

Entrega 5 Ejercicio 8.1

Augusto Souto

8 de mayo de 2019

Índice

1. Ejercicio 5.1	2
1.1. A	2
1.2. Numeros Aleatorios de Random.org	2
1.3. Numeros Aleatorios de Australia National University (ANU)	2
1.4. B	2
2. sorteo de valores en el hipercubo	3
3. centro	3
<pre>{r setup, include=FALSE} knitr::opts_chunk\$set(results="hide", message = FALSE, warning = FALSE, error=FALSE, fig.pos="H", options(xtable.comment = FALSE))</pre>	

1. Ejercicio 5.1

1.1. A

Elegir al menos dos fuentes de números aleatorios disponibles en Internet (sitio o tabla con valores). Explicar cómo funcionan, como se accede a los números, y qué características tienen.

1.2. Numeros Aleatorios de Random.org

Los numeros aleatorios de esta [pagina](#) provienen del ruido atómosférico, el cual es causado por procesos naturales, como las descargas de los rayos en una tormenta. Estas descargas son principalmente descargas que ocurren desde las nubes hacia la tierra.

Según [wikipedia](#), las sumas de estas descargas componen al ruido atmosférico. El mismo es medido a través de un receptor de radio. Dado que el proceso es de naturaleza estocástica, los numeros generados también lo son.

El acceso a los numeros se da mediante la [pagina](#), en donde se especifica, en nuestro caso, la necesidad de numeros ubicados entre 0 y 10, la cantidad, y numero de decimales.

Las características de estos numeros aleatorios son la independencia y uniformidad. La primer característica nos dice que cada numero sorteado no depende de ningun otro numero de la secuencia. La segunda característica nos dice que no existen puntos del intervalo establecido (en nuestro caso el intervalo 0-1) que tengan mayor chance de salir sorteados.

1.3. Numeros Aleatorios de Australia National University (ANU)

Estos numeros aleatorios también provienen de un proceso físico, en este caso los mismos son generados por la medición de las fluctuaciones cuanticas del vacío. Según [wikipedia](#), dicho proceso describe la variación de la cantidad de energía en un punto del espacio, lo que se da por principios físicos de incertidumbre. Esto hace que el proceso sirva para generar numeros aleatorios.

La obtención de dichos numeros se puede conseguir mediante varios canales, como explicita la [página del equipo de óptica cuántica de la ANU](#). En este caso se prefirió usar la libreria de R “qrandom”, la que nos deja acceder a dichos numeros en varias formas. En particular, mediante el comando qrandomunif se obtienen numeros aleatorios y uniformes entre 0 y 1.

Las propiedades de los numeros son las mismas descritas anteriormente, dado que son uniformemente distribuidos e independientes.

1.4. B

En base a este análisis, elegir una de las fuentes, fundamentar la selección, y modificar el ejercicio 3.1, parte a (visto en la sesión 3) para que emplee dichos números aleatorios (en lugar de los generados por bibliotecas como hasta el momento). Comparar si la salida obtenida es consistente o no con la obtenida en los experimentos de la parte a del ejercicio 3.1.

1.4.1. Resolucion

El ejercicio es hecho con los datos de Random.org para $n = 10^4$ y con los datos de la ANU para $n = 10^5$. Se elije esto dado que el limite de la primer fuente es el maximo numero de numeros aleatorios que nos dan ambas fuentes.

1.4.1.1. Random.org

```
x_randomorg<-read.csv("C:/Users/Usuario/Desktop/10_4.csv", header = F) #importa los valores de Random
y_randomorg<-read.csv("C:/Users/Usuario/Desktop/10_4_2.csv", header = F)

head(x_randomorg, header=F) #vista rapida#
head(y_randomorg)

“{r, results=“markup”}

estimar<-function(x_ran, y_ran, corridas){ t1<-Sys.time() #parametros# r<-0.4 n<-corridas
```

2. sorteo de valores en el hipercubo

```
x<-x_ran %>% as.matrix() y<-y_ran %>% as.matrix()
coor<-cbind(x, y)
```

3. centro

```
c_x<-rep(0.5, n) c_y<-rep(0.5, n) cent<-cbind(c_x, c_y)
valor<-vector(“numeric”, length=n) #valores que toma la función en los puntos sorteados dentro de la region# for
(i in 1:n) {
if(sum((coor[i,]-cent[i,])^2)<=r^2){ valor[i]=(200x[i]/y[i])/max(x[i],y[i]) }else{ valor[i]=0} }
s<-vector(“numeric”, length=n) t<-vector(“numeric”, length=n)
t[1]=0 s[1]=valor[1]
for (i in 2:n) { t[i]<-t[i-1]+(1-(1/i))*(valor[i] -( s[i-1] / (i-1)) )^2 s[i]<-s[i-1]+valor[i]
t2<-Sys.time() }
print(t2-t1)
int<-s[n]/n #INTEGRAL ESTIMADA# var_int<-t[n]/(n-1) #VARIANZA ESTIMADA# var_est<-var_int/n
#VARIANZA MUESTRA
options(“scipen”=-100, “digits”=4)
return(c(int, var_int, var_est)) }
estimar(x_ran=x_randomorg, y_ran=y_randomorg, corridas = 10^4)
““
```

En el caso de los valores de ANU, se simulan los valores de la siguiente manera

```
if (!require('qrandom')) install.packages('qrandom')

x_anu<-qrandom::qrandomunif(10^5, 0,1)
y_anu<-qrandom::qrandomunif(10^5, 0,1)

#mirada rapida a los numeros#
head(x_anu)
head(y_anu)

#estimaciones#
estimar(x_ran=x_anu, y_ran=y_anu, corridas = 10^5)
```