The ordinary scatterplot matrix is shown below:





The enhanced scatterplot matrix with estimated bivariate densities is shown below:



Compared to the simple scatterplot matrix, the enhanced scatterplot matrix can help us identify the potential distinct groups of observations.

Problem 2:



The multiple linear regression model seems not appropriate. We cannot find any indication of the appropriate model.

Problem3:

(1)

Part 1:

Scatterplot matrix:

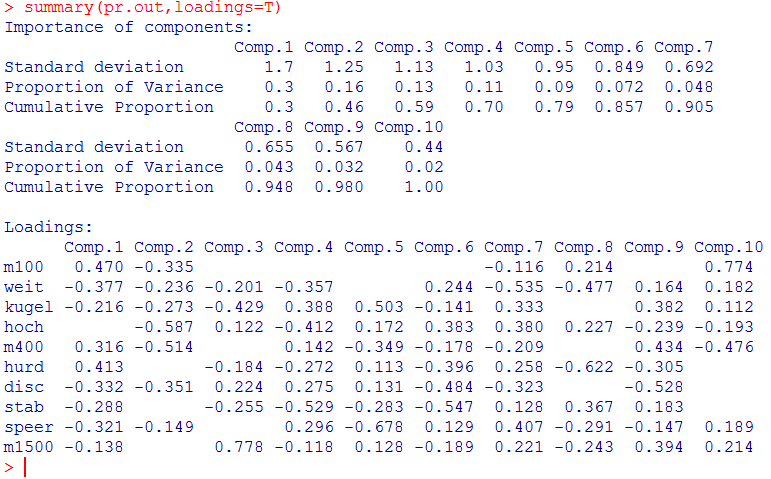


From the scatterplot matrix, we can find several important indications:

1)Stab and kugel seems have VERY weak relationship with other predictors.

2) The relationships between punkte and other variables seem not very linear.

PCA on correlation matrix:



Potential interpretation:

The first PC put much weight on the m100 and hurd

and the second PC put much weight on the m400 and hoch

The third on the kugel and m1500

the fourth on the hoch, stab

and the fifth on the hoch and speer

the sixth on the disc and stab.

Scree plot:

For every PC:

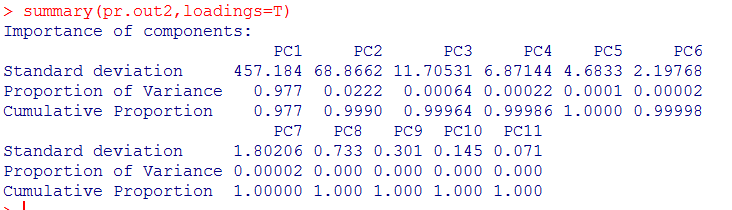


For searching the number of PCs:



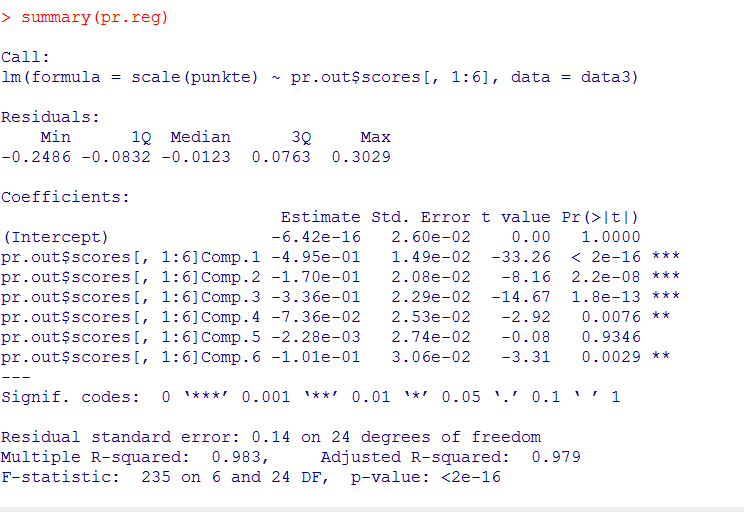
It seems that six PCs are enough to capture much information contained in the data.

PCA on covariance matrix:

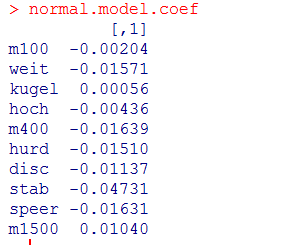


We can find from the “importance of components” on the pcs from covariance matrix, that the first pc contains 97.7% variance of all predictors, which is caused by different scales on every predictor. So, we’d better do PCA on correlation matrix.

(b) to draw some inference from this dataset, what we should do is regression on these predictors. However, the dataset has too less observations but so many predictors. So, we need some predictor shrinkage method. And PCA is such a method. And the issue is not about predictions but only the inference, so we just regress outcomes on these shrunk predictors ( means pcs here).



We do regression on the first six pcs. And the result is shown above. Now we tend to convert this regression model into the normalized model.



We can compare these coefficients to determine the importance to the final score.



For the “all do poor or all do well” questions, we cannot answer them because of lack of criterion of whether their scores are low or high.

1. The m100 , hoch and kugel seem have less important for the final score. We can see from the boxplot that the score range of athletes do not vary much. This is very reasonable.
2. It seems the most important item to determine the final score is stab. We can find some indication in the scatterplot and the boxplot where the 0 score is eliminated.

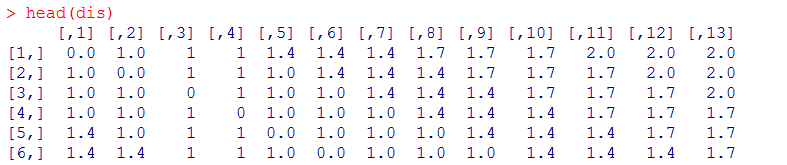
We can see from the scatterplot, the strong positive relationship between the final and the stab score. What is more, the stab score is very variable from athlete to athlete.

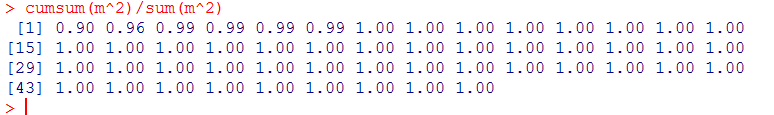
1. M400, weit and hurd have a relatively strong weight to the final score. However, their ranges are small. This might be an indication that these three items will make a big difference on the only small changes. In other words, these three items might be the most difficult to improve.
2. From the original scatterplot matrix, because Stab and kugel seems have VERY weak relationship with other predictors, it is obvious that we cannot use these two scores to make some explorations on the other scores.

Question (3)

I cannot quiet understand this question, what do you mean by “average athlete”?

Problem 4:





We can find that two eigenvectors are enough.

The recovered data is shown below:



We can find that this plot is symmetric. We can use two directions to determine the structure of the data. As the similarity matrix is also symmetric, which means the points in the original space are also symmetric distributed. So, this representation of the original data might be reasonable.

Code:

#problem 1#

library(MVA)

data=heptathlon

pairs(data,panel=function(x,y,...){

points(x,y,...)

abline(lm(y~x),col="grey")

},

pch=".",

cex=2.5)

panel.hist <- function(x, ...)

{

usr <- par("usr"); on.exit(par(usr))

par(usr = c(usr[1:2], 0, 1.5) )

h <- hist(x, plot = FALSE)

breaks <- h$breaks; nB <- length(breaks)

y <- h$counts; y <- y/max(y)

rect(breaks[-nB], 0, breaks[-1], y, col = "grey", ...)

}

pairs(data,

diag.panel = panel.hist,

panel = function (x,y) {

data <- data.frame(cbind(x,y))

par(new = TRUE)

den <- bkde2D(data, bandwidth = sapply(data, dpik))

contour(x = den$x1, y = den$x2,

z = den$fhat, axes = FALSE)

}

)

#problem 2#

data2=USairpollution;

pairs(data2,panel=function(x,y,...){

points(x,y,...)

abline(lm(y~x),col="grey")

lines(lowess(y~x),col="red")

},

pch=".",

cex=2.5)

problem 3:

pairs(data3)

pr.out=princomp(data3[,-11],cor=T)

pr.out2=princomp(data3[,-11],cor=F)

names(pr.out)

summary(pr.out,loadings=T)

var=(pr.out$sdev)^2

plot(var~seq(1,10),type="b")

plot(cumsum(var)~seq(1,10),type="b")

text(seq(1,10),cumsum(var),labels=round(cumsum(var)/sum(var),3))

summary(pr.out2,loadings=T)

var2=(pr.out2$sdev)^2

plot(var2~seq(1,10),type="b")

plot(cumsum(var2)~seq(1,10),type="b")

text(seq(1,10),cumsum(var2),labels=round(cumsum(var2)/sum(var2),3))

names(pr.out)

pairs(cbind(scale(data3$punkte),pr.out$scores[,1:6]))

pr.reg=lm(scale(punkte)~pr.out$scores[,1:6],data=data3)

summary(pr.reg)

coef=as.matrix(summary(pr.reg)$coefficients)

coef=coef[,2]

coef=coef[-1]

names(coef)=NULL

pc=as.matrix(pr.out$loadings)[,1:6]

normal.model.coef=pc%\*%coef

normal.model.coef

par(mfrow=c(2,5))

for (i in 1:10){

boxplot(data3[,i],main=names(data3)[i],ylim=c(min(data3[,i]),max(data3[,i])))}

plot(data3[,11]~data3$stab)

boxplot(data3$stab[-which.min(data3$stab)])

#question 3#

# what means "average athletes"?#

#last problem#

o=seq(1:51)

s=function(x,y){

(x==y)\*9+

(abs(x-y)>=1&abs(x-y)<=3)\*8+

(abs(x-y)>=4&abs(x-y)<=6)\*7+

(abs(x-y)>=7&abs(x-y)<=9)\*6+

(abs(x-y)>=10&abs(x-y)<=12)\*5+

(abs(x-y)>=13&abs(x-y)<=15)\*4+

(abs(x-y)>=16&abs(x-y)<=18)\*3+

(abs(x-y)>=19&abs(x-y)<=21)\*2+

(abs(x-y)>=22&abs(x-y)<=24)\*1+

(abs(x-y)>=25)\*0}

sim=matrix(rep(0),51,51)

for (i in 1:51){

for (j in 1:51){

sim[i,j]=s(o[i],o[j]);

}

}

delta=function(i,j){

sqrt(sim[i,i]+sim[i,j]-2\*sim[i,j])}

dis=matrix(rep(0),51,51)

for (i in 1:51){

for (j in 1:51){

dis[i,j]=delta(i,j);

}

}

head(dis)

# the result should be similar to the matrix decomposition#

msd=eigen(dis)

names(msd)

eivalue=msd$values

m=sort(abs(eivalue),decreasing=T)

cumsum(m^2)/sum(m^2)

eivector=msd$vectors

choose1=eivector[,1]

choose2=eivector[,51]

choose=cbind(choose1,choose2)

choose

sigma=matrix(c(eivalue[1],0,0,eivalue[51]),2,2)

recover=choose%\*%sigma

recover

# using cmdscale() the result will be less different but the configuration should be the same#

cmd=cmdscale(dis,k=2,eig=T)

plot(cmd$points[,1]~cmd$points[,2])