MAT-468: Sesión 8, Elementos de Simulación I

Felipe Osorio

http://fosorios.mat.utfsm.cl

Departamento de Matemática, UTFSM

Definición

Simulación: [Del lat. simulatio]

1. Acción de simular. **2.** Alteración aparente de la causa, la índole o el objeto verdadero de un acto o contrato.

Modelación. El acto de imitar el comportamiento de alguna situación o proceso por medio de algo análogo de forma adecuada.

Computación. Técnica de representación del mundo real usando una rutina computacional.

Simulación Estocástica

Conjunto de herramientas (estadísticas) para generar muestras aleatorias por medio de un computador con el fin de usarlas para obtener resultados aproximados.

Definición

Simulación: [Del lat. simulatio]

1. Acción de simular. **2.** Alteración aparente de la causa, la índole o el objeto verdadero de un acto o contrato.

Modelación. El acto de imitar el comportamiento de alguna situación o proceso por medio de algo análogo de forma adecuada.

Computación. Técnica de representación del mundo real usando una rutina computacional.

Simulación Estocástica

Conjunto de herramientas (estadísticas) para generar muestras aleatorias por medio de un computador con el fin de usarlas para obtener resultados aproximados.

Definición

Simulación: [Del lat. simulatio]

1. Acción de simular. **2.** Alteración aparente de la causa, la índole o el objeto verdadero de un acto o contrato.

Modelación. El acto de imitar el comportamiento de alguna situación o proceso por medio de algo análogo de forma adecuada.

Computación. Técnica de representación del mundo real usando una rutina computacional.

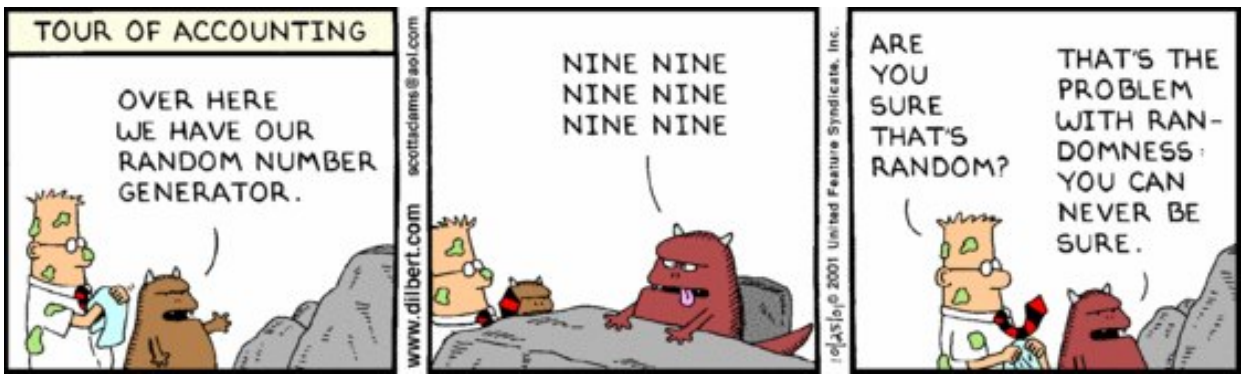
Simulación Estocástica

Conjunto de herramientas (estadísticas) para generar muestras aleatorias por medio de un computador con el fin de usarlas para obtener resultados aproximados.

Aleatoriedad y Predictibilidad

- Un computador no puede generar dígitos aleatorios.
- Nos contentaremos con:

  reglas para obtener números pseudo aleatorios, que parecen haber sido tomados al azar desde una distribución dada.



Generador de dígitos aleatorios

Definición 1

Una colección de v.a. $X_1, \ldots, X_n$ es una muestra aleatoria si son independientes y tienen la misma distribución de probabilidad.

Idea:

La clave de los métodos de simulación es la producción de números (pseudo)aleatorios. Esto es, un procedimiento que produce una colección

$$U_1, U_2, \cdots \overset{\text{iid}}{\sim} F.$$

La producción de v.a. será formalizada mediante definir un generador de dígitos pseudo-aleatorios (RNG).

RNG: U$(0, 1)$

Cuando $F \overset{\text{d}}{=} \text{U}(0, 1)$, el RNG se dice un generador de dígitos aleatorios uniformes (en el intervalo $(0, 1)$).

Definición 1

Una colección de v.a. $X_1, \ldots, X_n$ es una muestra aleatoria si son independientes y tienen la misma distribución de probabilidad.

Idea:

La clave de los métodos de simulación es la producción de números (pseudo)aleatorios. Esto es, un procedimiento que produce una colección

$$U_1, U_2, \cdots \overset{\text{iid}}{\sim} F.$$

La producción de v.a. será formalizada mediante definir un generador de dígitos pseudo-aleatorios (RNG).

RNG: U$(0, 1)$

Cuando $F \overset{\text{d}}{=} \text{U}(0, 1)$, el RNG se dice un generador de dígitos aleatorios uniformes (en el intervalo $(0, 1)$).

Generador de dígitos aleatorios

Observación:

Muchos lenguajes de programación disponen de RNGs implementados (suelen no ser recomendables).

Objetivo:

- Se desea una secuencia determinista de valores en $(0, 1)$ que imite una secuencia de v.a. IID U$(0, 1)$.
- RNGs requieren de valores iniciales o semillas para iniciar una recursión.

Definición 2 (RNG)

Un RNG es un algoritmo que partiendo de una semilla (o semillas) u_0 y una transformación D produce una secuencia

$$u_i = D^i(u_0), \qquad i = 1, \ldots, n,$$

de valores en $(0, 1)$ tal que para todo n, la secuencia $\{u_1, \ldots, u_n\}$ se comporta como una muestra desde U$(0, 1)$.

Generador de dígitos aleatorios

Observación:

La validez del algoritmo

$$u_i = D(u_{i-1}), \qquad i = 1, \ldots, n,$$

depende de verificar que la secuencia $u_1, \ldots, u_n$ permite aceptar la hipótesis,

$$H_0 : U_1, \ldots, U_n \text{ son iid } U(0,1).$$

Observación:

Es importante notar que en la práctica las secuencias generadas toman valores sobre el conjunto de enteros $\{0, 1, \ldots, M\}$, donde M se escoge como el mayor entero que se puede representar en el computador (en arq. de 32 bits, $M = 2^{31} - 1$).

Observación:

Una manera de caracterizar el desempeño de un algoritmo RNG es a través de su periodo.

Definición 3

El periodo T_0 de un generador es el entero más pequeño T tal que

$$u_{i+T} = u_i, \qquad \forall i.$$

Características de un buen RNG

- Pasar test estadísticos (baterías de test Knuth-TAOCP, DIEHARD, TestU01).
- Soporte teórico.
- Reproducible.
- Rápido y eficiente.
- Periodo (extremadamente) grande.
- Múltiples hebras.
- No producir 0 ó 1.

Definición 4

Sea M un entero positivo ≥ 2, una secuencia $\{x_1, x_2, \dots\}$ en $\{0, 1, \dots, M-1\}$ se dice generada por un método congruencial lineal (Lehmer, 1949) de parámetros a y b con semilla x_0 si

$$x_i = (ax_{i-1} + b) \mod M,$$

para a, b y x_0 enteros en $\{0, 1, \dots, M-1\}$.

a es llamado multiplicador, b incremento y M el módulo. Cuando $b = 0$ el RNG es llamado congruencial multiplicativo.

Para obtener dígitos en el intervalo $(0, 1)$ hacemos

$$u_i = \frac{x_i}{M}, \qquad i = 1, \dots, n.$$

Ejemplo

Considere la secuencia:

$$0, 1, 6, 15, 12, 13, 2, \ldots$$

generada con un RNG congruencial. ¿Cuál es el próximo número entre 0 y 15?

Mientras que la secuencia

$$1, 12, 1, 12, 1, 12, 1, \ldots$$

también generada usando un RNG congruencial. Fácilmente intuímos que el próximo número es 12.

Observación:

Existe condiciones bastantes específicas para las cuales un RNG congruencial produce secuencias de números satisfactorias.

Previo

Dos números son relativamente primos si ellos tienen a 1 como su divisor común.

Resultado 1

Si $b \neq 0$, el periodo del RNG congruencial $x_i = (ax_{i-1} + b) \mod M$ es igual a M sólo si

(a) b es relativamente primo a M.

(b) $a - 1$ es un múltiplo de todo primo que divide a M.

(c) $a - 1$ es múltiplo de 4, si M es múltiplo es multiplo de 4.

Resultado 2

Sea $x_i = (ax_{i-1} + b) \mod M$ con $b = 0$ y $M > 2$ es un número primo. Entonces

(a) el periodo máximo es $M - 1$.

(b) el periodo máximo es alcanzado si $a \mod M \neq 0$ y $a^{(M-1)/q} \mod M \neq 1$ para todo divisor primo q de $M - 1$.

Resultado 3

Sea $\{x_1, x_2, \dots\}$ una secuencia obtenida usando un generador con parámetros a, b y M, Sea $k \geq 1$ y

$$A = \{(x_i, \dots, x_{i+k-1})^\top : i = 0, 1, \dots, M - k\}.$$

Entonces, A está contenido en una familia de a lo más $(k!M)^{1/k}$ hiperplanos paralelos.

Ejemplo: RANDU

El pésimo y (infelizmente) muy usado generador RANDU es un RNG tipo Lehmer con $a = 2^{16} + 3$ y $M = 2^{31}$, es decir

$$x_i = 65539 x_{i-1} \mod 2^{31},$$

en cuyo caso tenemos que las tripletas (u_{i+1}, u_i, u_{i-1}) están en no más que 15 planos en $\mathbb{R}^3$.

Estructura de un RNG congruencial

Considere el generador congruencial

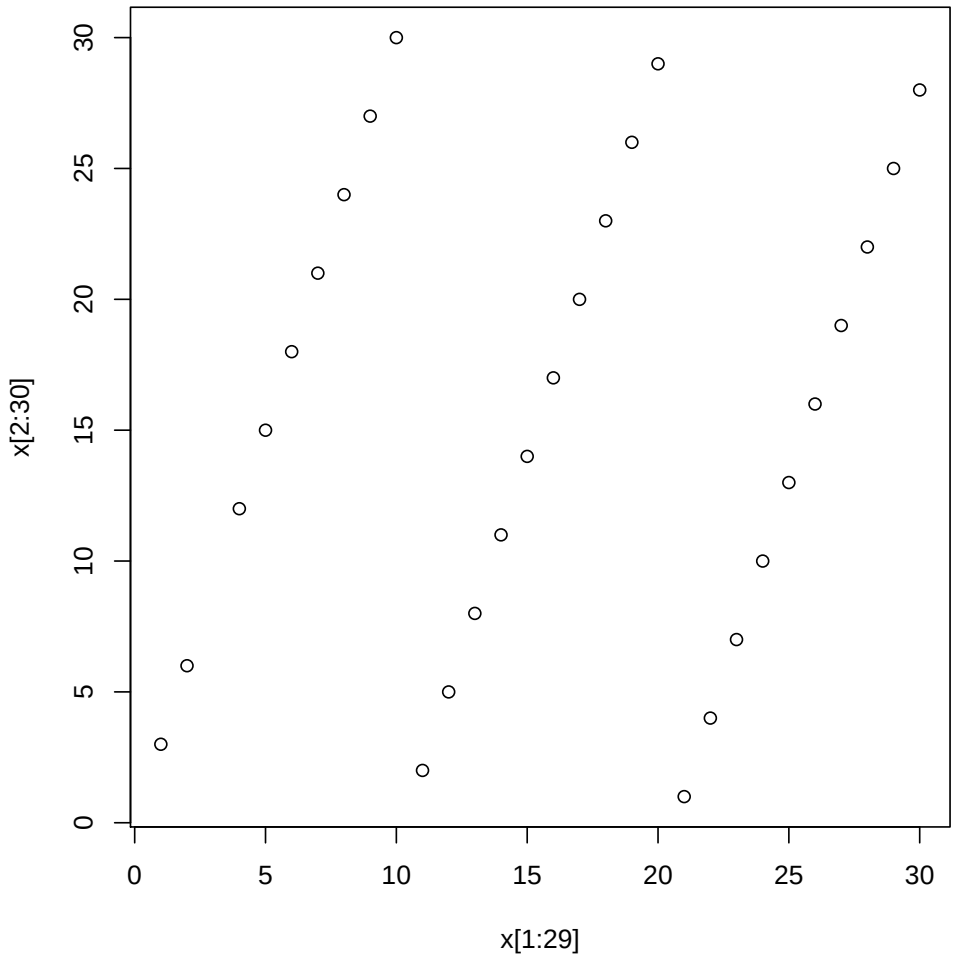
$$x_i = 3x_{i-1} \mod 31$$

usando como semilla $x_0 = 9$. Usamos los siguientes conmandos en R para simular 30 dígitos.

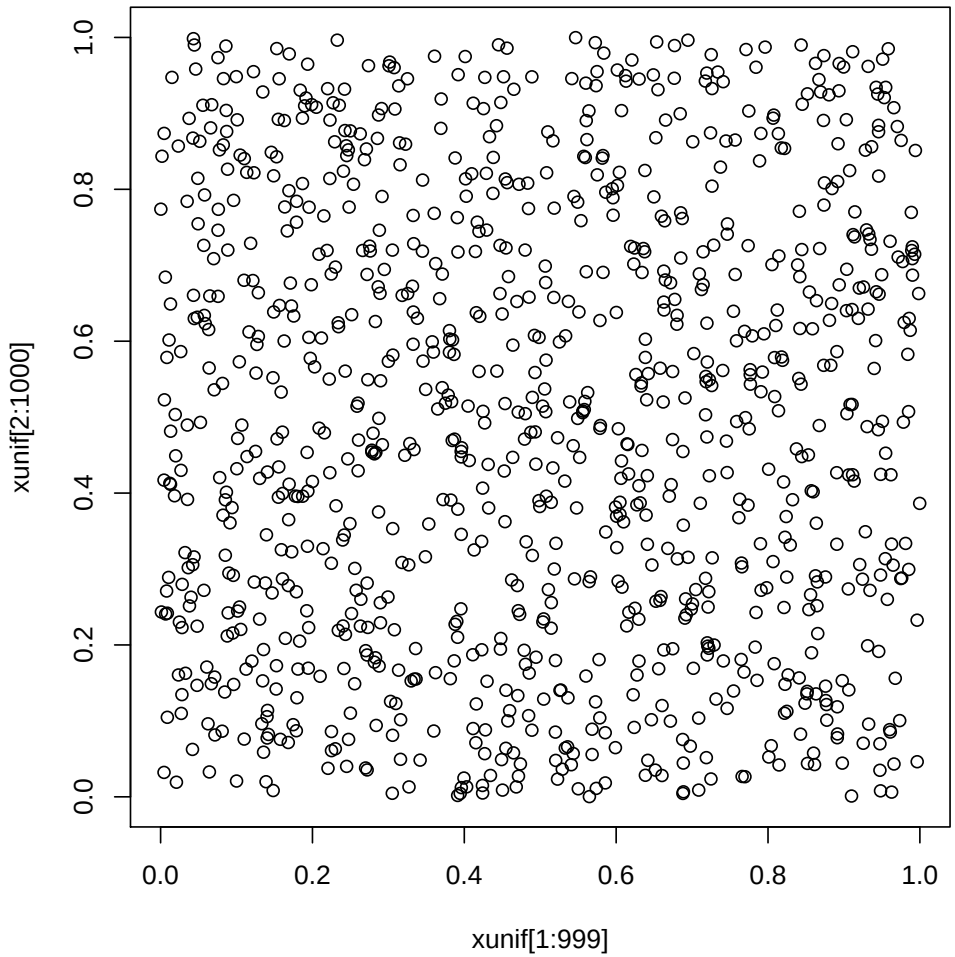
```
> x <- rep(0,30)
> x[1] <- 9
> for (i in 2:30) x[i] = (3 * x[i-1]) %% 31

> x
 [1]  9 27 19 26 16 17 20 29 25 13  8 24 10 30 28 22
[17]  4 12  5 15 14 11  2  6 18 23  7 21  1  3

> u <- x / 31
> mean(u)
[1] 0.5
> var(u)
[1] 0.08064516

> plot(x[1:29], x[2:30])
```

Para $Z \sim \mathrm{U}(0,1)$ tenemos $\mathrm{E}(Z) = \frac{1}{2}$ y $\mathrm{var}(Z) = 1/12 = 0.08\overline{3}$.

Estructura de un RNG congruencial



Estructura del RNG en `runif`



Definición 5

Un generador recursivo múltiple (MRG) de orden k es un generador tal que el conjunto de estados $(x_{i-k+1}, \ldots, x_i)^\top \in \{0, 1, \ldots, M-1\}^k$ y es definido como:

$$x_i = (a_1 x_{i-1} + a_2 x_{i-2} \cdots + a_k x_{i-k}) \mod M,$$

para $i = k, k+1, \ldots$ y los multiplicadores $\{a_i\}_1^k$ están en el conjunto $\{0, \ldots, M-1\}$. La salida debe ser transformada como

$$u_i = \frac{x_i}{M}, \qquad i = 1, \ldots, n.$$

El periodo máximo del generador es $M^k - 1$ (bajo ciertas condiciones).

Observación:

Para obtener algoritmos rápidos sólo unos pocos a_i's deberían ser no nulos.

Generador congruencial matricial

Observación:

MRGs son un caso particular de los generadores congruenciales matriciales, tal que

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix}, \qquad \boldsymbol{x}_i = \begin{pmatrix} x_i \\ x_{i+1} \\ \vdots \\ x_{i+k-1} \end{pmatrix}.$$

Observación:

La implementación de este tipo de generadores matriciales debe ser muy cuidadosa con el espacio de almacenamiento requerido.

Generador Wichman-Hill

Definición 6

Wichman y Hill (1982) describen un RNG que es una combinación de generadores congruenciales

$$\begin{aligned} x_i &= 171 x_{i-1} \mod 30269 \\ y_i &= 172 y_{i-1} \mod 30307 \\ z_i &= 170 z_{i-1} \mod 30323, \end{aligned}$$

y estos son combinados como

$$u_i = \left(\frac{x_i}{30269} + \frac{y_i}{30307} + \frac{z_i}{30323} \right) \mod 1$$

Observación:

Este RNG requiere una semilla (x_o, y_0, z_0) y su periodo es

$$(30269 - 1)(30307 - 1)(30323 - 1)/4 \approx 6.95 \cdot 10^{12}.$$

Además, Zeisel (1986) mostró que el generador Wichman-Hill es equivalente a un RNG congruencial con

$$a = 16\,555\,425\,264\,690, \quad M = 27\,817\,185\,604\,309$$

Combinando generadores

- El periodo y aleatoriedad de un RNGs puede ser mejorado mediante combinar más de un generador.

- Algunos generadores de este tipo:
  - Super-Duper (Reeds et al., 1982-4).
  - Wichmann-Hill (1982).
  - KISS - Keep It Simple, Stupid (Marsaglia and Zaman, 1993).
  - MRG32ka (L'Ecuyer, 1999).

Generador KISS

Definición 7 (Marsaglia y Zaman, 1993)

KISS está basado en combinar un generador congruencial y dos LFSR, del siguiente modo:

a) Un generador congruencial:

$$w_{n+1} = (69\,069 w_n + 23\,606\,797) \mod 2^{32}$$

b) Dos generadores LFSR

$$x_{n+1} = (\boldsymbol{I} + \boldsymbol{L}^{15})(\boldsymbol{I} + \boldsymbol{R}^{17}) x_n \mod 2^{32}$$
$$y_{n+1} = (\boldsymbol{I} + \boldsymbol{L}^{13})(\boldsymbol{I} + \boldsymbol{R}^{18}) y_n \mod 2^{32}$$

c) Los que son combinados como:

$$z_{n+1} = (w_{n+1} + x_{n+1} + y_{n+1}) \mod 2^{32}.$$

El periodo de KISS es de orden 2^{95} y ha sido probado exitosamente por los criterios disponibles en Die Hard.

Generador KISS

La definición del generador KISS está basado en las siguientes matrices:

$$\boldsymbol{T}_L = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ & & & \ddots & \\ & & & 1 & 1 \\ 0 & \dots & & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{T}_R = \boldsymbol{T}_L^\top,$$

que, en efecto, está relacionado con la operación

$$\boldsymbol{R}(e_1, \dots, e_k)^\top = (0, e_1, \dots, e_{k-1})^\top, \qquad \boldsymbol{L}(e_1, \dots, e_k)^\top = (e_2, \dots, e_k, 0)^\top$$

y además

$$\boldsymbol{T}_L = \boldsymbol{I} + \boldsymbol{L}, \qquad \boldsymbol{T}_R = \boldsymbol{I} + \boldsymbol{R}.$$

Definición 8 (L'Ecuyer, 1999)

Este generador es definido como:

a) Dos generadores recursivos:

$$x_i = (1\,403\,580 x_{i-2} - 810\,728 x_{i-3}) \mod M_1$$
$$y_i = (527\,612 y_{i-1} - 1\,370\,589 y_{i-3}) \mod M_2$$

donde $M_1 = 2^{32} - 209$ y $M_2 = 2^{32} - 22\,853$.

b) La salida es definida mediante la combinación:

$$z_i = \begin{cases} \dfrac{x_i - y_i + M_1}{M_1 + 1}, & x_i \leq y_i \\ \dfrac{x_i - y_i}{M_1 + 1}, & x_i > y_i \end{cases}$$

El periodo del generador es aproximadamente 2^{191}. MRG32ka supera todos los test estadísticos conocidos (TestU01: L'Ecuyer y Simard, 2007).

Métodos para generar variables no uniformes

- Sea F función de distribución definida en $\mathbb{R}$ y $\{X_1, X_2, \dots\}$ v.a. IID desde F.
- Se desea generar una realización $\{x_1, x_2, \dots\}$ desde $\{X_1, X_2, \dots\}$.
- Suponga que tenemos una muestra $\{u_1, u_2, \dots\}$ generada desde $\mathsf{U}(0, 1)$.
- Los siguientes métodos permiten transformar $\{u_1, u_2, \dots\}$ en $\{x_1, x_2, \dots\}$.

Definición 9

Para F función no decreciente, la inversa (generalizada) de F, denotada por F^- está definida por:

$$F^-(u) = \inf\{x : F(x) \geq u\}.$$

Lema

Si $U \sim \mathrm{U}(0,1)$, entonces la variable aleatoria $F^-(U)$ tiene distribución F. (

En efecto, basta notar que:

$$P(X \leq x) = P(F^-(U) \leq x) = P(U \leq F(x)) = F(x).$$

Definición 10

De este modo, para generar una v.a. $X \sim F$ es suficiente generar $U \sim \mathsf{U}(0,1)$ y hacer la transformación

$$x = F^{-}(u).$$

Ejemplo: $X \sim \mathsf{Exp}(1)$

Suponga que $X \sim \mathsf{Exp}(1)$, es decir $F(x) = 1 - e^{-x}$. Entonces, resolviendo para x en:

$$u = 1 - e^{-x},$$

tenemos que $x = -\log(1-u)$. Note además que $U \overset{\mathsf{d}}{=} 1 - U$.

Por tanto, para generar una observación desde $\mathsf{Exp}(1)$, basta hacer $u \sim \mathsf{U}(0,1)$ y luego tomar $x = -\log(u)$.

Observación:

Para $X \sim \mathsf{Exp}(\lambda)$, tenemos

$$F^{-1}(u) = -\frac{1}{\lambda}\log(1-u).$$

Observación:

Usualmente la evaluación de F^- es lenta y computacionalmente costosa, así que otros métodos son preferidos.

Ejemplo: $X \sim \mathsf{N}(0,1)$

Si $X \sim \mathsf{N}(0,1)$ podríamos usar

$$X = \Phi^{-1}(U), \qquad U \sim \mathsf{U}(0,1).$$

Sin embargo, tanto Φ como su inversa Φ^{-1} (usualmente) son aproximados usando la función error

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, \mathrm{d}t.$$

que es evaluada usando fracciones continuas o expansiones en series.

Observación:

Otra alternativa es resolver la ecuación no lineal $F(x) - u = 0$, usando métodos iterativos.

Métodos de Inversión: Caso discreto

Si X es una v.a. discreta, y

$$\cdots < x_{i-1} < x_i < x_{i+1} < \ldots$$

son los puntos de discontinuidad de F. Entonces la transformación inversa es dada por:

$$F^-(u) = x_i, \quad \text{donde } F(x_{i-1}) < u \leq F(x_i).$$

El algoritmo se reduce a los siguientes pasos:

(a) Generar u desde $\mathrm{U}(0,1)$.

(b) Devolver x_i donde $F(x_{i-1}) < u \leq F(x_i)$.[1]

[1] Esto requiere de un algoritmo de búsqueda

Generando dígitos por su definición*

Este procedimiento es útil cuando F está relacionado con una distribución simple de simular. Lamentablemente este método es bastante caso-específico (no es recomendable).

Ejemplo: basados en v.a. Exponenciales

Sabemos que es fácil generar v.a. $X \sim \text{Exp}(1)$. De este modo,

$$Y = 2\sum_{j=1}^{\nu} X_j \sim \chi^2_{2\nu}, \quad \nu \in \mathbb{N},$$

$$Y = \beta\sum_{j=1}^{a} X_j \sim \text{Gama}(a, \beta), \quad a \in \mathbb{N}, \beta > 0,$$

$$Y = \frac{\sum_{j=1}^{a} X_j}{\sum_{k=1}^{a+b} X_k} \sim \text{Beta}(a, b), \quad a, b \in \mathbb{N},$$

Generando dígitos por su definición*

Ejemplo: Método de Box-Muller

Considere generar v.a. desde $\mathcal{N}(0,1)$ y suponga que R y θ son las coordenadas polares de (X_1, X_2). Entonces, dado que (X_1, X_2) es invariante por rotaciones

$$R^2 = X_1^2 + X_2^2 \sim \chi^2(2) \overset{\mathsf{d}}{=} \mathsf{Exp}(\tfrac{1}{2})$$
$$\theta \sim U(0, 2\pi).$$

El algoritmo se puede escribir como:[2]

(a) Generar u_1 y u_2 v.a. iid desde $U(0,1)$.

(b) Hacer

$$x_1 = \sqrt{-2\log(u_1)}\cos(2\pi u_2)$$
$$x_2 = \sqrt{-2\log(u_1)}\sin(2\pi u_2).$$

[2] Usar una Transformación inversa 2D