1. **VETTORI ALEATORI GAUSSIANI NEL PIANO**

n=3000

X=rnorm(n)

Y=rnorm(n)

plot(X,Y)



N=3000

X=rnorm(n,5,5)

Y=rnorm(n)

plot(X,Y,asp=1)



Nota: si provi senza asp=1.

n=3000

X1=rnorm(n,5,5)

X2=rnorm(n)

a=pi/4

X=cos(a)\*X1 - sin(a)\*X2

Y=sin(a)\*X1 + cos(a)\*X2

plot(X,Y,asp=1)



Nota: questo disegno viene bene anche senza asp=1.

1. **VETTORI GAUSSIANI NELLO SPAZIO**

require(rgl)

(la prima volta bisogna preliminarmente installare il package rgl, usando il menù “pacchetti” dalla schermata di R).

n=1000

X=rnorm(n); Y=rnorm(n); Z=rnorm(n)

plot3d(X,Y,Z,box=FALSE)

axes3d(c('x','y','z'),pos=c(0,0,0),labels=FALSE,tick=FALSE,col=2)

Si veda la figura 3D. La si può prima ingrandire poi ruotare per vedere la nuvola di punti sotto varie angolature.

Provare poi con:

X=rnorm(n,0,3); Y=rnorm(n,0,1); Z=rnorm(n,0,0.1)

plot3d(X,Y,Z,xlim=c(-10,10),ylim=c(-10,10),zlim=c(-10,10),box=FALSE)

axes3d(c('x','y','z'),pos=c(0,0,0),labels=FALSE,tick=FALSE, col=2)

osservando come i punti si vedono più o meno distinti a seconda delle angolature.

1. **SULLA MATRICE DI COVARIANZA**

Poniamoci il problema di calcolare la matrice di covarianza del vettore aleatorio (X,Y,Z) della sezione precedente.

A=matrix(nrow=n,ncol=3); A[,1]=X;A[,2]=Y;A[,3]=Z

Q=cov(A)

> Q

 [,1] [,2] [,3]

[1,] 4.9018360 3.94096757 -0.02597830

[2,] 3.9409676 4.97743108 -0.01343499

[3,] -0.0259783 -0.01343499 0.99159805

Dalla teoria sappiamo che esiste una base ortonormale di auto vettori di Q, con auto valori non negativi. Ecco come si trovano:

> eigen(Q)

$values

[1] 8.8808805 1.0046489 0.9853357

$vectors

 [,1] [,2] [,3]

[1,] 0.703706348 0.5848376 0.4034382

[2,] 0.710482185 -0.5820647 -0.3954944

[3,] -0.003527116 -0.5649475 0.8251193

La matrice eigen(Q)$vectors è quella di cambio di base, aventi come componenti quelle dei vettori di una base rispetto all’altra (ci riferiamo alla base canonica ed alla base degli auto vettori), a seconda che si leggano le colonne o le righe della matrice. Essa è una matrice ortogonale, come si verifica coi seguenti comandi:

U=eigen(Q)$vectors

> U

 [,1] [,2] [,3]

[1,] 0.703706348 0.5848376 0.4034382

[2,] 0.710482185 -0.5820647 -0.3954944

[3,] -0.003527116 -0.5649475 0.8251193

> t(U)

 [,1] [,2] [,3]

[1,] 0.7037063 0.7104822 -0.003527116

[2,] 0.5848376 -0.5820647 -0.564947544

[3,] 0.4034382 -0.3954944 0.825119284

> solve(U)

 [,1] [,2] [,3]

[1,] 0.7037063 0.7104822 -0.003527116

[2,] 0.5848376 -0.5820647 -0.564947544

[3,] 0.4034382 -0.3954944 0.825119284

Le matrici t(U) e solve(U) coincidono.

1. **IL METODO DELLE COMPONENTI PRINCIPALI**

 IB <- read.table ('clipboard', header=TRUE)

 PLIC SC SA.SC TD TMI

Piem 0.088 0.471 -0.707 -0.607 -0.3950

Vaos -1.545 0.348 -0.642 -0.813 1.5780

Lomb 0.202 1.397 -0.836 -0.790 -0.5380

TrAA 0.677 0.435 -1.269 -0.966 -0.0750

Vene 0.088 1.334 -1.210 -0.848 -0.4970

FrVG 0.639 -0.005 -1.028 -0.804 -1.3010

Ligu 1.190 -0.247 0.470 -0.429 -0.3540

EmRo 0.658 1.177 -1.315 -0.863 -0.3470

Tosc 0.126 1.092 -0.795 -0.644 -1.3550

Umbr -1.431 0.675 -0.140 -0.524 -1.2870

Marc 0.278 1.090 -0.265 -0.702 -0.0006

Lazi 2.329 0.546 -0.080 -0.113 -0.0140

Abru 0.335 -0.373 0.402 -0.456 0.0400

Moli 0.658 -1.289 0.065 0.451 -1.1510

Camp -1.811 -1.314 2.031 1.664 0.4140

Pugl -0.766 -0.926 1.038 0.648 1.1090

Basi -0.747 -1.154 0.661 0.844 2.0010

Cala -0.500 -1.727 1.571 2.153 0.6320

Sici -0.918 -1.130 1.332 1.517 1.7830

Sard 0.449 -0.403 0.717 1.285 -0.2380

Ciascuna regione è come un punto in uno spazio a 5 dimensioni. In analogia coi disegni precedenti in 3 dimensioni, cerchiamo un’angolazione secondo la quale i 20 punti sono particolarmente sparpagliati e ben visibili, in modo da cogliere eventuali strutture.

P = princomp(IB)

biplot(P)



Si possono fare considerazioni su tre livelli: i) circa la collocazione dei punti (le regioni), gli uni rispetto agli altri; ii) circa l’allineamento o meno tra le frecce rosse (le variabili PLIC ecc.); iii) circa la posizione dei punti rispetto alle frecce rosse.