STAT 488: Multivariate Statistical Analysis — Homework 4

Charles Hwang

4/10/2022

Problem 8.1

$|\mathbf{A}-\lambda\mathbf{I}| = 0$

$(5-\lambda)(2-\lambda)-(2)(2) = 0$

$\lambda^2 - 7\lambda + 6 = 0$

$(\lambda - 1)(\lambda - 6) = 0$

$\lambda = 1, 6$

$\mathbf{Ax} = \lambda\mathbf{x}$

$5x_1 + 2x_2 = 1x_1$

$2x_1 + 2x_2 = 1x_2$

$-2x_1 = x_2$

We can see there are an infinite number of solutions. If we arbitrarily pick $x_1 = 1$ and $x_2 = -2$, we can see that $\mathbf{e}' = [\frac{1}{\sqrt{5}}, \frac{-2}{\sqrt{5}}] = [\ 0.4472136, \text{-}0.8944272]$.

$5x_1 + 2x_2 = 6x_1$

$2x_1 + 2x_2 = 6x_2$

$x_1 = 2x_2$

We can see there are an infinite number of solutions. If we arbitrarily pick $x_1 = 2$ and $x_2 = 1$, we can see that $\mathbf{e}' = [\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}] = [\ 0.8944272, 0.4472136]$.

We can see the population principal components are $Y_1 = e_1'\mathbf{X} = \frac{1}{\sqrt{5}}X_1 - \frac{2}{\sqrt{5}}X_2$ and $Y_2 = e_2'\mathbf{X} = \frac{2}{\sqrt{5}}X_1 + \frac{1}{\sqrt{5}}X_2$.

The proportion of the total population variance explained by the first principal component is $\frac{6}{6+1} = \frac{6}{7}$.

Problem 8.2

Problem 8.2(a)

```
rm(list=ls())
p12<-2/sqrt(5)*sqrt(1)/sqrt(matrix(c(5,2,2,2),nrow=2)[2,2]) # (8-8), page 433
p<-matrix(c(1,p12,p12,1),nrow=2)
1/sqrt(dim(p)[1]) # (8-17), page 440
```

```
## [1] 0.7071068
```

```
(p[1,1]+p12)/(p[1,1]+p[2,2])
```

```
## [1] 0.8162278
```

We can see from (8-15) on page 440 that the population principal components are $Y_1 = \frac{\sqrt{2}}{2}X_1 + \frac{\sqrt{2}}{2}X_2$ and $Y_2 = \frac{\sqrt{2}}{2}X_1 - \frac{\sqrt{2}}{2}X_2$. The proportion of the total population variance explained by the first principal component is 0.8162278.

Problem 8.2(b)

We can see the two pairs of principal components are different. They should not be the same because they are derived from different matrices and have different variances.

Problem 8.2(c)

```
1/sqrt(dim(p)[1])*sqrt(1+(dim(p)[1]-1)*sum(p[1,2])) # Result 8.4, page 437
```

```
## [1] 0.9034532
```

```
1/sqrt(dim(p)[1])*sqrt(1+(dim(p)[1]-1)*sum(p[1,2]))
```

```
## [1] 0.9034532
```

```
1/sqrt(dim(p)[1])*sqrt(diff(p[2,]))
```

```
## [1] 0.4286866
```

Problem 8.10

```
sp<-read.table("/Users/newuser/Desktop/Notes/Graduate/STAT 488 - Multivariate Statistical Analysis/T8-4
round(cov(sp),10) # Disabling scientific notation  # Problem 8.10(a)
```

```
##              V1           V2           V3           V4           V5
## V1 0.0004332695 0.0002756679 0.0001590265 0.0000641193 0.0000889662
## V2 0.0002756679 0.0004387172 0.0001799737 0.0001814512 0.0001232623
## V3 0.0001590265 0.0001799737 0.0002239722 0.0000734135 0.0000605461
## V4 0.0000641193 0.0001814512 0.0000734135 0.0007224964 0.0005082772
## V5 0.0000889662 0.0001232623 0.0000605461 0.0005082772 0.0007656742
```

```
colMeans(sp)
```

```
##           V1           V2           V3           V4           V5
## 0.0010627806 0.0006554204 0.0016260816 0.0040491252 0.0040386417
```

```
summary(prcomp(sp))
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5
## Standard deviation     0.03698 0.02648 0.01593 0.01194 0.01090
## Proportion of Variance 0.52926 0.27133 0.09822 0.05518 0.04601
## Cumulative Proportion  0.52926 0.80059 0.89881 0.95399 1.00000
```

```
prcomp(sp)
```

```
## Standard deviations (1, .., p=5):
## [1] 0.03698213 0.02647942 0.01593118 0.01194163 0.01090352
##
## Rotation (n x k) = (5 x 5):
##           PC1        PC2         PC3        PC4         PC5
## V1 -0.2228228  0.6252260 -0.32611218  0.6627590 -0.11765952
## V2 -0.3072900  0.5703900  0.24959014 -0.4140935  0.58860803
## V3 -0.1548103  0.3445049  0.03763929 -0.4970499 -0.78030428
```

```
## V4 -0.6389680 -0.2479475  0.64249741  0.3088689 -0.14845546
## V5 -0.6509044 -0.3218478 -0.64586064 -0.2163758  0.09371777
```

```
sum(prcomp(sp)$sdev[1:3]^2)/sum(prcomp(sp)$sdev^2) # Problem 8.10(b)
```

```
## [1] 0.8988095
```

```
prcomp(sp)$rotation[,c("PC1","PC2","PC3")]
```

```
##           PC1        PC2         PC3
## V1 -0.2228228  0.6252260 -0.32611218
## V2 -0.3072900  0.5703900  0.24959014
## V3 -0.1548103  0.3445049  0.03763929
## V4 -0.6389680 -0.2479475  0.64249741
## V5 -0.6509044 -0.3218478 -0.64586064
```

We can see approximately 89.8809485 percent of the total sample variance is explained by the first three principal components. We can see from the signs of the first component that it is a weighted sum of the five stocks (or as the textbook calls it, a "market component"). The signs of the second component indicate it compares banking stocks (JPMorgan, Citibank, Wells Fargo) to oil stocks (Royal Dutch Shell, Exxon-Mobil) which the textbook calls an "industry component". The third component is not easily interpretable, but it compares JPMorgan and Exxon-Mobil with the trio of Citibank, Wells Fargo, and Royal Dutch Shell. As the stock market is a highly-complex multi-dimensional system, there may be some relationship within these two groups. (See Example 8.5 (page 452) for more information.)

Problem 8.16

Problem 8.16(a)

The x_1, x_2, x_3, and x_4 variables have a weak positive correlation with one another. It does not appear the Walleye fish (x_5) groups with the other fish.

Problem 8.16(b-c)

```
# Setting both [1,3] and [3,1] = 0.2636 as they should be the same value but are
# listed differently in the textbook (page 475), likely due to a rounding error
R<-matrix(c(1,.4919,.2636,.4653,-.2277,.0652,.4919,1,.3127,.3506,-.1917,.2045,.2636,.3127,1,.4108,.0647
eigen(R[1:4,1:4]) # (8-28), page 450
```

```
## eigen() decomposition
## $values
## [1] 2.1539422 0.7875151 0.6156498 0.4428929
##
## $vectors
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.5265283  0.4571532  0.2491871  0.6720749
## [2,] -0.5032995  0.4120178 -0.6142318 -0.4468223
## [3,] -0.4428007 -0.7583919 -0.3680759  0.3054332
## [4,] -0.5228624 -0.2146951  0.6520316 -0.5053471
```

```
# We can see from the signs of the principal components that the first component is a
# weighted sum of bluegill (x1), black crappie (x2), and smallmouth (x3) and largemouth
# bass (x4). We can also see the second, third, and fourth principal components are
# comparing two pairs of fish with each other for the three possible combinations of pairs.
eigen(R) # Problem 8.16(c)
```

```
## eigen() decomposition
```

```
## $values
## [1] 2.3549437 1.0718555 0.9842359 0.6643850 0.5003684 0.4242116
##
## $vectors
##            [,1]        [,2]        [,3]        [,4]       [,5]        [,6]
## [1,] -0.4753480  0.02188533  0.47986753 -0.04564813 -0.3579790  0.64269727
## [2,] -0.4719275 -0.01924481  0.20903356 -0.70296427  0.1771671 -0.45571057
## [3,] -0.3931799 -0.56061449 -0.26453976  0.17550876  0.5973982  0.27101971
## [4,] -0.4963496 -0.07723144  0.03221512  0.60427412 -0.3238003 -0.52596650
## [5,]  0.2563101 -0.80502279  0.01266373 -0.21815808 -0.4823081 -0.07660978
## [6,] -0.2909991  0.17560090 -0.80922967 -0.24538541 -0.3822441  0.15266496
```

We can see the first component compares the Walleye fish (x_5) with the other fish. It is not a weighted sum as the first component usually is likely due to its negative correlation with other fish that we saw in Problem 8.16(a). The second, third, fourth, and fifth components compare pairs of fish (x_1/x_6, x_3/x_6, x_3/x_4, and x_2/x_3) with their four-fish complements. The sixth component compares the trio of x_1, x_3, and x_6 with the trio of x_2, x_4, and x_5.

Problem 8.18

```
ntr<-read.table("/Users/newuser/Desktop/Notes/Graduate/STAT 488 - Multivariate Statistical Analysis/T1-S
row.names(ntr)<-ntr$V1
ntr<-ntr[,c("V2","V3","V4","V5","V6","V7","V8")]
names(ntr)<-c("100m","200m","400m","800m","1500m","3000m","Marathon")
cor(ntr)                                    # Problem 8.18(a)
```

```
##               100m      200m      400m      800m     1500m     3000m  Marathon
## 100m     1.0000000 0.9410886 0.8707802 0.8091758 0.7815510 0.7278784 0.6689597
## 200m     0.9410886 1.0000000 0.9088096 0.8198258 0.8013282 0.7318546 0.6799537
## 400m     0.8707802 0.9088096 1.0000000 0.8057904 0.7197996 0.6737991 0.6769384
## 800m     0.8091758 0.8198258 0.8057904 1.0000000 0.9050509 0.8665732 0.8539900
## 1500m    0.7815510 0.8013282 0.7197996 0.9050509 1.0000000 0.9733801 0.7905565
## 3000m    0.7278784 0.7318546 0.6737991 0.8665732 0.9733801 1.0000000 0.7987302
## Marathon 0.6689597 0.6799537 0.6769384 0.8539900 0.7905565 0.7987302 1.0000000
```

```
eigen(cor(ntr))
```

```
## eigen() decomposition
## $values
## [1] 5.80762446 0.62869342 0.27933457 0.12455472 0.09097174 0.05451882 0.01430226
##
## $vectors
##            [,1]       [,2]       [,3]        [,4]        [,5]        [,6]
## [1,] -0.3777657 -0.4071756 -0.1405803  0.58706293 -0.16706891  0.53969730
## [2,] -0.3832103 -0.4136291 -0.1007833  0.19407501  0.09350016 -0.74493139
## [3,] -0.3680361 -0.4593531  0.2370255 -0.64543118  0.32727328  0.24009405
## [4,] -0.3947810  0.1612459  0.1475424 -0.29520804 -0.81905467 -0.01650651
## [5,] -0.3892610  0.3090877 -0.4219855 -0.06669044  0.02613100 -0.18898771
## [6,] -0.3760945  0.4231899 -0.4060627 -0.08015699  0.35169796  0.24049968
## [7,] -0.3552031  0.3892153  0.7410610  0.32107640  0.24700821 -0.04826992
##             [,7]
## [1,]  0.08893934
## [2,] -0.26565662
## [3,]  0.12660435
## [4,] -0.19521315
```

```
## [5,]  0.73076817
## [6,] -0.57150644
## [7,]  0.08208401
```

```
round(t(eigen(cor(ntr))$vectors[,1:2]),6) # Problem 8.18(b)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] -0.377766 -0.383210 -0.368036 -0.394781 -0.389261 -0.376094 -0.355203
## [2,] -0.407176 -0.413629 -0.459353  0.161246  0.309088  0.423190  0.389215
```

```
r1<-eigen(cor(ntr))$vectors[,1]*sqrt(eigen(cor(ntr))$values[1]) # (8-29), page 451
r2<-eigen(cor(ntr))$vectors[,2]*sqrt(eigen(cor(ntr))$values[2])
ry<-as.data.frame(rbind(round(r1,6),round(r2,6)))
row.names(ry)<-c("r_y1zi","r_y2zi")
names(ry)<-names(ntr)
ry
```

```
##             100m      200m      400m      800m     1500m     3000m  Marathon
## r_y1zi -0.910378 -0.923499 -0.886931 -0.951383 -0.938080 -0.906351 -0.856004
## r_y2zi -0.322850 -0.327967 -0.364222  0.127852  0.245076  0.335548  0.308610
```

```
sum(eigen(cor(ntr))$values[1:2])/sum(eigen(cor(ntr))$values)
```

```
## [1] 0.919474
```

Problem 8.18(c)

We can see from the signs of the first component that it is a weighted sum of all distances with each distance contributing about the same amount. We can see from the signs of the second component that it separates the times of the 100m, 200m, and 400m races from the times of the 800m, 1500m, 3000m, and marathon races. This distinction could make this a distance component.

Problem 8.18(d)

```
p1<-as.matrix(ntr)%*%eigen(cor(ntr))$vectors[,1] # Eigenvector is negative
rank<-as.data.frame(p1[order(p1,decreasing=TRUE),])
head(rank)
```

```
##     p1[order(p1, decreasing = TRUE), ]
## GBR                          -84.35454
## CHN                          -85.65724
## GER                          -85.66515
## USA                          -85.72607
## RUS                          -86.12949
## KEN                          -86.98079
```

Yes, these rankings appear to correspond with the intuitive notion of athletic performance for these 54 countries.