```
# Analisi Esplorativa
# Codice messo a disposizione per la prova scritta e di Laboratorio

n <- nrow(X)
p <- ncol(X)
t(X)
summary(X)
mean(X[,1])
apply(X, MARGIN=2, FUN="mean")
var(X)
cor(X)
plot(X)
boxplot(x)
boxplot.stats(x)
bagplot(X)
bagplot(X)$pxy.outlier

#=======================

cloud(x3 ~ x1 + x2, group=x4, data=dati)
faces(dati, scale=TRUE)
stars(dati, scale=TRUE)
co.intervals(X[,1], number=4, overlap=0)
coplot(x3 ~ x2 | x1, data = dati)

#=======================

t(x) %*% x
acos(cor(X)[1,2])
acos(cor(X)[1,2])*(180/pi)

#=======================

matrix(rep(1,n),ncol=1)
(1/n) * t(X) %*% one.n
diag(rep(1,n))
I.n - (1/n) * one.n %*% t(one.n)
H %*% X
diag(diag(S)^(1/2))
diag(diag(S)^(-1/2)) %*% S %*% diag(diag(S)^(-1/2))

#=======================

sum(diag(S))
det(S)
det(S)/prod(diag(S))
eigen(S)
eigen(S)$values
eigen(S)$vectors
eigen(S)$vectors[,1,drop=F]
bc = colMeans(X)
library(ellipse)
lines(ellipse(x=S, centre = bc))
```

```
#=======================

sum(eigen(S)$values)
prod(eigen(S)$values)
V %*% Lambda %*% t(V)
V %*% Lambda^(1/2) %*% t(V)
V %*% diag( 1/diag(Lambda) ) %*% t(V)
qr(X)$rank
svd(X)

#=======================

X %*% t(A) + one.n %*% t(b)
princomp(X)
princomp(X)$sdev
princomp(X)$loadings
princomp(X)$scores
princomp(X)$n.obs
princomp(X)$center
princomp(X)$scale
summary(princomp(X))
biplot(princomp(X))
prcomp(X)
prcomp(X)$rotation
prcomp(X)$x
prcomp(X)$sdev
prcomp(X)$center
prcomp(X)$scale
V[5,1]*princomp(X, cor=F)$sdev[1]/sqrt(diag(S)[5])
cor(Xtilde[,5],Y[,1])
image(X, col=gray(0:255/255), asp=p/n)
Yq = Y[,1:q]
Vq = V[,1:q]
Aq = Yq %*% t(Vq)
Aq + one.n %*% t(xbar)
prod(dim(face))
prod(dim(Yq)) + prod(dim(Vq)) + prod(dim(xbar))
object.size(X)
object.size(Yq) + object.size(Vq) + object.size(xbar)

#=======================

kmeans(X, centers = K)
kmeans(X, centers = K)$cluster
kmeans(X, centers = K)$centers
kmeans(X, centers = K)$totss
kmeans(X, centers = K)$withinss
kmeans(X, centers = K)$tot.withinss
kmeans(X, centers = K)$betweenss
kmeans(X, centers = K)$size
kmeans(X, centers = K)$iter
dist(X, method="euclidean")
library(cluster)
silhouette(x=clusters, dist=D)
```

```
summary(silhouette(x=clusters, dist=D))$avg.width

#======================

dist(X)
dist(X, diag=TRUE, upper=TRUE)
dist(X, method="manhattan")
dist(X, method="maximum")
dist(X, method="minkowski", p=m)
A = matrix(c(cos(theta), -sin(theta), sin(theta),
cos(theta)),byrow=TRUE,2,2)
Y = X%*%t(A)
t(X[1,] - colMeans(X)) %*% solve(S) %*% (X[1,] - colMeans(X))
apply(X, MARGIN=1, function(u) t(u-colMeans(X)) %*% solve(S) %*% (u
- colMeans(X)) )

#======================

hclust(D, method="single")
hclust(D, method="complete")
hclust(D, method="average")
plot(hclust(D))
plot(hclust(D), hang=-1)
rect.hclust(hclust(D),k=K)
cutree(hclust(D), k=K)
library(cluster)
daisy(X, metric="gower", type=list(symm=c(1,2),asymm=3))

#======================

factanal(X, factors = K)
factanal(X, factors = K, method="mle")
factanal(X, factors = K)$loadings
factanal(X, factors = K)$uniqueness
factanal(X, factors = K)$correlation
factanal(X, factors = K)$factors
factanal(X, factors = K)$dof
factanal(X, factors = K)$n.obs
factanal(X, factors = K)$rotmat
factanal(X, factors = K, rotation="none")
factanal(X, factors = K, rotation="varimax")
factanal(covmat=S, factors=K, n.obs=n)
factanal(X, factors=K)$PVAL
Lambda%*%t(Lambda) + Psi
n*log(det(fit)/det(R))
((p-k)^2 - p - k)/2
qchisq(1-alpha, df=((p-k)^2 - p - k)/2)
pchisq(t, lower.tail=TRUE, df=((p-k)^2 - p - k)/2)
1-1/(diag(solve(R)))
apply(abs( R - diag(rep(1,p)) ), 2, max)
```