

Página [www](#)

Página de Abertura

Contenido



Página 1 de 35

Regresar

Full Screen

Cerrar

Abandonar

Estadística Bayesiana: Clase 12

Juan Carlos Correa

13 de abril de 2021

Elementos de simulación

- Desde la aparición de los computadores una de sus aplicaciones fundamentales ha sido la simulación. Pero mucho antes de esto ya se habían realizado simulaciones manuales.
- Es famoso el caso de la aguja de Buffon, un matemático francés, quien en 1768 determinó el valor de π experimentalmente lanzando una aguja sobre una rejilla dibujada en papel.

[Página www](#)[Página de Abertura](#)[Contenido](#)[Página 3 de 35](#)[Regresar](#)[Full Screen](#)[Cerrar](#)[Abandonar](#)

Recordamos los pesados libros de números aleatorios cuya utilización era misteriosa para los profanos que en las primeras décadas del siglo XX ayudaron a realizar a los pioneros en trabajo de simulación, sin olvidar el formidable trabajo de W. S. Gosset, más conocido por Student (el de la famosa distribución t), quien utilizó el conjunto de números, supuestamente normales, obtenidos de la medición del dedo medio izquierdo de la mano y la estatura de 3000 criminales británicos. Cada pareja de números fueron escritas en una tarjeta. Las 3000 tarjetas fueron mezcladas para garantizar aleatoriedad y se extrajeron 4 tarjetas (una muestra de tamaño 4), para las cuales se calcularon la media y la desviación típica. Este procedimiento se repitió 750 veces.

Página www

Página de Abertura

Contenido



Página 4 de 35

Regresar

Full Screen

Cerrar

Abandonar

En los años 30 Enrico Fermi utilizó el método Monte Carlo en el cálculo de la difusión de neutrones, y posteriormente diseñó el Fermiac, un artefacto mecánico para el método Monte Carlo utilizado en el trabajo crítico de reactores nucleares. En los años 40 von Neumann desarrolló los fundamentos formales del método Monte Carlo, quien estableció sus bases matemáticas. El trabajo fue realizado con la colaboración de Stanislaw Ulam, quien notó la importancia de los computadores digitales en la implementación del método. Este trabajo se hizo dentro del proyecto Manhattan, donde el computador ENIAC jugó un papel importante.

Visualización del Teorema Central del Límite

Una de las aplicaciones más sencillas de la simulación consiste en mostrar el Teorema Central del Límite (parte central en el trabajo estadístico). Este teorema nos dice que la distribución de la media muestral se aproxima una normal si el tamaño muestral es lo suficientemente grande.

```
> numeros.aleatorios<-matrix(runif(12000),ncol=12)
> medias.muestrales<-apply(numeros.aleatorios,1,mean)
> hist(medias.muestrales,
      main='Distribución de la Media',ylab='',xlab='')
>
```

Qué es la simulación?

La simulación es esencialmente una técnica de muestreo estadístico controlado (experimento) que es utilizado, en conjunto con un modelo, para obtener respuestas aproximadas a preguntas acerca de complejos problemas probabilísticos multifactores. Su utilidad es mejor cuando no se pueden obtener respuestas analíticas o numéricas.

Los métodos de simulación se toman como una forma de método científico (Naylor, 1971), y como el método científico, es un instrumento para obtener conocimiento con un nivel de incertidumbre aceptable. Los experimentos de simulación constan de un procedimiento que consta de los siguientes pasos:

1. La formulación del problema.
2. El diseño de un problema matemático.
3. La preparación del programa de computador.
4. La validación.
5. El diseño experimental.
6. El análisis de datos.

Reglas de oro para la aplicación de la simulación

1. Nunca simule a menos que usted no tenga otra salida.
2. Una modelación cuidadosa es fundamental.
3. Validación del modelo y chequeo del programa de computador son tareas básicas.
4. Las soluciones halladas mediante la simulación son estimaciones, no son respuestas exactas.
5. Los supuestos y el modelo deben ser realistas.
6. Esté seguro antes de simular, que usted conoce las preguntas que usted quiere responder.

Generadores de Números Aleatorios

La generación de números que representen bien un comportamiento aleatorio es de extrema importancia en simulación estocástica. Ingenuamente podemos intentar un mecanismo físico que nos produzca números aleatorios de una forma natural, como por ejemplo las diez bolas en una bolsa que son muestreadas con remplazo. Dodge (1996) describe la utilización de la parte decimal de π como un generador natural de números aleatorios.

Métodos para generar números pseudoaleatorios

Método del medio del cuadrado

En 1951 John von Neumann propone el método del *medio del cuadrado* que fue utilizado desde 1949 y a mostrado ser muy pobre para generación de números aleatorios. Un número de n dígitos es elevado al cuadrado para producir un número de $2n$ dígitos del cual se toman los n dígitos del medio para generar el próximo número aleatorio y el proceso se repite. La sucesión cae fácilmente en ciclos cortos repetitivos, especialmente si un cero aparece como un elemento de la sucesión.

Orden	Número	Cuadrado	Próximo valor	Número pseudoaleatorio
1	81	6561	56	0.56
2	56	3136	13	0.13
3	13	0169	16	0.16
4	16	0256	25	0.25
5	25	0625	62	0.62
⋮	⋮	⋮	⋮	⋮

Página www

Página de Abertura

Contenido



Página 11 de 35

Regresar

Full Screen

Cerrar

Abandonar

Utilizando el número π

Jaditz (2000) reporta las pruebas realizadas para verificar si se puede asumir si los decimales de π conforman una sucesión independiente e idénticamente distribídas, sometiéndolo a varias pruebas: la de razón de verosimilitud, la prueba de colisión, la prueba BDSL y pruebas de momentos. Las hipótesis nulas se mantienen.

Método congruencial

Casi todos los generadores de variables uniformes que se usan hoy están basados en una relación de recurrencia congruente lineal de la forma

$$X_{n+1} = (a_0X_n + a_1X_{n-1} + \cdots + a_kX_{n-k} + c) \bmod m$$

a_0, a_1, \cdots, a_k, c y m son enteros conocidos que caracterizan el generador. El usuario debe proveer un vector semilla (X_0, X_1, \cdots, X_k) para comenzar la recurrencia. Entonces X_{k+1} puede ser generado, y la semilla para X_{k+2} se vuelve $(X_1, X_2, \cdots, X_{k+1})$, etc.

La mayoría a menudo usan $k = 1$ y la recurrencia se convierte en

$$X_{n+1} = (aX_n + c) \bmod m,$$

propuesto por Lehmer en 1951. Los parámetros son

- a : una constante multiplicadora
- c : la constante aditiva,
- X_0 : una semilla

Se hace

$$u_i = \frac{X_i}{m}$$

- Cuando $c = 0$, el generador es llamado *congruencial multiplicativo* y cuando $c \neq 0$, el generador es llamado *congruencial mixto*.
- Boswell et al. (1993) comentan sobre lo crítico de la selección de las constantes.
- Un generador muy popular ha sido RANDU basado en $X_{n+1} = (2^{16} + 3)(2^{31} - 1)$ y era proporcionado en los computadores IBM 360/370 y PDP11, el cual mostró ser un generador muy pobre.
- Uno de los más conocidos generadores de tipo multiplicativo está basado en la relación $X_{n+1} = (aX_n)(2^{31} - 1)$, donde $a = 14^{29}$ y ha mostrado tener buenas propiedades.

Ejemplo...

Asuma el generador congruencial $x_n = 5x_{n-1} + 3 \pmod{16}$, con $x_0 = 7$

n	x_n	u_n
0	7	-
1	6	0.375
2	1	0.0625
3	8	0.5
4	11	0.6875
5	10	0.625
6	5	0.3125
7	12	0.75
8	15	0.9375
9	14	0.875
10	9	0.563
11	0	0
12	3	0.1875
13	2	0.125
14	13	0.8125
15	4	0.25
16	7	0.4375
17	6	0.375
18	1	0.0625

Página [www](#)

Página de Abertura

Contenido



Página 16 de 35

Regresar

Full Screen

Cerrar

Abandonar

Método aditivo congruencial

Teorema

El generador congruencial mixto

$$x_{n+1} = ax_n + b(m)$$

con $a = 4c + 1$, b impar y $m = 2^k$, donde c , b y k son enteros positivos, tiene ciclo m .

El generador múltiple recursivo, MRG

El MRG es una extensión natural del generador lineal congruencial. El siguiente número aleatorio en el MRG se calcula como una combinación lineal de los k números aleatorios anteriores.

$$X_i = (\alpha_1 X_{i-1} + \cdots + \alpha_k X_{i-k}) \bmod(p) \text{ para } i \geq k$$

Se obtiene el máximo período de un polinomio primitivo de grado k

$$f(x) = x^k - \alpha_1 x^{k-1} - \cdots - \alpha_k$$

con período $p^k - 1$.

El FMRG, dicen los autores, tomaría, con $k = 4$ y $p = 2^{31} - 1$ un billón de años en completar su ciclo, que es aproximadamente $2,1 \times 10^{37}$, en el computador más rápido de hoy.

Propiedades de un buen generador

Aleatoriedad

Los números aleatorios se distribuyen uniformemente en el intervalo $(0, 1)$. Los números pseudoaleatorios deberían aproximarse a este ideal.

Independencia

Los verdaderos números aleatorios son independientes.

Reproducibilidad

Los números deben ser reproducibles. Los números pseudoaleatorios deben poder reproducirse sin tener que guardarlos, comenzando de la misma semilla, el primer número de la sucesión. Esta propiedad puede utilizarse para mejorar el diseño de ciertos experimentos de simulación. También en procesos de depuración de los programas de simulación, en especial cuando se presentan situaciones aberrantes.

Rapidez

El generador debe ser rápido, ya que con frecuencia se requieren cantidades enormes de números pseudoaleatorios,

Económico

El generador debe utilizar pocos recursos del equipo computacional (memoria) y debe ser independiente del tamaño de la simulación.

Pruebas para verificar si un generador es aceptable

Computacionales

Siempre es preferible un generador sencillo a uno complejo. Es preferible uno que ejecute rápidamente que uno lento.

Estadísticas

Pruebas de bondad de ajuste

Pruebas de bondad de ajuste es un tópico con un amplio desarrollo en estadística (Stephens, 1986).

Pruebas de aleatoriedad

Pruebas de independencia

Nota: El hecho que un generador pase todas las pruebas no implica que sea necesariamente un buen generador.

Generador de números aleatorios en R

R proporciona varios generadores de números aleatorios y, además, permite al usuario definir sus propios generadores si lo desea. Los generadores que posee son:

1. **Marsaglia-Multicarry:** Este es el generador por defecto. Tiene un período mayor a 2^{60} y ha pasado todas las pruebas existentes. La semilla consta de dos valores.
2. **Wichmann-Hill:** Este generador tiene un ciclo de $6,9536 \times 10^{12}$. La semilla es un vector que consta de 3 números primos.
3. **Super-Duper:** Este es un generador propuesto por Marsaglia en los 70's, pero no pasó la prueba MTUPLE de la batería Diehard. La semilla consta de dos enteros, de los cuales el segundo debe ser impar. La longitud del ciclo es aproximadamente $4,6 \times 10^{18}$
4. **Mersenne-Twister**
5. **Knuth-TAOCP:** Este es un generador del tipo rezagado Fibonacci. La recurrencia es

$$X_j = (X_{j-100} - X_{37}) (2^{30})$$

La semilla debe ser un conjunto de 100 números. El período es de alrededor 2^{129} .

Generación de variables no uniformes

Método de la transformación inversa

Sea X una variable aleatoria con función de distribución acumulada (F.D.) $F_X(x)$. Ya que $F_X(x)$ es una función no decreciente, la función inversa $F_X^{-1}(y)$ puede definirse para cualquier valor de y entre 0 y 1 como:

$$F_X^{-1}(y) = \inf\{x : F_X(x) \geq y\}, 0 \leq y \leq 1.$$

Resultado: Si U se distribuye uniformemente sobre el intervalo $(0, 1)$, entonces $X = F_X^{-1}(U)$ tiene función de distribución acumulada $F_X(x)$.

Generación de una exponencial

La distribución exponencial juega un papel fundamental en muchas aplicaciones estadísticas. Una variable aleatoria exponencial X tiene f.d.p.

$$f_X(x) = \frac{1}{\beta} e^{-x/\beta} I_{0 \leq x < \infty}(x)$$

para $\beta > 0$.

Por el método de la transformación inversa

$$U = F_X(X) = 1 - e^{-x/\beta}$$

de donde

$$X = -\beta \ln(1 - U).$$

Ya que $1 - U$ se distribuye de la misma forma que U , tenemos

$$X = -\beta \ln(U).$$

El algoritmo es

- Genere U de una $U(0, 1)$.
- $X \leftarrow -\beta \ln(U)$
- Presente X .

Generación usando la discretización del soporte de la distribución

- Si tenemos una variable aleatoria X con f.d.p. $f(x)$, el soporte de X es definido como el siguiente conjunto:

$$S = \{x : f(x) > 0\}$$

- Para generar valores al azar de esta distribución creamos una rejilla sobre S con valores $x_0, x_1, x_2, \dots, x_K$.
- Calculamos los valores $f(x_0), f(x_1), f(x_2), \dots, f(x_K)$.
- Luego extraemos muestras con reemplazo de $\{x_0, x_1, x_2, \dots, x_K\}$ con probabilidades proporcionales a $\{f(x_0), f(x_1), f(x_2), \dots, f(x_K)\}$.

- Para que este procedimiento sea bueno se requiere que la rejilla sea lo suficientemente fina.
- Un procedimiento consiste en refinar la rejilla en varias etapas, por ejemplo, para generar un valor se obtiene de la primera rejilla un valor, digamos x_j , en lugar de considerar este como el valor definitivo, se crea una segunda rejilla en el intervalo

$$\left(x_j - \frac{(x_j + x_{j-1})}{2}, x_j + \frac{(x_j + x_{j+1})}{2} \right)$$

- Se evalúa la f.d.p. en cada uno de estos valores y se extrae un elemento de esta nueva rejilla con probabilidades proporcionales a los valores de densidad evaluados en ella.

Página *www*

Página de Abertura

Contenido



Página 26 de 35

Regresar

Full Screen

Cerrar

Abandonar

Ejemplo

Suponga que el tiempo que gasta ud. en llegar de su casa a la universidad caminando es una distribución exponencial con media μ minutos. La distribución apriori del tiempo promedio es una normal con media apriori 35 minutos y nivel de seguridad de esta media apriori dado por la varianza como 4.

Suponga que ud. registra 5 tiempos que gastó caminando de su casa a la universidad y fueron (en minutos): 28.4, 25.3, 31.0, 27.4, 28.9.

■ Apriori

$$\xi(\mu) \propto \exp\left(-\frac{1}{2} \frac{(\mu - 35)^2}{4}\right)$$

■ Verosimilitud

$$L(\mu | \text{Datos}) \propto \frac{1}{\mu^5} \exp\left(-\frac{28,4 + 25,3 + 31,0 + 27,4 + 28,9}{\mu}\right)$$

■ Aposteriori

$$\xi(\mu | \text{Datos}) \propto \exp\left(-\frac{1}{2} \frac{(\mu - 35)^2}{4}\right) \frac{1}{\mu^5} \exp\left(-\frac{141,0}{\mu}\right)$$

Página www

Página de Abertura

Contenido



Página 28 de 35

Regresar

Full Screen

Cerrar

Abandonar

```
# Simulación de la posterior vía discretización
```

```
aposte<-function(m)exp(-0.5*(m-35)^2/4)*(1/m^5)*exp(-141.0/m)
```

```
mus<-seq(10,50,length=1000)
```

```
alturas<-sapply(mus,aposte)
```

```
Nsim<-10000
```

```
muestra.sim<-sample(mus,Nsim,replace=T,prob=alturas)
```

```
plot(density(muestra.sim),ylab='Densidad',xlab='Tiempo',main='')
```

```
title(main='Distribución aposteriori simulada\n Tiempo Medio para 1
```

[Página www](#)

[Página de Abertura](#)

[Contenido](#)



[Página 29 de 35](#)

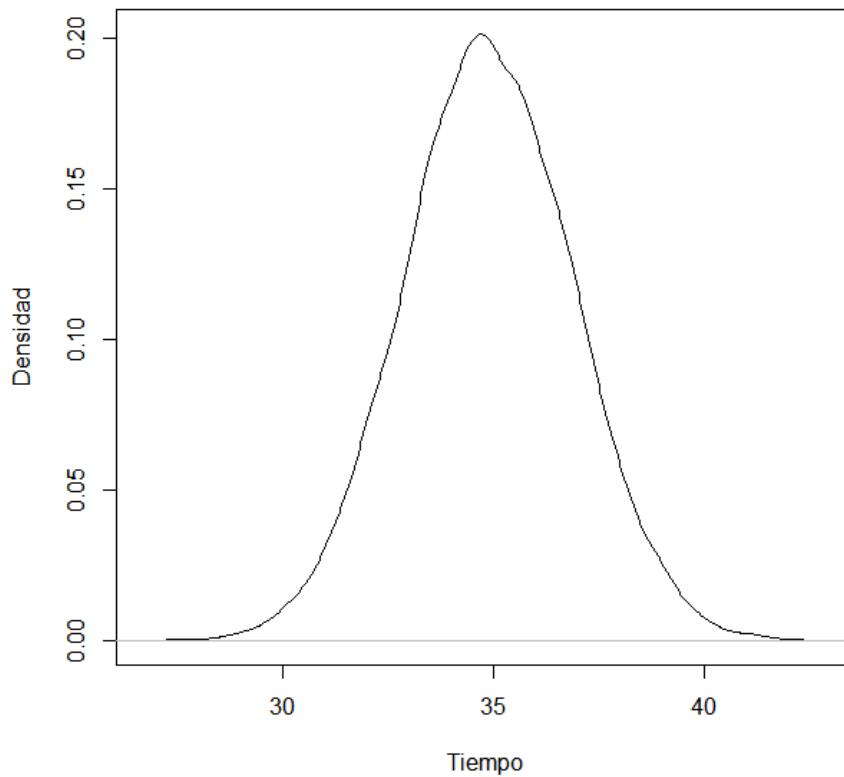
[Regresar](#)

[Full Screen](#)

[Cerrar](#)

[Abandonar](#)

Distribución aposteriori simulada Tiempo Medio para llegar a la Universidad



Medidas de resumen

```
> summary(muestra.sim)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
27.58	33.54	34.86	34.88	36.23	42.07

```
> sd(muestra.sim)
```

```
[1] 1.981128
```

```
> var(muestra.sim)
```

```
[1] 3.924867
```

```
> quantile(muestra.sim,probs=c(0.05,1:9/10,0.95))
```

```
> quantile(muestra.sim,probs=c(0.05,1:9/10,0.95))
```

5%	10%	20%	30%	40%	50%	60%
31.62162	32.33834	33.22322	33.82382	34.38438	34.86486	35.38539
35.90000						
80%	90%	95%				
36.58659	37.42743	38.14815				

Página www

Página de Abertura

Contenido



Página 31 de 35

Regresar

Full Screen

Cerrar

Abandonar

Estimación de la moda

```
> library(modeest)
> grenander(muestra.sim,p=2)
[1] 34.8788
> venter(muestra.sim)
[1] 34.72472
```

Una aproximación...

Podemos pensar en aproximar la aposteriori mediante una...

$$\xi(\mu | \text{Datos}) \approx \text{Normal}(\text{Moda}; \text{varianza})$$

$$\xi(\mu | \text{Datos}) \approx \text{Normal}(34,8788; 3,924867)$$

```
> alturas2<-dnorm(mus,mean=34.22422,sd=sqrt(4.070843))  
> points(mus,alturas2,type='l',col='red')
```


[Página www](#)

[Página de Abertura](#)

[Contenido](#)



[Página 33 de 35](#)

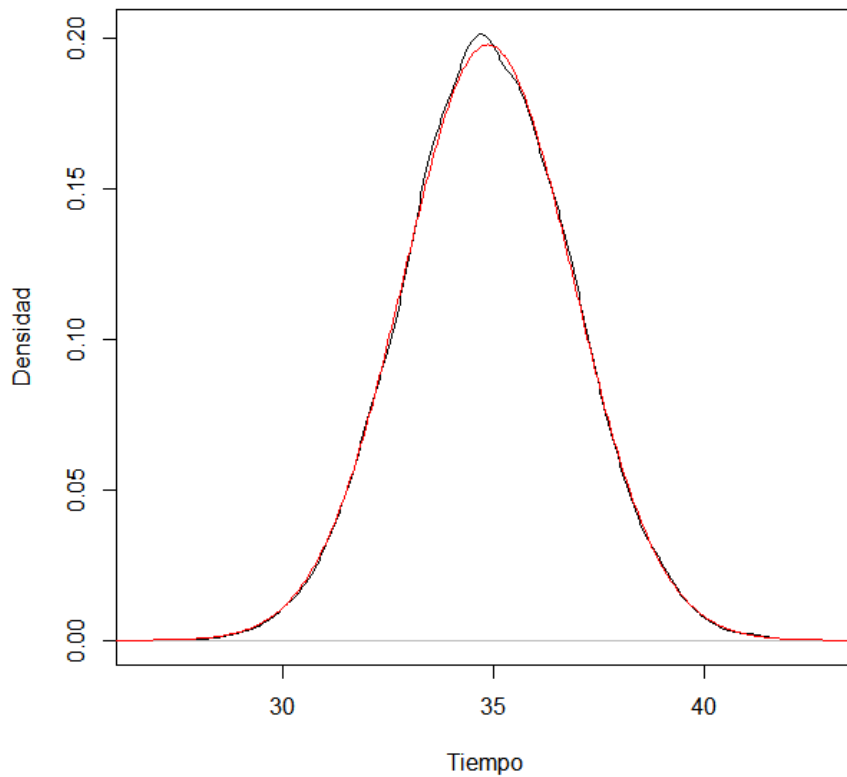
[Regresar](#)

[Full Screen](#)

[Cerrar](#)

[Abandonar](#)

Distribución aposteriori simulada Tiempo Medio para llegar a la Universidad



Método de aceptación y rechazo

Rubinstein (198*) le atribuye este método a von Neumann.

Sea X una variable aleatoria que se desea generar de una distribución con densidad $f_X(x)$, $x \in I$. El método consiste en representar $f_X(x)$ como

$$f_X(x) = C h(x)g(x),$$

donde $C \geq 1$, $h(x)$ es también una f.d.p., y $0 < g(x) \leq 1$. Generamos dos variables aleatorias U y Y de $U(0, 1)$ y $(h(y))$ respectivamente, y verificamos si se cumple la desigualdad $Ug(Y)$:

1. Si la desigualdad se cumple, entonces acepte Y como una variable generada de $f_X(x)$.
2. Si la desigualdad no se cumple, entonces rechace la pareja U y Y y trate de nuevo.

- La eficiencia de este método está determinada por la desigualdad $Ug(Y)$.
- Ya que los ensayos son independientes, la probabilidad de éxito en cada ensayo es $p = 1/C$.
- El número de ensayos antes de un par exitoso sigue una distribución geométrica:

$$p_X(x) = p(1 - p)^x, \quad x = 0, 1, \dots$$

con un número esperado de ensayos igual a C .