

Test Permutacions 17-18. Randomisation version of ANOVA 1 factor.

Jordi Ocaña, Sergi Civit,

7 de març de 2018

1 Randomisation version of ANOVA 1 factor

1.1 Enunciat de la Situació Experimental

Dades d'un assaig clínic AMB 3 TRACTAMENTS per a la hipertensió: un placebo, un diurètic i un beta-bloquejant.

Disseny balancejat, d'un total de 45 individus, 15 assignats **a l'atzar a cada tractament**.

Es tracta doncs de comparar un nou tractament contra la hipertensió arterial(basat en una substància beta-bloquejant) amb el tractament habitual (un diurètic) i una substància inocua o placebo

La hipòtesi a demostrar (alternativa) més lògica és que hi ha almenys alguna diferencia entre els diferents tractaments.

```
> # Lectura de les dades d'un assaig clínic on es volien comparar
> # a = 3 tractaments per a la hipertensió:
> # un placebo, un diurètic i un beta-bloquejant.
> # Disseny balancejat, d'un total de 45 individus, ASSIGNATS ALEATORIAMENT
> assaig <- read.table("BetaBloc.txt", header = TRUE)
> n <- length(assaig[, "Tratamiento"])
> n
```

```
[1] 45
```

```
> ntreat<- tapply(assaig[, "ReducTension"], assaig[, "Tratamiento"], length)
> n1 <- ntreat[1]
> n2 <- ntreat[2]
> n3<- ntreat[3]
```

1.2 Estadístic F i MODIFICACIONS

Elecció de l'ESTADÍSTIC. Aquí COMENÇAREM PER UN POSSIBLE COM
ÉS EL CLÀSSIC QUE ENS PROPORCIONA LA METODOLOGIA PARA-
MÈTRICA(Estadístic F)

```

> # Estadístic de test:
>
> # Una tria inicial molt lògica seria utilitzar l'estadístic
> # F de l'ANOVA paramètrica normal:
> estad.F = function(dat, testFactor = 1, response = 2) {
+   return(oneway.test(dat[,response] ~ dat[,testFactor], var.equal = TRUE)$statistic)
+ }

```

MODIFICACIÓ DE L'ESTADÍSTIC F. Cas balancejat: el RESULTATS SERIA EXACTAMENT el mateix FENT SERVIR l'estadístic, UNICAMENT EN EL CAS BALANCEJAT LA SUMA DE QUADRATS DEL FACTOR:

```

> # Estadístic de test: Modificació equivalent a l'estadístic F de l'ANOVA paramètrica
>
> estad <- function(dat, testFactor = 1, response = 2) {
+   y <- dat[,response]
+   fact <- dat[,testFactor]
+   yi. <- tapply(y, fact, sum)
+   return(sum(yi.*yi. / tapply(y, fact, length)))
+ }
> estad.balanc <- function(dat, testFactor = 1, response = 2) {
+   sums <- tapply(dat[,response], dat[,testFactor], sum)
+   return(sum(sums*sums))
+ }
> estad.tilde1<-estad.F(assaig)
> estad.tilde1

      F
14.19115

> estad.tilde2<-estad(assaig)
> estad.tilde2

[1] 14853.73

> estad.tilde <- estad.balanc(assaig)
> estad.tilde

[1] 222806

```

1.3 Permutacions de Monte Carlo VERSUS Test de Permutacions exacte

ENUMERAR TOTES LES POSSIBLES PERMUTACIONS de les dades ÉS computacionalment complex:

```

> factorial(n)/(factorial(n1)*factorial(n2)*factorial(n3))

      betablo
5.349498e+19

```

1.4 Test de PERMUTACIONS DE MONTE CARLO i Estadístic F

És més viable un ENFOC de TEST de PERMUTACIONS de Monte Carlo.

```
> #Nombre de permutacions
> nperm <- 19999
> assaig.perm <- assaig
> # Generació de 'nperm' permutacions aleatòries, i càlcul d'estad.F
> # sobre cada permutació:
> set.seed(23771)
> estad.perms <- replicate(nperm,
+ {
+   assaig.perm[,2] <- sample(assaig[,2], replace = FALSE)
+   estad.F(assaig.perm)
+ })
> # Triga bastant...
>
> # Visualització dels 20 primers valors F calculats sobre les permutacions:
> estad.perms[1:20]
```

	F		F		F		F		F		F		F
0.86756050	5.34791174	0.15575787	0.81840922	1.06801848	0.42690829	1.89144574							
	F		F		F		F		F		F		F
0.08017513	0.36984696	1.36072822	0.46741573	1.03327842	0.80418086	1.71433424							
	F		F		F		F		F		F		F
1.41820010	1.11667647	0.36453674	2.30360877	1.10205656	0.16468561								

1.5 P-VALOR

Calculem EL P-VALOR

```
> # p-valor:
> (sum(estad.perms >= estad.tilde1) + 1) / (nperm + 1)

[1] 5e-05
```

1.6 MODIFICACIÓ ESTADÍSTIC F amb Permutacions de Monte Carlo

Ara do duem a TERME A PARTIR DE LA MODIFICACIÓ de l'estadístic F

```
> # El mateix procés, però ara calculant l'estadístic més senzill i equivalent a F:
> set.seed(23771)
> estad.perms <- replicate(nperm,
+ {
+   assaig.perm[,2] <- sample(assaig[,2], replace = FALSE)
```

```

+      estad.balanc(assaig.perm)
+    }
+  )
> estad.perms[1:20]

[1] 201966 211326 200114 201842 202466 200834 204428 199910 200684 203180
[11] 200940 202380 201806 204018 203318 202586 200670 205358 202550 200138

```

1.7 P-VALOR

Calculem EL P-VALOR

```

> (sum(estad.perms >= estad.tilde) + 1) / (nperm + 1)

[1] 5e-05

```