577-HW1-LianZuo

LianZuo

2023-09-20

## Question 1

# a. the mean of each variable is as follows  
  
library(carData)  
names(Baumann)

[1] "group" "pretest.1" "pretest.2" "post.test.1" "post.test.2"  
[6] "post.test.3"

attach(Baumann)  
sapply(Baumann[,-1], mean)

pretest.1 pretest.2 post.test.1 post.test.2 post.test.3   
 9.787879 5.106061 8.075758 6.712121 44.015152

# b. the covariance matrix with 3 decimal places is as follows, the dimension of this matrix is 5\*5.  
covb <- round(cov(Baumann[,-1]),digits = 3)  
covb

pretest.1 pretest.2 post.test.1 post.test.2 post.test.3  
pretest.1 9.124 2.238 5.801 0.707 -0.751  
pretest.2 2.238 4.896 2.592 1.200 2.660  
post.test.1 5.801 2.592 11.517 0.576 10.599  
post.test.2 0.707 1.200 0.576 6.947 -0.734  
post.test.3 -0.751 2.660 10.599 -0.734 44.138

# c. The covariance value between pretest.1 and pretest.2 variables is 2.238  
cov12 <- covb[1,2]  
cov12

[1] 2.238

# d.the correlation matrix with 3 decimal places is as follows, the dimension of this matrix is 5\*5  
corrd <- round(cor(Baumann[,-1]),digits = 3)  
corrd

pretest.1 pretest.2 post.test.1 post.test.2 post.test.3  
pretest.1 1.000 0.335 0.566 0.089 -0.037  
pretest.2 0.335 1.000 0.345 0.206 0.181  
post.test.1 0.566 0.345 1.000 0.064 0.470  
post.test.2 0.089 0.206 0.064 1.000 -0.042  
post.test.3 -0.037 0.181 0.470 -0.042 1.000

# e.The correlation value between post.test.1 and post.test.2 variables is 0.335  
cor34 <- corrd[3,4]  
cor34

[1] 0.064

## Question 2

Draw a bivaritate boxplot for pretest.1 and pretest.2 variables. In your graph, pretest.1 and pretest.2 variables will go along x and y axis, respectively. Find the index of any outlier(s) that is outside the fence (the outer ellipse). Write down the text “Outlier” to the right of the outlier point in the graph. Make sure to have x and y axes titles as well as the graph title.

#install.packages("MVA")  
library(tools)  
library(HSAUR2)  
library(MVA)  
attach(Baumann)

The following objects are masked from Baumann (pos = 6):  
  
 group, post.test.1, post.test.2, post.test.3, pretest.1, pretest.2

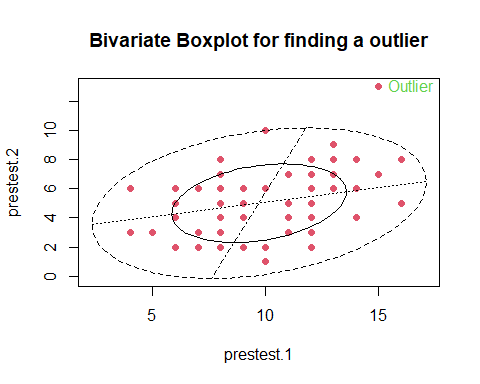
bvbox(cbind(pretest.1,pretest.2),xlab="prestest.1",ylab = "prestest.2",pch=16,cex=1,col=2,,main="Bivariate Boxplot for finding a outlier")  
n=nrow(Baumann)  
out=match(sort(pretest.2)[n],pretest.2)  
out

[1] 6

Baumann[out,2:3]

pretest.1 pretest.2  
6 15 13

text(pretest.1[out], pretest.2[out], labels = "Outlier", cex = 1, col=3, pos = 4)



## Question 3

Draw a convex hull for post.test.1 and post.test.2 variables. In your graph, post.test.1 and post.test.2 variables will go along x and y axis, respectively. Make sure to have x and y axes titles as well as the graph title. Report the index of extreme points in each direction.

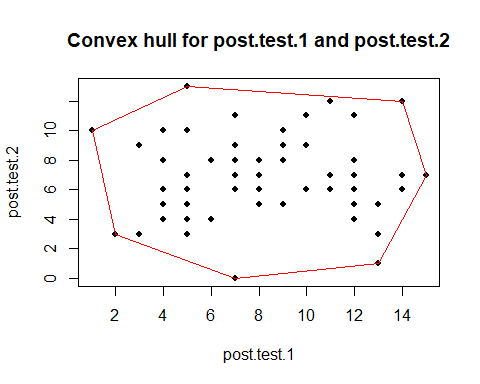
extr.s=chull(cbind(post.test.1, post.test.2))  
extr.s

[1] 58 34 20 57 60 52 63

cbind(post.test.1[extr.s], post.test.2[extr.s])#give extreme points

[,1] [,2]  
[1,] 13 1  
[2,] 7 0  
[3,] 2 3  
[4,] 1 10  
[5,] 5 13  
[6,] 14 12  
[7,] 15 7

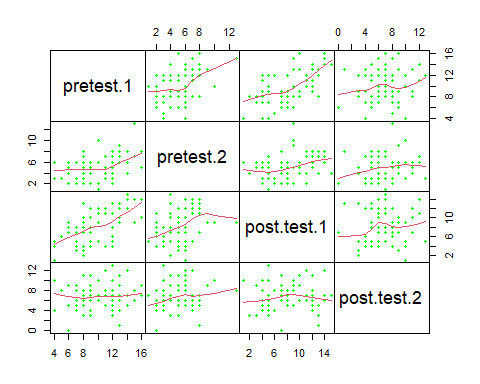
extr.s=c(extr.s, extr.s[1])  
plot(post.test.1, post.test.2,xlab = "post.test.1",ylab = "post.test.2", pch = 16, col = 1, cex = 0.8, main="Convex hull for post.test.1 and post.test.2")  
lines(post.test.1[extr.s], post.test.2[extr.s], type="l", col="red", lwd=1.2)



## Question 4

Draw a scatterplot matrix of the following four variables: pretest.1,pretest.2,post.test.1, post.test.2 and provide your graph below.

pairs(cbind(pretest.1, pretest.2, post.test.1, post.test.2), panel=panel.smooth, col="green",gap = 0, lwd = 1, pch = 16, cex = 0.6)



## Question 5

set.seed(1233)  
z=matrix(runif(30,1,2), nrow=6, ncol=5, byrow=F)  
z

[,1] [,2] [,3] [,4] [,5]  
[1,] 1.679906 1.059584 1.529674 1.827584 1.466673  
[2,] 1.253893 1.484312 1.842131 1.373377 1.799311  
[3,] 1.271116 1.420363 1.926901 1.396387 1.259994  
[4,] 1.464016 1.141275 1.021787 1.641406 1.338666  
[5,] 1.466820 1.047877 1.470556 1.333974 1.781985  
[6,] 1.790542 1.953523 1.318965 1.237703 1.627446

# a. z’z is symmetric  
A <- t(z)%\*%z  
A

[,1] [,2] [,3] [,4] [,5]  
[1,] 13.51101 12.15237 13.34348 13.14311 13.80931  
[2,] 12.15237 11.56014 12.37574 11.64739 12.58879  
[3,] 13.34348 12.37574 14.39255 13.28758 14.12087  
[4,] 13.14311 11.64739 13.28758 13.18173 13.49975  
[5,] 13.80931 12.58879 14.12087 13.49975 14.59231

# b. Find the eigenvalues of z’z.  
eigen(A)

eigen() decomposition  
$values  
[1] 65.56187151 0.71661607 0.64177513 0.26610296 0.05137859  
  
$vectors  
 [,1] [,2] [,3] [,4] [,5]  
[1,] -0.4503019 -0.30155476 0.44026414 -0.06896813 0.7125334  
[2,] -0.4116328 0.60689853 0.48535968 -0.33743589 -0.3358505  
[3,] -0.4611337 0.34781606 -0.74065195 -0.17255705 0.2967128  
[4,] -0.4422595 -0.64739336 -0.14727159 -0.34331472 -0.4957161  
[5,] -0.4685604 0.02539164 0.01841819 0.85658595 -0.2138404

eigen(A)$values

[1] 65.56187151 0.71661607 0.64177513 0.26610296 0.05137859

# c. Based on the eigenvalues of z’z, it is positive definite.  
  
# d. the determinant of z’z is reported as follows. the determinant of z’z is equal to the product of the eigenvalues  
det(A)

[1] 0.4122424

d2=prod(eigen(A)$values)  
all.equal(det(A),d2)

[1] TRUE

# e. the trace of z’z is equal to the sum of the eigenvalues.  
diag(A) #extracting diagonal elements of A

[1] 13.51101 11.56014 14.39255 13.18173 14.59231

sum(diag(A)) # trace of A

[1] 67.23774

sum(eigen(A)$values)

[1] 67.23774

all.equal(sum(diag(A)),sum(eigen(A)$values))

[1] TRUE

# f. the generalized inverse matrix of z’z is reported in two methods  
#generalized inverse of matrix A  
Lambda=eigen(A)$values  
P=eigen(A)$vectors   
gen.inv.A = P %\*%solve(diag(Lambda)) %\*%t(P)  
gen.inv.A

[,1] [,2] [,3] [,4] [,5]  
[1,] 10.331510 -4.4898139 3.508333 -6.6113226 -3.1824412  
[2,] -4.489814 3.5069017 -1.983412 3.0188567 0.3499958  
[3,] 3.508333 -1.9834117 2.852243 -2.7812946 -1.7960318  
[4,] -6.611323 3.0188567 -2.781295 5.8473834 0.9340608  
[5,] -3.182441 0.3499958 -1.796032 0.9340608 3.6521437

require(MASS) # the second method with ginv function

Loading required package: MASS

ginv(A)

[,1] [,2] [,3] [,4] [,5]  
[1,] 10.331510 -4.4898139 3.508333 -6.6113226 -3.1824412  
[2,] -4.489814 3.5069017 -1.983412 3.0188567 0.3499958  
[3,] 3.508333 -1.9834117 2.852243 -2.7812946 -1.7960318  
[4,] -6.611323 3.0188567 -2.781295 5.8473834 0.9340608  
[5,] -3.182441 0.3499958 -1.796032 0.9340608 3.6521437

# g. Find the square root matrix of z’z.  
sq.root.A=P%\*%diag(sqrt(Lambda))%\*%t(P)  
sq.root.A

[,1] [,2] [,3] [,4] [,5]  
[1,] 1.991644 1.474879 1.385387 1.657999 1.643424  
[2,] 1.474879 1.956797 1.435117 1.181682 1.449095  
[3,] 1.385387 1.435117 2.298973 1.545300 1.655435  
[4,] 1.657999 1.181682 1.545300 2.072399 1.534147  
[5,] 1.643424 1.449095 1.655435 1.534147 2.167377