Problema del cumpleaños / fechas de nacimiento

Alarcón González Edgar Gerardo

6 de Junio del 2018

El objetivo de este documento es trabajar con elementos básicos de R de manera didáctica y aplicada. Por esta razón, no entraremos en detalle de cómo funciona la teoría ni las diferentes variaciones que tiene el modelo, simplemente nos remitiremos a ejemplificar su uso en un contexto real. Algo importante qué destacar de éste documento es, que a pesar de prestarse la situación **no se utilizó un solo ciclo for o afín**.

Problema

Supongamos que tenemos una muestra de n personas, tomemos los días del año como $1 = 1/\text{Enero}$ y $365 = 31/\text{Diciembre}$ (suponiendo un año "normal", claro) entonces, considerando la hipótesis de que es equiprobable nacer cualquier día del año, ¿Cuál es la probabilidad de que **AL MENOS** dos de las n personas cumplan exactamente el mismo día?

Solución real aproximada

Se deja como ejercicio para el lector demostrar que, dadas n personas, la probabilidad real aproximada es:

$$p \approx 1 - exp\left(-\frac{n(n-1)}{730}\right)$$

Hint: Utiliza el complemento de lo que buscamos y en algún momento que $x \approx ln(1+x)$ si $x \approx 0$. No es tan trivial... lo siento.

Solución vía simulación

Supongamos primero $n = 180$. La primera pregunta que debemos hacernos es ¿cómo obtengo un éxito?, para solucionar esto, lo primero que haremos es generar una muestra de tamaño n (que representará a los sujetos) con reemplazo de números entre el 1 al 365, pues este es nuestro espacio muestral Ω.

```
n<-180 ; set.seed(21)
M<-sample(x = 1:365,   #Del vector 1:365
          size = n,    #Toma una muestra de tamaño n
          replace = T) #Con reemplazo.
```

Dentro de éste vector, debemos ver si tenemos duplicado al menos un valor, lo cual significaría que dos personas cumplen años exactamente el mismo día. Para lograr esto, utilizaremos la función `anyDuplicated` la cual, de ser el caso, arroja el id (o la posición dentro del vector) del primer valor duplicado, es decir si ya salió un "x" antes, arroja el id del inmediato siguiente idéntico a "x"; si no hay duplicados, entonces arrojará un "0".

```
id<-anyDuplicated(x = M) ; id
```

```
## [1] 31
```

```r
#¿En dónde salió el número que se duplicó primero?
which(M==M[id])
```

```
## [1] 12 31
```

En este caso, el número en la entrada 12, se repitió en la entrada 31, valor que nos arroja la función `anyDuplicated`.

Por lo tanto, tendremos un éxito si `anyDuplicated`$\neq 0$ (pues eso indicaría que al menos dos personas nacieron el mismo día en nuestra muestra). Marquemos como 1 si hay un éxito y como 0 si no lo hay:

```r
prueba<-anyDuplicated(x = M)

#Si hubo éxito, marca 1, si no, marca 0.
Exito<-ifelse(test = prueba!=0,yes = 1,no = 0) ; Exito
```

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

Lo anterior lo hicimos únicamente una vez, pero nos gustaría estimar la probabilidad en cuestión, por lo cual debemos realizar el experimento anterior "muchas veces" para tener un número considerable de éxitos y ensayos, lo cuál al tomar su cociente, obtendríamos la probabilidad empírica (estimada) solicitada. Para lograr repetir "varias" veces lo anterior, necesitamos "varias" muestras, por lo que haremos uso de la función `replicate`, la cual hace cierto experimento el número de veces indicados, para este caso, consideremos un número de personas $n = 7$ y un número de ensayos Ensayos $= 100{,}000$:

```r
set.seed(6); n<-7 ; Ensayos<-100000
#Replicate hace un experimento varias veces.
M<-replicate(n = Ensayos, #Realiza "Ensayos" veces, la siguiente expresión:
             expr = sample(x = 1:365, size = n, replace = T))
```

En este caso, nuestro objeto `M` es una matriz de $n \times$ Ensayos llena de números entre el 1 y el 365 (distribuidos equiprobablemente), lo cual nos dice que tenemos "Ensayos" muestras de n sujetos cada una, donde cada muestra es una columna. Por este motivo, debemos aplicar la función `anyDuplicated` por columnas como muestra el siguiente código:

```r
#Realizamos la prueba por columnas:
prueba<-apply(X = M,                #A la matriz M,
              MARGIN = 2,           #aplica por columnas,
              FUN = anyDuplicated)  #la función anyDuplicated.
```

Finalmente, convertimos en 1 los éxitos, en 0 los fracasos y estimamos la probabilidad de la siguiente manera:

```r
#Entrada a entrada, si hubo éxito, marca 1, si no, marca 0.
Exitos<-ifelse(test = prueba!=0,yes = 1,no = 0)
#length(Éxitos)

#Probabilidad estimada:
no.Exitos<-sum(Exitos)
no.Exitos/Ensayos
```

```
## [1] 0.05568
```

En la teoría, para n personas, la probabilidad real aproximada es:

```r
1-exp(-n*(n-1)/(2*365))
```

```
## [1] 0.05591044
```

Ahora, hagamos lo anterior para diferentes valores de n creando dos funciones, una que estime la probabilidad y otro que calcule el valor aproximado real.

```r
#Podemos crear una función entonces que, dada cierta "n"
#nos arroje la probabilidad estimada como sigue:

prob.estimada<-function(n=100,Ensayos=10000){

  M<-replicate(n = Ensayos, #Realiza "Ensayos" veces, la siguiente expresión:
               expr = sample(x = 1:365, size = n, replace = T))
  #Realizamos la prueba por columnas:
  prueba<-apply(X = M,                #A la matriz M,
                MARGIN = 2,           #aplica por columnas,
                FUN = anyDuplicated)  #la función anyDuplicated.

  #Entrada a entrada, si hubo éxito, marca 1, si no, marca 0.
  Exitos<-ifelse(test = prueba!=0,yes = 1,no = 0)

  #Probabilidad estimada:
  no.Exitos<-sum(Exitos)
  return(no.Exitos/Ensayos)

}

#Podemos crear una función entonces que, dada cierta "n"
#nos arroje la probabilidad real aproximada como sigue:

prob.aprox<-function(n){

  return(1-exp(-n*(n-1)/(730)))

}
```

Tomando entonces diferentes valores de n obtenemos los siguientes resultados:

```r
#A diferentes valores de n, vamos el resultado:
n<-seq(from = 5, #Del 5
       to = 50,  #al 50
       by = 5)   #De 5 en 5.

#Con este vector, calculamos valores estimados y aprox. reales.

#Aprox. Reales
Aprox.Reales<-sapply(X = n, #Al vector "n" , entrada a entrada
                     FUN = prob.aprox) #aplicale la función dada.

#Estimados
set.seed(21)
Estimados<-sapply(X = n, #Al vector "n" , entrada a entrada
                  FUN = prob.estimada) #aplicale la función dada.

#Mostramos los resultados:
Resultados<-data.frame(n,Aprox.Reales,Estimados)
library(knitr) ; kable(Resultados)
```

n	Aprox.Reales	Estimados
5	0.0270254	0.0260

n	Aprox.Reales	Estimados
10	0.1159907	0.1146
15	0.2499919	0.2516
20	0.4058051	0.4047
25	0.5604122	0.5676
30	0.6963200	0.7061
35	0.8040973	0.8097
40	0.8819900	0.8938
45	0.9336180	0.9383
50	0.9651313	0.9691

¿Cuánto tardó en generar todo?

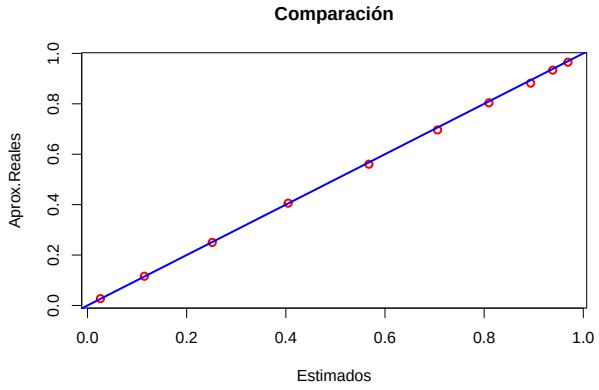
```
set.seed(21)
system.time(expr =  sapply(X = n, FUN = prob.estimada))
```

```
##    user  system elapsed
##    4.53    0.00    4.59
```

En estimar para todos los valores dada cada n, tardó menos de 5 segundos. Por último, graficando nuestros resultados.

```
#Grafica:
plot(Aprox.Reales~Estimados, #Estimados Vs. Aprox.
     main="Comparación", #Pon como título,
     col="red", #Color rojo,
     lwd=2)     #con ancho 2.

#Agrega la siguiente línea:
abline(a=0,        #ordenada de origen 0,
       b = 1,      #pendiente 1,
       col="blue", #color azul,
       lwd=2)      #ancho 2
```



De donde vemos que dado que los puntos parecen encontrarse de manera adecuada en la recta identidad, los valores aproximados reales son muy similaes a los valores estimados, por lo que la probabilidad simulada es correcta.