

Guia 6

Agustín Muñoz González

4/5/2020

Preparamos el entorno.

Limpiamos los registros.

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

Ejercicio 1

El comando para simular una elección al azar de un número de la tira (1:6) es

```
sample((1:6),1)
```

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

Ejercicio 2

Vamos a definir dos funciones: una que devuelva la cantidad de figuritas que debemos comprar para completar el álbum y el álbum per sé; y otra que devuelva sólo la cantidad de figuritas.

Lo que vamos a hacer es entender al álbum como una tira de 6 elementos, que en tiempo 0 es la tira `c(0,0,0,0,0,0)`, y en cada paso vamos a elegir una muestra al azar de la tira (1:6) y vamos a sumarle un 1 en la coordenada correspondiente al número que hayamos obtenido.

La iteración terminará cuando el álbum no tenga ninguna componente nula.

Algo práctico de esta forma de resolver el problema es que no sólo obtenemos la cantidad de figuritas que hay que comprar (i.e. la cantidad de números al azar que hay que elegir) sino que el vector final del álbum especifica cuántas figuritas tenemos de cada número.

```
figus.y.album=function(){
  album=c(0,0,0,0,0,0)
  figus_compradas=0
  contador=0
  while(sum(album==0)!=0){
    figu_nueva=sample(1:6,1)
    album[figu_nueva]=album[figu_nueva]+1
    contador=contador+1
  }
  return(list('Intentos totales hasta completar: '=contador,'Album: '=album))
}
```

```
figus=function(){
  album=c(0,0,0,0,0,0)
  figus_compradas=0
  contador=0
  while(sum(album==0)!=0){
    figu_nueva=sample(1:6,1)
    album[figu_nueva]=album[figu_nueva]+1
    contador=contador+1
  }
  return(contador)
}
```

Ejercicio 3

Vamos ahora a generalizar las funciones del item anterior para que tomen como input un parámetro `figusTotal` que va a representar la longitud del álbum.

```
cuantasFigus_etiqueta=function(figusTotal){  
  album=rep(0,figusTotal)  
  contador=0  
  while(sum(album==0)!=0){  
    figu_nueva=sample(1:figusTotal,1)  
    album[figu_nueva]=album[figu_nueva]+1  
    contador=contador+1  
  }  
  return(list('Intentos totales hasta completar: '=contador,'Album: '=album))  
}
```

```
cuantasFigus=function(figusTotal){  
  album=rep(0,figusTotal)  
  contador=0  
  while(sum(album==0)!=0){  
    figu_nueva=sample(1:figusTotal,1)  
    album[figu_nueva]=album[figu_nueva]+1  
    contador=contador+1  
  }  
  return(contador)  
}
```

Ejercicio 4

Simularemos la función del item anterior $N_rep=1000$ veces para $figusTotal=6$. Vamos a definir una función que tome como input N_rep y devuelva un vector formado por los resultados de cada experimento.

```
n_albums=function(figusTotal,N_rep){
  resultado=c()
  for(i in (1:N_rep)){
    resultado=c(resultado,cuantasFigus(figusTotal))
  }
  #names(resultado)=paste('Experimento n° ',(1:N_rep))
  resultado
}
```

Ejecutamos `n_albums` para $figusTotal=6$ y $N_rep=1000$.

```
N_rep=1000
cantidades=n_albums(6,N_rep)
```

Vamos a resolver los siguientes items para el experimento `cantidades`.

Item 4.a

Calculemos la media de figuritas que hay que comprar.

```
mean(cantidades)
```

```
## [1] 14.905
```

Item 4.b

Calculemos la probabilidad de completar el álbum comprando 16 figuritas de dos formas: una dividiendo 16 por la media del item anterior y otra dividiendo 16 por `figus()` (i.e. por la cantidad de `figus` necesarias para completar 1 album).

```
proba_1=function(){
  aux=16/(mean(cantidades))
  if(1<=aux){1}else{aux}
}
```

```
proba_2=function(){
  aux=16/figus()
  if(1<=aux){1}else{aux}
}
```

LAS ANTERIORES ESTÁN MAL! PENSAR POR QUÉ! O MEJOR, QUÉ SIGNIFICA EL NÚMERO QUE CALCULÉ ANTES? PRIMERO NO TIENE SENTIDO QUE EL NUMERADOR SEA MAS GRANDE QUE EL DENOMINADOR EN EL CALCULO DE UNA PROBA. ADEMÁS, EL ERROR CREO QUE ES QUE SI DIVIDIS POR `figus()` O POR LA MEDIA ESTAMOS DICIENDO QUE ESE DENOMINADOR ES LA CANTIDAD TOTALES DE EXPERIMENTOS, PERO ESOS NUMEROS EN REALIDAD SON LA CANTIDAD QUE HABRIA QUE SACAR PARA TENER EXITO, PERO UNO TIENE QUE DIVIR POR TODOS LOS EXPERIMENTOS NO SOLO POR LOS EXITOSOS. LA FORMA CORRECTA ES LA DEL 'OJO!'

Ojo: Hay que fijarse cuantos son ≤ 16 en `cantidades` y dividido por n_rep .

```
proba_3=function(){
  sum(cantidades<=16)/N_rep
}
```

```
proba_N_rep=function(N_rep){
  sum(n_albums(6,N_rep)<=16)/N_rep
}
```

Para calcular la frecuencia relativa de los casos en los que tenemos ≤ 16

```
mean(cantidades<=16)
```

```
## [1] 0.681
```

Item 4.c

Calcularemos el número de figuritas que hay que comprar para tener una probabilidad de 0.9 de completar el álbum.

```
n=figus()*0.9
```

LO ANTERIOR ESTÁ MAL! PENSAR POR QUÉ! O MEJOR, QUÉ SIGNIFICA EL NÚMERO QUE CALCULÉ ANTES? DE NUEVO, figus() NO ES LA CANTIDAD TOTAL DE EXPERIMENTOS.

Ojo! Usamos quantil. O sea, quantil(cantidades,0.9) devuelve el elemento de cantidades que deja el 90% de los datos a su izquierda. ¿Por qué este número es la cantidad de figus que hay que comprar para tener 90% de completar el álbum? Pensemoslo al revés, si compro quantile(cantidades,0.9) figuritas, entonces hay un 10% de cantidad de figuritas que hay que comprar para completar el album, entonces hay sólo un 10% de chances de que nuestro número no sea la cantidad que había que comprar. Digamos nuestro número ‘incluye’ el 90% de los numeros anteriores con lo cual vamos a tener éxito salvo que JUSTO la cantidad que habia que comprar era la del 10% restante. O sea tenemos 10% de pifiar y entonces 90% de tener éxito.

```
n=quantile(cantidades,0.90)
#buscamos la posicion mas parecida a 0.9*N_rep
```

otra forma:

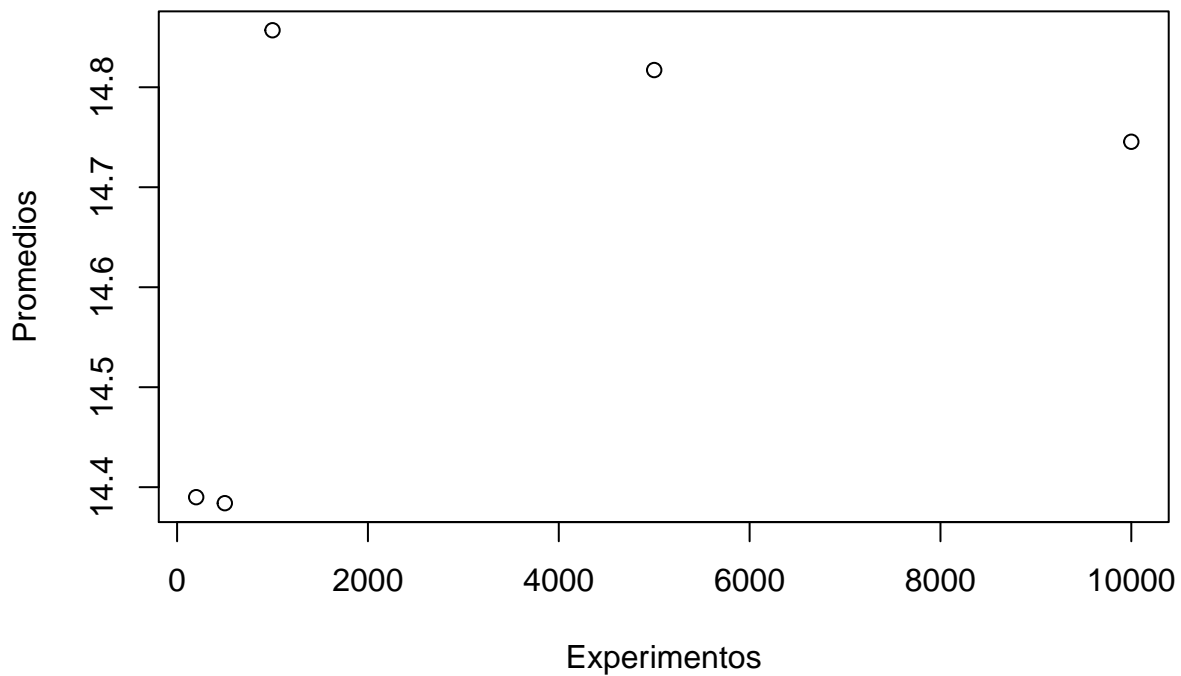
```
c=sort(cantidades)[0.9*N_rep]
```

Sobre variabilidad y convergencia

Ejercicio 5

Queremos repetir el ítem 4.a para $N_{rep}=200, 500, 1000, 5000$ y 10000 . Y graficar N_{rep} vs los promedios calculados para cada valor de N_{rep} .

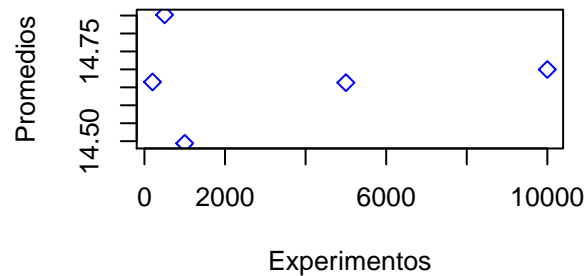
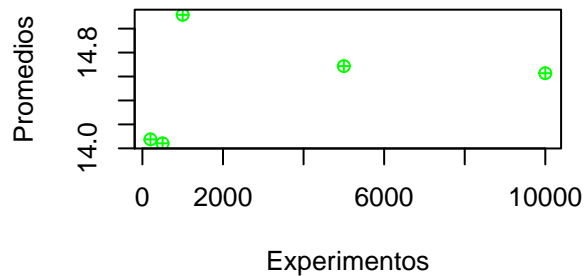
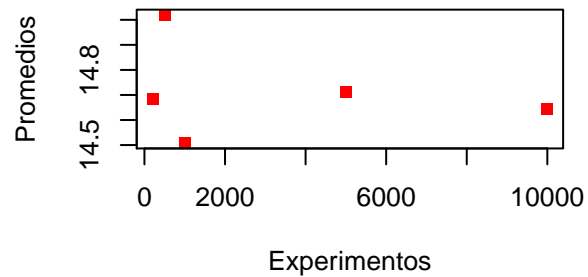
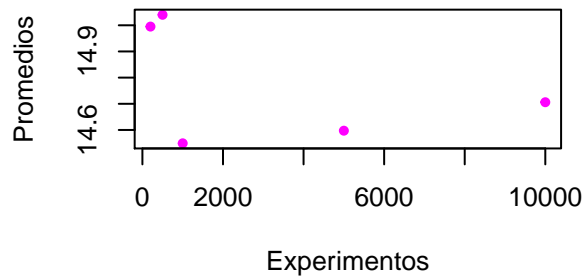
```
N_reps=c(200,500,1000,5000,10000)
medias=function(experimentos_totales){
  resultado=c()
  for (i in experimentos_totales){
    resultado=c(resultado,mean(n_albums(6,i)))
  }
  resultado
}
plot(N_reps,medias(N_reps), xlab='Experimentos',
     ylab='Promedios')
```



Ejercicio 6

Repetimos el Ejercicio anterior varias veces pero poniendole otro color a los puntos.

```
par(mfrow=c(2,2))
plot(N_reps,medias(N_reps), xlab='Experimentos',
     ylab='Promedios',
     pch=20,col='magenta')
plot(N_reps,medias(N_reps), xlab='Experimentos',
     ylab='Promedios',
     pch=15,col='red')
plot(N_reps,medias(N_reps), xlab='Experimentos',
     ylab='Promedios',
     pch=10,col='green')
plot(N_reps,medias(N_reps), xlab='Experimentos',
     ylab='Promedios',
     pch=5,col='blue')
```



```
par(mfrow=c(1,1))
```

Otra forma de mostrar varios

```
# library(gridExtra)
# g1=plot(N_reps,medias(N_reps), xlab='Experimentos',
#         ylab='Promedios',
#         pch=20,col='magenta')
# g2=plot(N_reps,medias(N_reps), xlab='Experimentos',
#         ylab='Promedios',
#         pch=15,col='red')
# g3=plot(N_reps,medias(N_reps), xlab='Experimentos',
#         ylab='Promedios',
#         pch=10,col='green')
# g4=plot(N_reps,medias(N_reps), xlab='Experimentos',
```

```
#      ylab='Promedios',  
#      pch=5,col='blue')  
# grid.arrange(g1,g2,g3,g4,ncol=2,nrow=2)
```

Notamos que en los últimos valores de N_rep, es decir para cantidades grandes de experimentos, el promedio tiende a valores cercanos a 14.7.

Caso intermedio

Ahora `figusTotales=640`.

Ejercicio 7

Calculemos `cuantasFigus(figusTotal=640)` `N_rep=100` veces, es decir, ejecutemos la función `n_albums(figusTotal=640,N_rep=100)` y calculemos la media de figus necesarias para completar el álbum.

```
mean(n_albums(640,100))
```

```
## [1] 4459.72
```

Con paquetes

Ejercicio 8

Para simular ahora una compra de un paquete de 5 figuritas (que pueden estar repetidas) usamos el comando

```
sample((1:640),5,replace = T)
```

```
## [1]  91 145 498 568 276
```

Ejercicio 9

Simulamos la compra de un paquete con `figusPaquete` figuritas de un album con `figusTotal` cantidad de figuritas distintas.

```
genPaquete=function(figusTotal, figusPaquete){  
  sample(1:figusTotal,figusPaquete,replace = T)  
}
```

Ejercicio 10

Definimos una función que toma como input la cantidad de figuritas en cada paquete `figusPaquete` y la cantidad total de figuritas en el álbum `figusTotal` y devuelva la cantidad de paquetes que debemos comprar para completar el álbum.

```
cuantosPaquetes=function(figusTotal, figusPaquete){
  album=rep(0,figusTotal)
  contador=0
  while(sum(album==0)!=0){
    paquete_nuevo=genPaquete(figusTotal, figusPaquete)
    for (i in paquete_nuevo){
      album[i]=album[i]+1
    }
    contador=contador+1
  }
  return(contador)
}
```

Ejercicio 11

Definimos una función que tome como parámetros la cantidad de figuritas en cada paquete `figusPaquete`, la cantidad total de figuritas en el álbum `figusTotal` y la cantidad de experimentos a simular `N_rep` y devuelva un vector formado por la cantidad de paquetes que debemos comprar para completar el álbum en cada simulación del experimento. Es decir, definimos una función análoga a la del Ejercicio 4 pero en el contexto de paquetes en vez de figuritas individuales.

```
n_albums_por_paquete=function(figusTotal,figusPaquete,N_rep){  
  resultado=c()  
  for(i in (1:N_rep)){  
    resultado=c(resultado,cuantosPaquetes(figusTotal, figusPaquete))  
  }  
  #names(resultado)=paste('Experimento n° ',(1:N_rep))  
  resultado  
}
```

Ejecutamos la función para `figusTotal=640`, `figusPaquete=5` y `N_rep=100`.

```
resultados=n_albums_por_paquete(640,5,100)
```