point 12 and point 15 are connected, point 14 and point 6 are connected and point 10 and point 6 are connected, while in the second figure, point 12 and point 5 are connected, point 14 and point 7 are connected and point 11 and point 9 are connected, this is because in the 2nd figure, we use the correct distance for the new coordinates.