

Pràctica 4: Inferència en components principals.

En aquesta pràctica es demana crear el codi de tres funcions de R, per a tres apartats d'inferència en components principals (document: *Discussió sobre components principals*). Seguidament, s'aplicaran a les dades Decathlon i caldrà interpretar els resultats. **Atenció:** La mida de la mostra n i el nombre de variables p són les de les dades "bàsiques", les que determinen les components principals.

[Recordatori general: *Acceptar la hipòtesis nul·la d'un test només significa que no hi ha prou evidència per acceptar la hipòtesi alternativa.*]

```
require(FactoMineR)
data(decathlon, package="FactoMineR")
deca<-decathlon
# transformació variables de temps
temps<- deca[c(1,5,6,10)]
maxt<-apply(temps,2,max)
transf<- sweep(temps,2,maxt,FUN="-")
deca[c(1,5,6,10)]<-transf

basic<-deca[-(39:41),-(11:13)]
n<-nrow(basic); p<-ncol(basic)

res<-PCA(deca,quali.sup=13,quanti.sup=11:12,ind.sup=39:41,graph=F)
## s'ha aplicat PCA al fitxer bàsic per determinar les direccions de les components;
### els objectes suplementaris només es projecten en l'espai creat
vaps<-res$eig[,1]
```

1. Fes la funció que doni l'interval de confiança per a un valor propi λ_j , amb j fixat. Recordeu que, com passa amb tots els arguments de totes les funcions, el nivell de significació 0.05 que es posa com a valor per defecte es pot canviar al cridar la funció (no cal fer una funció nova per a cada nivell de significació!). Aplica la funció per determinar un interval de confiança per al valor propi més petit de les dades Decathlon, amb nivell de confiança **0.90**. Interpreta el resultat.

```
ci.lambda<-function(vaps,n,j,alpha=0.05)
{
  # vaps: vector numeric de valors propis
  # n: mida mostral del fitxer que genera els vaps i veps
  # j: posició del valor propi del qual es vol obtenir l'interval
  # alpha: 1-"nivell de confiança"
  ...

  list(lamb_j=,nc= ,liminf= ,limsup= ) # valor propi en qüestió # nivell conf. # límits de l'interval
}

ci.lambda(vaps=,n=,j=,alpha=)
```

```
## $lam_j
##   comp 10
## 0.2013808
##
## $nc
## [1] 0.9
##
## $liminf
##   comp 10
## 0.1373836
##
## $limsup
##   comp 10
## 0.2951898
```

2. Fes una funció per fer el test d'incorrelació de les variables inicials, que doni com a resultat una llista amb: el valor de l'estadístic de contrast, els graus de llibertat i el p -valor del test. Escriu la hipòtesi nul·la per a les dades Decahtlon. Aplica el test a les dades i extreu-ne la conclusió (per defecte, el nivell de significació és 0.05). Si no es pogués descartar la H_0 , quines implicacions pràctiques tindria?

```
incor<-function(dades)
{
  # dades: matriu o dataframe de dades numèriques bàsiques
  ...

  list(u= ,gl= ,pvalor= ) # valor estadístic # gl·lib # pvalor
}

incor( ..... )
```

```
## $u
## [1] 113.8331
##
## $gl
## [1] 45
##
## $pvalor
## [1] 6.937864e-08
```

3. Crea una funció pel test d'esfericitat parcial: Igualtat dels $k = p - r$ últims valors propis, considerats com a 'soroll' que no aporta cap direcció principal. El resultat de la funció serà una llista amb: els k últims valors propis, el valor de l'estadístic de contrast, els graus de llibertat i el p -valor. Aplica-ho a les dades Decahtlon, per tal de decidir si les últimes 6 components són soroll o no. Escriu la hipòtesi nul·la del test, fes-lo i digues la conclusió.

```
esf<-function(dades,k)
{
  # dades: bàsiques, sense elements suplementaris
  # k: nombre de vaps finals
  ...

  list(k.vaps.finals= ,u= ,gl= ,pvalor= ) # llistat dels valors propis en discussió
                                          # valor estadístic # gl·lib # pvalor
}

esf(dades=,k=)
```

```
## $k.vaps.finals
## [1] 0.6709757 0.6347902 0.4797817 0.3715403 0.2581859 0.2013808
##
## $u
## [1] 18.23764
##
## $gl
## [1] 20
##
## $pvalor
## [1] 0.5717572
```