**PRIMA PARTE**: Esplorazione approfondita 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")

B<-princomp(A)

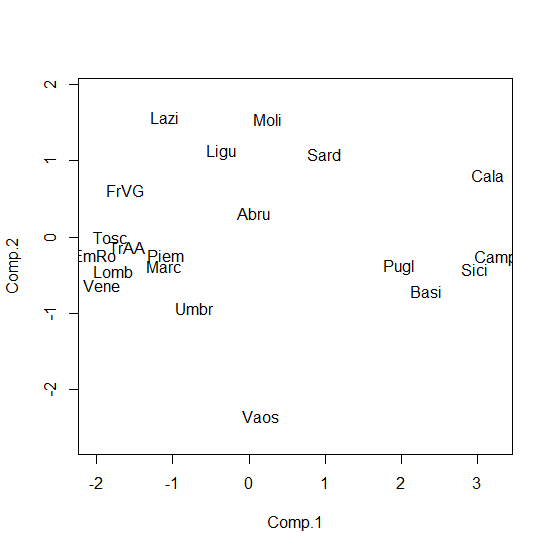
C<-predict(B)

Usiamo i seguenti comandi con varie combinazioni di i e j, per visualizzare la struttura ellissoidale dei dati, il loro allungamento, immaginare come si collocano nello spazio a 5 dimensioni (almeno nei vari spazi a tre dimensioni):

i=1; j=2

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

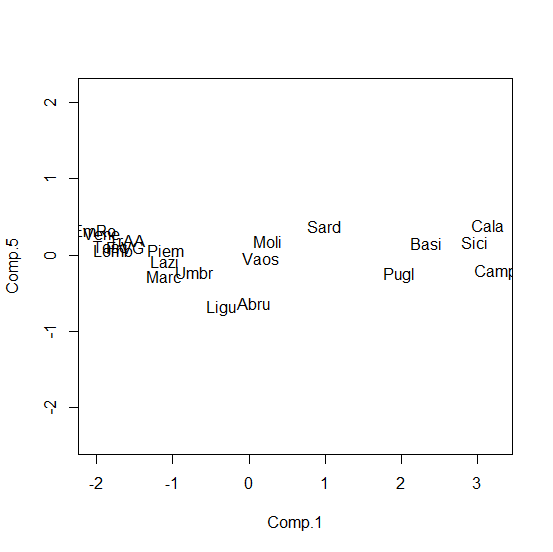
text(C[,c(i,j)],labels=as.character(row.names(A)))



i=1; j=5

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

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



E così via. I seguenti comandi non impongono la stessa unità di misura sugli assi e quindi distorcono l’idea dell’ellissoide:

i=1; j=5

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

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

Proviamo invece a disegnare punti e nomi:

i=1; j=2

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

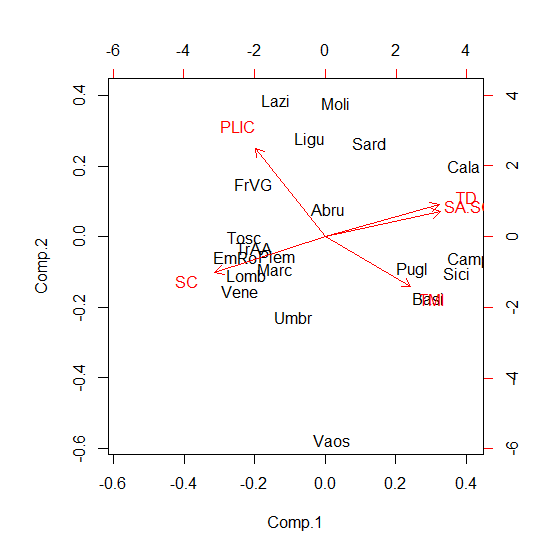
text(C[,c(i,j)],labels=as.character(row.names(A)))

**SECONDA PARTE**: Esempio di esplorazione: FRVG

Tramite questi comandi possiamo farci un’idea di dove sia collocato FRVG, riguardo al quale biplot sembra dare un assurdo.

Il comando B$loadings ed il solito biplot(B) permettono di ragionare congiuntamente con i comandi precedenti, per farsi un’idea sempre più precisa.

biplot(B)



Sembra mostrare FRVG nella zona positiva di SC, ma

> A[6,]

PLIC SC SA.SC TD TMI

FrVG 0.639 -0.005 -1.028 -0.804 -1.301

>

Il valore è circ zero, persino negativo. Come si spiega? Vediamo come sono collocati, rispetto al piano principale (non su di esso) SC e FRVG.

> 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

mostra che SC ha un’elevata componente (negativa) nella direzione 4; d’altro canto

> C

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

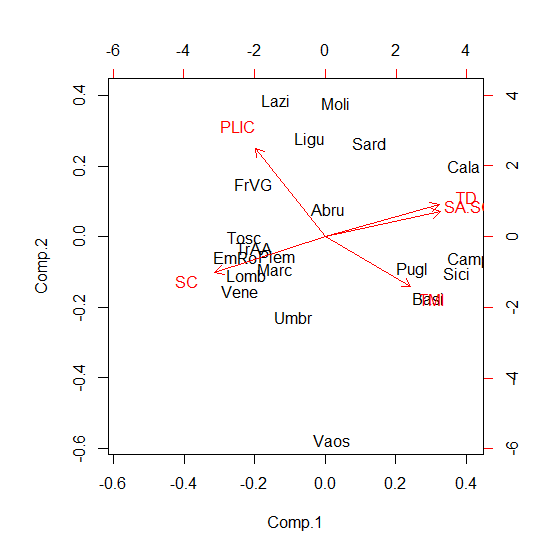
Cala 3.13730003 0.813742449 0.26220721 -0.07269702 0.39375787

Sici 2.96621213 -0.421620637 -0.57037426 -0.20691747 0.16803474

Sard 0.98574841 1.086416139 0.19494534 -0.50271921 0.37799612

>

FRVG ha una componente marcatamente positiva nella direzione 4. Si immagini lo spazio a tre dimensioni formato dalle componenti 1, 2 4. Torniamo a biplot(B):



e capiamo che SC è diretto in buona misura perpendicolarmente al piano principale e FRVG si trova dall’altro lato, quindi nella parte negativa di SC. Tuttavia, la proiezione di entrambi sul piano è orientata dalla stessa parte.

La posizione di FRVG rispetto a SC è spiegata. Ma allora, il cluster TOSC, TRAA ecc. può essere un’analoga finzione dovuta alla proiezione? Magari sono punti lontani tra loro, che sembrano vicini a causa della proiezione. Vediamo. Aggiungiamo una colorazione per distinguere i disegni.

PLIC SC SA.SC TD TMI col

Piem 0.088 0.471 -0.707 -0.607 -0.395 1

Vaos -1.545 0.348 -0.642 -0.813 1.578 2

Lomb 0.202 1.397 -0.836 -0.790 -0.538 1

TrAA 0.677 0.435 -1.269 -0.966 -0.075 1

Vene 0.088 1.334 -1.210 -0.848 -0.497 1

FrVG 0.639 -0.005 -1.028 -0.804 -1.301 3

Ligu 1.190 -0.247 0.470 -0.429 -0.354 2

EmRo 0.658 1.177 -1.315 -0.863 -0.347 1

Tosc 0.126 1.092 -0.795 -0.644 -1.355 1

Umbr -1.431 0.675 -0.140 -0.524 -1.287 2

Marc 0.278 1.090 -0.265 -0.702 -0.0006 1

Lazi 2.329 0.546 -0.080 -0.113 -0.014 2

Abru 0.335 -0.373 0.402 -0.456 0.040 2

Moli 0.658 -1.289 0.065 0.451 -1.151 2

Camp -1.811 -1.314 2.031 1.664 0.414 2

Pugl -0.766 -0.926 1.038 0.648 1.109 2

Basi -0.747 -1.154 0.661 0.844 2.001 2

Cala -0.500 -1.727 1.571 2.153 0.632 2

Sici -0.918 -1.130 1.332 1.517 1.783 2

Sard 0.449 -0.403 0.717 1.285 -0.238 2

A.col <- read.table("clipboard")

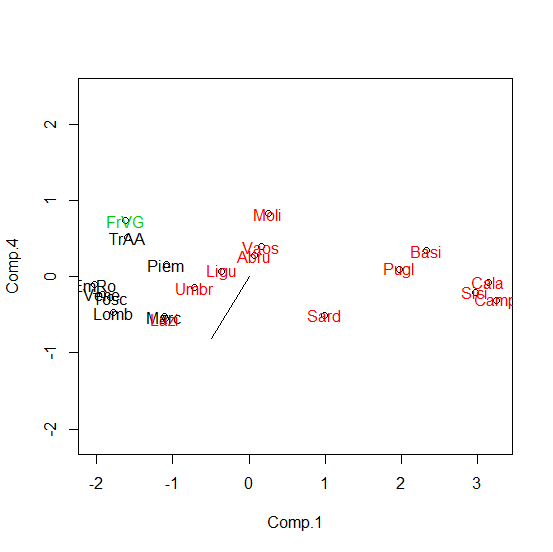
Ad esempio:

i=1; j=4

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

text(C[,c(i,j)],labels=as.character(row.names(A)),col=A.col[,6])

lines(c(0,B$loadings[2,i]), c(0,B$loadings[2,j]))



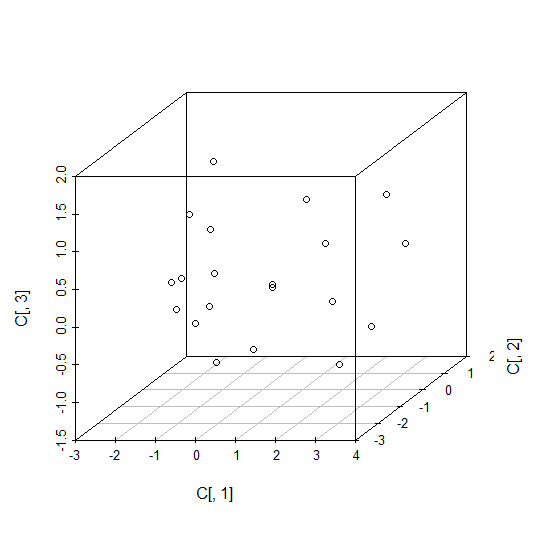
**TERZA PARTE**: scatterplot 3D

Bisogna innanzi tutto scaricare il package scatterplot3D. Dopo averlo trovato su R (alla pag. <http://cran.r-project.org/web/packages/scatterplot3d/index.html>) si esegue il download e lo si inserisce in library. A questo punto è apribile. Serve il comando

Require(scatterplot3D)

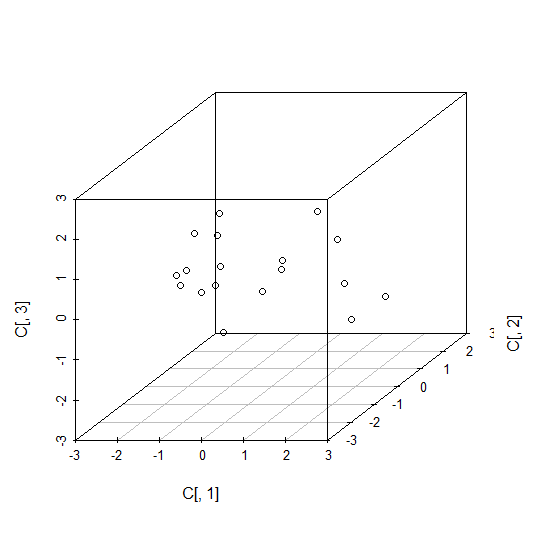
Dopo il quale funzionano i suoi comandi. Ecco il risultato:

scatterplot3d(C[,1], C[,2], C[,3])



La scala non è uniforme. Scriviamo allora

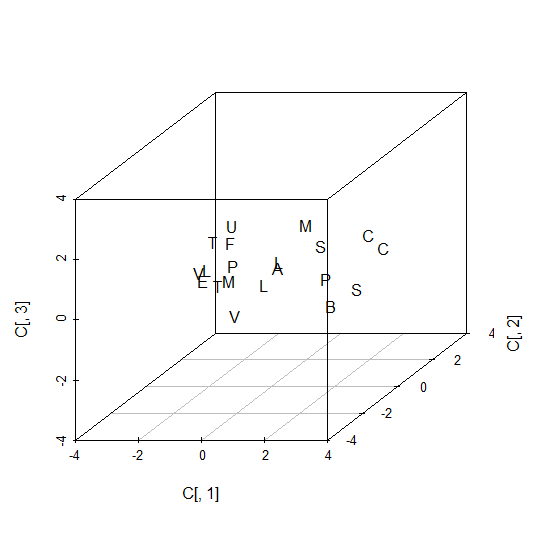
scatterplot3d(C[,1], C[,2], C[,3], xlim=c(-3,3), ylim=c(-3,3), zlim=c(-3,3))



Il problema sono i nomi.

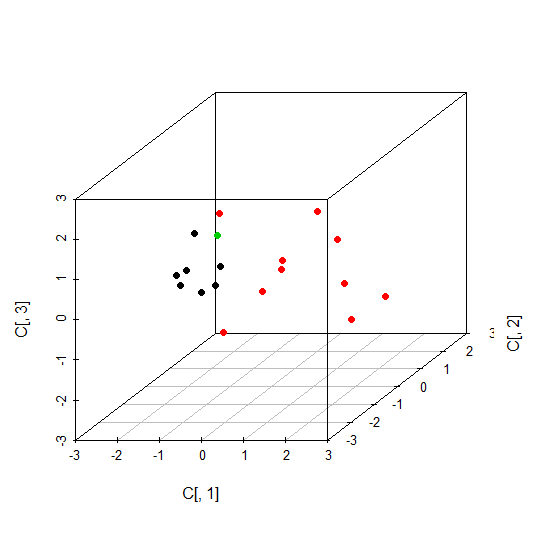
scatterplot3d(C[,1], C[,2], C[,3], pch=row.names(A), xlim=c(-4,4), ylim=c(-4,4), zlim=c(-4,4))

rimedia un po’ ma il risultato è scarso.



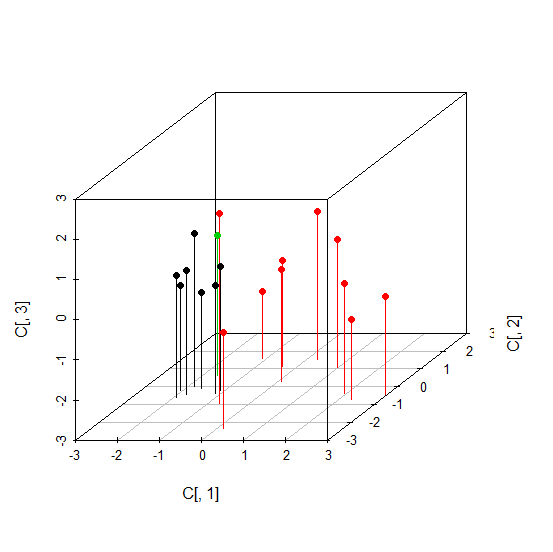
Invece purtroppo l’aggiunta dei nomi con text… funziona male.

scatterplot3d(C[,1], C[,2], C[,3], xlim=c(-3,3), ylim=c(-3,3), zlim=c(-3,3), pch=16,A.col[,6])



aggiunge i colori, riempie i pallini ed aiuta un po’. Sicuramente conferma l’esistenza di un cluster.

scatterplot3d(C[,1], C[,2], C[,3], xlim=c(-3,3), ylim=c(-3,3), zlim=c(-3,3), pch=16, type="h", A.col[,6])



È forse la più ricca di informazione.