

# Guia 4 Ejercicio 6

Agustin Muñoz Gonzalez

30/4/2020

## Preparamos el entorno.

Limpiamos los registros.

```
rm(list=ls())
```

Vamos a necesitar la función `exito_fracaso()` del Ejercicio 4. Recordemos que si queremos simular un experimento cuyo éxito tiene probabilidad  $p$ , lo que podemos hacer es tomar una muestra al azar de una variable uniformemente distribuida en el intervalo  $[0,1]$  y pedirle que esa muestra sea mayor o igual que  $1-p$ .

```
exito_fracaso=function(p){  
  as.integer(runif(1) >= 1-p)  
}
```

## Ejercicio 6

### Item a)

Definimos una función que devuelve la cantidad `N_rep` de experimentos necesarios hasta obtener el primer éxito

```
perseverancia_exito=function(p){  
  N_rep=1  
  while(exito_fracaso(p) != 1){  
    N_rep=N_rep+1  
  }  
  N_rep  
}
```

Como el experimento depende de la probabilidad  $p$ , esa probabilidad debe ser ingresada como input de la función.

### Item b)

En primer lugar definimos una función que devuelve el vector de resultados de  $n$  ejecuciones de la función `perseverancia_exito(p)`.

```
n_perseverancia_exito=function(n,p){  
  resultado=c()  
  for(i in (1:n)){  
    resultado=c(resultado,perseverancia_exito(p))  
  }  
  resultado  
}
```

Ejecutamos la función `perseverancia_exito` 1000 veces para  $p=0.8$ . Esto es aplicar `n_perseverancia_exito(1000,0.8)`

```
muchas_perseverancia_exito=n_perseverancia_exito(1000,0.8)
```

Calculemos la frecuencia relativa a cada valor (contado con repeticiones) de muchas\_perseverancias. Devolvemos el resultado como un vector de listas donde el nombre de cada lista es el número i de la iteración, es decir el valor que toma cada componente de muchas\_perseverancias.

```
aux=c()
for(i in muchas_perseverancia_exito){
  aux=c(aux,list(i=sum(muchas_perseverancia_exito==i)/1000))
  # otra forma: aux[i]=sum(muchas_perseverancia_exito==i)/1000
}
frec_muchas_perseverancia_exito=unique(aux)
frec_muchas_perseverancia_exito
```

```
## [[1]]
## [1] 0.176
##
## [[2]]
## [1] 0.786
##
## [[3]]
## [1] 0.028
##
## [[4]]
## [1] 0.008
##
## [[5]]
## [1] 0.001
```

Por último calculamos el promedio de muchas\_perseverancias.

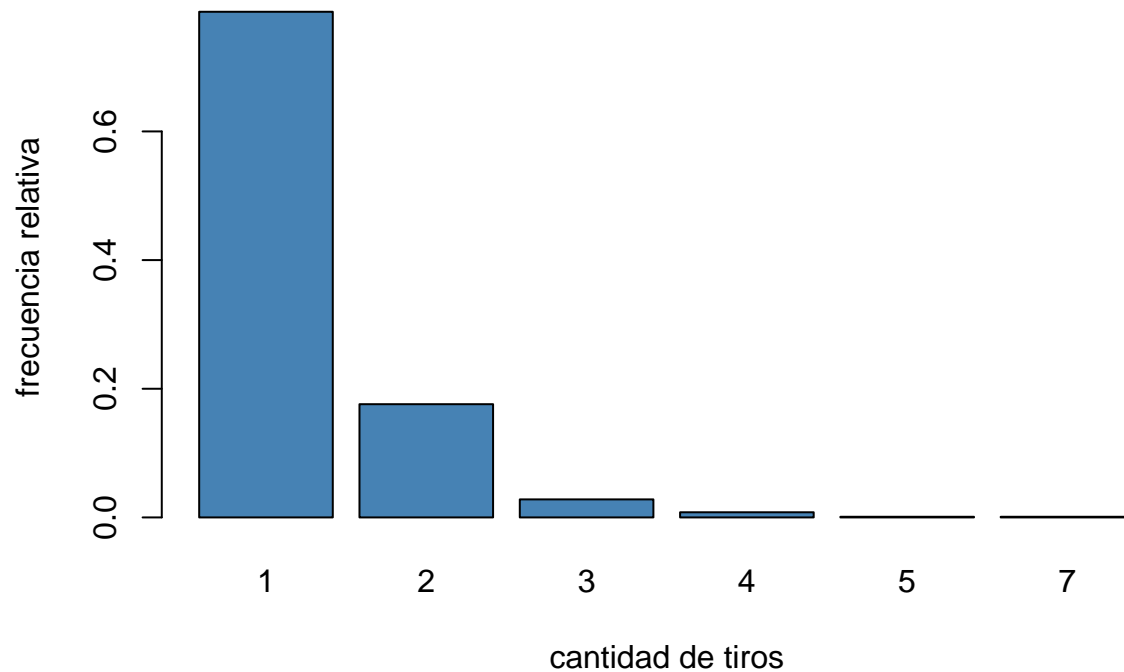
```
promedio_muchas_perseverancia_exito=sum(muchas_perseverancia_exito)/1000
promedio_muchas_perseverancia_exito
```

```
## [1] 1.266
```

Otra forma de hacer la frec relativa:

Hagamos una tablita con las frec relativas.

```
tablita=table(muchas_perseverancia_exito)/1000
barplot(tablita,col='steelblue',xlab='cantidad de tiros',
        ylab='frecuencia relativa')
```



#### Item c)

Repetimos el item anterior para la grilla siguiente

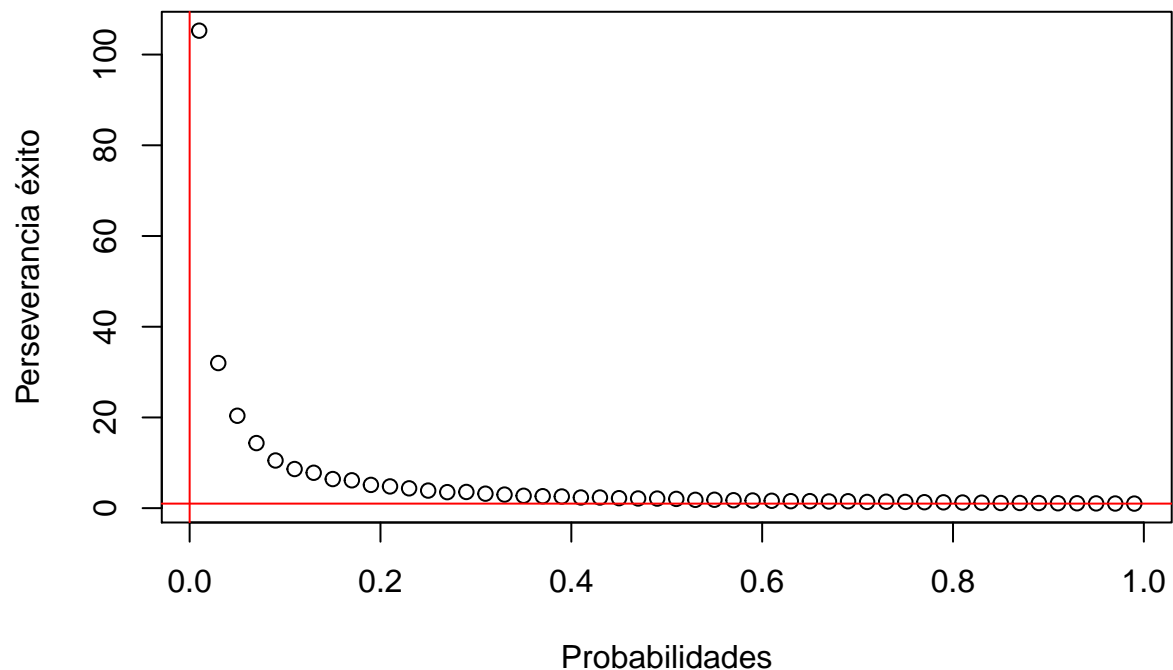
```
grilla=seq(0.01,0.99,by=0.02)
```

De nuevo, devolvemos un vector de listas donde el nombre de cada lista es la probabilidad respecto de la cual estamos realizando los 1000 experimentos y cuyo valor son los 1000 experimentos.

```
resultado_grilla=c()
for (i in grilla){
  resultado_grilla=c(resultado_grilla,list(n_perseverancia_exito(1000,i)))
}
names(resultado_grilla)=paste('p=',grilla)
```

Graficamos p vs el promedio de `n_perseverancia_exito(1000,p)`. Definimos una función para calcular el promedio de la función `n_perseverancia_exito(1000,-)` y de paso nos sirve para poder usar el comando `lapply` a la grilla.

```
promedio_1000_ejecuciones=function(p){
  x=n_perseverancia_exito(1000,p)
  sum(x)/1000
}
y=lapply(grilla,promedio_1000_ejecuciones)
plot(grilla,y,xlab='Probabilidades',
     ylab='Perseverancia éxito')
abline(h=1,v=0,col='red')
```



Vemos que tiene una forma asintótica. Siendo un poco más formales podemos decir que cuando  $p$  tiende a 0 la perseverancia tiende a infinito, lo cual es intuitivo ya que si  $p=0$  nunca vamos a obtener un éxito y por tanto vamos a repetir el experimento infinitas veces. Análogamente si  $p$  tiende a 1 la perseverancia tiende a 1, de hecho cuando  $p$  ES 1 la perseverancia ES 1, porque si la probabilidad de que el experimento sea exitoso es 1 entonces al primer intento obtendremos un éxito.

Proponemos la función  $f(x) = \frac{1}{x}, x > 0$ . Grafiquemos esta curva superpuesta con el gráfico anterior. Primero definamos la función  $f$  y la imagen de la grilla vía  $f$

```
f=function(x){1/x}
f_grilla=lapply(grilla,f)
```

Grafiquemos

```
plot(grilla,y,xlab='Probabilidades',
     ylab='Perseverancia éxito')
abline(h=1,v=0,col='red')
lines(grilla,f_grilla,col='blue')
```

