HW2-Multivariate

Benjan

11/4/2022

# Problem 1.

Lets read the file first.

urlremote <- urlRemote <- "https://raw.githubusercontent.com/"  
pathGithub <- pathGithub <- "EricBrownTTU/ISQS6350/main/"  
filename <- "fish.csv"  
  
fish <- read.csv(paste0(urlremote, pathGithub, filename))  
fish <- fish[,2:7]  
fish <- fish[complete.cases(fish),]

## a.

To get the mahalanobis distance, we need to get the mean vector and covariance matrix first.

xbar <- colMeans(fish) # mean vector  
S <- cov(fish) #covariance matrix

Now, the mahalanobis distance is:

d\_mahal <- round(mahalanobis(fish, xbar, S),4)  
head(d\_mahal)

## 1 2 3 4 5 6   
## 3.6358 4.3665 4.3261 4.5427 3.4045 5.4503

## b.

The outliers determined by chi\_squared distribution are:

out <- which(1 - pchisq(d\_mahal, 2) < 0.05)  
out

## 10 24 26 29 30 31 32 34 36 37 38 39 40 41 62 63 65 69 73 74   
## 10 23 25 28 29 30 31 33 35 36 37 38 39 40 61 62 64 68 72 73   
## 75 76 77 78 79 80 81 82 83 84 87 88 89 90 92 93 94 95 96 97   
## 74 75 76 77 78 79 80 81 82 83 86 87 88 89 91 92 93 94 95 96   
## 98 99 100 101 102 103 104 110 133 141 143 145 148 150 153 155 156 157 159   
## 97 98 99 100 101 102 103 109 132 140 142 144 147 149 152 154 155 156 158

## c.

First we need to load the library for kernel density evaluation.

library(ks)

Now, we need to evaluate the kde for the fish dataset.

fish.kde <- kde(fish, eval.points = fish)

Finally, lets find out the outliers.

out\_kde <- which(fish.kde$estimate <= quantile(fish.kde$estimate, .05))  
out\_kde

## 34 39 40 41 97 100 104 141   
## 33 38 39 40 96 99 103 140

## d.

We have already printed out the outliers given by each method. Just by visaully looking, we can see that mahalanobis gave us extra outliers. To confirm lets see the length of both vectors.

length(out)

## [1] 59

length(out\_kde)

## [1] 8

The elements that lie in both of them are:

intersect(out,out\_kde)

## [1] 33 38 39 40 96 99 103 140

As we can see that the intersect has 8 elements which is the same length of kde outliers and the elements are the same as well. This means that all the outliers given by kde are also given by mahalanobis with some extra outliers.

## e.

First, we will scale the data to represent them as z-scores.

fish\_sc <- scale(fish)

Now, we will make a dataframe of outliers:

fish\_outlier <- fish\_sc[row.names(fish\_sc) %in% out\_kde,]  
fish\_outlier

## species weight length1 length2 length3 hgtpct  
## 33 -1.4751429 1.4656771 0.99487008 1.03353537 1.21263045 1.58426643  
## 38 -1.0559476 -0.2581429 -0.06250294 -0.03657334 -0.03339411 0.02937076  
## 39 -1.0559476 0.3935112 0.22677835 0.24258546 0.24159062 0.40302786  
## 40 -1.0559476 1.1175713 0.74548965 0.74507128 0.72281390 0.17401222  
## 96 0.6208335 0.2821173 1.57343267 1.54532649 1.44464882 -1.65811291  
## 99 0.6208335 1.5352983 2.20187135 2.16878113 2.05477119 -1.45320418  
## 103 0.6208335 3.4846909 3.26921959 3.25750042 3.16330338 -1.48936455  
## 140 1.0400288 -0.2191551 0.15695183 0.14953253 0.03535207 -0.50098125

As we can see, the first four and last row have z-scores greater than 1 in the *species* column. The other 3 columns have high z-scores in at least four variables which makes them outliers.

## f.

The difference in the results is that Mahalanobis gives all the outliers obtained from kde along with some extra ones. This could be because Mahalanobis uses a stricter method of determing outliers.

# Problem 2.

Lets initialize the correlation matrix first.

P2\_cor <-matrix(c(1.000,0.402,0.396,0.301,0.305,0.339,0.340,  
 0.402,1.000,0.618,0.150,0.135,0.206,0.183,  
 0.396,0.618,1.000,0.321,0.289,0.363,0.345,  
 0.301,0.150,0.321,1.000,0.846,0.759,0.661,  
 0.305,0.135,0.289,0.846,1.000,0.797,0.800,  
 0.339,0.206,0.363,0.759,0.797,1.000,0.736,  
 0.340,0.183,0.345,0.661,0.800,0.736,1.000),nrow=7,byrow=TRUE)  
row.names(P2\_cor) <- c('Head length','Head breadth','Face breadth','Left finger length','Left forearm length','Left foot length','Height')  
colnames(P2\_cor) <- c('Head length','Head breadth','Face breadth','Left finger length','Left forearm length','Left foot length','Height')

The PCA of the correlation matrix is:

P2\_cor.pca <- princomp(covmat=P2\_cor)

We will use the summary of the PCA to see how many components are significant to explain 80 percent of the variance.

summary(P2\_cor.pca)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.9492241 1.2256950 0.80610632 0.6000474 0.58237656  
## Proportion of Variance 0.5427821 0.2146183 0.09282963 0.0514367 0.04845178  
## Cumulative Proportion 0.5427821 0.7574004 0.85023003 0.9016667 0.95011851  
## Comp.6 Comp.7  
## Standard deviation 0.48502898 0.33751644  
## Proportion of Variance 0.03360759 0.01627391  
## Cumulative Proportion 0.98372609 1.00000000

So, three components are significant.

The first component is:

P2\_cor.pca$loadings[,1]

## Head length Head breadth Face breadth Left finger length   
## 0.2763037 0.2118636 0.2951449 0.4375581   
## Left forearm length Left foot length Height   
## 0.4557045 0.4502341 0.4356893

The second component is:

P2\_cor.pca$loadings[,2]

## Head length Head breadth Face breadth Left finger length   
## 0.3647677 0.6392041 0.5123928 -0.2349399   
## Left forearm length Left foot length Height   
## -0.2766674 -0.1784374 -0.1795404

The first component is a weighted sum of all variables.The second component is most influenced by *Head breadth* and *Face breadth*.

# Problem 3.

Lets read the file.

urlremote <- urlRemote <- "https://raw.githubusercontent.com/"  
pathGithub <- pathGithub <- "EricBrownTTU/ISQS6350/main/"  
filename <- "protein.csv"  
  
protein <- read.csv(paste0(urlremote, pathGithub, filename), row.names = "Country")  
protein <- protein[,1:9]

## a.

Lets conduct a principle component analysis on the correlation matrix.

r <- cor(protein)  
protein\_pca <- princomp(covmat=r)

The results is:

summary(protein\_pca)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 2.0237432 1.2747169 1.0417887 0.9513238 0.6532516  
## Proportion of Variance 0.4550596 0.1805448 0.1205915 0.1005574 0.0474153  
## Cumulative Proportion 0.4550596 0.6356044 0.7561959 0.8567534 0.9041687  
## Comp.6 Comp.7 Comp.8 Comp.9  
## Standard deviation 0.58901631 0.51915697 0.36677320 0.33390912  
## Proportion of Variance 0.03854891 0.02994711 0.01494695 0.01238837  
## Cumulative Proportion 0.94271757 0.97266468 0.98761163 1.00000000

## b.

We can use the ‘cumulative proportion’ section of summary to know how much variance is explained and how significant the components are. We can see that 63.4 percent variance is explained by the first two principal components.

## c.

The first principal component is:

protein\_pca$loadings[,1]

## Red\_Meat White\_Meat Eggs   
## 0.3106693 0.3159279 0.4205930   
## Milk Fish Cereals   
## 0.3788776 0.1341071 -0.4298291   
## Starchy\_Foods Pulses\_nuts\_oilseeds Fruits\_Vegetables   
## 0.2959618 -0.4218085 -0.1223681

The second principal component is:

protein\_pca$loadings[,2]

## Red\_Meat White\_Meat Eggs   
## 0.06957085 0.21457197 0.09986721   
## Milk Fish Cereals   
## 0.16867961 -0.65161517 0.25366332   
## Starchy\_Foods Pulses\_nuts\_oilseeds Fruits\_Vegetables   
## -0.38888491 -0.12932932 -0.50377330

## d.

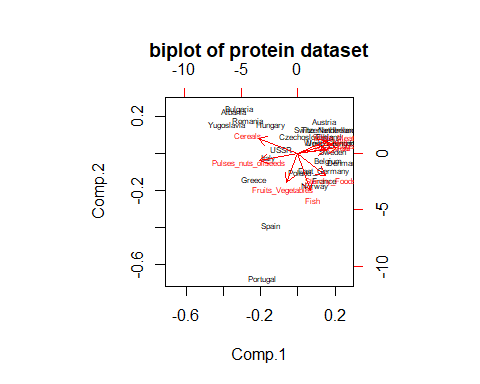
The first component is influenced by all original variables except *Fish* and *Fruits-Vegetables*. *Cereals* and *Pulses-nuts-oilseeds* contribute in the opposing direction.

The second component, in contrast to the first component, is most influenced by *Fish* and *Fruits-Vegetables* with both contributing in the opposing direction.

## e.

Lets create a biplot.

protein.pca <- princomp(protein,cor=TRUE)  
biplot(protein.pca, col = c("black","red"), cex = 0.5,main='biplot of protein dataset')



Lets look at ‘Yugoslavia’ and ‘Portugal’. ‘Yugoslavia’ has the most negative x-component or the lowest first principal component.From the interpretation of first component, it should have low values in variables that represent the first component, so thus we low values in *Eggs*, *Milk* and high values in *Cereal*. ‘Portugal’ has lowest second component. Thus we expect high values in *Fish* and *Fruits-Vegetables* .

Let us see the scaled values for validation.

round(scale(protein),2)

## Red\_Meat White\_Meat Eggs Milk Fish Cereals Starchy\_Foods  
## Albania 0.06 -1.85 -1.87 -1.17 -1.23 0.88 -2.03  
## Austria -0.24 1.63 0.83 0.38 -0.66 -0.39 -0.22  
## Belgium 1.23 0.29 0.83 0.10 0.21 -0.48 0.99  
## Bulgaria -0.53 -0.51 -0.97 -1.31 -0.95 2.24 -2.03  
## Czechoslovakia 0.06 0.82 -0.07 -0.60 -0.66 0.15 0.39  
## Denmark 0.35 0.82 0.83 1.09 1.65 -0.94 0.39  
## East\_Germany -0.53 1.09 0.83 -0.88 0.21 -0.66 1.59  
## Finland 0.06 -0.78 -0.07 2.36 0.50 -0.57 0.39  
## France 2.41 0.56 -0.07 0.38 0.50 -0.39 0.39  
## Greece 0.06 -1.32 -0.07 0.10 0.50 0.88 -1.43  
## Hungary -1.41 1.09 -0.07 -1.03 -1.23 0.70 -0.22  
## Ireland 1.23 0.56 1.72 1.23 -0.66 -0.76 0.99  
## Italy -0.24 -0.78 -0.07 -0.46 -0.37 0.43 -1.43  
## The\_Netherlands 0.06 1.63 0.83 0.81 -0.37 -0.94 -0.22  
## Norway -0.24 -0.78 -0.07 0.81 1.65 -0.85 0.39  
## Poland -0.82 0.56 -0.07 0.24 -0.37 0.33 0.99  
## Portugal -1.12 -1.05 -1.87 -1.73 2.80 -0.48 0.99  
## Romania -1.12 -0.51 -0.97 -0.88 -0.95 1.61 -0.82  
## Spain -0.82 -1.32 -0.07 -1.17 0.78 -0.30 0.99  
## Sweden 0.06 0.02 0.83 1.09 1.07 -1.12 -0.22  
## Switzerland 0.94 0.56 -0.07 0.95 -0.66 -0.57 -0.82  
## United\_Kingdom 2.12 -0.51 1.72 0.52 -0.08 -0.76 0.39  
## USSR -0.24 -0.78 -0.97 -0.04 -0.37 1.06 0.99  
## West\_Germany 0.35 1.36 0.83 0.24 -0.37 -1.21 0.39  
## Yugoslavia -1.70 -0.78 -1.87 -1.03 -0.95 2.15 -0.82  
## Pulses\_nuts\_oilseeds Fruits\_Vegetables  
## Albania 1.45 -1.15  
## Austria -1.03 -0.10  
## Belgium -0.53 -0.10  
## Bulgaria 0.46 -0.10  
## Czechoslovakia -1.03 -0.10  
## Denmark -1.03 -1.15  
## East\_Germany -1.03 -0.10  
## Finland -1.03 -1.67  
## France -0.53 1.46  
## Greece 2.44 1.46  
## Hungary 0.95 -0.10  
## Ireland -0.53 -0.63  
## Italy 0.46 1.46  
## The\_Netherlands -0.53 -0.10  
## Norway -0.53 -0.63  
## Poland -0.53 1.46  
## Portugal 0.95 1.98  
## Romania 0.95 -0.63  
## Spain 1.45 1.46  
## Sweden -1.03 -1.15  
## Switzerland -0.53 0.42  
## United\_Kingdom -0.04 -0.63  
## USSR -0.04 -0.63  
## West\_Germany -0.53 -0.10  
## Yugoslavia 1.45 -0.63  
## attr(,"scaled:center")  
## Red\_Meat White\_Meat Eggs   
## 9.80 7.92 3.08   
## Milk Fish Cereals   
## 17.28 4.28 32.32   
## Starchy\_Foods Pulses\_nuts\_oilseeds Fruits\_Vegetables   
## 4.36 3.08 4.20   
## attr(,"scaled:scale")  
## Red\_Meat White\_Meat Eggs   
## 3.403430 3.740766 1.115049   
## Milk Fish Cereals   
## 7.097652 3.470351 11.010298   
## Starchy\_Foods Pulses\_nuts\_oilseeds Fruits\_Vegetables   
## 1.655295 2.019076 1.914854

As we can see from the table, ‘Yugoslavia’ scores the joint lowest in *Eggs*, lowest in *Red-Meat*, very low in *Milk* and highest in *Cereal*. Portugal scores the highest in both *Fish* and *Fruits-Vegetables* . Thus our observation from biplot is validated.

# Problem 4.

Lets read and look at the heptathlon data.

data("heptathlon", package = "HSAUR2")  
head(heptathlon)

## hurdles highjump shot run200m longjump javelin run800m  
## Joyner-Kersee (USA) 12.69 1.86 15.80 22.56 7.27 45.66 128.51  
## John (GDR) 12.85 1.80 16.23 23.65 6.71 42.56 126.12  
## Behmer (GDR) 13.20 1.83 14.20 23.10 6.68 44.54 124.20  
## Sablovskaite (URS) 13.61 1.80 15.23 23.92 6.25 42.78 132.24  
## Choubenkova (URS) 13.51 1.74 14.76 23.93 6.32 47.46 127.90  
## Schulz (GDR) 13.75 1.83 13.50 24.65 6.33 42.82 125.79  
## score  
## Joyner-Kersee (USA) 7291  
## John (GDR) 6897  
## Behmer (GDR) 6858  
## Sablovskaite (URS) 6540  
## Choubenkova (URS) 6540  
## Schulz (GDR) 6411

A heptathlon data is recorded such that in 100 meter, 200 meter and 800 meter , the smallest value is superior and in others largest value is superior. So, we will reverse the value for standard measure of the data.

heptathlon$hurdles <- max(heptathlon$hurdles) - heptathlon$hurdles  
heptathlon$run200m <- max(heptathlon$run200m) - heptathlon$run200m  
heptathlon$run800m <- max(heptathlon$run800m) - heptathlon$run800m  
head(heptathlon)

## hurdles highjump shot run200m longjump javelin run800m  
## Joyner-Kersee (USA) 3.73 1.86 15.80 4.05 7.27 45.66 34.92  
## John (GDR) 3.57 1.80 16.23 2.96 6.71 42.56 37.31  
## Behmer (GDR) 3.22 1.83 14.20 3.51 6.68 44.54 39.23  
## Sablovskaite (URS) 2.81 1.80 15.23 2.69 6.25 42.78 31.19  
## Choubenkova (URS) 2.91 1.74 14.76 2.68 6.32 47.46 35.53  
## Schulz (GDR) 2.67 1.83 13.50 1.96 6.33 42.82 37.64  
## score  
## Joyner-Kersee (USA) 7291  
## John (GDR) 6897  
## Behmer (GDR) 6858  
## Sablovskaite (URS) 6540  
## Choubenkova (URS) 6540  
## Schulz (GDR) 6411

## a.

We will scale the data first.

heptathlon\_sc <- scale(heptathlon)

The distance matrix is:

d <- round(dist(heptathlon\_sc,diag=TRUE,upper=TRUE),2)  
head(d)

## [1] 2.17 2.14 3.39 3.48 4.04 3.91

## b.

Lets find the covariance matrix and its eigenvalues.

hep\_s <- cov(heptathlon\_sc) # covariance  
hep\_eigen <- eigen(hep\_s)  
lambda <- hep\_eigen$values

To see the cumulative proportion of eigenvalues:

100 \* round(cumsum(lambda)/sum(lambda),4)

## [1] 68.08 83.10 89.61 95.33 98.42 99.34 99.95 100.00

We need atleast **2** dimensions to account for 80 percent of eigenvalues.

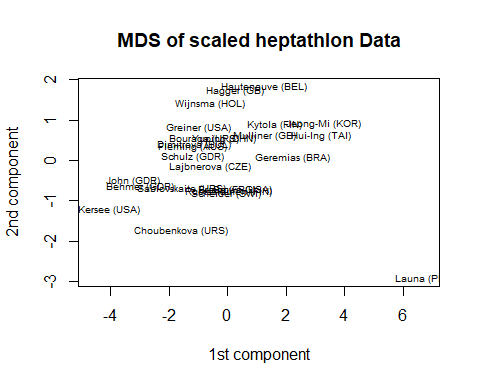
## c.

We will use the distance ‘d’ to perform MDS analysis.

hep.mds <- cmdscale(d)

Let us plot the cmd data.

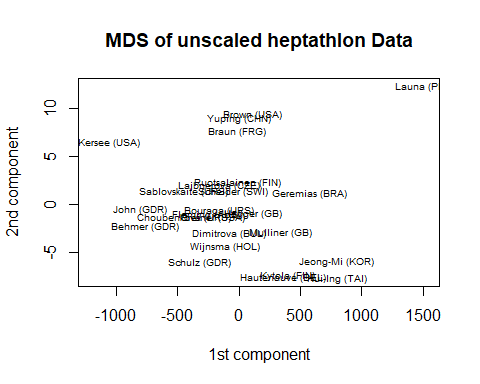
plot(hep.mds, type = "n",  
 xlab = "1st component", ylab = "2nd component",  
 main = "MDS of scaled heptathlon Data")  
text(hep.mds, labels = rownames(heptathlon),cex = 0.6)

 The athlete from Finland, ‘Ruotsalainer’ is most similar to Scheider.

## d.

Here, we will use the unscaled data to plot.

d\_unscaled <- round(dist(heptathlon,diag=TRUE,upper=TRUE),2)  
unscaled.mds <- cmdscale(d\_unscaled )  
plot(unscaled.mds, type = "n",  
 xlab = "1st component", ylab = "2nd component",  
 main = "MDS of unscaled heptathlon Data")  
text(unscaled.mds, labels = rownames(heptathlon),cex = 0.6)



As we can see that the plots are not the same.

Now, we can look at the actual data as well.

head(hep.mds)

## [,1] [,2]  
## Joyner-Kersee (USA) -4.638754 -1.20349360  
## John (GDR) -3.215774 -0.50081518  
## Behmer (GDR) -2.981587 -0.64977011  
## Sablovskaite (URS) -1.560959 -0.68367078  
## Choubenkova (URS) -1.586395 -1.74164043  
## Schulz (GDR) -1.186890 0.08794781

head(unscaled.mds)

## [,1] [,2]  
## Joyner-Kersee (USA) -1200.4171 6.515087  
## John (GDR) -806.4679 -0.513956  
## Behmer (GDR) -767.4894 -2.330502  
## Sablovskaite (URS) -449.4220 1.379192  
## Choubenkova (URS) -449.4764 -1.336971  
## Schulz (GDR) -320.4952 -6.042793

The data are not the same for both cases. Hence, scaling impacts on MDS.This should be because when we scale a dataset, the center of the dataset changes along with the value of each observation.

## e.

The correlation matrix is:

hep\_r <- cor(heptathlon\_sc)

The correlation distance matrix is:

cor\_dist <- round(1-abs(hep\_r),2)  
cor\_dist

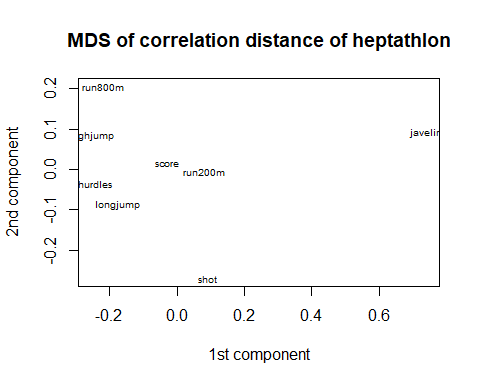
## hurdles highjump shot run200m longjump javelin run800m score  
## hurdles 0.00 0.19 0.35 0.23 0.09 0.99 0.22 0.08  
## highjump 0.19 0.00 0.56 0.51 0.22 1.00 0.41 0.23  
## shot 0.35 0.56 0.00 0.32 0.26 0.73 0.58 0.20  
## run200m 0.23 0.51 0.32 0.00 0.18 0.67 0.38 0.14  
## longjump 0.09 0.22 0.26 0.18 0.00 0.93 0.30 0.05  
## javelin 0.99 1.00 0.73 0.67 0.93 0.00 0.98 0.75  
## run800m 0.22 0.41 0.58 0.38 0.30 0.98 0.00 0.23  
## score 0.08 0.23 0.20 0.14 0.05 0.75 0.23 0.00

This matrix shows how dissimilar the variables are in terms of correlation with the value ‘1’ representing most unrelated variables. As we can see that the diagonal values are zero which means that the variables have zero correlation distance with themselves and this makes sense.

## f.

The MDS for correlation distance is:

cor.mds <- cmdscale(cor\_dist)  
plot(cor.mds, type = "n",  
 xlab = "1st component", ylab = "2nd component",  
 main = "MDS of correlation distance of heptathlon")  
text(cor.mds, labels = colnames(heptathlon),cex = 0.6)



The variables that are most similar are score and 200 meters run. However, from the correlation-distance matrix, score and longjump are most similar. The conclusion is not clear from these two observations.