

## El problema del cavaller De Méré

Considerem un joc de daus que va jugar un important paper en el desenvolupament històric de la probabilitat. Les famoses cartes entre Pascal i Fermat, que molts pensen que comencen un estudi seriós de la probabilitat, van ser instigades per una demanda d'ajuda d'un jugador i noble francès, el cavaller de Méré. Es deia que de Méré havia estat apostant que, en quatre tirades d'un dau, al menys sortia un sis. Ell guanyava sistemàticament i, per a que hi jugués més gent, canvià el joc per apostar que, en 24 tirades de dos daus, sortirien un parell de sisos. Es reclamà que de Méré perdia amb 24 i va trobar que eren necessàries 25 tirades per fer el joc favorable. Va ser *un grand scandale* que les matemàtiques s'equivoquessin.

Què és més probable, que surti al menys un doble sis en 24 tirades de dos daus o obtenir com a mínim un sis en 4 tirades d'un únic dau?

La comparació es pot fer matemàticament, però ara utilitzarem R per a contestar aquesta qüestió. En comptes de fer servir les probabilitats exactes, que en aquest cas es poden calcular, farem servir les freqüències relatives per a un número  $n$  de tirades prou gran.

Llencem 4 vegades un únic dau i volem obtenir al menys un sis.

La següent instrucció genera un vector aleatori de quatre valors entre 1 i 6

```
> exp1 <- sample(1:6,4,replace=T)
```

Si volem saber si tenim cap sis, podem fer

```
> max(exp1)==6
```

La resposta és TRUE o FALSE.

Ara anem a repetir això per a un número  $n$  elevat de tirades.

```
> n<-10000  
> valors<-rep(0,n) # vector de zeros  
> for (i in 1:n) {valors[i]<-max(sample(1:6,4,replace=T))==6}  
> sum(valors)/n
```

Això ens diu la freqüència relativa de la simulació.

Ara podem repetir aquestes instruccions per un  $n$  més gran i tindrem una millor aproximació de la probabilitat de l'esdeveniment considerat.

D'altra banda, podem simular 24 tirades de dos daus i comprovar si la suma és 12.

```
> exp2 <- matrix(sample(1:6,48,replace=T),nrow=24,ncol=2)  
> aux <- apply(exp2,1,sum)  
> max(aux)==12
```

També podem repetir les instruccions anteriors:

```
> n<-10000
> valors<-rep(0,n)
> for (i in 1:n) {
+   exp2 <- matrix(sample(1:6,48,replace=T),nrow=24,ncol=2)
+   aux <- apply(exp2,1,sum)
+   valors[i]<- max(aux)==12}
> sum(valors)/n
```

La comparació de freqüències ens donarà una pista sobre la resposta a la pregunta inicial.

Podem escriure una funció:

```
> demere1 <- function(nsimul){
+   valors<-rep(0,nsimul)
+   for (i in 1:nsimul) {valors[i]<-max(sample(1:6,4,replace=T))==6}
+   sum(valors)/nsimul}
```

I així podem fer

```
> demere1(100000)
```

De forma similar podem definir la funció demere2.