

Pràctica 6: MDS-2na part: Anàlisi de correspondències

Tenim resultats de les eleccions municipals (2019) a Barcelona ciutat, els resultats s'han agrupat per districtes: Ciutat Vella(CV), Eixample (Eix), Sants-Montjuïc (SM), (LC) Les Corts, (SSG) Sarrià-Sant Gervasi, (Gr) Gràcia, (HG) Horta-Guinardó, (NB) Nou Barris, (StA) Sant Andreu, (StM) Sant Martí. Font: <https://www.bcn.cat/estadistica/catala/dades/inf/ele/ele46/t35.htm>

Notacions: ab: abstencions, b blancs, n nuls i els vots a candidatures:

erc, bec(BEC-ECG), psc, cs, jts, pp, cup, bcp(BCAP), vox, pcm(PACMA), alt(Altres: resta de candidatures).

Més info: <https://www.bcn.cat/estadistica/catala/dades/inf/ele/ele46/t24.htm>

```
require(analogue);require(readr); require(FactoMineR)
bcn2019 <- as.data.frame(read_table2("MunicBcn2019_districtes.txt",col_names = FALSE))
distr <- as.vector(bcn2019[,1])
bcn0 <- as.matrix(bcn2019[-1])
rownames(bcn0) <- distr
dim(bcn0)

# tallem i considerem "altres"= candidatures a partir col 17 amb pocs vots
altres <- apply(bcn0[,17:30],1,sum)
bcn0 <- data.frame(bcn0[,1:16],altres)
names(bcn0) <- c("Elecs","Votants", "ab","n","Val","b","erc","bec","psc","cs","jts","pp","cup","bcp","vox","pcm","alt")

# comprovem que suma els "Elecs" (electors):
all.equal(apply(bcn0[,c(3,4,6:17)],1,sum), as.numeric(bcn0[,1]), check.attributes=F)
# comprovem que suma els "Votants" (exclou abs):
all.equal(apply(bcn0[,c(4,6:17)],1,sum), as.numeric(bcn0[,2]), check.attributes=F)
# comprovem que suma els "Val" (vàlids -no inclouen abs ni nuls, però sí blancs i vots a candidatures -):
all.equal(apply(bcn0[,c(6:17)],1,sum), as.numeric(bcn0[,5]), check.attributes=F)

# eliminem les columnes "Elecs", "Votants" i "Val" i ens quedem amb la resta:
bcn <- bcn0[, -c(1,2,5)]
```

Diguem N la matriu de recomptes n_{ij} (sense totals, és a dir bcn). Denotem: n_{++} el nombre total d'observacions, n_{i+} el total de la fila i , n_{+j} el total d'una columna j . Guardem els objectes N (matriu de recomptes N), nip , el vector amb els totals de fila $\{n_{i+}\}$, npj , el vector amb els totals de columna $\{n_{+j}\}$, i npp el nombre total de casos n_{++} . També obtenim la taula afegint els totals.

```
N<-bcn
npp<-sum(N) # nombre de casos
nip<-apply(bcn, 1, sum) # totals files
npj<-apply(bcn, 2, sum) # totals columnes
tbcn<-cbind(bcn, nip) ## totals files afegits
tbcn<-rbind(tbcn, c(npj,npp)) ## totals columnes afegits
```

Atenció: Per operar no s'usa la matriu amb els totals afegits, sinó la de recomptes purs: N , que en aquest cas és bcn .

Exercicis: Recorda operar matricialment amb vectors i matrius (no cal usar for()) !

- Assegura't que N sigui de classe matriu, per operar. Calcula la matriu de proporcions (freq. relatives del total) i guarda-les en l'objecte P : $P = (p_{ij})_{ij}$, on $p_{ij} = \frac{n_{ij}}{n_{++}}$.
- Marginals: Calcula i guarda el vector c que conté les masses de les columnes (o perfil mitjà de fila): $c = (p_{+1}, \dots, p_{+p})$, on $p_{+j} = \frac{n_{+j}}{n} = \sum_{i=1}^n p_{ij}$. Fes el mateix amb el vector r que conté les masses de les files (o perfil columna mitjà): $r = (p_{1+}, \dots, p_{n+})$, on $p_{i+} = \frac{n_{i+}}{n} = \sum_{j=1}^p p_{ij}$. Mostra els vectors r i c .
- Calcula els perfils fila i guarda'ls en una matriu R : són les proporcions de vots als partits, condicionats a cada districte $R = (r_{ij})_{ij}$, on $r_{ij} = \frac{p_{ij}}{p_{i+}}$. **Indicació:** Pots usar que $R = D_r^{-1} P$, on D_r^{-1} és la matriu diagonal amb els recíprocs del vector r ((també es pot obtenir usant la funció `sweep()`). Comprova que totes les files de la matriu R sumen 1.
- Idem amb la matriu C de perfils columna.
- Fes una funció `dr(N,i,h)` amb R que, donada la matriu de recomptes, calculi la distància χ^2 entre dues files i , h . Recorda que aquesta distància al quadrat és:

$$d_{\chi^2}^2(i, h) = \sum_j \frac{(r_{ij} - r_{hj})^2}{p_{+j}}$$

Cal assegurar-se que la classe dels objectes que multipliquem matricialment sigui matriu (as.matrix) també dins de les funcions !

6. Aplica la funció `dr()` a calcular la distància entre CV i SSG, i entre CV i SM.
A quin dels dos altres districtes s'assembla més CV, en termes de la distància khi-quadrat ?
7. Fes una funció `dchi(N)` que calculi tota la matriu de distàncies khi-quadrat entre tots els perfils fila d'una matriu de recomptes $N = (n_{ij})$. La funció `dchi()` ha de donar una llista amb: la matriu de distàncies $D = (d_{\chi^2}(i, h))_{i, h}$, el valor de l'estadístic U i el seu p-valor, on:

$$U = n_{++} \sum_i \sum_j \frac{(p_{ij} - p_{i+}p_{+j})^2}{p_{i+}p_{+j}} = n_{++} \sum_i \sum_j z_{ij}^2; \quad \text{amb} \quad z_{ij} = \frac{p_{ij} - p_{i+}p_{+j}}{\sqrt{p_{i+}p_{+j}}}$$

Segueix les indicacions:

Abans de calcular Y , cal obtenir: npp , P i R , i assegurar-se que la classe dels objectes que multipliquem matricialment sigui matriu (`as.matrix`)

- Calcula primer $Y = R D_c^{-1/2}$, on D_c^{-1} és la matriu diagonal amb els recíprocs del vector c .
- Calcula $B = Y Y^t$.
- Calcula D^2 tenint en compte que: $d_{ij}^2 = -2b_{ij} + b_{ii} + b_{jj}$, i després: $D = \sqrt{D^2}$.
- Per obtenir el valor de l'estadístic, tingues en compte que $U = n_{++} \text{Tr}(Z Z^t)$, és a dir, on $Z = (z_{ij})_{ij}$ i `tr()` significa traça. Pots usar:

$$Z = D_r^{-1/2} (P - \text{rcol} \%*\% \text{crow}) D_c^{-1/2}, \quad \text{tots els productes són matricials, no només l'indicat amb } \%*\%$$

on `rcol` és el vector r en forma de matriu d'una columna i `crow` és el vector c en forma de matriu d'un fila.

- El p-valor és la cua dreta de la llei χ^2 amb graus de llibertat $d = (n - 1)(p - 1)$, n = nombre de files i p = nombre de columnes.

```
dchi<-function(N,dig=2){
  # N taula de recomptes, classe 'matrix'
  # dig: nom. decimals output, 2 per defecte
  ....
  ....
  list(val.U=U,gl=gl,dchi=round(D,dig),pval=pval)
}
```

8. Aplica la funció `dchi()`, amb 3 dígits, per calcular la matriu de distàncies khi-quadrat dels districtes de BCN segons les municipals de 2019. Quina és la parella de districtes amb perfils més diferents segons els vots i aquesta distància. El test: és significatiu? com s'interpreta?
9. Calcula la mateixa matriu de distàncies amb la funció `distance()` del paquet `analogue`, especificant el tipus de distància amb l'argument `method`.
10. Aplica ara la funció (CA) del paquet `FactoMineR`. *Remarca:* Aquesta funció fa l'anàlisi de correspondències en la qual s'analitzen perfils fila entre sí, perfils columna entre sí i la correspondència files-columnes. L'anàlisi de la correspondència es basa en la distància khi-quadrat i la inèrcia que resulta de multiplicar distància al quadrat per massa ($d_{\chi^2}^2((y_{i1}, \dots, y_{i1}), G) * w_i$) on w_i és la freq. relativa de la fila i (abans li hem dit p_{i+}).

```
require(FactoMineR)
###?CA
bcn.CA<-CA(bcn,ncp=2,graph=F)
par(mfrow=c(1,1))
plot(bcn.CA,cex=.6)
#
bcn.CA$row$coord
bcn.CA$row$cos2
apply(bcn.CA$row$cos2,1,sum) ## qualitat repre 2D
iner<-bcn.CA$row$inertia
iner_perc<-100*iner/sum(iner)
names(iner)<-rownames(bcn)
bcn.CA$row$contrib
## idem per a columnes ..... !!
apply(bcn.CA$col$cos2,1,sum) ## qualitat repre 2D
```

Interpreta la gràfica.

11. Repeteix el procediment amb la funció `CA` però fent que ara l'abstenció i els vots nuls i blancs siguin columnes suplementàries. De vegades, per tal de comparar-lo amb el resultat anterior, pot ser convenient canviar el signe de les coordenades((recorda que el signe és arbitrari)).