```
#-------------------------------------------------
# CODICE R MESSO A DISPOSIZIONE PER LA PROVA D'ESAME
#-------------------------------------------------

rm(list=ls())
X <- matrix(
  c(3,4,2,6,8,2,5,
    5,5.5,4,7,10,5,7.5),
  nrow=7,ncol=2,
  byrow=FALSE)
n <- nrow(X)
p <- ncol(X)
colnames(X)<-c("x1","x2")
rownames(X)<-paste("u",1:n, sep="")
X
t(X)
summary(X)
mean(X[,1])
((n-1)/n) * var(X[,1])
apply(X,MARGIN=2,FUN="mean")
S = ((n-1)/n)*var(X)
S
R = cor(X)
R
plot(X)
set.seed(123)
X[,"x1"] <- sample(X[,"x1"])
X
plot(X)
summary(X)
((n-1)/n)*var(X)
cor(X)
n <- 20
x1 <- -1 + 2* ((1:n) -1 )/(n-1)
x2 <- x1^2
X <- cbind(x1,x2)
plot(x1,x2)
cor(X)
round(cor(X),1)

#-------------------------------------------------

rm(list=ls())
library("MASS")
data(Animals)
row.names(Animals)
with(Animals,
boxplot(brain, horizontal=TRUE)
)
boxplot.stats(Animals$brain)
baffo.sx <- boxplot.stats(Animals$brain)$stats[1]
baffo.sx
baffo.dx <- boxplot.stats(Animals$brain)$stats[5]
baffo.dx
```

```
outs <- boxplot.stats(Animals$brain)$out
outsTRUE <- Animals$brain %in% outs
outsTRUE
which.outs <- which(outsTRUE)
which.outs
rownames(Animals)[which.outs]
plot(brain~body, Animals)
cor(Animals)
plot(log(brain)~log(body), Animals)
with(Animals,
     text(log(brain)~log(body), labels = row.names(Animals), pos=4)
     )
cor(log(Animals))
library(aplpack)
bag <- bagplot(log(Animals))
bag$pxy.outlier
which.out<-which( rownames(Animals) %in% c("Brachiosaurus",
"Triceratops", "Dipliodocus"))
plot(log(brain)~log(body), Animals[-which.out,])
cor(log(Animals[-which.out,]))
hull <- with(Animals[-which.out,], chull(log(body),log(brain)))
with(Animals, plot(log(body),log(brain)))
with(Animals[-which.out,], polygon(log(body)[hull],log(brain)[hull],
density = 15, angle=30))

#-------------------------------------------------

rm(list=ls())
Tizio = c(180,70)
Caio = c(160,50)
X = rbind(Tizio,Caio)
colnames(X) = c("Altezza","Peso")
X
plot(X, xlim=c(0,200), ylim=c(0,200))
text(x=X[,"Altezza"], y=X[,"Peso"], labels = row.names(X), pos=3)
barx = matrix(colMeans(X), ncol=1)
barx
baricentro = t(barx)
baricentro
points(baricentro, pch=19)
tX = t(X)
plot(tX, xlim=c(0,200), ylim=c(0,200), pch=".")
text(x=tX[,"Tizio"], y=tX[,"Caio"], labels = row.names(tX), pos=4)
arrows(x0=0,y0=0,x1=tX[,"Tizio"], y1=tX[,"Caio"])
n = 3
p = 2
X = matrix(c(4,1,-1,3,3,5) ,byrow=TRUE, ncol=p, nrow=n)
X
tildex1 = matrix(X[,1] - mean(X[,1]), ncol=1)
tildex1
tildex2 = matrix(X[,2] - mean(X[,2]), ncol=1)
tildex2
ns11 = t(tildex1) %*% tildex1
ns11
```

```
(n-1)*var(X[,1])
ns22 = t(tildex2) %*% tildex2
ns22
(n-1)*var(X[,2])
ns12 = t(tildex1) %*% tildex2
ns12
(n-1)*var(X)[1,2]
r12 = ns12/sqrt(ns11*ns22)
r12
cor(X)[1,2]
acos(r12)
acos(r12)*(180/pi)

#-------------------------------------------------

rm(list=ls())
n = 10
p = 2
X <-
matrix(c(2,3,3,4,4,5,6,6,7,8,7,8,10,6,8,10,12,13,11,12),nrow=n,ncol=
p)
X
plot(X,xlim=c(-4,13),ylim=c(-4,13),asp=1,
bg=heat.colors(n),pch=21,cex=2)
abline(h=0)
abline(v=0)
one.n <- matrix(rep(1,n),ncol=1)
one.n
xbar <- (1/n) * t(X) %*% one.n
xbar
I.n <- diag(rep(1,n))
I.n
H <- I.n - (1/n) * one.n %*% t(one.n)
H
sum( t(H) - H )
sum( H %*% H - H )
Xtilde <- H %*% X
plot(Xtilde, xlim=c(-4,13),ylim=c(-4,13),
     bg=heat.colors(n),pch=21,cex=2,asp=1)
abline(h=0)
abline(v=0)
sum( H%*%Xtilde - Xtilde )
S <- (1/n) * t(H%*%X) %*% (H%*%X)
S
D <- diag(diag(S)^(-.5))
D
R <- D %*% S %*% D
R
D2 <- diag(diag(S)^(.5))
D2
S = D2 %*% R %*% D2
S
Z = Xtilde %*% D
plot(Z,xlim=c(-4,13),ylim=c(-4,13),
```

```
bg=heat.colors(n),pch=21,cex=2,asp=1)
abline(h=0)
abline(v=0)
( S_Xtilde <- (1/n) * t(H%*%Xtilde) %*% (H%*%Xtilde) )
( S_Z <- (1/n) * t(H%*%Z) %*% (H%*%Z) )
( R_Xtilde <-  diag(diag(S_Xtilde)^(-.5)) %*% S_Xtilde %*%
diag(diag(S_Xtilde)^(-.5)) )
( S_Z <- diag(diag(S_Z)^(-.5)) %*% S_Z %*% diag(diag(S_Z)^(-.5)) )

#-------------------------------------------------

rm(list=ls())
S <- matrix(c(2.2, 0.4, 0.4, 2.8),nrow=2,ncol=2)
S
sum(diag(S))
det(S)
det(S) / prod(diag(S))
R = diag(diag(S)^(-1/2)) %*% S %*% diag(diag(S)^(-1/2))
det(R)
eigenS <- eigen(S)
lambda1 <- eigenS$values[1]
lambda1
lambda2 <- eigenS$values[2]
lambda2
v1 <- eigenS$vectors[,1, drop=F]
v1
v2 <- eigenS$vectors[,2, drop=F]
v2
t(v1) %*% v1
crossprod(v1,v2)

#-------------------------------------------------

rm(list=ls())
library(MASS)
data(Animals)
X = log(Animals)
n = nrow(X)
S = var(X) * (n-1)/n
S
sum(diag(S))
det(S)
det(S) / prod(diag(S))
eigenS <- eigen(S)
lambda1 <- eigenS$values[1]
lambda2 <- eigenS$values[2]
v1 <- eigenS$vectors[,1, drop=F]
v2 <- eigenS$vectors[,2, drop=F]
plot(X)
bc = colMeans(X)
points(bc[1],bc[2], pch=19)
library(ellipse)
lines(ellipse(x=S,centre = bc, t = 1))
arrows(x0 = bc[1], y0 = bc[2], x1 = sqrt(lambda1)*v1[1] + bc[1],
```

```
y1=sqrt(lambda1)*v1[2] + bc[2], length = 0.1, col=2)
arrows(x0 = bc[1], y0 = bc[2], x1 = sqrt(lambda2)*v2[1] + bc[1],
y1=sqrt(lambda2)*v2[2] + bc[2], length = 0.1, col=2)

#-------------------------------------------------

rm(list=ls())
( S <- matrix(c(2.2, 0.4, 0.4, 2.8),nrow=2,ncol=2) )
eigen <- eigen(S)
( Lambda = diag(eigen$values) )
V = eigen$vectors
colnames(V) = c("v1","v2")
round(
  S - V %*% Lambda %*% t(V)
  , 8)
V %*% t(V)  - diag(c(1,1))
( SqrtS = V %*% Lambda^(1/2) %*% t(V) )
( InvS = V %*% diag( 1/diag(Lambda) ) %*% t(V) )
round(
  InvS %*% S
  , 8)
sum(diag(Lambda))
prod(diag(Lambda))

#-------------------------------------------------

rm(list=ls())
( X <-
matrix(c(2,3,3,4,4,5,6,6,7,8,7,8,10,6,8,10,12,13,11,12),nrow=10,ncol
=2) )
n <- nrow(X)
p <- ncol(X)
one.n <- matrix(rep(1,n),ncol=1)
I.n <- diag(rep(1,n))
H <- I.n - (1/n) * one.n %*% t(one.n)
Xtilde <- H %*% X
S = (1/n)* t(Xtilde) %*% Xtilde
eigen = eigen(S)
Lambda = diag(eigen$values)
V = eigen$vectors
( InvSqrtS = V %*% diag(diag(Lambda)^(-.5)) %*% t(V) )
Ztilde = Xtilde %*% InvSqrtS
plot(Ztilde, xlim=c(-4,13),ylim=c(-4,13),
     bg=heat.colors(n),pch=21,cex=2,asp=1)
abline(h=0)
abline(v=0)
round(
  (1/n) * t(Ztilde) %*% one.n
  ,8)
round(
  (1/n)* t(H %*% Ztilde)%*% (H %*% Ztilde)
  , 8)
( r = qr(Xtilde)$rank )
SVD=svd(Xtilde)
```

```
( Ur = SVD$u )
( Vr = SVD$v )
( Deltar = diag(SVD$d) )
round( Xtilde - Ur %*% Deltar %*% t(Vr) , 8)

#-----------------------------------------------

rm(list=ls())
marks <- read.table("http://www.maths.leeds.ac.uk/~charles/mva-data/
openclosedbook.dat",header = TRUE)
X = as.matrix(marks)
colnames(X) <- c("Mechanics", "Vectors", "Algebra", "Analysis",
"Statistics")
head(X)
n = nrow(X)
p = ncol(X)
A = diag(rep(1,p))
one.n = matrix(rep(1,n))
b = (1/n) * t(X) %*% one.n
Xtilde = X %*% t(A) + one.n %*% (-t(b))
a = matrix(rep(1/p, p), ncol=1)
y = X %*% a
S = (1/n) * t(Xtilde) %*% Xtilde
eigen = eigen(S)
Lambda = diag(eigen$values)
V = eigen$vectors
v1 = V[,1, drop=FALSE]
v1
y1 = Xtilde %*% v1
var(y1) * (n-1)/n # coincide con Lambda[1,1]
a = matrix(rep(1/sqrt(p), p), ncol=1)
y = Xtilde %*% a
var(y) * (n-1)/n
Y = Xtilde %*% V
round(
  (1/n) * t(Y) %*% one.n
  , 8)
S_Y = (1/n) * t(Y) %*% Y
round(
  S_Y
  , 8)
sum(diag(S_Y))
det(S_Y)
pca = princomp(X)
summary(pca)
pca$loadings[,]
head( pca$scores )
pca2 = prcomp(X, center = TRUE)
summary(pca2)
pca2$rotation[,]
head( pca2$x )
V[5,1]*pca$sdev[1]/sqrt(diag(S)[5])
cor(Xtilde[,"Statistics"],Y[,1])
summary(pca)
```

```
c = mean(pca$sdev^2)
pca$sdev^2 > c
plot(pca, type="line")
biplot(pca)


#-------------------------------------------------
rm(list=ls())
wine <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-
databases/wine/wine.data", header  = FALSE)
colnames(wine)<-c("Type","Alcohol",
                  "Malic","Ash",
                  "Alcalinity","Magnesium",
                  "Phenols","Flavanoids",
                  "Nonflavanoids","Proanthocyanins",
                  "Color_int","Hue","Dilution","Proline")
wine$Type = factor(wine$Type)
X = as.matrix(wine[,-1])
n = nrow(X)
p = ncol(X)
pca = princomp(X, cor=T)
V = pca$loadings
Y = pca$scores
q <- 10
cumsum(pca$sdev^2/sum(pca$sdev^2))[1:q]
plot(Y[,1:2], col=as.numeric(wine$Type), asp=1)

#-------------------------------------------------

rm(list=ls())
face <- read.table("https://raw.githubusercontent.com/aldosolari/AE/
master/docs/dati/face.txt", header=FALSE)
X = as.matrix(face)
n = nrow(face)
p = ncol(face)
image(X, col=gray(0:255/255), asp=p/n)
pca = princomp(X, cor=F)
V = pca$loadings
Y = pca$scores
xbar = matrix(pca$center, ncol=1)
q = 10
Yq = Y[,1:q]
Vq = V[,1:q]
Aq = Yq %*% t(Vq)
one.n = matrix(rep(1,n), ncol=1)
face2 = Aq + one.n %*% t(xbar)
face2 <- pmax(pmin(face2, 1), 0)
image(face2, col=gray(0:255/255), asp=p/n)
pixels = prod(dim(face))
pixels2 = prod(dim(Yq)) + prod(dim(Vq)) + prod(dim(xbar))
round(pixels/pixels2, 2)

#-------------------------------------------------
```

```
rm(list=ls())
X <- USArrests
n <- nrow(X)
p <- ncol(X)
states <- row.names(X)
names(X)
head(X)
apply(X, 2, mean)
apply(X, 2, var) * (n-1)/n
devstd <- sqrt( apply(X, 2, var) * (n-1)/n )
Z <- scale(X, center=TRUE, scale = devstd)
R <- cor(X)
R
eigenR <- eigen(R)
Lambda <- diag(eigenR$values)
Lambda
V <- eigenR$vectors
V
Y <- Z %*% V
head(Y)
ACP <- princomp(X, cor=TRUE)
names(ACP)
summary(ACP)
ACP$loadings[,]
ACP$scores
ACP2 <- prcomp(X, scale=TRUE)
names(ACP2)
ACP2$scale
devstd * sqrt(n/(n-1))
summary(ACP2)
ACP2$rotation[,]
ACP2$x
ACP3 <- prcomp(X, scale=devstd)
summary(ACP3)
ACP3$rotation[,]
ACP3$x
DVS = svd(Z)
U = DVS$u
Delta = diag(DVS$d)
diag(Delta)/sqrt(n)
DVS$v
U %*% Delta
biplot(ACP , scale = 0)
k = 1
sapply(1:4, function(j) V[j,k]*sqrt(diag(Lambda)[k]) )
cor(Y[,k], X)
q = 2
Yq = Y[,1:q, drop=FALSE]
Vq = V[,1:q, drop=FALSE]
Zapp = Yq %*% t(Vq)
sum(c(Z - Zapp)^2)
n * sum(diag(Lambda)[(q+1):p])
plot(UrbanPop ~ Assault, X, asp=1)
biplot(princomp(X), scale=0)
```

```
nomit <- 20
set.seed(123)
ina <- sample(seq(n), nomit)
inb <- sample(1:p, nomit, replace = TRUE)
Zna <- Z
index.na <- cbind(ina, inb)
Zna[index.na] <- NA
Zhat <- Zna
zbar <- colMeans(Zna, na.rm = TRUE)
Zhat[index.na] <- zbar[inb]
ismiss <- is.na(Zna)
fit.svd <- function(X, q = 1){
  DVS <- svd(X)
  with(DVS,
       u[,1:q, drop = FALSE] %*%
         (d[1:q] * t(v[,1:q, drop = FALSE]))
  )
}
cor(Zhat[ismiss], Z[ismiss])
plot(Zhat[ismiss], Z[ismiss], asp=1)
abline(a=0,b=1)
n_iter <- 10
for (iter in 1:n_iter){
  Zapp <- fit.svd(Zhat, q = 1)
  Zhat[ismiss] <- Zapp[ismiss]
  e <- mean(((Zna - Zapp)[!ismiss])^2)
  cat("Iter :", iter, " Errore :", e, "\n")
}
cor(Zhat[ismiss], Z[ismiss])
plot(Zhat[ismiss], Z[ismiss], asp=1)
abline(a=0,b=1)

#-------------------------------------------------

rm(list=ls())
paesi <- read.table("https://raw.githubusercontent.com/aldosolari/
AE/master/docs/dati/paesi.txt", header=TRUE)
X = paesi[,-c(1,11)]
n = nrow(X)
p = ncol(X)
km = kmeans(X, centers = X[c(1,25,26),], algorithm = "Lloyd")
table(km$cluster)
table(km$cluster, paesi$blocco)
km$centers
( W = km$tot.withinss )
( B = km$betweenss )
K = 3
(B/(K-1)) / (W/(n-K))
library(cluster)
D = dist(X, method="euclidean")
sil <- silhouette(x=km$cluster, dist=D)
row.names(sil) <- paesi$Country
plot(sil)
```

```
#-------------------------------------------------

rm(list=ls())
face <- read.table("https://raw.githubusercontent.com/aldosolari/AE/
master/docs/dati/face.txt", header=FALSE)
X = as.matrix(face)
n = nrow(face)
p = ncol(face)
X1 = matrix(c(X),ncol=1,nrow=n*p)
K = 3
set.seed(123)
km = kmeans(X1, centers = K)
for (k in 1:K){
  X1[km$cluster==k] = km$centers[k]
}
X2 = matrix(X1, ncol=p,nrow=n)
image(X2, col=gray(0:255/255), asp=p/n)

#-------------------------------------------------

rm(list=ls())
X <-
matrix(c(2,3,3,4,4,5,6,6,7,8,7,8,10,6,8,10,12,13,11,12),nrow=10,ncol
=2)
n <- nrow(X)
p <- ncol(X)
( D2 = dist(X, method="euclidean") )
b = matrix(c(-5,0.1),ncol=1)
one.n = matrix(rep(1,n),ncol=1)
Y = X + one.n%*%t(b)
plot(Y,xlim=c(-4,13),ylim=c(-4,13),asp=1,
     bg=heat.colors(n),pch=21,cex=2)
abline(h=0)
abline(v=0)
m = sqrt(2)
sum( dist(Y, method="minkowski", p=m) - dist(X, method="minkowski",
p=m) )
A=matrix(c(0,1,1,0),2,2)
Y = X%*%t(A)
plot(Y,xlim=c(-4,13),ylim=c(-4,13),asp=1,
     bg=heat.colors(n),pch=21,cex=2)
abline(h=0)
abline(v=0)
sum( dist(Y, method="euclidean") - D2 )
gradi = 30
theta = (pi/180)*gradi
A = matrix(c(cos(theta), -sin(theta), sin(theta),
cos(theta)),byrow=T,2,2)
Y = X%*%t(A)
plot(Y,xlim=c(-4,13),ylim=c(-4,13),asp=1,
     bg=heat.colors(n),pch=21,cex=2)
abline(h=0)
abline(v=0)
sum( dist(Y, method="euclidean") - D2 )
```

```
sum( dist(Y, method="manhattan") - dist(X, method="manhattan")  )

#-------------------------------------------------

rm(list=ls())
require("MASS")
X = as.matrix( log(Animals[-c(6,16,26),]) )
colnames(X) = c("logbody","logbrain")
n = nrow(X)
p = ncol(X)
xbar = matrix(colMeans(X), nrow=p, ncol=1)
S = var(X) * ((n-1)/n)
require(ellipse)
plot(X)
lines(ellipse(S,centre = t(xbar), t = 2.447))
InvS = solve(S)
t(X[1,] - xbar) %*% InvS %*% (X[1,] - xbar)
dM2 = apply(X,MARGIN=1, function(u) t(u-xbar) %*% InvS %*% (u -
xbar) )
q.95 = qchisq(0.95, df=2)
plot(dM2, xlab="unità statistica", ylab="distanza di Mahalanobis al
quadrato")
abline(h=q.95)
which(dM2 > q.95)
n * 0.05
Xtilde <- scale(X,center=TRUE,scale=FALSE)
eigenS = eigen(S)
InvSqrtS = eigenS$vectors %*% diag(eigenS$values^(-1/2)) %*%
t(eigenS$vectors)
Ztilde = Xtilde %*% InvSqrtS
dE2.Ztilde = apply(Ztilde,MARGIN=1, function(u) t(u) %*% u )
sum(dE2.Ztilde  - dM2)
Y = X
Y[,"logbody"] = X[,"logbody"] + log(1000)
n = nrow(Y)
p = ncol(Y)
ybar = matrix(colMeans(Y), nrow=p, ncol=1)
S.Y = var(Y) * ((n-1)/n)
InvS.Y = solve(S.Y)
dM2.Y = apply(Y,MARGIN=1, function(u) t(u-ybar) %*% InvS.Y %*% (u -
ybar) )
sum(dM2.Y - dM2)

#-------------------------------------------------

rm(list=ls())
X = matrix(c(1,3,2,4,1,5,5,5,5,7,4,9,2,8,3,10), ncol=2, nrow=8,
byrow=T)
n = nrow(X)
colnames(X) = c("x1","x2")
rownames(X) = 1:n
( D = dist(X,method="euclidean") )
hc.single=hclust(D, method="single")
hc.complete=hclust(D, method="complete")
```

```
hc.average=hclust(D, method="average")
op <- par(mfrow = c(1, 2))
plot(X)
text(x2~x1, X, labels=rownames(X), pos=3)
plot(hc.single, hang=-1)
par(op)
op <- par(mfrow = c(1, 2))
plot(X)
text(x2~x1, X, labels=rownames(X), pos=3)
plot(hc.average, hang=-1)
par(op)
op <- par(mfrow = c(1, 2))
plot(X)
text(x2~x1, X, labels=rownames(X), pos=3)
plot(hc.complete, hang=-1)
par(op)

#-------------------------------------------------

rm(list=ls())
data(eurodist)
hc.single=hclust(eurodist, method="single")
plot(hc.single, hang=-1)
hc.complete=hclust(eurodist, method="complete")
plot(hc.complete, hang=-1)
rect.hclust(hc.complete,k=5)
hc.average=hclust(eurodist, method="average")
plot(hc.average, hang=-1)

#-------------------------------------------------

rm(list=ls())
require(cluster)
data(flower)
str(flower)
row.names(flower) = c("begonia","broom","camellia","dahlia","forget-
me-not","fuchsia",

"geranium","gladiolus","heather","hydrangea","iris","lily",
                      "lily-of-the-valley","peony","pink
carnation","red rose","scotch rose", "tulip")
D = daisy(flower, metric="gower", type=list(symm=c(1,2),asymm=3))
hc.complete = hclust(D, method="complete")
plot(hc.complete, hang=-1)
K = 3
rect.hclust(hc.complete,k=K)

#-------------------------------------------------

rm(list=ls())
C1 <- read.table("http://azzalini.stat.unipd.it/Libro-DM/C1.dat",
sep="", header=TRUE)
plot(x2 ~ x1, col=gruppo, C1, pch=19)
D = dist(C1[,c("x1","x2")], method = "euclidean")
```

```
hc = hclust(D, method="complete")
plot(hc, labels=FALSE, hang=-1)
rect.hclust(hc,k=3)
g3 = cutree(hc, k=3)
plot(x2 ~ x1, col=g3, C1, pch=19)
table(g3, C1$gruppo)

#-------------------------------------------------

rm(list=ls())
C2 <- read.table("http://azzalini.stat.unipd.it/Libro-DM/C2.dat",
sep="", header = TRUE)
D = dist(C2[,c("x1","x2")], method = "euclidean")
hc = hclust(D, method="complete")
plot(hc, labels=FALSE, hang=-1)
rect.hclust(hc,k=3)
g3 = cutree(hc, k=3)
plot(x2 ~ x1, col=g3, C2, pch=19)

#-------------------------------------------------

rm(list=ls())
data(iris)
head(iris)
X = iris[,1:4]
n = nrow(X)
D = dist(X, method="euclidean")
hc = hclust(D, method="average")
plot(hc, hang= -1, label=iris$Species)
rect.hclust(hc, k=3)
hc3 = cutree(hc, k=3)
table(hc3, iris$Species)
require(cluster)
sil <- silhouette(hc3, dist=D)
plot(sil)
plot(sil, col=iris$Species)
K = 3
km <- kmeans(X, centers = X[c(25,75,125),], nstart=1, algorithm =
"Lloyd")
km$centers
W = km$tot.withinss
B = km$betweenss
km$totss
km3 = km$cluster
table(km3, iris$Species)
plot(X[c("Sepal.Length", "Sepal.Width")], col=km$cluster)
points(km$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3,
pch=23, cex=3)
S = var(X)*((n-1)/n)
Z = scale(X, center=TRUE, scale=diag(S)^(1/2))
pca <- prcomp(Z, center = FALSE, scale. = FALSE)
Y <- pca$x[,1:2]
km.pca <- kmeans(Y, centers = Y[c(25,75,125),], nstart=1, algorithm
= "Lloyd")
```

```
table(km.pca$cluster, iris$Species)
```