**PRIMA PARTE**: Esempio di PCA

Consideriamo nuovamente la tabella

 PLIC SC SA.SC TD TMI

Piem 0.088 0.471 -0.707 -0.607 -0.395

Vaos -1.545 0.348 -0.642 -0.813 1.578

Lomb 0.202 1.397 -0.836 -0.790 -0.538

TrAA 0.677 0.435 -1.269 -0.966 -0.075

Vene 0.088 1.334 -1.210 -0.848 -0.497

FrVG 0.639 -0.005 -1.028 -0.804 -1.301

Ligu 1.190 -0.247 0.470 -0.429 -0.354

EmRo 0.658 1.177 -1.315 -0.863 -0.347

Tosc 0.126 1.092 -0.795 -0.644 -1.355

Umbr -1.431 0.675 -0.140 -0.524 -1.287

Marc 0.278 1.090 -0.265 -0.702 -0.0006

Lazi 2.329 0.546 -0.080 -0.113 -0.014

Abru 0.335 -0.373 0.402 -0.456 0.040

Moli 0.658 -1.289 0.065 0.451 -1.151

Camp -1.811 -1.314 2.031 1.664 0.414

Pugl -0.766 -0.926 1.038 0.648 1.109

Basi -0.747 -1.154 0.661 0.844 2.001

Cala -0.500 -1.727 1.571 2.153 0.632

Sici -0.918 -1.130 1.332 1.517 1.783

Sard 0.449 -0.403 0.717 1.285 -0.238

Che carichiamo in R col solito comando A <- read.table("clipboard")

La tabella è già standardizzata in origine, altrimenti era preferibile standardizzarla, prima di eseguire PCA.

A differenza della regressione, non serve isolare le variabili in diversi vettori.

Calcoliamo B=princomp(A); biplot(B)



Possiamo fare le varie osservazioni riportate sulle dispense.

Col comando plot(B) si ottiene un grafico delle varianze:



Che indica come 2 o 3 dimensioni descrivano perfettamente i dati. Numericamente si leggono i valori con summary(B)

> summary(B)

Importance of components:

 Comp.1 Comp.2 Comp.3 Comp.4 Comp.5

Standard deviation 1.7779900 0.9103965 0.7114299 0.40172430 0.29690293

Proportion of Variance 0.6661239 0.1746455 0.1066499 0.03400577 0.01857485

Cumulative Proportion 0.6661239 0.8407694 0.9474194 0.98142515 1.00000000

>

Può anche essere carino mostrare il grafico di questi valori:

plot(c(0.6661239, 0.1746455, 0.1066499, 0.03400577, 0.01857485),type="b")



plot(c(0,0.6661239, 0.8407694, 0.9474194, 0.98142515, 1.00000000),type="b")



Nel grafico cumulativo abbiamo artificialmente inserito uno zero iniziale, per ovvie ragioni.

**SECONDA PARTE**: punteggi secondo la componente principale

Q=cov(A); e1 = eigen(Q)$vector[,1]

> e1

[1] -0.3100934 -0.4911865 0.5122889 0.5060663 0.3795192

>

Ora bisogna moltiplicare le righe della tavola A (cioè gli individui, le regioni) per e1. Il problema è che A è una tavola, non una matrice e gli usuali comandi di moltiplicazione vettoriale non funzionano. Bisogna tradurla in matrice. Basta introdurre:

AA <- as.matrix(A).

Visualizzando AA si vede che non è cambiato nulla, apparentemente, ma ora le moltiplicazioni vettoriali funzionano. Se calcoliamo

AA%\*%e1

Stiamo moltiplicando ogni riga di AA per e1, cioè che desideriamo. Otterremo i punteggi desiderati.

> AA%\*%e1

 [,1]

Piem -1.07791766

Vaos 0.16672139

Lomb -1.78107369

TrAA -1.59101800

Vene -1.92016589

FrVG -1.62295857

Ligu -0.35836460

EmRo -2.02425630

Tosc -1.82287237

Umbr -0.71314763

Marc -1.11284211

Lazi -1.09387733

Abru 0.06968593

Moli 0.25380602

Camp 3.24667234

Pugl 1.97294394

Basi 2.32362993

Cala 3.13754862

Sici 2.96646072

Sard 0.98599700

>

Osserviamo che, secondo la nostra logica, avremmo dovuto ottenere elevati punteggi per le regioni del nord. Ma la direzione del vettore e1 non è dettata dal significato che gli attribuiamo, solo da accidenti matematici. Ad esempio, in e1 confluisce TD che attribuisce valori bassi alle regioni del nord. C’è quindi una discrepanza tra la nostra attesa interpretativa di e1 ed il segno dei suoi valori. Basta cambiare di segno:

> -AA%\*%e1

 [,1]

Piem 1.077

Vaos -0.166

Lomb 1.781

TrAA 1.591

Vene 1.920

FrVG 1.622

Ligu 0.358

EmRo 2.024

Tosc 1.822

Umbr 0.713

Marc 1.112

Lazi 1.093

Abru -0.069

Moli -0.253

Camp -3.246

Pugl -1.972

Basi -2.323

Cala -3.137

Sici -2.966

Sard -0.985

>

(lo scorciamento tre cifre decimali, che aumenta la leggibilità, è stato fatto a mano).

Vediamo che EmRo 2.024, Vene 1.920, Tosc 1.822, Lomb 1.781, FrVG 1.622, TrAA 1.591 spiccano in positivo, mentre Camp -3.246, Cala -3.137, Sici -2.966, Basi -2.323, Pugl -1.972 in negativo.