

Ejemplos en R - práctica 1

Nicolás Villagrán dos Santos - 2do cuat. 2020

Ej 1.1

El comando sample permite tomar una muestra aleatoria de un vector de números o letras. Escribimos primero de qué vector tomamos la muestra, cuántos objetos extraemos y si es con reposición (replace=TRUE) o no (FALSE).

Simulamos una moneda de dos caras. Y hacemos una tirada de la moneda

```
moneda <- c("cara", "ceca")
tirada1 <- sample(moneda, 1, replace = TRUE)
```

Se podría hacer con

```
monedanum <- c(0,1)
```

para hacer una moneda que dé resultados numéricos.

x es un dado normal de 6 caras

```
x <- c(1,2,3,4,5,6)
```

Podemos hacer un dado de n caras haciendo esto. Primero, creamos un dado vacío. Hacemos loop para llenar el dado con los números de 1 hasta n

```
n <- 100
x <- rep(0, n)
for (i in 1:n)
{
  x[i] <- i
}
```

Vamos a lo que nos pide el ejercicio. Tiramos 2 veces con el comando sample (x es nuestro dado). La escritura de sample es *sample(de dónde elegimos, cuántos valores, si es con reposición, TRUE, o no, FALSE)*. Ponemos muestra2 al final para ver qué dio.

```
muestra2 <- sample(x, 2, replace = TRUE)
muestra2
```

```
## [1] 64 54
```

Repetimos para tirar 1000 veces

```
muestra1000 <- sample(x, 1000, replace = TRUE)
```

Se puede hacer, si no, con un loop. i es el índice de las tiradas, empieza en 1 y vamos subiendo de a 1.

```
i <- 1
muestra <- sample(x, 1, replace = TRUE)
while(i<1000){
  muestra <- c(muestra,sample(moneda, 1, replace = TRUE))
  i <- i+1
}
```

Para estimar las probabilidades a partir de una simulación, podemos contar cuántas veces dio un resultado dado, y dividir por el número total de experimentos.

```
muestra1000[muestra1000==1]

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1
length(muestra1000[muestra1000==1])

## [1] 13
```

`muestra1000[muestra1000==1]` me devuelve sólo aquellas tiradas que salieron 1. El igual igual es para hacer una comparación lógica (si pongo sólo un igual, sirve para asignar un valor, como la flechita `<-`). *Length* me da la longitud de un vector. Acá me dice cuánto mide el vector obtenido de `muestra1000` considerando sólo los 1. O sea, cuántos 1 hubo.

Armo un vector `p` donde voy a almacenar las estimaciones de la proba poniendo 6 veces el 0 (no importa que haya un 0, porque después lo reescribimos).

```
p <- rep(0,6)
p[1] <- length(muestra1000[muestra1000==1])/1000
```

Asignamos a la primera coordenada el resultado de contar cuántos 1 hubo, dividido el número de tiradas. Debería dar cerca de $1/6=0,167$. Si lo quiero hacer para todos los números del dado, para no hacerlo a mano, vamos de 1 a 6 y ponemos ese resultado en la coordenada correspondiente de `p`

```
for (i in 1:6){
  p[i] <- length(muestra1000[muestra1000==i])/1000
}
p

## [1] 0.013 0.015 0.007 0.006 0.009 0.010
```

Juntando todo esto, podemos armar un experimento donde tomamos progresivamente más tiradas, y debería acercarse cada vez más al resultado teórico. Armo un dado y una matriz de estimaciones de la probabilidad de obtener cada número, para 10, 100, ..., 100000 tiradas

```
x <- c(1,2,3,4,5,6)
P <- matrix(nrow = 5, ncol = 6)
for (i in 1:5) {
  n <- 10^i
  muestra <- sample(x, n, replace = TRUE)
  for (j in 1:6){
    P[i,j] <- length(muestra[muestra==j])/n
  }
}
```

```

}
P

##           [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
## [1,] 0.00000 0.3000 0.1000 0.2000 0.4000 0.0000
## [2,] 0.18000 0.1300 0.1700 0.1800 0.1600 0.1800
## [3,] 0.17600 0.1600 0.1630 0.1710 0.1750 0.1550
## [4,] 0.16850 0.1644 0.1617 0.1682 0.1724 0.1648
## [5,] 0.16416 0.1680 0.1678 0.1668 0.1660 0.1671

```

Primero definimos el número de tiradas como 10 a la algo, voy de 10 a 100000. Hago esas tiradas. El segundo loop escribe la estimación de la proba (casos de cada número dividido casos totales) en la fila i de la matriz de estimaciones.

¿Qué le sucede a nuestras estimaciones a medida que aumenta el número de tiradas (o sea, a medida que vamos bajando en la matriz P)?

Ej 1.4

El experimento que hacemos aquí sería: definir una urna con números del 1 al 5, tomar uno de ellos al azar, extraer las bolitas de número mayor a ese, repetir. Para escribirlo en R, empezamos definiendo la urna y un vector bol de 3 elementos que almacenará los resultados de las extracciones. Después de cada extracción, guardamos ese resultado en una coordenada de bol, y redefinimos la urna.

```

urna <- c(1,2,3,4,5)
bol <- rep(0,3)
bol[1] <- sample(urna, 1, TRUE)
urna <- urna[urna>=bol[1]]
bol[2] <- sample(urna, 1, TRUE)
urna <- urna[urna>=bol[2]]
bol[3] <- sample(urna, 1, TRUE)
bol

## [1] 2 2 2

```

En este caso, el experimento dio (3,2,2). Si queremos estimar las probabilidades de obtener cada resultado, seguiremos la misma idea que con el dado del ejercicio 1.1. Repetimos muchas veces el experimento, contamos cuántas veces apareció cada resultado, y esa es nuestra estimación de la probabilidad.

Definimos la urna original, el número N de repeticiones del experimento, el vector bol que representará un experimento dado, y B que almacenará los resultados de todos los experimentos.

```

urna <- c(1,2,3,4,5)
N <- 10000
bol <- rep(0,3)
B <- matrix(nrow = N, ncol = 3)

for (i in 1:N) {
  bol[1] <- sample(urna, 1, TRUE)
  urna2 <- urna[urna<=bol[1]]
  bol[2] <- sample(urna2, 1, TRUE)
  urna3 <- urna[urna<=bol[2]]
  bol[3] <- sample(urna3, 1, TRUE)
  B[i,] <- bol
}

```

Hacemos N experimentos. En cada uno el primer elemento de bol es la primera bolita. Redefinimos la urna sacando los mayores al resultado de ese. Repetimos. Escribimos el vector bol a la fila i-ésima de B. B tiene entonces los resultados de los 10000 experimentos que hicimos (no pedimos mostrar la matriz B porque es gigante). Ahora tenemos que ir contando las filas que corresponden a cada resultado.

```

probas <- matrix(nrow = 35, ncol = 4)
orden <- 1

```

probas es una matriz que almacenará las ternas y los resultados de las estimaciones de las probabilidades (hay 35 resultados posibles, contadas a mano (o con un loop adecuado se pueden contar también)). Orden indica el orden en que almacenaremos las probas, es un contador que sirve para saber en qué fila de probas estamos.

```

for (i in 1:5){
  for (j in 1:i){
    for (k in 1:j) {
      s <- 0
      for (l in 1:N){
        if (sum(B[l,]==c(i,j,k))==3) {
          s <- s + 1
        }
      }
      probas[orden,] <- c(i, j, k, s/N)
      orden <- orden + 1
    }
  }
}

```

En este loop definimos i, j, k como los resultados de las extracciones 1, 2 y 3. Para cada terna posible (i,j,k) definimos s, que va a contar cuántas veces apareció esa terna en los resultados. El último for recorre las filas de B, chequeando si una fila es igual a (i,j,k). Cuando eso sucede sale TRUE TRUE TRUE. La suma de las coordenadas de TRUE TRUE TRUE es 3 porque cada TRUE se toma como que vale 1. Si esa suma da 3, o sea, coincide con (i,j,k), sumamos 1 a s, si no, no. La estimación de la proba es s/N. Escribimos en la fila número orden de la matriz probas la terna i,j,k y la estimación para la probabilidad de obtener esa terna. El contador orden avanza 1 y se repite la indicación. Veamos qué dio (chequeamos que la suma de la fila 4 dé 1, porque estamos estimando probabilidades):

```
probas
```

```
##      [,1] [,2] [,3] [,4]
```

```
## [1,] 1 1 1 0.1983
## [2,] 2 1 1 0.0994
## [3,] 2 2 1 0.0462
## [4,] 2 2 2 0.0480
## [5,] 3 1 1 0.0630
## [6,] 3 2 1 0.0328
## [7,] 3 2 2 0.0373
## [8,] 3 3 1 0.0234
## [9,] 3 3 2 0.0222
## [10,] 3 3 3 0.0218
## [11,] 4 1 1 0.0510
## [12,] 4 2 1 0.0247
## [13,] 4 2 2 0.0232
## [14,] 4 3 1 0.0164
## [15,] 4 3 2 0.0172
## [16,] 4 3 3 0.0185
## [17,] 4 4 1 0.0140
## [18,] 4 4 2 0.0130
## [19,] 4 4 3 0.0109
## [20,] 4 4 4 0.0135
## [21,] 5 1 1 0.0416
## [22,] 5 2 1 0.0214
## [23,] 5 2 2 0.0201
## [24,] 5 3 1 0.0145
## [25,] 5 3 2 0.0121
## [26,] 5 3 3 0.0132
## [27,] 5 4 1 0.0100
## [28,] 5 4 2 0.0089
## [29,] 5 4 3 0.0137
## [30,] 5 4 4 0.0099
## [31,] 5 5 1 0.0095
## [32,] 5 5 2 0.0078
## [33,] 5 5 3 0.0070
## [34,] 5 5 4 0.0073
## [35,] 5 5 5 0.0082
```

```
sum(probas[,4])
```

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

Ej 1.17

Tenemos un dado normal y dos urnas con bolitas rojas, R, y blancas, B. Definimos cada una de ellas. El experimento consiste en tirar el dado, y si sale 3 o 6, sacar una bolita de A y ponerla en B, y si no, al revés. Nos interesan los resultados de estas tres acciones: el dado (que irá a resdado), la primera bolita (resbol1) y la segunda bolita (resbol2). Repetimos el experimento muchas veces (exps es el número de repeticiones).

```
dado <- c(1,2,3,4,5,6)
urnaA <- c(rep("R",5), rep("B",3))
urnaB <- c(rep("R",1), rep("B",2))

exps <- 100000
resdado <- rep(0,exps)
```

```

resbol1 <- rep("0",exps)
resbol2 <- rep("0",exps)

for (i in 1:exps){
  t <- sample(dado, 1, TRUE)
  resdado[i] <- t
  if (t==3 | t==6){
    ext <- sample(urnaA, 1, TRUE)
    resbol1[i] <- ext
    urnaB1 <- c(urnaB, ext)
    resbol2[i] <- sample(urnaB1, 1, TRUE)
  } else {
    ext <- sample(urnaB, 1, TRUE)
    resbol2[i] <- ext
    urnaA1 <- c(urnaA, ext)
    resbol1[i] <- sample(urnaA1, 1, TRUE)
  }
}

```

En el loop armamos un experimento y lo repetimos hasta llegar a `exps`. El condicional `if` nos indica que si sale 3 o 6 en el dado, la primera bolita se extraerá de A. Armamos una urna B1, que tiene las bolitas de B y la que recién extrajimos, y sacamos la segunda de ahí. Escribimos todo a los vectores correspondientes. Si el dado no da 3 o 6, el segundo caso es igual al primero pero al revés (está en la parte `else` del condicional).

Ahora procesamos los resultados. Armamos un dataframe `res` a partir de los vectores con la información de los resultados (no podemos escribirlos directamente a una matriz como en los anteriores porque el resultado del dado es un número, pero el de las bolitas es una letra).

Contamos en ese dataframe cuántas veces vale que `resbol1` y `resbol2` coinciden teniendo una bolita roja (si eso sucede, da `TRUE`. El comando `sum` está para contar todos esos `TRUE`; cada uno vale 1). Este valor es `RR`.

```

res <- data.frame(resdado, resbol1, resbol2)

RR <- sum(res$resbol1 == "R" & res$resbol2 == "R")
probaRR <- RR/exps
probaRR

```

```
## [1] 0.24925
```

Dividimos `RR` por la cantidad total de experimentos. El valor “verdadero” de esta probabilidad es $109/432$, aproximadamente 0,2523. ¿Qué tan cerca nos dio? ¿Qué pasaría si aumento `exps`, el número de experimentos?

Para la segunda pregunta, nos piden la probabilidad de que haya salido un 3 o un 6 en el dado, considerando que ambas bolitas fueron rojas. Para estimar esta probabilidad condicional, tenemos que contar cuántas veces sucedió que `resdado` valió 3 o 6 y salió `RR`, pero dividiendo por el total de veces que salió `RR` (no dividiendo por `exps`). En el fondo, esto no es más que seguir la definición de la proba condicional.

```

o3o6dadoRR <- sum((res$resdado == 3 | res$resdado == 6) & res$resbol1 == "R" & res$resbol2 == "R")
proba3o6dadoRR <- o3o6dadoRR/RR
proba3o6dadoRR

```

```
## [1] 0.4157673
```

El valor teórico es $45/109$, o sea, aproximadamente 0,4128. ¿La aproximación es buena?