Capítulo 2. Generación de número aleatorios

Referencias:
Kennedy, W and Gentle, J. (1980) Statistical Computing. Marcel Dekker
Press, et.al (1992) Numerical Recipes in C. Cambridge University Press
Ross, S. (1990). A course in Simulation. Macmillan Publishing Company.

2.1. Motivación:

Números aleatorios son usados en muchas aplicaciones tales como:

1- Para simulación de sistemas (colas de espera, loterias, juegos).
2- Boostrappping, Pruebas no paramétricas
3- Integración numérica usando Monte Carlo.
4- En optimización: Algoritmos Genéticos y Simulated Annealing

En la mayoría de estas aplicaciones se usan variables aleatorias uniformes como punto de partida.

In 1927, L.H.C. Tippett publicó una tabla de 41,600 números aleatorios obtenidos de los dígitos del medio de las areas de parcelas en Inglaterra. En 1955, la Corporación RAND usó una computadora para generar electrónicamente una tabla de un millón de numeros aleatorios.
Las computadoras tienen capacidad limitada y son esencialmente deterministicas en consecuencia nunca se consigue una secuencia de números puramente aleatorios solo se puede obtener una secuencia que aparente ser aleatoria.

2.2 Generadores de números aleatorios Uniformes.

Una variable aleatoria discreta X se dice que es uniforme en el intervalo [1,n] si la probabilidad de ocurrencia es la misma para cada valor de la variable. Es decir, $P[X=j]=1/n$ para $j=1,\ldots n$. A nivel muestral habría uniformidad si cada valor se repite con la misma frecuencia. Si X es una variable aleatoria continua entonces se distribuye uniformente en (0,1) si la probabilidad de ocurrencia de que X caiga en un intervalo (a,b), $0<a<b<1$, es igual a la longitud de dicho intervalo. Una variable aleatoria uniforme continua puede ser discretizada asumiendo que n es grande.

Los números aleatorios constituyen una secuencia de números, que aunque han sido obtenidos determinísticamente tienen la apariencia de ser distribuidos aleatoriamente en un intervalo. Un algoritmo de computadoras que sirve para generar una secuencia de números aleatorios es llamado un **generador de números aleatorios**. Lo primero que hay que notar acerca de los generadores de números aleatorios es que los números que ellos generan no son aleatorios. Los generadores usan algoritmos deterministicos y mas propiamente deberían llamarse generadores de números seudo aleatorios, ya que generan

una secuencia de números que eventualmente pueden volver a repetirse y solamente se aproxima a una secuencia aleatoria.

Los generadores de números aleatorios requieren que el usuario le especifique un valor inicial, llamado la “semilla”. La semilla es un número primo bien grande o un número par elegido al azar. Inicializando un generador con la misma semilla producirá la misma secuencia de números aleatorios. Algunos generadores usan el tiempo que da el reloj de la computadora como semilla. Por ejemplo, Anderson (1990) propuso

X_o=iyear+100*(imonth-1 +12*(iday-11+31*(ihour+24*(imin-+60*isec))))

Pero esto no es muy adecuado ya que algunas veces hay que reportar la semilla usada al tiempo de correr el generador, y en este caso eso no seria posible.

Muchos generadores de números aleatorios disponible en algunos programas de aplicación han mostrado que tienen poca aleatoriedad.

Las funciones **rand** que vienen con los programas de computadoras son por lo general malos. Por ejemplo cada vez que se entra a MATLAB y se llama a la función rand se obtiene.

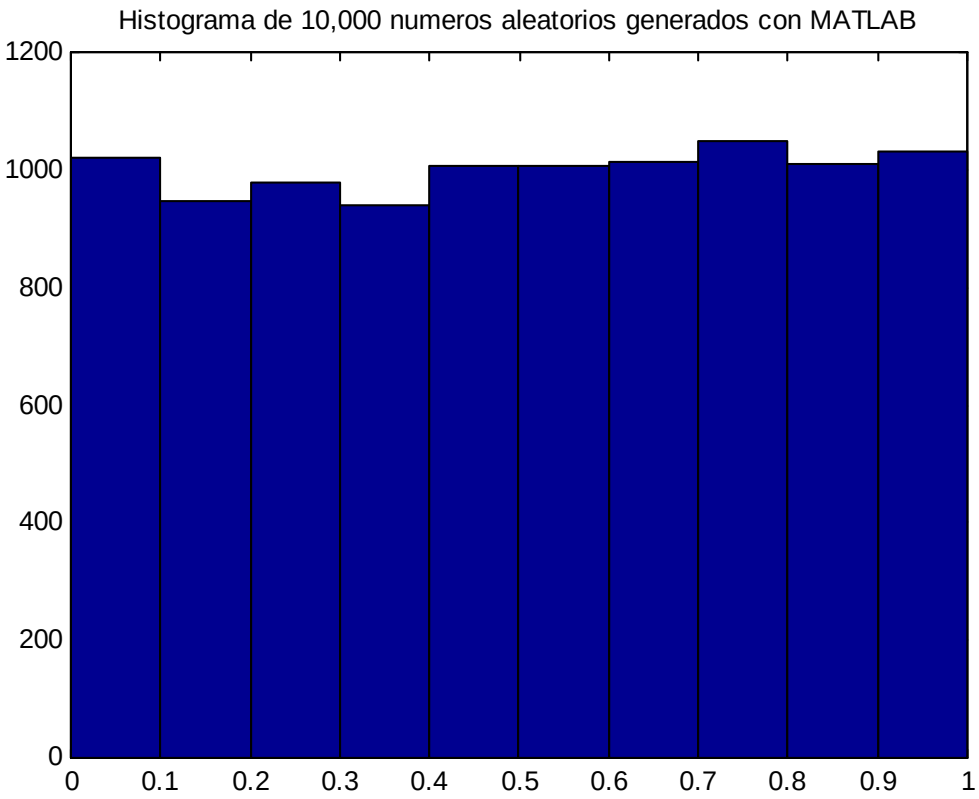
```
>rand
ans =    0.9501
```

Igualement en R,

```
> runif(1)
[1] 0.5036409
```

Sin embargo, si se genera una gran cantidad de datos y se hace su histograma se observa un comportamiento uniforme de los datos. La siguiente secuencia de comandos genera 10,000 números aleatorios y luego hace su histograma

```
>u=rand(10000,1)
>hist(u)
>title(“Histograma de 10,000 numeros aleatorios generados con MATLAB’)
```



Recomendaciones para usar un generador de números aleatorios:

1. Asegurarse de que el programa llama a la rutina de inicialización del generador antes de que ejecute el mismo.
2. Usar semillas que son de alguna manera aleatoria. Por ejemplo, 2731774 y 10293082 son mas seguros que 1 o 4096 (or de alguna otra potencia de 2). También dos semillas similares como 23612 y 23613 pueden producir secuencias correlacionadas.
3. Nunca confié en el generador de números aleatorios que viene con la computadora.
4. Si es posible hacer comparaciones para su aplicación usando dos distintos generadores para verificar si ellos dan los mismos resultados desde el punto de vista estadístico.
5. Cuando publique sus resultados especificar que generador de números aleatorios usó.

2.3 Métodos congruenciales

Los generadores de números aleatorios mas usados son los **generadores congruenciales lineales** (LCG). Los generadores de C standard y de UNIX: RAND (32-bit de precisión), DRAND48 y RANF (48-bit de precisión) son de este tipo.

Los generadores LCG fueron introducidos por Lehmer en 1951. Aquí se produce una secuencia de números aleatorios enteros, donde el siguiente número aleatorio es generado usando el anterior y 3 constantes a, c y M a través de la siguiente relación

$X_n=(aX_{n-1}+c) \pmod M$

Donde a es llamado el multiplicador c la constante aditiva y M el modulo. La expresión y=z (mod m) significa que existe un entero k tal que mk=y-z.. Por ejemplo, 11=3(mod 4) puesto que 4(2)=11-3.
Los parámetros a, c y M deben ser elegidos cuidadosamente para asegurar un largo periodo y que la secuencia generada tenga buena propiedad de aleatoriedad y uniformidad.
Cuando c=0, el generador es llamado **congruencial multiplicativo** y si c≠0 es llamado el **congruencial mixto.**
Para empezar el algoritmo requiere de un valor inicial X_o, llamado la semilla La secuencia generada es denotada por LCG(a,c,M,X_o), donde LCG significa Linear, Congruential Generator.
La apariencia de aleatoriedad es obtenida cuando se hace la aritmética de módulo, por ejemplo la secuencia 0, 1,2,3,4,5 modulo 3 es lo mismo que 0,1,2,0,1,2.
Una vez que se obtiene X_n se puede obtener números reales "aleatorios" dividiendo entre n si se quiere valores distribuidos uniformemente en [0,1), o dividiendo entre n-1 si se quiere valores uniformes en [0,1].

Ejemplo 1. LCG (5, 1, 16, 1)

Consideremos un simple ejemplo con a=5, c=1 M=16 y Xo=1. La sucesión de pseudo números aleatorios generados por este algoritmo es

1,6,15,12,13,2,11,8,9,14,7,4,5,10,3,0,1,6,15,12,13,2,11,8,9,14,……

Notar lo siguiente

i) El periodo P de la secuencia (esto es el número de enteros antes de que la secuencia se repita) es 16 igual al modulo. Cuando el siguiente resultado depende solo del entero anterior el periodo más grande posible es igual al modulo
ii) Notar que los números impares y pares se intercalan. En la siguiente tabla se muestra la representación binaria y los números reales generados por la secuencia aleatoria.

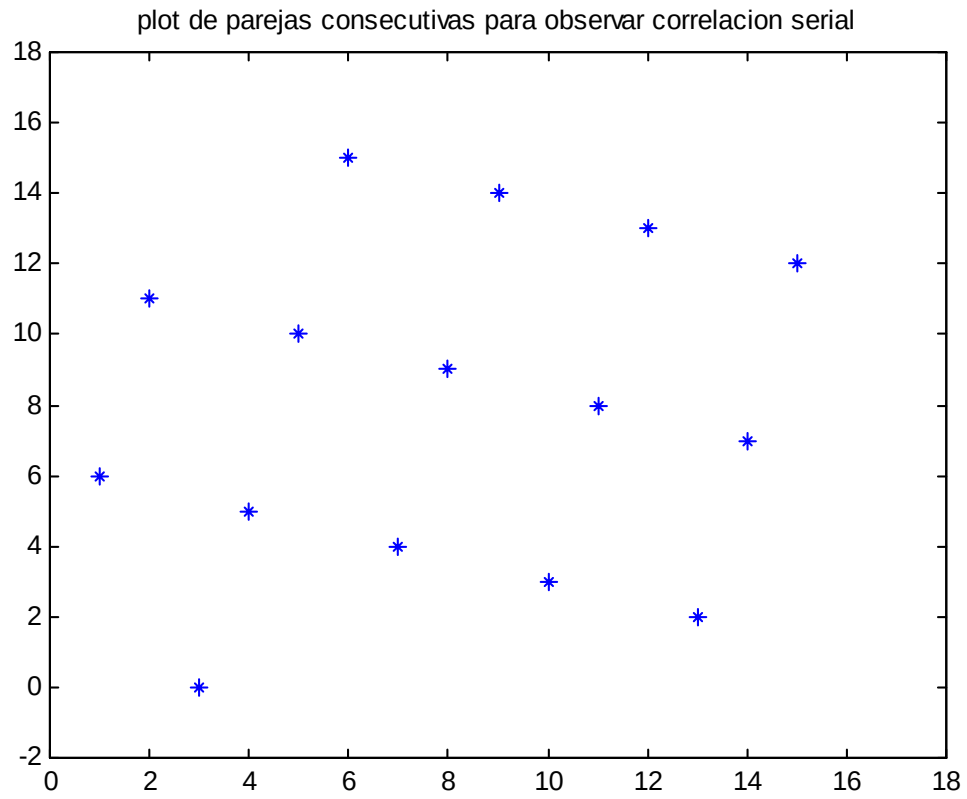
N	X_n	Binario X_n	Real X_n
0	1	0001	0.0625
1	6	0110	0.3750
2	15	1111	0.9375
3	12	1100	0.7500
4	13	1101	0.8125
5	2	0010	0.1250

6	11	1011	0.6875
7	8	1000	0.5000
8	9	1001	0.5625
9	14	1110	0.8750
10	7	0111	0.4375
11	4	0100	0.2500
12	5	0101	0.3125
13	10	1010	0.6250
14	3	0011	0.1875
15	0	0000	0.0000
Mean			0.4688
Varianza			0.0830

Notar que los digitos que están mas a la derecha en el sistema binario forman la secuencia 1,0,1,0,1,0,1,0,1,0,1,0…..lo que implica que hay correlación serial y por lo tanto la aleatoriedad se pierde. El coeficiente de correlación serial de lag j, está dada por

$$r_j = \frac{\sum_{i=j+1}^{n} (x_i - \bar{x})(x_{i-j} - \bar{x})}{\sum_{i=j+1}^{n} (x_i - \bar{x})^2}$$

Existen varias pruebas estadísticas para detectar correlación serial, siendo la más conocida la de Durbin-Watson. En el siguiente plot de las parejas consecutivas se puede observar el patrón periódico que existe, lo cual indica presencia de correlación serial



La presencia de correlación serial alta se debe a que se ha usado como modulo una potencia de 2. A pesar de todo, este tipo de modulo es usado bastante porque la aritmética de módulo se hace más eficientemente y porque los números reales que se obtiene muestran cierta aleatoriedad.

iii) Si se selecciona una semilla entre 0 y 15 el efecto que ocurre es trasladar ciclicamente la anterior secuencia.
iv) Notar que el promedio de los números reales es 0.4688, y la varianza =0.0830, mientras que los valores teóricos para una uniforme son ½=.5 y 1/12=.08333 respectivamente. No hay mucha diferencia entre ellos, pero esto solo son condiciones necesaria pero no suficientes para que el método sea un buen generador de números aleatorios.

Se puede mostrar que cuando c=0 y el modulo es 2^M una potencia de dos entonces el periodo es 2^{M-2}. Para cualquier c, si M es un número primo entonces el periodo es p=M-1. (Ver detalles en Hull and Dohell, 1962,1964).

MATLAB usa LCG con a= 7^5 =16807 ,c=0 y M=2^{31}-1=2147483647. Estas constantes fueron propuestas por Lewis, Godman y Miller en 1969.

LCG's trabaja bien para la mayoría de las aplicaciones pero puede producir secuencias de números que tiene correlación serial alta. Otro problema es que los más usados LCGs, incluyendo DRAND48 y RANF, usan un modulo M que es una potencia de 2 ya que es

más rápido de implementarlo en una computadora. Sin embargo hacer esto puede causar correlaciones altas para números generados en intervalos, cuyos tamaños son potencias de 2. Para evitar estos problemas se recomienda usar como modulo un número primo.
Existen varios métodos alternativos para remover la correlación serial, tales como
El metodo de "Shuffling" (Bays y Durham, 1976), combinar dos generadores (Marsaglia, 1965), El método "Feedback shift register" (Tausworthe, 1965), y el método substractivo (Knuth, 1981).

2.4 Cálculo de Integrales usando Monte Carlo

Consideremos el cáculo de la integral.

$$I = \int_o^1 g(x)dx$$

Supongamos que U es una variable aleatoria distribuida uniformemte en el intervalo (0,1). Es decir, su función de densidad está dada por $f(u)=I_{(0,1)}$, donde $I_{(0,1)}$ representa la función indicadora en el intervalo (0.1). Entonces, el valor esperado de g(U) está dado por

$$E[G(U)]= \int_o^1 g(u)du$$

Sea U1, U2,……Un, una muestra aleatoria de U (es decir, cada una de ellas es uniforme y además son independientes). Entonces por la ley fuerte de los grandes números, se tiene que con probabilidad 1.

$$\frac{\sum_{i=1}^{n} g(U_i)}{n} \to E[(g(U)] = I \text{ , cuando n}\to\infty.$$

Esta forma de calcular integrales es llamada Integración Monte Carlo.

Ejemplo: Calcular por Integración Monte Carlo

$$\int_o^1 x^3(1-x)^2\,dx$$

Solución:

Haciendo un programa en MATLAB para generar 10,000 números aleatorios y aplicando luego la función mean se obtiene un valor aproximado de 0.0165, el valor exacto es 1/60=0.0166.

En R,

```
> x=runif(10000)
```

```
> y=x^3*(1-x)^2
> int1=mean(y)
> int1
[1] 0.01671438
```

Si se desea calcular $I = \int_a^b g(x)dx$ entonces hay que hacer la sustitución y=(x-a)/(b-a) para cambiar el dominio de integración al intervalo (0,1), y la integral I se transforma en $I = \int_o^1 h(y)dy$, con h(y)=(b-a)g(a+(b-a)y).

Similarmente si se desea calcular $I = \int_o^\infty g(x)dx$, se hace la sustitución y=1/(x+1) y la integral se transforma en $I = \int_o^1 h(y)dy$, con $h(y) = \frac{g(\frac{1}{y} - 1)}{y^2}$

Ejemplo: Calcular por integracion Monte Carlo

$$I = \int_o^\infty e^{-x^2} dx$$

Solución: El valor exacto de la integral es $\frac{\sqrt{\pi}}{2}$= 0.886227. Haciendo la sustitución y=1/(x+1), la integral dada se convierte en $I = \int_0^1 \frac{e^{(\frac{1-y}{y})^2}}{y^2} dy$. Siguiendo el mismo procedimiento del ejemplo anterior resulta

```
> y=(1/x^2)*exp(-((x-1)/x)^2)
> mean(y)
[1] 0.8845605
>
```