Test Permutacions: Exacte i Montecarlo

By: S. Civit, MNPR, 17-18. 1,2 sessiÃ³Â³

Feb, 2018

Contents

Resampling

- **Resampling/remostreig** terme que empreem i que porta implícit una varietat de mètodes estadístics basats en les dades disponibles (mostres) en lloc d'un conjunt d'assumpcions estàndard sobre població/ns subjacent/s.

- Aquest mètodes inclouen **permutation tests, bootstrap and jackknife**

Permutation tests

Indicis

Situació Experimental

En un context biomèdic i a partir de N = 22 dones, aleatòriament assignades n = 11 a rebre la droga D i n = 11 un placebus P. Totes 22 prenien un anticonceptiu amb dos components: etinil estradiol (EE) i noretindrona (NET). Es volia estudiar si la presència de la droga D influia en en els nivells de EE i NET (i, per tant, en l'efectivitat i/o seguretat de l'anticonceptiu). El nivell dels components de l'anticonceptiu es mesurava mitjançant la variable "àrea sota la corba" (AUC) que designarem EEAUC i NETAUC respectivament per EE i NET (i també amb la concentració màxima Cmax, que no considerem aquí)

Aproximación descriptiva

A tot el que segueix ens centrarem en la variable EEAUC.

```
dat<-read.table("E:/MNPR 17-18/anticonceptive.txt", header=TRUE)
dat
```

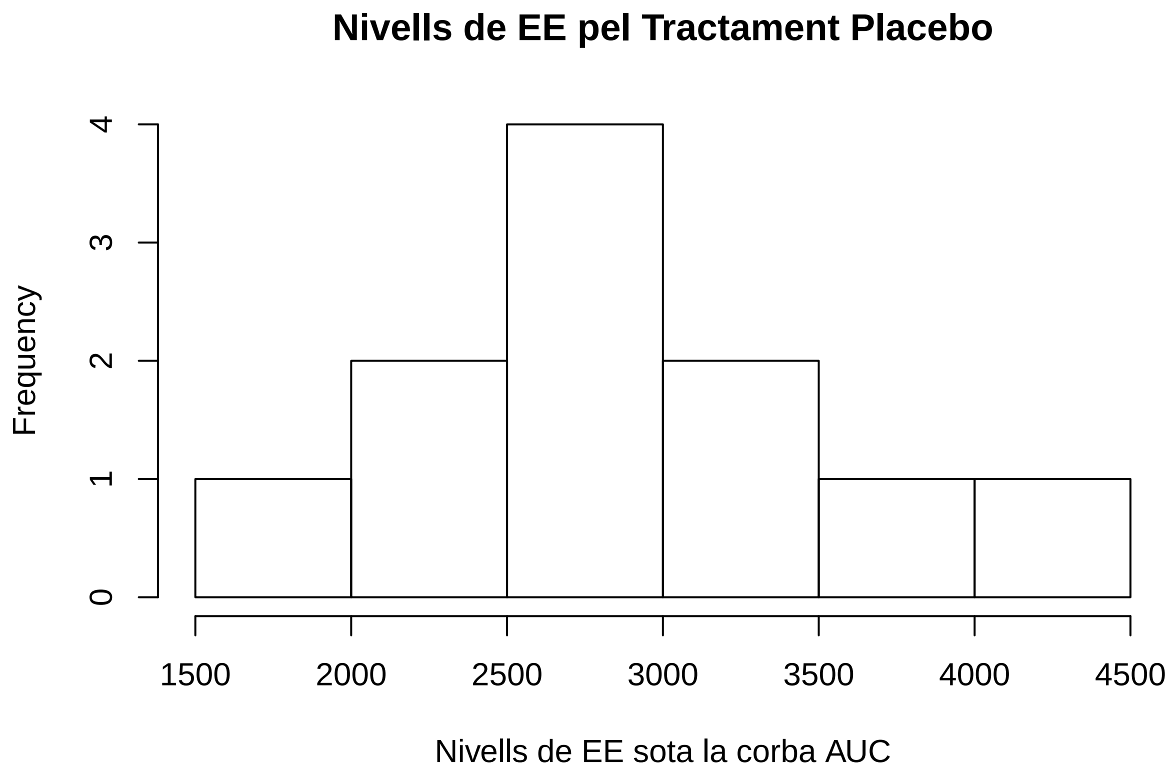
```
##    Suj tratam  EEAUC   NETAUC
## 1    1      P 2623.0 197525.0
## 3    2      D 3756.2 176487.5
## 5    3      P 2227.7 151957.5
## 7    4      D 2986.9 116305.0
## 9    5      D 3858.2 125869.0
## 11   6      P 4493.7 156367.5
## 13   7      P 2985.3 184942.5
## 15   8      D 3328.1 118340.0
## 17   9      D 3005.8 176590.0
## 19  10      P 1919.7 123012.5
## 21  11      P 2496.1 122987.5
## 23  12      D 5624.5 207327.5
## 25  13      P 3016.0 173565.0
## 27  14      D 3354.9 197912.5
## 29  15      P 2985.2 156367.5
## 31  16      D 3282.8 182250.0
## 33  17      P 3385.5 201557.5
## 35  18      D 5018.0 278532.5
## 37  19      P 2622.0 155790.0
## 39  20      D 2472.0  86467.5
## 41  21      D 3819.6 142359.8
## 43  22      P 3863.5 232197.5
```

```
class(dat$EEAUC)
```

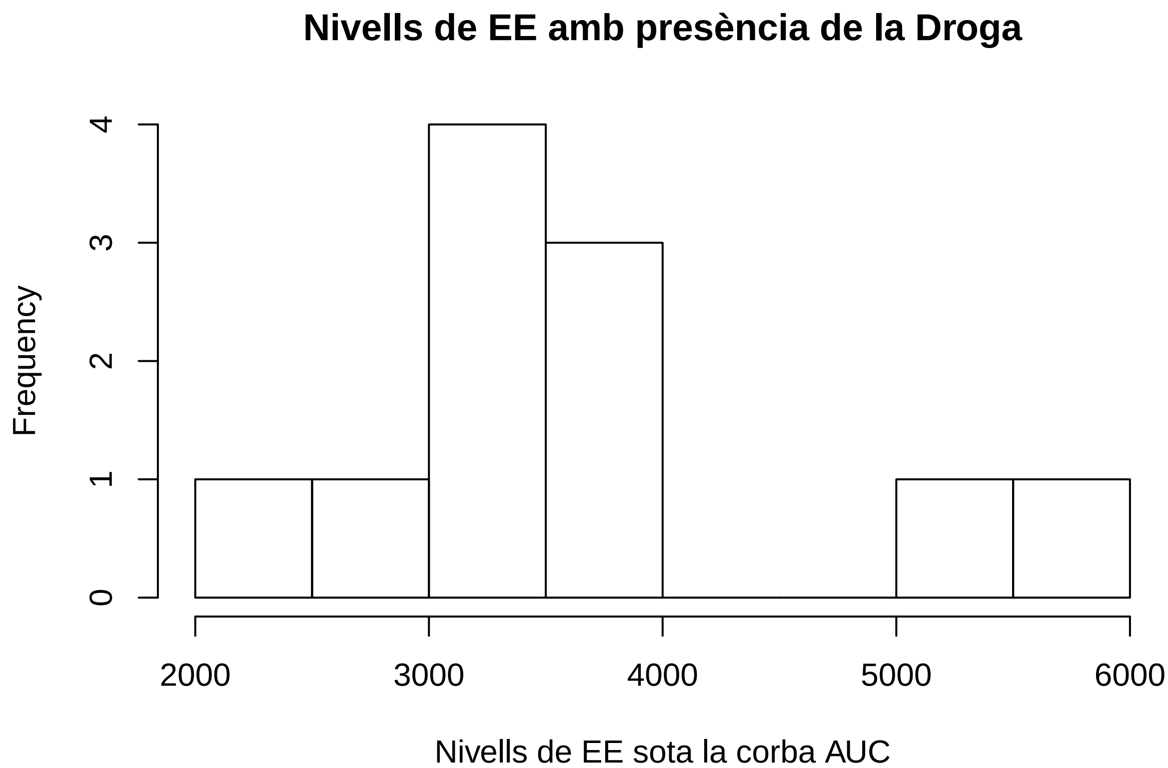
```
## [1] "numeric"
```

Indicis de normalitat (o no) de les dades:

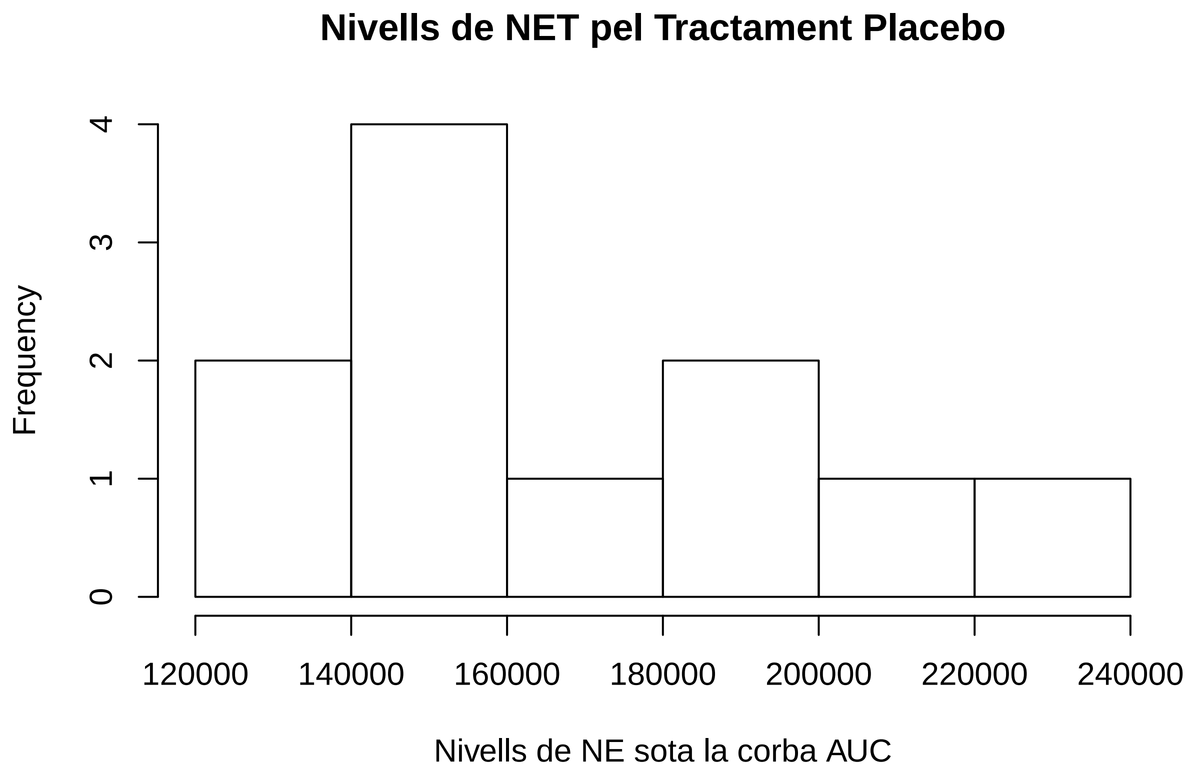
```
hist(dat[dat[,"tratam"]=="P","EEAUC"], main="Nivells de EE pel Tractament Placebo",  xlab= "Nivells de I
```



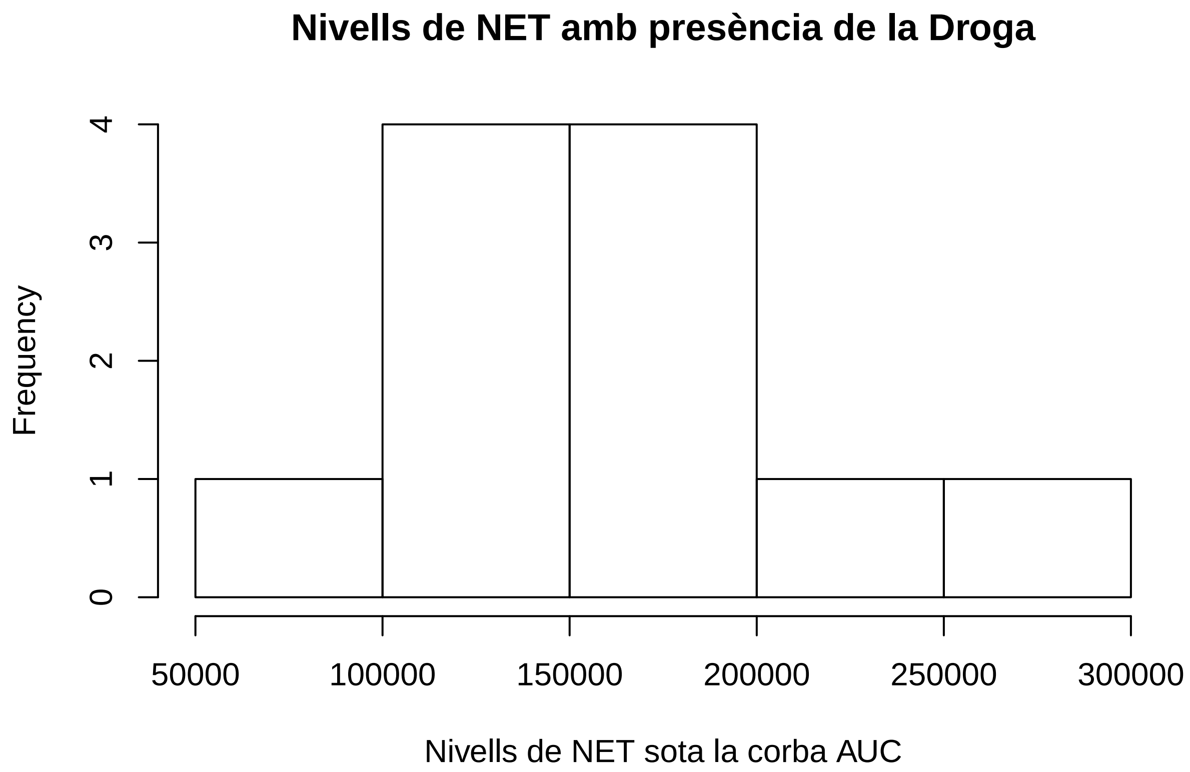

```
hist(dat[dat[,"tratam"]=="D","EEAUC"],main="Nivells de EE amb presència de la Droga",  xlab= "Nivells de
```




```
hist(dat[dat[,"tratam"]=="P","NETAUC"],main="Nivells de NET pel Tractament Placebo",  xlab= "Nivells de
```

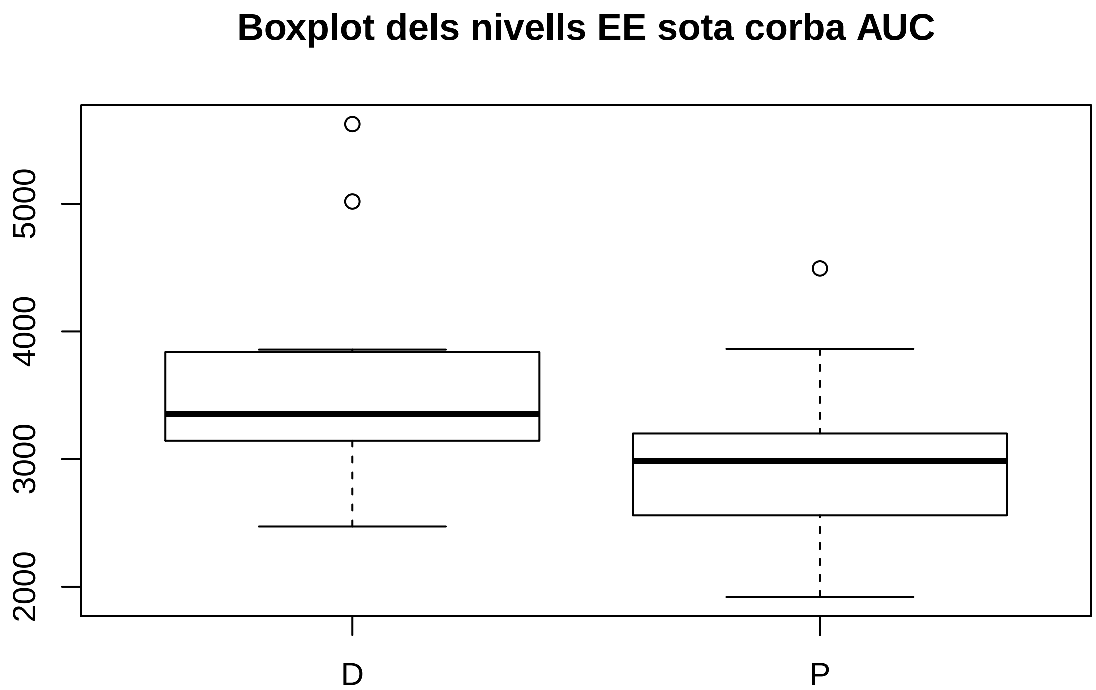



```
hist(dat[dat[,"tratam"]=="D","NETAUC"],main="Nivells de NET amb presència de la Droga",  xlab= "Nivells
```

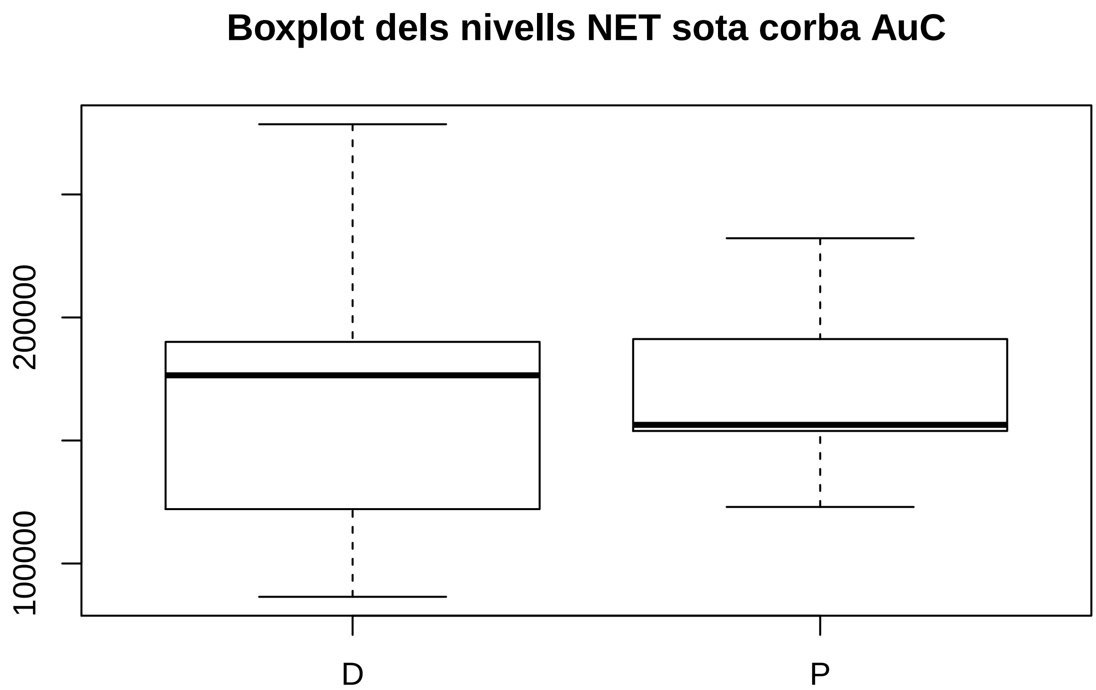


Indicis d'igualtat de la dispersió (o no) de les dades P vs D

```
boxplot(EEAUC ~ tratam, main="Boxplot dels nivells EE sota corba AUC",data = dat)
```




```r
boxplot(NETAUC ~ tratam, main="Boxplot dels nivells NET sota corba AuC",data = dat)
```



Una mica de R (R packages and dataset)

Per no escriure tant, guardarem en variables separades algunes informacions útils:

Recuperant les grandàries mostrals

```
n <- tapply( dat[,"EEAUC"], dat[,"tratam"], length) # Mides mostrals
n                  # Ara 'n' conté les mides mostrals de cada grup, n[1] == n[2] == 11
```

```
##  D  P
## 11 11
```

```
N <- sum(n)        # 'N' serà el total d'observacions, 22
n1 <- n[1]
n2 <- n[2]
```

Reordenant els valors d'EEAUC (els 11 primers corresponen a "D")

```
auc <- dat[order(dat$tratam), "EEAUC"]
auc
```

```
##  [1] 3756.2 2986.9 3858.2 3328.1 3005.8 5624.5 3354.9 3282.8 5018.0 2472.0
```

```
## [11] 3819.6 2623.0 2227.7 4493.7 2985.3 1919.7 2496.1 3016.0 2985.2 3385.5
## [21] 2622.0 3863.5
```

```r
auc[1:n1]          # Valors EEAUC per tots els casos "D"
```

```
##  [1] 3756.2 2986.9 3858.2 3328.1 3005.8 5624.5 3354.9 3282.8 5018.0 2472.0
## [11] 3819.6
```

```r
auc[(n1+1):N]      # i per tots els casos "P"
```

```
##  [1] 2623.0 2227.7 4493.7 2985.3 1919.7 2496.1 3016.0 2985.2 3385.5 2622.0
## [11] 3863.5
```

Comparació de mitjanes de 2 grups: t-Student

La prova t de Student anterior també es podria demanar així, indicant els dos grups de valors per separat:

```r
t.test(auc[1:n1], auc[(n1+1):N], var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  auc[1:n1] and auc[(n1 + 1):N]
## t = 2.0231, df = 20, p-value = 0.05664
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   -22.29917 1456.71735
## sample estimates:
## mean of x mean of y
##  3682.455  2965.245
```

TEST DE PERMUTACIONS: Idea conceptual

Una idea intuïtiva similar a l'empleada en l'enfocament basat en rangs seria la següent:

Si les dues mostres que comparem (D i P) són iguals pel que fa a la variable que estem estudiant (AUC), **PERTANYER** a D o a P és purament una etiqueta, no té cap relació amb que AUC sigui sistemàticament gran o petit. És a dir estem establint la **Ho**

Com a conseqüència, **és igualment probable** el que hem observat (certs valors de AUC associats a D i certs valors associats a P) que **qualsevol permutació de les dades**, en la qual els valors de AUC corresponguessin d'una altra manera a D i a P.

Per establir **la significació de la diferència que hem observat** en els nostres dades reals, COMPAREM aquest valor amb el **valor de la diferència** per a totes les POSSIBLES PERMUTACIONS de les dades.

Un mètode de "pura força bruta" seria

1. enumerar TOTES les **possibles permutacions de les dades**
2. sobre cadascuna d'elles calcular un ESTADÍSTIC ADEQUAT (per exemple, l'estadístic t del test paramètric) i
3. COMPTAR quantes vegades **aquest valor és més extrem** que el valor d'aquest estadístic calculat sobre les dades reals.

Test de Permutacions: Pas a pas

En el nostre cas la LLISTA de TOTES LES PERMUTACIONS (P~21= 1.124E+21) és, pràcticament impossible de realitzar, per temps i espai de memòria.

Però si **l'estadístic** que anem a calcular NOMÉS DEPÈN de quines observacions queden en cada grup, però no de **l'ordre** de les observacions dins de cada grup, PODEM EMPRAR com a possibilitat més raonable la funció **'combn'** de la llibreria estàndard 'utils' (es carrega per defecte) que enumera totes les combinacions possibles (sense importar l'ordre dins del grup). Per exemple

```
combn(5,3)  # equivalente a
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    1    1    2    2    2     3
## [2,]    2    2    2    3    3    4    3    3    4     4
## [3,]    3    4    5    4    5    5    4    5    5     5
```

```
combn(1:5,3)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    1    1    2    2    2     3
## [2,]    2    2    2    3    3    4    3    3    4     4
## [3,]    3    4    5    4    5    5    4    5    5     5
```

Per exemple en aquest segon cas enumera tots els índex 1:5 en grups de 3 i a cada una d'aquestes combinacions s'aplica la funció 'sum'

```
combn(5,3, sum)
```

```
##  [1]  6  7  8  8  9 10  9 10 11 12
```

Si només volem comptar el nombre de combinacions,:

```
choose(5,3)
```

```
## [1] 10
```

En el nostre cas, el nombre de combinacions de N (= 22) escollits en grups de 11 és molt, pero "es pot treballar":

```
choose(N,n1)
```

```
## [1] 705432
```

Del vector de 22 valors de EEAUC, enumerarem TOTS els possibles grups de 11 (TOTES les possibles formes en que a 11 d'ells els podem asignar a la categoría "D" -es pot suposar que per defecte els restants "reben" l'etiqueta "P") (Fixeu-vos que **cada una d'aquestes combinacions té igual probabilitat**, i per tant tenim 11! ordenacions possibles d'aquests valors)

Suposem que **hem escollit com estadístic** el de la t de student (VEUREM, QUE HI HAN ALTRES OPCIONS EQUIVALENTS I MES EFICIENTS COMPUTACIONALMENT, I ALTRE OPCIONS DIFERENTS).

TEST DE PERMUTACIONS: Pas 0

Creem la funció que esn RETORNA L'ESTADÍSTIC (t de Student). A partir de la llei del "mínim esforç", aprofitarem la funció 't.test'

```
tStat <- function(indexs, vector.dades) {
  t.test(vector.dades[indexs], vector.dades[-indexs], var.equal = TRUE)$statistic
}
```

Fixem-nos que aquesta funció, - Com a primer argument, rep els índexos dels valors que constitueixen el PRIMER GRUP i como segon argument el vector AMB tots els valors.

TEST DE PERMUTACIONS: Pas 1

1. CÀLCUL de l'estadístic t per LES DADES ORIGINALS:

```
auc
```

```
##  [1] 3756.2 2986.9 3858.2 3328.1 3005.8 5624.5 3354.9 3282.8 5018.0 2472.0
## [11] 3819.6 2623.0 2227.7 4493.7 2985.3 1919.7 2496.1 3016.0 2985.2 3385.5
## [21] 2622.0 3863.5
```

```
tauc <- tStat(1:n1, auc)
tauc
```

```
##        t
## 2.023063
```

TEST DE PERMUTACIONS: Pas 2

2. ENUMEREM TOTES les possibles combinacions dels índex 1:22 escollits en grups de 11:

```
cindexs <- as.data.frame(combn(1:N, n1))
```

combn(N, n1)

Tarda un rato...

Visualitzem una qualsevol d'aquestes combinacions, per exemple la número 12235:

```
cindexs[,12235]
```

```
##  [1]  1  2  3  4  5 13 14 18 20 21 22
```

Fixem-nos que

```
tStat(cindexs[,12235], auc)
```

```
##            t
## -0.008761422
```

correspon al càlculo de l'estadístic t de Student on el grupo "D" el formen els INDIVIDUS 1 2 3 4 5 13 14 18 20 21 22 (I no 1:11 com SUCCEIX en les DADES REALS)

TEST DE PERMUTACIONS: Pas 3

3. CÀLCUL de l'estadístic t de Student PER CADASCUNA DE LES COMBINACIONS (reordenacions dels indexs) (**¡AIXÒ SI QUE TARDA ESTONA!**):

```
tPerms <- vapply(cindexs, tStat, auc, FUN.VALUE = 0.0)
```

(en tota la "segona dimensió" -les columnes- de 'cindexs' aplico 'tStat', que com a SEGON ARGUMENT espera REBRE el vector complet dels valors, 'auc')

TEST DE PERMUTACIONS: Pas 4

4. p-valor: **proporció de vegades** en que l'estadístic calculat sobre les en el pas anterior ES TANT O MÉS EXTREM que el calculat sobre les DADES ORIGINALS:

```
sum(tPerms >= tauc) / length(tPerms)
```

```
## [1] 0.0282125
```

TEST DE PERMUTACIONS: Pas 5

5. El p-valor ES REFEREIX al test UNILATERAL en el que consideren la hipòtesis alternativa

H1 : mitja de D > mitja de P. Podem CONCLOURE que el resultat és significatiu amb un nivell de significació del 0.05.

Exercici 1: Plantejar el test bilateral en el pas 4.

TEST DE PERMUTACIONS: Consideracions (1/3)

- En un test de permutacions és MOLT MÉS CÓMODE calcular el p-valor, i decidir a partir d'aquest. Equivalentement podriem ESTABLIR una taula de VALORS CRÍTICS.
- Per exemple en el test unilateral, ELS VALOR CRÍTICS associats a diferents nivells de significació, **PER AQUESTES DADES**,SERIAN:

```
alphas <- c(0.10, 0.05, 0.025, 0.01)
tabla <- quantile( tPerms, probs = 1 - alphas)
names(tabla) <- alphas
tabla
```

```
##      0.1     0.05    0.025     0.01
## 1.334805 1.725701 2.079062 2.477088
```

(se podrían comparar con los valores críticos del test t de Student):

```
qt(1 - alphas, df = N - 2)
```

```
## [1] 1.325341 1.724718 2.085963 2.527977
```

- Fixeu-vos que hem destacat la frase "PER AQUESTES DADES". En un test de permutacions, la taula dels valors crítics s'OBTÉ UN COP TENIM/CONEIXEM LES DADES, per tant es **CONDICIONAL A LES DADES**. Per alguns autors això consitueix **un problema filosofic important**.

TEST DE PERMUTACIONS: Consideracions (2/3). Cas balancejat

- En un TEST DE PERMUTACIONS el que realment IMPORTA ÉS L'ORDRE en el que quedan ELS VALORS DE L'ESTADÍSTIC.
- En aquest sentit el RESULTATS SERIA EXACTAMENT el mateix FENT SERVIR **l'estadístic t-student, o bé la diferencia de mitjes**, o bé, UNICAMENT EN EL CAS BALANCEJAT (como el del exemple n1 == n2 == 11), SENZILLAMENT la **SUMA DELS VALORS D'UN DELS GRUPS**, per exemple "D":

```
combs <- combn(auc, n1)
combs[,1:20]
```

```
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##  [1,] 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2
##  [2,] 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9
##  [3,] 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2
##  [4,] 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1
##  [5,] 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8
##  [6,] 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5
##  [7,] 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9
##  [8,] 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8
##  [9,] 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0
## [10,] 2472.0 2472.0 2472.0 2472.0 2472.0 2472.0 2472.0 2472.0 2472.0
## [11,] 3819.6 2623.0 2227.7 4493.7 2985.3 1919.7 2496.1 3016.0 2985.2
##        [,10]  [,11]  [,12]  [,13]  [,14]  [,15]  [,16]  [,17]  [,18]
##  [1,] 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2 3756.2
##  [2,] 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9 2986.9
##  [3,] 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2 3858.2
##  [4,] 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1 3328.1
##  [5,] 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8 3005.8
##  [6,] 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5 5624.5
##  [7,] 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9 3354.9
##  [8,] 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8 3282.8
##  [9,] 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0 5018.0
## [10,] 2472.0 2472.0 2472.0 3819.6 3819.6 3819.6 3819.6 3819.6 3819.6
## [11,] 3385.5 2622.0 3863.5 2623.0 2227.7 4493.7 2985.3 1919.7 2496.1
##        [,19]  [,20]
##  [1,] 3756.2 3756.2
##  [2,] 2986.9 2986.9
##  [3,] 3858.2 3858.2
##  [4,] 3328.1 3328.1
##  [5,] 3005.8 3005.8
##  [6,] 5624.5 5624.5
##  [7,] 3354.9 3354.9
##  [8,] 3282.8 3282.8
##  [9,] 5018.0 5018.0
## [10,] 3819.6 3819.6
## [11,] 3016.0 2985.2
```

visualitzo les 20 primeres (de 705432) combinacions dels 22 valors de EEAUC escollits en grups de 11

```r
sums <- combn(auc, n1, sum)
```

A cadascuna de les 705432 combinacions CALCULO l'ESTADÍSTIC DE TEST (FAIG SERVIR LA SUMA, donat que es EQUIVALENT a altres MÉS COMPLICATS). En una única operació puc GENERAR LES COMBINACIONS I CALCULAR L'ESTADÍSTIC.

```r
suma.D <- sum(auc[1:n1])
sum(sums >= suma.D) / length(sums)
```

```
## [1] 0.02821392
```

Valor de la SUMA de les 11 dades que son "D" (test unilateral): **PER OBTENIR EL P.VALOR EXACTE**

TEST DE PERMUTACIONS: Consideracions (3/3). Cas NO BALANCEJAT

- En un cas NO BALANCEJAT tindriem que emprar un **ESTADÍSTIC com la DIFERENCIA DE MITJES**. Seria MES RÀPID que calcular CADA COP l'estadistic t-Student, pero no tant com l'anterior (la suma del valors de "D"):

```
diff.means <- function(indexs, vector.dades) {
  mean(vector.dades[indexs]) - mean(vector.dades[-indexs])
}
```

- Calculem PAS1 (CÀLCUL DE L'ESTADÍSTIC DIFERENCIA DE MITJES entre D y P per les DADES ORIFINALS)

```
dmeansReal <- diff.means(1:n1, auc)
```

- Calculem PAS 3 (CÀLCUL DE L'ESTADÍSTIC DIFERENCIA DE MITJES PER CADASCUNA DE LES COMBINACIONS)

```
dmeansPerm <- vapply(cindexs, diff.means, vector.dades = auc, FUN.VALUE = 0.0)
```

- Calculem PAS 4 (P-VALOR)

Test unilateral: D > P ?

```
sum(dmeansPerm >= dmeansReal) / length(dmeansPerm)
```

```
## [1] 0.0282125
```

Test bilateral: D != P ?

```
sum(abs(dmeansPerm) >= abs(dmeansReal)) / length(dmeansPerm)
```

```
## [1] 0.056425
```

- Calculem PAS 5 (CONCLUSIONS)
- Fixeu-vos que els **P-VALORS son exactament ELS MATEIXOS QUE FENT SERVIR l'estadístic t-Student**

TEST DE PERMUTACIONS "de Montecarlo" o aproximado

Idea conceptual

- Per MIDES MOSTRAL mitjanes o GRANS els tests de PERMUTACIONS EXACTES, com els anteriors, són impossibles de realitzar. La solució més habitual és fer un **test de PERMUTACIONS APROXIMAT o de MONTECARLO**.
- En aquest enfoc, es **SIMULA UNA MOSTRA ALEATÒRIA DE PERMUTACIONS**, normalment molt gran però de mida inferior a tota la "població" de possibles permutacions, que pot ser enorme (per exemple, si tinguèssim unes mostres de mida doble a les de l'exemple, el nombre de combinacions possibles, el de permutacions encara seria molt més gran, seria de més de 2.1 x 10^12).
- Procedint d'aquesta manera s'obté una **ESTIMACIÓ del p-valor**, que es pot suposar molt precisa.

Versió de Montecarlo del test anterior (farem servir l'estadístic "diferència de mitjanes"):

PAS 1

Fixem el nombre de permutacions aleatòries a generar, per exemple 99999 (+ 1 = 100000)

```
nperm <- 99999
```

PAS 2

Una permutació aleatòria dels índexs 1:N

```
sample(1:N, replace = FALSE)  # replace = FALSE és el valor per defecte, no cal indicar-ho
```

```
##  [1] 16  2 13  1 21  4 15 11  3 12 20 14 17 22  6 10  8  9 19  5  7 18
```

PAS 3

Tal com hem definit la funció 'diff.means', només espera rebre els índexos, permutats, corresponents al grup "D", i el vector amb les dades originals.

```
sample(1:N, size = n1)
```

```
##  [1] 16 21 11 14  1  7  4 19  3  5 10
```

PAS 4

Generem 'nperm' permutacions aletòries i sobre cada una d'elles calculem 'diff.means'

```
set.seed(1324)
dmeansPerm <- replicate(nperm, diff.means(sample(1:N, size = n1), auc))
dmeansPerm[1:10]
```

```
##  [1]  437.20909  296.88182  190.39091  703.59091  650.59091  437.44545
##  [7]  -18.42727 -482.84545  407.48182 -351.97273
```

PAS 5

Valor de diferència de mitjanes sobre la mostra real

```
dmeansReal <- diff.means( 1:n1, auc)
```

PAS 6

Un estimador no esbiaixat del p-valor seria la freqüència relativa: Test unilateral: D > P ?

```
sum(dmeansPerm >= dmeansReal) / nperm
```

```
## [1] 0.02749027
```

Test bilateral: D != P ?

```r
sum(abs(dmeansPerm) >= abs(dmeansReal)) / nperm
```

```
## [1] 0.05578056
```

PAS 6 (CONSIDERACIÓ IMPORTANT)

De tota manera és preferible l'estimador proposat per Dwass(1957), que és esbiaixat negativament però assegura que es respectarà el nivell de significació nominal (de tota forma, si el nombre de permutacions aleatòries és gran, la diferència és imperceptible)

Test unilateral: D > P ?

```r
(sum(dmeansPerm >= dmeansReal) + 1) / (nperm + 1)
```

```
## [1] 0.0275
```

Test bilateral: D != P ?

```r
(sum(abs(dmeansPerm) >= abs(dmeansReal)) + 1) / (nperm + 1)
```

```
## [1] 0.05579
```

Exercici 2: Repetir-ho per NETAUC

SITUACIÓ A RESOLDRE

Per exemple, podria passar perfectament que sospitèssim de la presència de dades extremes, **outliers**. Potser, si en lloc de calcular les mitjanes calculèssim les medianes seria més robust.

Com funcionaria un test basat en la diferència de medianes AMB LES DADES DE NETAUC?