

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE MATEMÁTICA



Licenciatura en Estadística

Control Estadístico del Paquete R

”UNIDAD TRES”

Alumna:
Erika Beatríz Guillén Pineda

Fecha de elaboración
Santa Ana - 27 de noviembre de 2015

GENERACIÓN DE ESPACIOS MUESTRALES Y DE MUESTRAS ALEATORIAS.

La función `sample()`: permite seleccionar una muestra aleatoria de tamaño `n`, especificado el vector `x` desde el cual tomará la muestra (normalmente es un vector de caracteres aunque no es indispensable), la selección puede ser con o sin reemplazo. La sintaxis general de esta función es:

`sample(X, size, replace = FALSE, prob = NULL)`

donde

- `X`: es el vector del cual se seleccionan la muestra (podría decirse que representa el marco muestral).
- `size`: es el tamaño de la muestra.
- `replace = FALSE` indica que la muestra es sin reposición, si fuera `TRUE` sería con reposición.
- `prob`: vector de pesos o probabilidad de obtener los elementos del vector `X` que esté siendo muestreado (en caso de que los elementos tengan distintas probabilidades).

1. Activa tu directorio de trabajo

```
getwd()

## [1] "C:/Users/User/Documents/TODAS_PRACTICAS"

setwd("C:/Users/User/Documents/TODAS_PRACTICAS")
```

2. Crea un nuevo Script y llamarle "Script13-Probabilidades1"

3. Simular 10 lanzamientos de una moneda

```
# vector del cual se tomara la muestra

moneda <- c("C", "+"); moneda

## [1] "C" "+"
```

```
# Tamaño de la muestra

n <- 10; n

## [1] 10
```

```
# Generando la muestra aleatoria con reemplazamiento (replace=TRUE)

lanzamientos <- sample(moneda, n, replace=TRUE); lanzamientos

## [1] "+" "+" "+" "C" "C" "+" "+" "+" "C" "+"
```

4. Elegir 6 números de una lotería de 54 números

```
# se define el espacio muestral del cual se tomara la muestra

espacio <- 1:54;espacio

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
## [47] 47 48 49 50 51 52 53 54

# Se define el tamaño de la muestra

n <- 6; n

## [1] 6
```

```
# seleccionando la muestra sin reposición

muestra <- sample(espacio, n); muestra

## [1] 23 3 4 29 16 13
```

OBSERVACIÓN: por defecto la selección es sin reemplazo o sin reposición, pero no se reduce el espacio muestral; en otras palabras lo que esto significa es que a pesar de que la muestra se selecciona sin reposición, el vector (del cual se selecciona la muestra) permanece sin cambio alguno; para nuestro ejemplo en particular en el vector muestra se almacenan los 6 elementos seleccionados del vector espacio, sin embargo, en el vector espacio estos elementos se conservan; esto presentan un inconveniente si se desea seleccionar una segunda muestra pero en la cual no se encuentre ningún elemento contenido en la primera, tendrían que descartarse primero antes de tomar una segunda muestra.

5. Simular 4 lanzamientos de dos dados

```
# genera el espacio muestral del lanzando de los dos dados

espacio = as.vector(outer(1:6, 1:6, paste)); espacio
```

```
## [1] "1 1" "2 1" "3 1" "4 1" "5 1" "6 1" "1 2" "2 2" "3 2" "4 2" "5 2"
## [12] "6 2" "1 3" "2 3" "3 3" "4 3" "5 3" "6 3" "1 4" "2 4" "3 4" "4 4"
## [23] "5 4" "6 4" "1 5" "2 5" "3 5" "4 5" "5 5" "6 5" "1 6" "2 6" "3 6"
## [34] "4 6" "5 6" "6 6"
```

```
# la funci\on outer genera un arreglo (una matriz) de caracteres en el cual el
# primer elemento es un n\umero entre 1 y 6 (obtenido por la instruccion 1:6),
# mientras que en el segundo tambi\en es un n\umero entre 1 y 6 (obtenido por
# la instrucc\on 1:6). Es un arreglo de orden 6 x 6.
```

```
# con la instrucc\on as.vector se convierte en un vector el arreglo.
```

```
# se define el tama\~no de la muestra
```

```
n <- 4; n
```

```
## [1] 4
```

```
# finalmente se selecciona la muestra
```

```
muestra <- sample(espacio, n, replace=TRUE); muestra
```

```
## [1] "5 4" "3 4" "6 6" "5 2"
```

6. Seleccionar cinco cartas de un naipes de 52 cartas

```
#genera el espacio muestral de las 52 cartas
```

```
naipes = paste(rep(c("A", 2:10, "J", "Q", "K"), 4),
                c("OROS", "COPAS", "BASTOS", "ESPADAS"));naipes
```

```
## [1] "A OROS" "2 COPAS" "3 BASTOS" "4 ESPADAS" "5 OROS"
## [6] "6 COPAS" "7 BASTOS" "8 ESPADAS" "9 OROS" "10 COPAS"
## [11] "J BASTOS" "Q ESPADAS" "K OROS" "A COPAS" "2 BASTOS"
## [16] "3 ESPADAS" "4 OROS" "5 COPAS" "6 BASTOS" "7 ESPADAS"
## [21] "8 OROS" "9 COPAS" "10 BASTOS" "J ESPADAS" "Q OROS"
## [26] "K COPAS" "A BASTOS" "2 ESPADAS" "3 OROS" "4 COPAS"
## [31] "5 BASTOS" "6 ESPADAS" "7 OROS" "8 COPAS" "9 BASTOS"
## [36] "10 ESPADAS" "J OROS" "Q COPAS" "K BASTOS" "A ESPADAS"
## [41] "2 OROS" "3 COPAS" "4 BASTOS" "5 ESPADAS" "6 OROS"
## [46] "7 COPAS" "8 BASTOS" "9 ESPADAS" "10 OROS" "J COPAS"
## [51] "Q BASTOS" "K ESPADAS"
```

```
# con la instrucció\ 'on rep(c("A", 2:10, "J", "Q", "K"), 4) se crea un vector de
# caracteres, el primer elemento es "A", los elementos de segundo al undécimo
# son n\ 'umero del 2 al 10, los siguientes elementos son "J", "Q" y "K";
# y los elementos se repiten en este orden cuatro veces.
```

```
#con la funció\ 'on paste se crea un vector en el que sus elementos son: un
# elemento del vector (rep(c("A", 2:10, "J", "Q", "K"), 4)
# concatenado con uno del vector c("OROS", "COPAS", "BASTOS", "ESPADAS"))
```

- El primer elemento de `rep(c("A", 2:10, "J", "Q", "K"), 4)` con el primero de `c("OROS", "COPAS", "BASTOS", "ESPADAS")`.

```
naipe1 = paste(rep(c("A"), 4),
               c("OROS"));naipe1

## [1] "A OROS" "A OROS" "A OROS" "A OROS"
```

- El segundo elemento de `rep(c("A", 2:10, "J", "Q", "K"), 4)` con el segundo de `c("OROS", "COPAS", "BASTOS", "ESPADAS")`.

```
naipe2 = paste(rep(c( 2:10), 4),
               c("COPAS"));naipe2

## [1] "2 COPAS" "3 COPAS" "4 COPAS" "5 COPAS" "6 COPAS" "7 COPAS"
## [7] "8 COPAS" "9 COPAS" "10 COPAS" "2 COPAS" "3 COPAS" "4 COPAS"
## [13] "5 COPAS" "6 COPAS" "7 COPAS" "8 COPAS" "9 COPAS" "10 COPAS"
## [19] "2 COPAS" "3 COPAS" "4 COPAS" "5 COPAS" "6 COPAS" "7 COPAS"
## [25] "8 COPAS" "9 COPAS" "10 COPAS" "2 COPAS" "3 COPAS" "4 COPAS"
## [31] "5 COPAS" "6 COPAS" "7 COPAS" "8 COPAS" "9 COPAS" "10 COPAS"
```

- El tercer elemento de `rep(c("A", 2:10, "J", "Q", "K"), 4)` con el tercero de `c("OROS", "COPAS", "BASTOS", "ESPADAS")`.

```
naipe3 = paste(rep(c("J"), 4),
               c("BASTOS"));naipe3

## [1] "J BASTOS" "J BASTOS" "J BASTOS" "J BASTOS"
```

- El cuarto elemento de `rep(c("A", 2:10, "J", "Q", "K"), 4)` con el cuarto de `c("OROS", "COPAS", "BASTOS", "ESPADAS")`.

```
naipe4 = paste(rep(c("Q"), 4),
               c("ESPADAS"));naipe4

## [1] "Q ESPADAS" "Q ESPADAS" "Q ESPADAS" "Q ESPADAS"
```

```
# se define el tamaño de la muestra
```

```
n <- 5; n
```

```
## [1] 5
```

```
# se obtiene la muestra sin reemplazo (aunque no se especifique  
# con replace=FALSE)
```

```
cartas <- sample(naipes, n) ; cartas
```

```
## [1] "3 COPAS" "2 COPAS" "5 BASTOS" "8 COPAS" "2 OROS"
```

7. Generar una muestra aleatoria de tamaño 120, con los números del 1 al 6 en el que las probabilidades de cada uno de los números son respectivamente los siguientes valores: 0.5, 0.25, 0.15, 0.04, 0.03 y 0.003.

```
sample(1:6, 120, replace=TRUE, c(0.5, 0.25, 0.15, 0.04, 0.03, 0.03))
```

```
## [1] 1 1 1 2 2 1 5 1 3 2 2 1 2 2 1 4 2 1 1 1 1 1 4 1 4 3 4 1 1 2 1 3 2 2 5  
## [36] 2 6 1 1 1 1 1 2 2 1 2 1 1 3 1 3 1 2 1 1 1 1 2 2 1 1 2 3 1 3 1 2 1 1 3  
## [71] 3 3 5 1 1 1 2 1 2 3 6 1 1 3 1 4 1 2 1 2 2 1 3 1 1 1 1 2 1 3 1 2 4 1 1  
## [106] 3 1 1 2 2 5 1 2 1 3 1 1 1 6 3
```

```
# note que en el vector c(0.5, 0.25, 0.15, 0.04, 0.03, 0.03) se especifican  
# las probabilidades de cada uno de los elementos, por lo que la suma de cada  
# uno de los elementos del vector debe ser uno.
```

```
# note que R siempre generara la muestra sin importar si la suma es o no la  
# unidad, sin embargo, para que la muestra sea verdaderamente aleatoria la  
# suma de las probabilidades debe ser igual a la unidad.
```

8. Escriba una función que reciba los números enteros entre 1 y 500 inclusive, la función retornará el espacio formado por los números divisibles entre 7. Después de llamar a esta función se extraerá aleatoriamente 12 de estos números, con reemplazo.

```
# definiendo la función que generara el espacio formado
```

```
espacio <- function(num)
```

```
{
```

```
  numDiv7 <- numeric(0)
```

```
  ind <- 0
```

```
for(i in 1:length(num))
if ((num[i] %% 7)==0)

{

  ind <- ind+1
  numDiv7[ind]=num[i]

}

return(numDiv7)

}
```

```
numeros <- 1:500
```

```
# generando el espacio muestral
```

```
s <- espacio(numeros); s
```

```
## [1] 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119
## [18] 126 133 140 147 154 161 168 175 182 189 196 203 210 217 224 231 238
## [35] 245 252 259 266 273 280 287 294 301 308 315 322 329 336 343 350 357
## [52] 364 371 378 385 392 399 406 413 420 427 434 441 448 455 462 469 476
## [69] 483 490 497
```

```
# seleccionando la muestra
```

```
muestra <- sample(s, 12, replace=TRUE); muestra
```

```
## [1] 455 322 378 14 105 182 371 126 343 112 112 315
```