HW6

Keith Mitchell

3/7/2020

Question 1:

a) Give the formula for the first and second principle components

```
X <- as.matrix(cbind(c(2,2,2,0,-1,-2,-3), c(2,4,6,0,-4,-4,-4)))
X
```

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    2    4
## [3,]    2    6
## [4,]    0    0
## [5,]   -1   -4
## [6,]   -2   -4
## [7,]   -3   -4
```

```
help <- (1/6)*t(X)%*%X
help
```

```
##          [,1]     [,2]
## [1,] 4.333333  8.00000
## [2,] 8.000000 17.33333
```

```
cov <- cov(X)
cov
```

```
##          [,1]     [,2]
## [1,] 4.333333  8.00000
## [2,] 8.000000 17.33333
```

```
ev <- eigen(cov)
ev$values
```

```
## [1] 21.1410974  0.5255693
```

```
ev$vectors
```

```
##           [,1]       [,2]
## [1,] 0.4297717 -0.9029376
## [2,] 0.9029376  0.4297717
```

So the first principle component is the eigen vector corresponding to the largest eigenvalue which is the first column above

```
ev$vectors[,1]
```

```
## [1] 0.4297717 0.9029376
```

So the second principle component is the eigen vector corresponding to the second largest eigenvalue which is the second column above

```r
ev$vectors[,2]
```

```
## [1] -0.9029376  0.4297717
```

b) Determine the proportion of total sample variance due to the first sample principal component.

```r
ev$values[1]/sum(ev$values)
```

```
## [1] 0.975743
```

c) Compare the contributions of the two variates to the determination of the first sample principal component based on loadings

- here we see that the first value of the eigen vector is the contribution of the first variable towards the principle component while the second value is the contribution of the second variable to the principle component

```r
ev$vectors[,1]
```

```
## [1] 0.4297717 0.9029376
```

d) Compare the contributions of the two variates to the determination of the first sample principal component based on sample correlations

```r
corr_1 <- ev$vectors[,1][1]*sqrt(ev$values[1]/cov[,1][1])
corr_1
```

```
## [1] 0.9492716
```

```r
corr_2 <- ev$vectors[,1][2]*sqrt(ev$values[1]/cov[,2][2])
corr_2
```

```
## [1] 0.9971958
```

e) Redo (a)-(d) on the standardized dataset.

a) Give the formula for the first and second principle components

```r
X_stan <- scale(X)
X_stan[is.nan(X_stan)] <- 0
X_stan
```

```
##            [,1]       [,2]
## [1,]  0.9607689  0.4803845
## [2,]  0.9607689  0.9607689
## [3,]  0.9607689  1.4411534
## [4,]  0.0000000  0.0000000
## [5,] -0.4803845 -0.9607689
## [6,] -0.9607689 -0.9607689
## [7,] -1.4411534 -0.9607689
```

```
## attr(,"scaled:center")
## [1] 0 0
## attr(,"scaled:scale")
## [1] 2.081666 4.163332
```

```r
cov <- cov(X_stan)
cov
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.9230769
## [2,] 0.9230769 1.0000000
```

```r
ev <- eigen(cov)
ev$values
```

```
## [1] 1.92307692 0.07692308
```

```r
ev$vectors
```

```
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

So the first principle component is the eigen vector corresponding to the largest eigenvalue which is the first column above

```r
ev$vectors[,1]
```

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

So the second principle component is the eigen vector corresponding to the second largest eigenvalue which is the second column above

```r
ev$vectors[,2]
```

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

b) Determine the proportion of total sample variance due to the first sample principal component.

```r
ev$values[1]/sum(ev$values)
```

```
## [1] 0.9615385
```

c) Compare the contributions of the two variates to the determination of the first sample principal component based on loadings

- here we see that the first value of the eigen vector is the contribution of the first variable towards the principle component while the second value is the contribution of the second variable to the principle component

```r
ev$vectors[,1]
```

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

d) Compare the contributions of the two variates to the determination of the first sample principal component based on sample correlations

```r
corr_1 <- ev$vectors[,1][1]*sqrt(ev$values[1]/cov[,1][1])
corr_1
```

```
## [1] 0.9805807
```

```r
corr_2 <- ev$vectors[,1][2]*sqrt(ev$values[1]/cov[,2][2])
corr_2
```

```
## [1] 0.9805807
```

Question 3:

Consider the air polution in table 1.5. Summarize the data in fewer the p=7 dimensions if possible. Conduct a PCA of the data using both the covariance matrix S and the correlation matrix R. What have you learned? Does it make any difference which matrix is chose for analysis? Can the data be summarized in thre or fewer dimensions? Can you interpret the principal components?

```r
data <- read.table("T1-5.DAT",
          header=FALSE)
data <- as.matrix(data)
```

```r
cov <- cov(data)
cor <- cor(data)

cov
```

```
##            V1          V2         V3         V4         V5         V6        V7
## V1  2.5000000  -2.7804878 -0.3780488 -0.4634146 -0.5853659 -2.2317073 0.1707317
## V2 -2.7804878 300.5156794  3.9094077 -1.3867596  6.7630662 30.7909408 0.6236934
## V3 -0.3780488   3.9094077  1.5220674  0.6736353  2.3147503  2.8217189 0.1416957
## V4 -0.4634146  -1.3867596  0.6736353  1.1823461  1.0882695 -0.8106852 0.1765389
## V5 -0.5853659   6.7630662  2.3147503  1.0882695 11.3635308  3.1265970 1.0441347
## V6 -2.2317073  30.7909408  2.8217189 -0.8106852  3.1265970 30.9785134 0.5946574
## V7  0.1707317   0.6236934  0.1416957  0.1765389  1.0441347  0.5946574 0.4785134
```

```r
cor
```

```
##            V1          V2         V3          V4         V5         V6
## V1  1.0000000 -0.10144191 -0.1938032 -0.26954261 -0.1098249 -0.2535928
## V2 -0.1014419  1.00000000  0.1827934 -0.07356907  0.1157320  0.3191237
## V3 -0.1938032  0.18279338  1.0000000  0.50215246  0.5565838  0.4109288
## V4 -0.2695426 -0.07356907  0.5021525  1.00000000  0.2968981 -0.1339521
## V5 -0.1098249  0.11573199  0.5565838  0.29689814  1.0000000  0.1666422
## V6 -0.2535928  0.31912373  0.4109288 -0.13395214  0.1666422  1.0000000
## V7  0.1560979  0.05201044  0.1660323  0.23470432  0.4477678  0.1544506
##            V7
## V1 0.15609793
## V2 0.05201044
## V3 0.16603235
## V4 0.23470432
## V5 0.44776780
## V6 0.15445056
## V7 1.00000000
```

The eigenvalues and vectors based on the covariance matrix are:

```
ev <- eigen(cov)
ev$values
```

```
## [1] 304.2578640  28.2761046  11.4644830   2.5243296   1.2795247   0.5287288
## [7]   0.2096157
```

```
ev$vectors
```

```
##              [,1]        [,2]        [,3]          [,4]          [,5]
## [1,]  0.010039244  0.07622439  0.03087761  0.9203045748  0.3423859285
## [2,] -0.993199405  0.11615518  0.00659069 -0.0002118679  0.0022391022
## [3,] -0.014062314 -0.09956775 -0.18282641 -0.1382922410  0.6500776063
## [4,]  0.004710175  0.01320423 -0.13021553 -0.3277842624  0.6431560485
## [5,] -0.024255644 -0.15038113 -0.95526318  0.1023719020 -0.2065840405
## [6,] -0.112429558 -0.97335904  0.16981025  0.0632480276 -0.0002935726
## [7,] -0.002340785 -0.02382046 -0.08519558  0.1095073458  0.0619613872
##              [,6]         [,7]
## [1,]  0.011779079 -0.169729925
## [2,]  0.003353218 -0.001781987
## [3,] -0.563893916  0.443577538
## [4,]  0.497513370 -0.462855916
## [5,] -0.009009299 -0.105029951
## [6,]  0.051067254 -0.066992404
## [7,]  0.657012233  0.738019426
```

Then to calculate the proportion of variance in the first two pcs

```
(sum(ev$values[1:2]))/sum(ev$values)
```

```
## [1] 0.9540751
```

So the first and second principal components summarize 95.4% of the variation in the data based on the covariance matrix.

In comparison, the eigenvalues and vectors based on the correlation matrix are:

```
ev <- eigen(cor)
sum(ev$values)
```

```
## [1] 7
```

```
ev$values
```

```
## [1] 2.3367826 1.3860007 1.2040659 0.7270865 0.6534765 0.5366888 0.1558989
```

```
ev$vectors
```

```
##            [,1]         [,2]       [,3]         [,4]        [,5]         [,6]
## [1,]  0.2368211  0.278445138  0.6434744  0.172719491  0.56053441 -0.223579220
## [2,] -0.2055665 -0.526613869  0.2244690  0.778136601 -0.15613432 -0.005700851
## [3,] -0.5510839 -0.006819502 -0.1136089  0.005301798  0.57342221 -0.109538907
## [4,] -0.3776151  0.434674253 -0.4070978  0.290503052 -0.05669070 -0.450234781
## [5,] -0.4980161  0.199767367  0.1965567 -0.042428178  0.05021430  0.744968707
## [6,] -0.3245506 -0.566973655  0.1598465 -0.507915905  0.08024349 -0.330583071
## [7,] -0.3194032  0.307882771  0.5410484 -0.143082348 -0.56607057 -0.266469812
```

```
##             [,7]
## [1,] -0.24146701
## [2,] -0.01126548
## [3,]  0.58524622
## [4,] -0.46088973
## [5,] -0.33784371
## [6,] -0.41707805
## [7,]  0.31391372
```

Then to calculate the proportion of variance in the first two pcs and 3 pcs

```r
(sum(ev$values[1:2]))/sum(ev$values)
```

```
## [1] 0.5318262
```

```r
(sum(ev$values[1:3]))/sum(ev$values)
```

```
## [1] 0.7038356
```

So the first and second principal components summarize 53.2% of the variation in the data based on the corrlation matrix. The first 3 pcs are ~70%.

Based on these results the choice of the covariance vs the correlation matrix makes a difference. The correlation matrix can be summarized in the first 3 principal components where as the covariance matrix can be summarized in the first two principla components fairly effectively. The data can be summarized in 3 or fewer dimensions, but it is not very effective in comparison.

Question 7

```r
data <- read.table("T1-6.DAT",
           header=FALSE, colClasses = c("integer", "double", "double", "double", "double", "factor"))
colnames(data) <- c("Age", "S1L+S1R", "S1L-S1R", "S2L+S2R", "S2L-S2R", "Group")
levels(data$Group)[levels(data$Group)=="0"] <- "Non-MS"
levels(data$Group)[levels(data$Group)=="1"] <- "MS"
summary(data)
```

```
##       Age           S1L+S1R         S1L-S1R          S2L+S2R
##  Min.   :18.00   Min.   :125.4   Min.   : 0.000   Min.   :169.2
##  1st Qu.:25.25   1st Qu.:141.4   1st Qu.: 0.800   1st Qu.:188.2
##  Median :36.00   Median :148.8   Median : 1.600   Median :200.6
##  Mean   :39.19   Mean   :156.5   Mean   : 4.733   Mean   :207.8
##  3rd Qu.:49.75   3rd Qu.:162.9   3rd Qu.: 3.350   3rd Qu.:217.4
##  Max.   :79.00   Max.   :238.4   Max.   :90.200   Max.   :328.0
##     S2L-S2R           Group
##  Min.   : 0.000   Non-MS:69
##  1st Qu.: 0.400   MS    :29
##  Median : 1.600
##  Mean   : 5.012
##  3rd Qu.: 3.350
##  Max.   :83.000
```

```r
group_1 <- data[data$Group == "Non-MS",]
```

```r
print("Summary and dimensions of the first group of individuals.")
```

```
## [1] "Summary and dimensions of the first group of individuals."
```

```r
summary(group_1)
```

```
##       Age           S1L+S1R         S1L-S1R         S2L+S2R         S2L-S2R
##  Min.   :18.00   Min.   :125.4   Min.   :0.000   Min.   :169.2   Min.   :0.00
##  1st Qu.:24.00   1st Qu.:139.2   1st Qu.:0.400   1st Qu.:185.6   1st Qu.:0.20
##  Median :31.00   Median :146.0   Median :1.600   Median :194.2   Median :1.60
##  Mean   :37.99   Mean   :147.3   Mean   :1.562   Mean   :195.6   Mean   :1.62
##  3rd Qu.:54.00   3rd Qu.:152.0   3rd Qu.:2.400   3rd Qu.:203.6   3rd Qu.:2.80
##  Max.   :79.00   Max.   :176.8   Max.   :5.600   Max.   :235.6   Max.   :6.00
##     Group
##  Non-MS:69
##  MS    : 0
##
##
##
##
```

```r
dim(group_1)
```

```
## [1] 69  6
```

```r
group_2 <- data[data$Group == "MS",]

print("Summary and dimensions of the second group of individuals.")
```

```
## [1] "Summary and dimensions of the second group of individuals."
```

```r
summary(group_2)
```

```
##       Age           S1L+S1R         S1L-S1R         S2L+S2R
##  Min.   :23.00   Min.   :134.4   Min.   : 0.00   Min.   :176.8
##  1st Qu.:34.00   1st Qu.:158.0   1st Qu.: 1.60   1st Qu.:214.4
##  Median :44.00   Median :166.4   Median : 6.80   Median :228.4
##  Mean   :42.07   Mean   :178.3   Mean   :12.28   Mean   :236.9
##  3rd Qu.:47.00   3rd Qu.:199.8   3rd Qu.:18.40   3rd Qu.:254.0
##  Max.   :59.00   Max.   :238.4   Max.   :90.20   Max.   :328.0
##     S2L-S2R          Group
##  Min.   : 0.00   Non-MS: 0
##  1st Qu.: 0.80   MS    :29
##  Median : 6.00
##  Mean   :13.08
##  3rd Qu.:15.60
##  Max.   :83.00
```

```r
dim(group_2)
```

```
## [1] 29  6
```

```r
X <- as.matrix(cbind(data[1:5]))
group_list <- data[6]
X_1 <- as.matrix(cbind(group_1[1:5]))
X_2 <- as.matrix(cbind(group_2[1:5]))
```

Some general setup for fishers rule

```
S1 <- cov(X_1)
S2 <- cov(X_2)
n1 <- nrow(X_1)
n2 <- nrow(X_2)
n<-c(n1,n2)
xmean1 <- colMeans(X_1)
xmean2 <- colMeans(X_2)
d<-xmean1-xmean2
Sp<-((n[1]-1)*S1+(n[2]-1)*S2)/(sum(n)-2)
Sp
```

```
##                Age   S1L+S1R  S1L-S1R   S2L+S2R    S2L-S2R
## Age     231.987996  82.97242 -2.09989  93.34313  -6.401513
## S1L+S1R  82.972416 325.90734 72.55294 341.76042  32.586132
## S1L-S1R  -2.099890  72.55294 93.81391  69.35624  87.073236
## S2L+S2R  93.343134 341.76042 69.35624 475.37981  25.318765
## S2L-S2R  -6.401513  32.58613 87.07324  25.31877 104.057218
```

Construct Fisher's rule. Moreover,calculate theapparent error rate.

```
w <- solve(Sp)%*%(xmean1-xmean2)
correct <- 0
incorrect <- 0
for (val in 1:nrow(X))
{
  row <- X[val,]
  group <- group_list[val,]
  #print("-----")
  #print(group)
  left_side <- t(w)%*%row
  right_side <- 0.5*t(w)%*%(xmean1+xmean2)
  #print(left_side)
  #print(right_side)
  if (left_side >= right_side){
    if (group == "Non-MS"){
      #print("Correct, should be Non-mS")
      correct <- correct + 1
    }
    else{
      #print("incorrect, should be MS")
      incorrect <- incorrect + 1
    }
  }
  else{
    if (group == "Non-MS"){
      #print("incorrect, should be MS")
      incorrect <- incorrect + 1
    }
    else{
      #print("Correct, should be ms")
      correct <- correct + 1
    }
```

```
  }
}
print("Error rate for Lachenbruch's holdout:")
```

```
## [1] "Error rate for Lachenbruch's holdout:"
```

```
incorrect/(nrow(X))
```

```
## [1] 0.1020408
```

```
print("Number correct:")
```

```
## [1] "Number correct:"
```

```
correct
```

```
## [1] 88
```

```
print("Number incorrect:")
```

```
## [1] "Number incorrect:"
```

```
incorrect
```

```
## [1] 10
```

Finally the expected actual error rate by Lachenbruch's holdout.

```
# this was fun :)
incorrect <- 0
correct <- 0
for (val in 1:nrow(data))
{
  # first we want to take out a row in the dataframe
  data_check <- data[c(val),]
  data_seg <- data[-c(val),]

  # then we want to seperate the data into the two groups
  group_1 <- data_seg[data_seg$Group == "Non-MS",]
  group_2 <- data_seg[data_seg$Group == "MS",]

  # we also want a matrix of the data and group_list calculate w based on s_pooled (since size of xmean
  X_1 <- as.matrix(cbind(group_1[1:5]))
  X_2 <- as.matrix(cbind(group_2[1:5]))
  S1 <- cov(X_1)
  S2 <- cov(X_2)
  n1 <- nrow(X_1)
  n2 <- nrow(X_2)
  n<-c(n1,n2)

  xmean1 <- colMeans(X_1)
  xmean2 <- colMeans(X_2)
  d<-xmean1-xmean2
  Sp<-((n[1]-1)*S1+(n[2]-1)*S2)/(sum(n)-2)
  w <- solve(Sp)%*%(xmean1-xmean2)

  # lets set row to the row we took out of the dataframe (to match our code from earlier) probably shou
  row <- as.matrix(cbind(data_check[1:5]))[1,]
```

```r
  group <- data_check[6][1,]
  left_side <- t(w)%*%row
  right_side <- 0.5*t(w)%*%(xmean1+xmean2)
  if (left_side >= right_side){
    if (group == "Non-MS"){
      #print("Correct, should be Non-mS")
      correct <- correct + 1
    }
    else{
      #print("incorrect, should be MS")
      incorrect <- incorrect + 1
    }
  }
  else{
    if (group == "Non-MS"){
      #print("incorrect, should be MS")
      incorrect <- incorrect + 1
    }
    else{
      #print("Correct, should be ms")
      correct <- correct + 1
    }
  }
}
apparent_error_rate <- incorrect/(incorrect+correct)
print("Error rate for Lachenbruch's holdout:")
```

```
## [1] "Error rate for Lachenbruch's holdout:"
```

```r
incorrect/(nrow(X))
```

```
## [1] 0.1326531
```

```r
print("Number correct:")
```

```
## [1] "Number correct:"
```

```r
correct
```

```
## [1] 85
```

```r
print("Number incorrect:")
```

```
## [1] "Number incorrect:"
```

```r
incorrect
```

```
## [1] 13
```