1. **CLUSTER ANALYSIS**

Carichiamo la tabella in rete relativa a questa esercitazione (è un progettino di Matteo Vicini e Andrea Ammannati, gentilmente offerto per lo svolgimento di corsi successivi):

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

Parte 1: analisi preliminari, di tipo grafico

plot(E)



PCA = princomp(E); biplot(PCA)



Non è molto “trasparente”, forse per mancanza di standardizzazione (es. le variabili accidents ed export non si vedono).

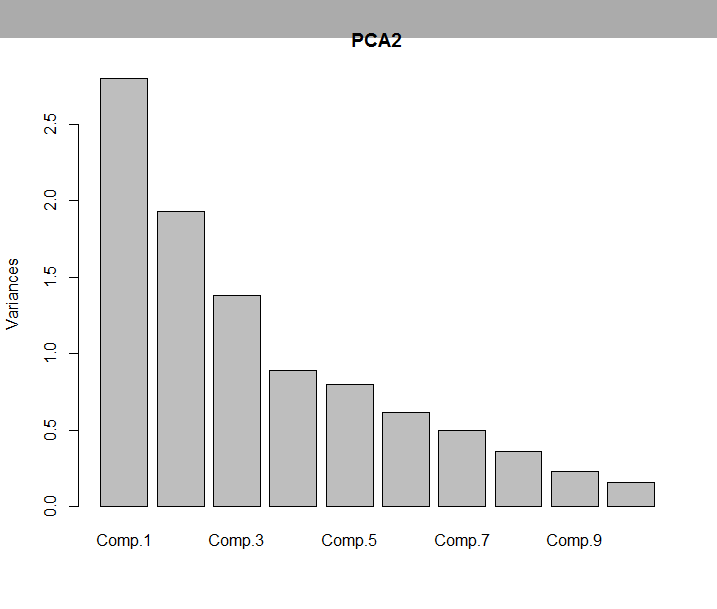
E2 =E

for (i in 1:10) {E2[,i] = (E[,i] - mean(E[,i])) / (sd(E[,i]))}

PCA2 = princomp(E2); biplot(PCA2)



plot(PCA2)



Parte 2: cluster analysis

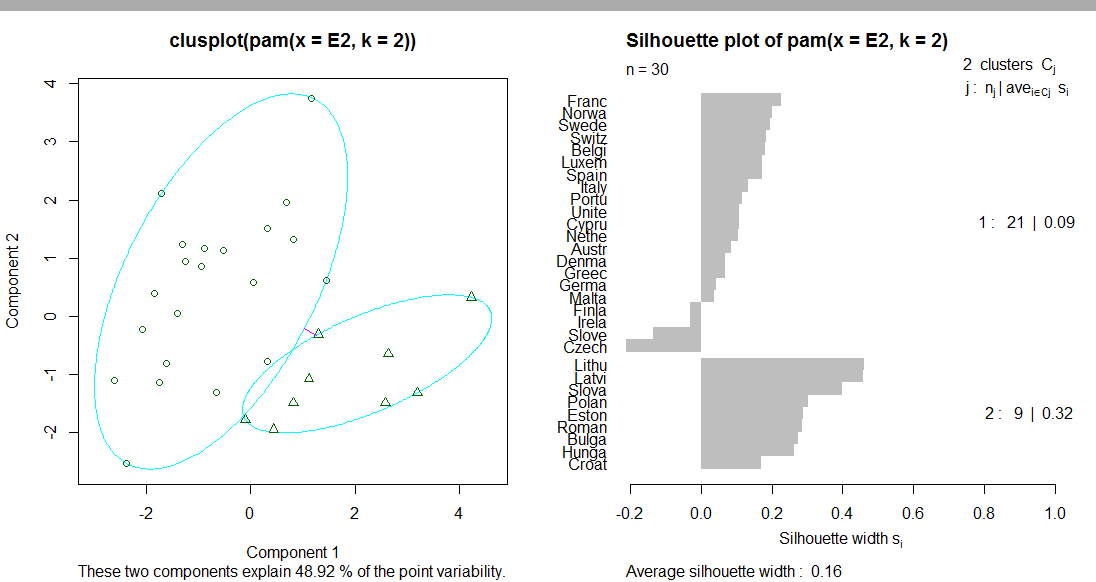
require(cluster)

PAM=pam(E2,2)

plot(PAM)

par(mfrow=c(1,2))

plot(pam(E2,2))



Chiedendo 2 clusters, abbiamo una silhouette media pari solo a 0.16. Le prime due componenti principali spiegano il 48.9% della variabilità (pam esegue una PCA). Vediamo se la silhouette media migliora aumentando il numero di clusters. Prima però vediamo anche se la suddivisione in 2 insegna qualcosa. Il blocco dell’EST è separato da quello più occidentale.

plot(pam(E2,3))



La silhouette media non è cambiata. Ora si è separato il blocco del sud da quello principale!

Provando con

plot(pam(E2,4))

e così via si vede che c’è un lievissimo miglioramento verso k=6-7, ma d’altra parte l’interpretazione non è più così evidente quindi non è chiaro che sia davvero meglio.

Altra considerazione: alcune nazioni hanno una pessima silhouette, cioè sono classificate male, ad es. Czech Rep. e Slovenia. Il fenomeno scompare solo per k=7, ma lì peggiora ad es. la Svizzera, quindi non è chiaro se convenga separare in più blocchi.

In definitiva l’utilità della cluster analysis in questo esempio è stata di identificare tre blocchi socio-economici: l’Est europeo, il Sud e il resto. Ovviamente lo sapevamo già, ma si immagini di esaminare un fenomeno nuovo e di scoprire in modo così veritiero la sua scomposizione.

**ESERCIZIO**: effettuare una cluster analysis relativamente alla tabella PLIC-TD\_ecc.

1. **CLASSIFICAZIONE TRAMITE REGRESSIONE**

Carichiamo la tabella con le classi. Esse sono state scelte piuttosto arbitrariamente, diciamo seguendo il senso comune, secondo il seguente criterio: quali sono le nazioni più ricche, dove si pensa che la gente sia ricca, ci sia lavoro ecc. (classe 1). La logica è: contiamo sul fatto che questi indicatori socio-economici spieghino la ricchezza.

E’ stato commesso volutamente un errore rispetto al senso comune, così vediamo se emerge.

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

Eseguiamo una regressione lineare multipla usando come output la classe:

fit = lm(classe ~ Education+Health+Accidents+Employment+Export+HealthyYears+GenderPaymentGap+ IndustrialProduction+ TaxesLabour+ TaxesTotal, data=Ec)

Notare che basta scrivere

fit = lm(classe ~ ., data=Ec)

> summary(fit)

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -4.687731 1.428508 -3.282 0.00393 \*\*

Education 0.013568 0.007776 1.745 0.09716 .

Health 0.004052 0.007225 0.561 0.58149

Accidents 0.205919 0.115303 1.786 0.09009 .

Employment 0.027607 0.010904 2.532 0.02033 \*

Export 0.046917 0.038747 1.211 0.24080

HealthyYears 0.019378 0.016938 1.144 0.26680

GenderPaymentGap 0.006630 0.010268 0.646 0.52619

IndustrialProduction 0.003202 0.008859 0.361 0.72175

TaxesLabour -0.002333 0.011677 -0.200 0.84377

TaxesTotal 0.014260 0.008119 1.756 0.09513 .

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2927 on 19 degrees of freedom

Multiple R-squared: 0.7559, Adjusted R-squared: 0.6274

F-statistic: 5.884 on 10 and 19 DF, p-value: 0.0004776

Forse il modello senza intercetta ha più senso, ma proseguiamo così a titolo di prova.

Il comando

predict(fit)

fornisce i valori della y dati dal modello. Essi possono contraddire in qualche misura la nostra scelta iniziale, indicando delle possibili correzioni. Il risultato, osservato attentamente, è molto interessante.

Si osservi anche

round(predict(fit))

fornisce la classificazione in forma perentoria. (Non è sempre vero: round approssima al’intero più vicino ed in questo esempio siamo fortunati che l’interro più vicino è 0 o 1.)

Ovviamente i metodi di classificazione hanno come scopo primario la classificazione di nuovi individui. Potremmo artificialmente eliminare delle nazioni e vedere come vengono classificate.

Più realisticamente, nel nostro caso, possiamo argomentare che l’utilizzo di ciascuna nazione ha influenzato il modello e quindi lo ha “spostato” nella direzione che favorisce la classificazione giusta. Pertanto, anche se magari una certa nazione AA sarebbe stata più naturalmente classificata come 0, avendola usata per costruire il modello stesso ne risulta che il modello aumenterà la sua vicinanza alla classe 1.

Pertanto, se dall’analisi precedente sono emerse delle situazioni impreviste o strane, si può consolidare la loro analisi eliminandole e poi applicando ad esse il modello sviluppato solo sulle altre. Vediamo un esempio: Denmark (riga 4) ha un punteggio un po’ diverso dalle aspettative; togliamola e vediamo che succede.

Ec2 = Ec[-4,]

fit2 = lm(classe ~ ., data=Ec2)

Ora, purtroppo, il comando predict(fit2) fornisce le predizioni di Ec2. Sono molto istruttive, perché diverse dalle precedenti. Ma a noi ora interessa la predizione di Denmark, secondo questo nuovo modello, per esse “neutro”.

predict(fit2,Ec[4,]) ;

fornisce il valore (controllare come conferma con predict(fit2,Ec[1,]) ).

Il risultato conferma che Denmark è di classe 1. La sua vicinanza a 1 è aumentata, come dev’essere: nel primo modello la sua posizione a 0 “trascinava” il modello in una direzione che rendeva Denmark più vicina a 0 (perché lo stavamo imponendo).

1. **CLASSIFICAZIONE NON PERENTORIA**

Iniziamo rendendo non perentoria la classificazione precedente. Usiamo la funzione logistica, che R calcola in automatico con la funzione plogis. Essa vuole in input il valore da trasformare, 0.5 perché esso è il valore della y a metà strada tra 0 ed 1, ed infine 1 (è una sorta di deviazione standard). Poi abbreviamo i numeri perché hanno troppe cifre:

round(plogis(predict(fit), location = 0.5, scale = 1),3)

Come si può vedere, i numeri risultano molto schiacciati vicino a 0.5: si guardi ad esempio Germany.

1. **REGRESSIONE LOGISTICA**

Ora svolgiamo una classificazione non perentoria tramite il metodo della regressione logistica.

Rlog =glm(classe ~ Education+Health+Accidents+Employment+Export+HealthyYears+GenderPaymentGap+ IndustrialProduction+ TaxesLabour+ TaxesTotal, family = binomial, data=Ec)

oppure

Rlog =glm(classe ~., family = binomial, data=Ec)

Purtroppo fornisce errore. La cosa non è infrequente quando ci sono troppe variabili: si pensi che si chiede di minimizzare una funzione molto non lineare di 10 variabili. Un trucco per saltarci fuori è di ridurre la variabili e per farlo in fretta osserviamo la matrice di correlazione cor(Ec). Vediamo che la classe è più correlata a Employment, Export, TaxesTotal (se ne potrebbero aggiungere altre ma rischiamo di ricreare il problema).

Vediamo ora:

Rlog2 =glm(classe ~ Employment+Export+ TaxesTotal, family = binomial, data=Ec)

Ora ha funzionato. Vediamo i punteggi e le probabilità:

round(predict(Rlog2) ,3)

fornisce i punteggi del modello di regressione lieare multipla sottostante. Vediamo che i valori dell’output sono più accentuati, rispetto al caso della regressione fatta sopra. Ecco le relative probabilità:

round(predict(Rlog2,type="response"),3)

Il metodo ora è severissimo! Non dà scampo.