

Programación en R para Estadística

Simulación

J. Elías Rodríguez M.

Facultad de Matemáticas

Universidad de Guanajuato

XXIII Foro Nacional de Estadística

11 de septiembre del año 2008

Boca del Río, Veracruz, México

Temario

1. Breve introducción a la generación de valores de

variables aleatorias

2. Integración usando métodos Monte Carlo

Bibliografía

Rizzo, M. L. (2008). Statistical computing with R. Chapman & Hall/CRC.

1 Breve introducción a la generación de valores de variables aleatorias

Primero lo primero, ¿por qué no generación de variables aleatorias?

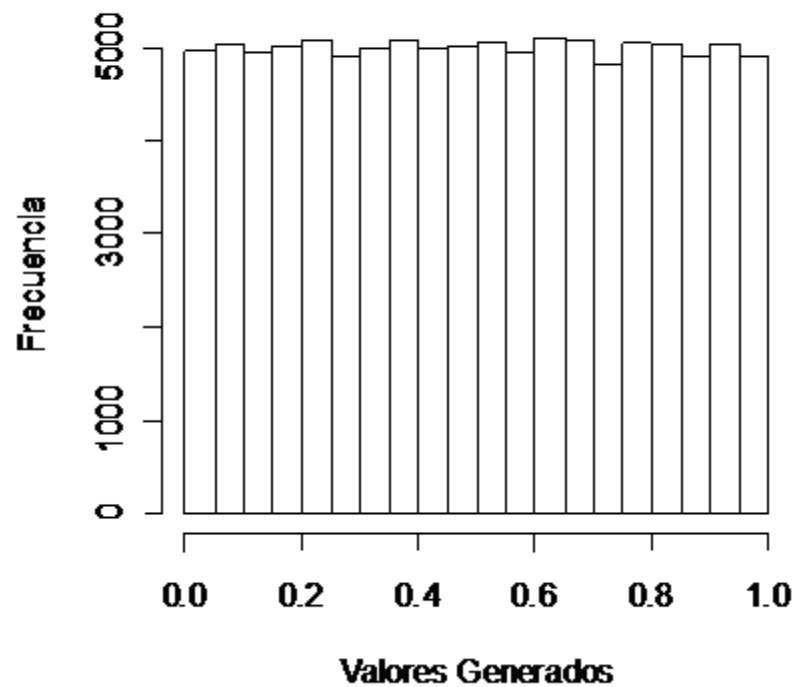
Bueno y ¿qué “onda” con números aleatorios o pseudo aleatorios?

Bueno dejemos que el expositor se haga “bolas” con la explicación. Nosotros a la generación de valores de variables aleatorias.

Para empezar imaginemos que necesitamos generar 100 000 valores de una variable aleatoria con función de distribución *Uniforme*(0,1). Esto lo podemos hacer con el siguiente código (simulacion1.r).

```
set.seed(112358);  
x=runif(100000);  
hist(x, breaks=17,  
      xlab="Valores Generados",  
      main=" ", ylab="Frecuencia");
```

Gráficamente podemos apreciar el resultado de los valores generados:



Ok, y de qué otras distribuciones podemos generar

valores:

- `runif(n, min=0, max=1)`; uniforme
- `rnorm(n, mean = 0, sd = 1)`; normal
- `rlnorm(n, meanlog = 0, sdlog = 1)`;
lognormal
- `rbeta(n, shape1, shape2, ncp = 0)`, beta
- `rgamma(n, shape, rate = 1, scale =
1/rate)`, gamma
- `rchisq(n, df, ncp=0)`, chi-cuadrada
- `rexp(n, rate = 1)`, exponencial
- `sample(x, size, replace =TRUE, prob =q)`,

distribución discreta finita

- `rbinom(n, size, prob)`, binomial
- `rnbinom(n, size, prob, mu)`, binomial negativa
- `rpois(n, lambda)`, poisson
- etc, etc

Y si R no lo tiene, todavía nos quedan los métodos conocidos en la literatura de generación de valores de variables aleatorias:

- Transformación inversa
- Aceptación y Rechazo

○ Cualquier otro tipo de transformación

§ Sumas y Mezclas

○ Etc, etc

¡Más acción por favor!

A manera de ejemplo simulemos el lanzamiento de una moneda pero de forma diferente a como no lo enseñaron en la primaria (quiero ver como les explica el expositor esta cosa de águila y sol, ignorancia y caos)

El siguiente código ([simulacion2.r](#)) simula los resultados

(águilas y soles como 1's y 0's) de lanzar m veces una moneda:

```
m=10000;  
exito=numeric(m);  
for(i in 1:m)  
{  
  alpha=rexp(1, rate=1);  
  q=rbeta(1, alpha, alpha);  
  exito[i]=rbinom(1, 1, q);  
}  
table(exito)/m;  
pie(table(exito),
```

```
col=c("grey","blue"),  
main="1=águila 0=sol",  
xlab="Fracción de 0's y 1's");
```

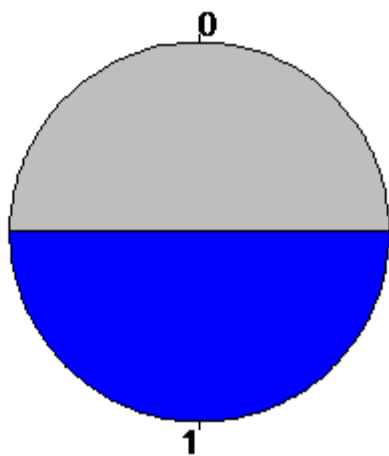
Para una repetición de este experimento obtenemos lo siguiente:

```
exito
```

```
0      1
```

```
0.4989 0.5011
```

1=águila 0=sol



Fracción de 0's y 1's

2 Integración usando métodos Monte Carlo

Imaginemos que por alguna razón queremos calcular la siguiente integral

$$\theta = \int_A \phi(x) dx.$$

Por otro lado tenemos a la variable aleatoria X con función de densidad g y soporte A . Entonces la anterior integral la podemos escribir como

$$\begin{aligned}\theta &= \int_A \phi(x) dx \\ &= \int_A \frac{\phi(x)}{g(x)} g(x) dx \\ &= E\left(\frac{\phi(X)}{g(X)}\right)\end{aligned}$$

Lo anterior nos lleva a proponer el siguiente método Monte Carlo para aproximar el valor de la integral de interés.

1. Generar valores de las variables aleatorias

X_1, \dots, X_m independientes e idénticamente distribuidas con función de densidad g ;

2. Estimar θ con

$$\hat{\theta} = \frac{1}{m} \sum_{j=1}^m \frac{\varphi(X_j)}{g(X_j)}$$

(¿Sí? A ver un ejemplito, ¿no? ¿Quién es g ?)

A manera de ejemplo supongamos que queremos estimar la función de distribución normal estándar en 2. Antes observemos que

$$\Phi(2) = 0.5 + \int_0^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

El siguiente código ([simulacion3.r](#)) nos puede ayudar a estimar la anterior cantidad:

```
m=10000;  
x=runif(m, min=0, max=2);  
theta=0.5+mean(2*exp(-x*x/2))/sqrt(2*pi);  
c(theta,pnorm(2));  
abs(theta-pnorm(2))/pnorm(2);
```

Una realización de este experimento nos produce:

```
c(theta, pnorm(2));
```

```
[1] 0.9780748 0.9772499
```

```
abs(theta-pnorm(2))/pnorm(2);
```

```
[1] 0.0008441782
```

(Bueno ¿y?, ¿Qué sigue?, varianza del estimador, un intervalo de de confianza par el estimador, reducción de la varianza, etc, etc)


```

### Simulacion 1

set.seed(112358); #Establece semilla

x=runif(100000); #Genera valores de una uniforme

hist(x, breaks=17, xlab="Valores Generados", main="",
ylab="Frecuencia");

### Simulacion 2

m=10000; #Número de lanzamientos (simulaciones)
exito=numeric(m);

for(i in 1:m)
{ alpha=rexp(1, rate=1); #genera el parámetro de la Beta
  q=rbeta(1,alpha,alpha); #genera la probabilidad de éxito
  exito[i]=rbinom(1,1,q); #lanza la moneda
}

table(exito)/m;
pie(table(exito), col=c("grey","blue"), main="1=águila 0=sol",
xlab="Fracción de 0's y 1's"); #Gráfica de pie

### Simulacion 3

m=100000;

x=runif(m, min=0, max=2);

theta=0.5+mean(2*exp(-x*x/2))/sqrt(2*pi);

c(theta,pnorm(2));
abs(theta-pnorm(2))/pnorm(2);

```