**PRIMA PARTE**: comando predict per 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")

Ricordiamo i comandi:

B<-princomp(A)

biplot(B)

Ricordiamo anche (vista l’ultima lezione) che plot(B) mostra graficamente le varianze ovvero gli auto valori di Q, mentre summary mostra varie cose collegate; ed insieme danno l’idea della dimensione dei dati.

Abbiamo visto anche come dare i punteggi rispetto ad una componente. C’è però un comando che trova tutti i punteggi contemporaneamente:

predict(B)

> predict(B)

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

Piem -1.07816625 -0.227975683 0.10130876 0.15732378 0.07060086

Vaos 0.16647280 -2.347119067 -0.61724620 0.38991111 -0.03377744

Lomb -1.78132227 -0.442861548 0.08302592 -0.47516356 0.06210484

TrAA -1.59126658 -0.124984546 -0.59244133 0.51158847 0.19172147

Vene -1.92041447 -0.625980285 0.06342820 -0.23174255 0.28429627

FrVG -1.62320716 0.612348748 0.49322164 0.73632698 0.09902350

Ligu -0.35861319 1.127220367 -0.35178036 0.07369344 -0.69230494

EmRo -2.02450489 -0.231519801 -0.38220653 -0.10240697 0.32362954

Tosc -1.82312096 -0.002101653 0.82495472 -0.27443740 0.11231684

Umbr -0.71339622 -0.924685298 1.73816079 -0.14177582 -0.22233672

Marc -1.11309070 -0.375421935 -0.30701486 -0.52591362 -0.27968433

Lazi -1.09412591 1.579134605 -1.31879585 -0.55255650 -0.08270458

Abru 0.06943734 0.315193975 -0.20725227 0.27469434 -0.63242045

Moli 0.25355743 1.544286535 0.68456477 0.82302965 0.17844701

Camp 3.24642376 -0.263411524 1.14783883 -0.31074836 -0.21808732

Pugl 1.97269536 -0.380289416 -0.24709228 0.09005900 -0.25517075

Basi 2.32338134 -0.710371426 -0.99945223 0.34045171 0.15455746

Cala 3.13730003 0.813742449 0.26220721 -0.07269702 0.39375787

Sici 2.96621213 -0.421620637 -0.57037426 -0.20691747 0.16803474

Sard 0.98574841 1.086416139 0.19494534 -0.50271921 0.37799612

Il comando predict ha bisogno di due input: il modello, che qui è B, ed i dati da predire. Non scrivendo niente, si sottintende A. Provare con predict (B,A) il risultato è identico. Se invece ritagliamo una regione:

molise= A[14,]

> molise

 PLIC SC SA.SC TD TMI

Moli 0.658 -1.289 0.065 0.451 -1.151

molise.basi= A[c(14,17),]

> molise.basi

 PLIC SC SA.SC TD TMI

Moli 0.658 -1.289 0.065 0.451 -1.151

Basi -0.747 -1.154 0.661 0.844 2.001

>

Poi possiamo eseguire predict (B,molise)

> predict (B,molise)

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

Moli 0.2535574 1.544287 0.6845648 0.8230296 0.178447

>

Ecc.

I nome “predict” può apparire innaturale, usato nel modo precedente (prediciamo cose già note).

Sviluppiamo però un esempio con dati “training” e dati “test”. Supponiamo di conoscere inizialmente solo i valori sperimentali per alcune regioni italiane:

> A.meno = A[c(1,2,3,4,5,7,8,9,11,12,13,15,16,17,18,19,20),]

> A.meno

 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

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

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

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

>

eseguiamo PCA con esse

B.meno<-princomp(A.meno)

biplot(B.meno)



 e troviamo quindi un modello, le direzioni principali associate a tali regioni e tali variabili; la prima ad es. è il “benessere economico”. Poi, successivamente, riusciamo a misurare le 5 variabili anche per alcune delle altre regioni. Invece che creare un nuovo modello PCA includendo anche queste, usiamo il vecchio modello e facciamo una “previsione” del benessere economico, basata su tale modello, per le nuove regioni, cioè calcoliamo la proiezione di esse su e1. Ma usando il comando predict:

molise= A[14,]

> molise

 PLIC SC SA.SC TD TMI

Moli 0.658 -1.289 0.065 0.451 -1.151

predict (B.meno,molise)

> predict (B.meno,molise)

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

Moli 0.1837829 1.45079 -0.5641611 0.4922538 1.232639

**SECONDA PARTE**: Loadings di PCA

Ora vediamo i loadings

B$loadings

> B$loadings

Loadings:

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

PLIC -0.310 0.769 -0.553

SC -0.491 -0.309 -0.813

SA.SC 0.512 0.216 0.120 -0.433 -0.699

TD 0.506 0.279 0.115 -0.381 0.713

TMI 0.380 -0.435 -0.816

Discutiamoli, sempre alla luce dell’interpretazione delle componenti principali.

Infine, esaminiamo le componenti principali con un nuovo sistema.

**TERZA PARTE**: Esplorazione approfondita di PCA

Infine, esaminiamo le componenti principali con un nuovo sistema.

C<-predict(B)

i=1; j=5

plot(C[,c(i,j)],type="n",asp=1)

text(C[,c(i,j)],labels=as.character(row.names(A)))

i=1; j=5

plot(C[,c(i,j)],type="n")

text(C[,c(i,j)],labels=as.character(row.names(A)))