



Departamento de Ciencias Básicas y Modelado

SIMULACION ESTOCASTICA

Tema 1

Presentación
Sistemas, modelos y simulación
Repaso probabilidad
Números aleatorios
Generación de variables aleatorias

Profesor: Javier Riascos Ochoa, MSc, PhD
(javier.riascos@utadeo.edu.co)

Programa del curso

Los procesos estocásticos permiten describir y cuantificar la dinámica de variables aleatorias siendo fundamentales para realizar predicciones basadas en conceptos probabilísticos. Por tanto, son la base para la modelación en diferentes áreas de las ciencias naturales, ciencias sociales y económicas, ciencias médicas e ingenierías, en donde se presenten fenómenos de carácter aleatorio.

La simulación de variables aleatorias y de procesos estocásticos, a través de los llamados métodos de Monte Carlo, son entonces la herramienta para reproducir y estudiar estos fenómenos, utilizando las facilidades de la computación moderna.

La asignatura está diseñada como una introducción a la simulación estocástica, los métodos de Monte Carlo y sus aplicaciones.

Programa del curso

Objetivos

1. Introducir los principales modelos probabilísticos y de procesos estocásticos, sus procedimientos analíticos y las técnicas computacionales de simulación (métodos de Monte Carlo) para su solución.
2. Desarrollar en el estudiante la capacidad de reconocer, plantear y solucionar los modelos probabilísticos y/o estocásticos más adecuados en la descripción de una situación real.

Programa del curso

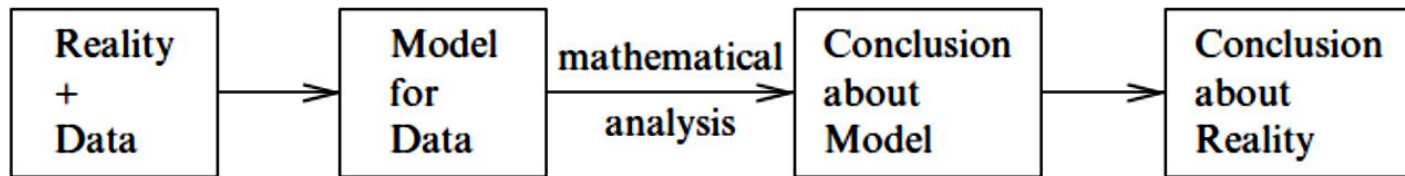
Evaluación

- 4 Talleres con ejercicios teóricos y computacionales - en parejas (60%)
- 1 proyecto de aplicación – en parejas (40%)
 - Propuesta (15%)
 - Exposición final (25%)

Sistemas, modelos y simulación

Estructura del modelado matemático y simulación

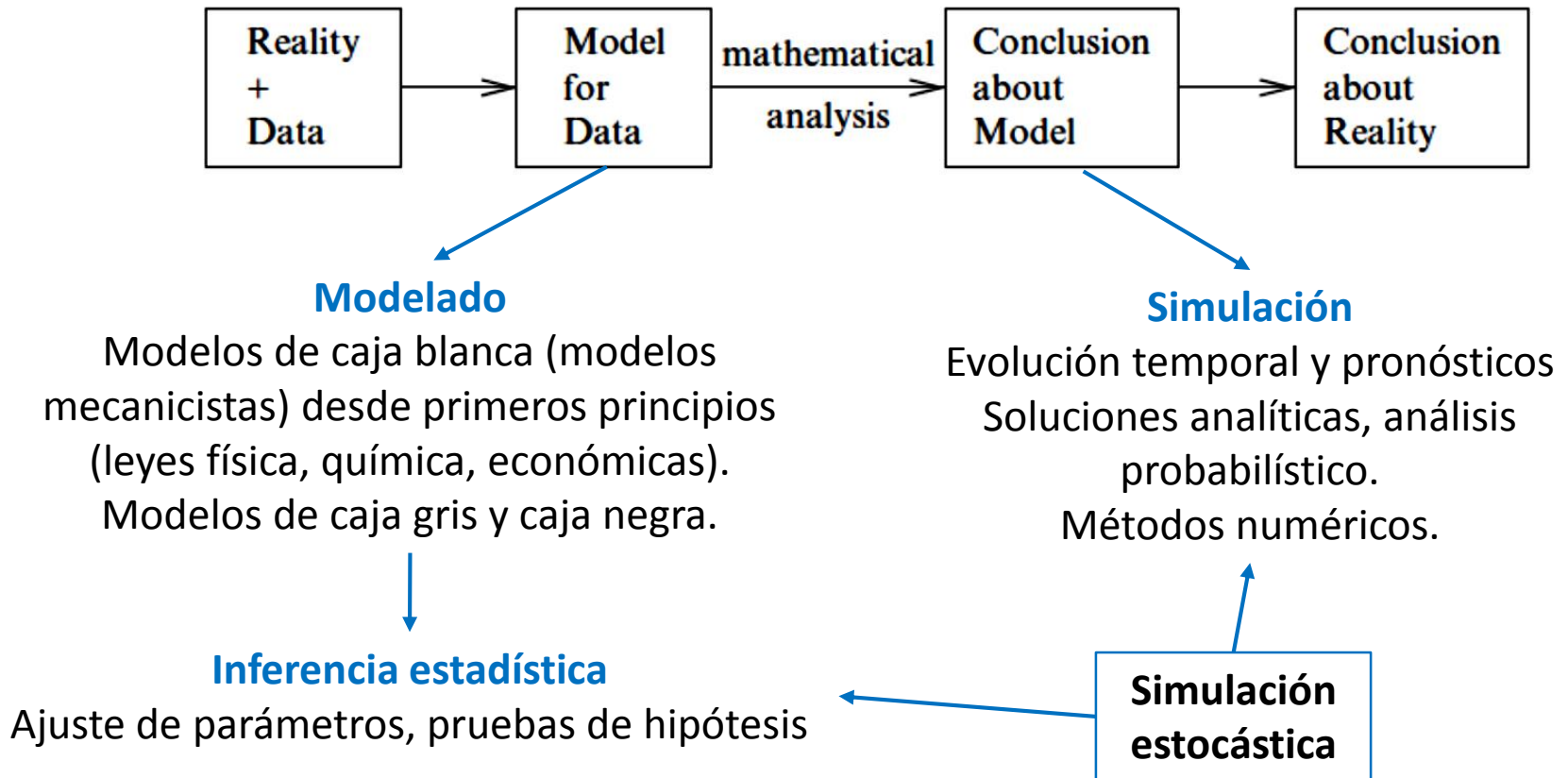
Kroese, Chan, 2014



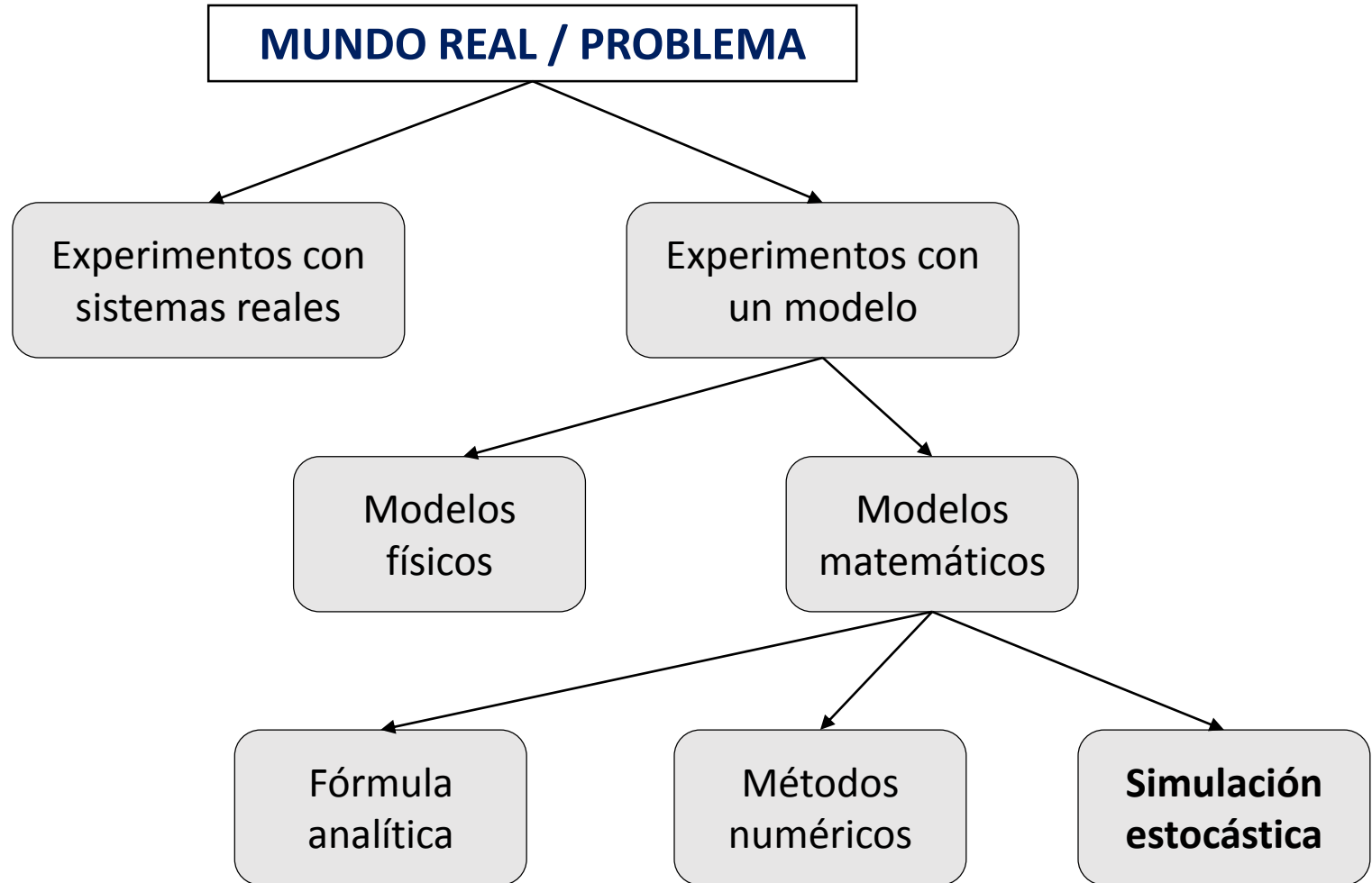
Sistemas, modelos y simulación

Estructura del modelado matemático y simulación

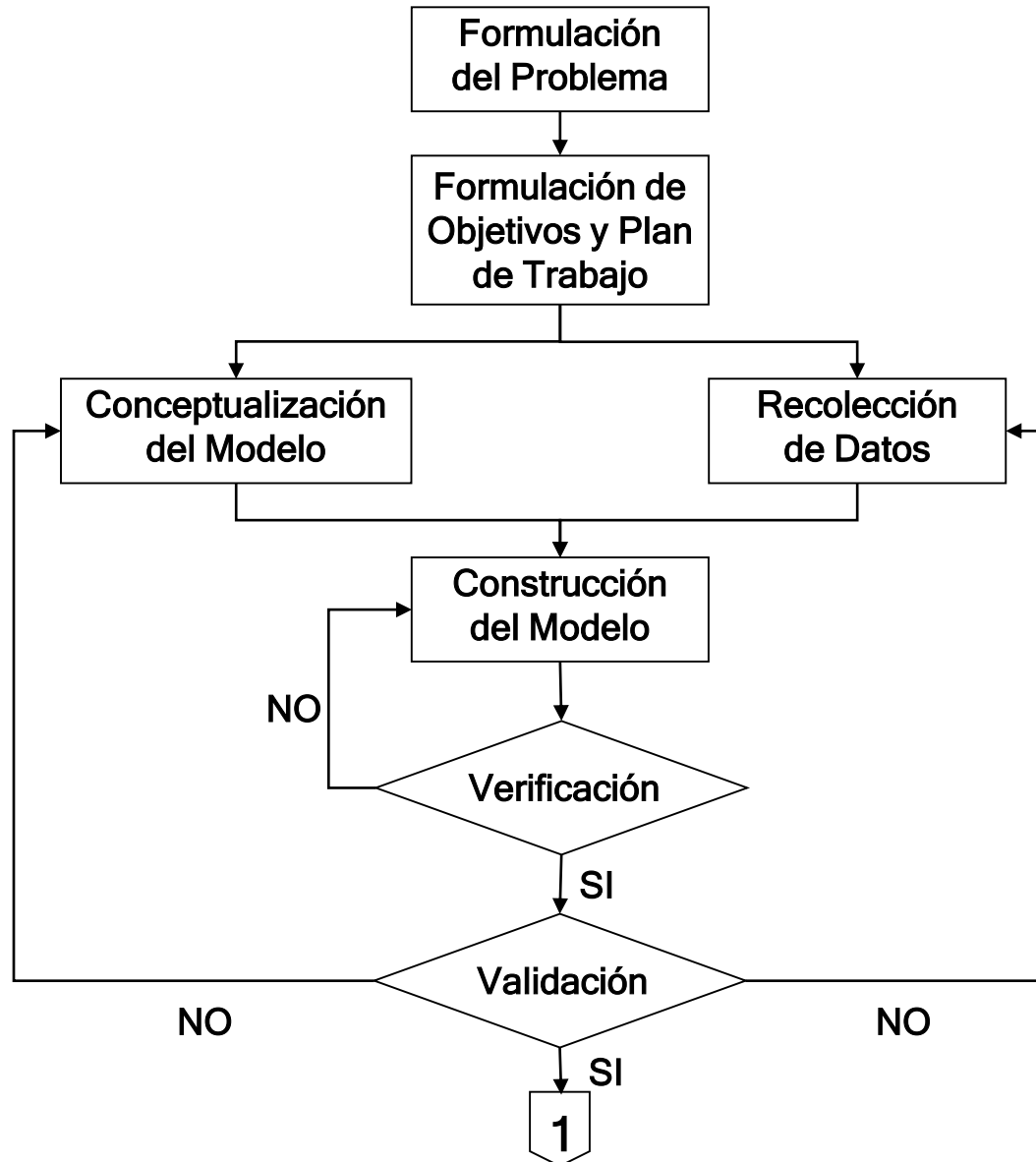
Kroese, Chan, 2014



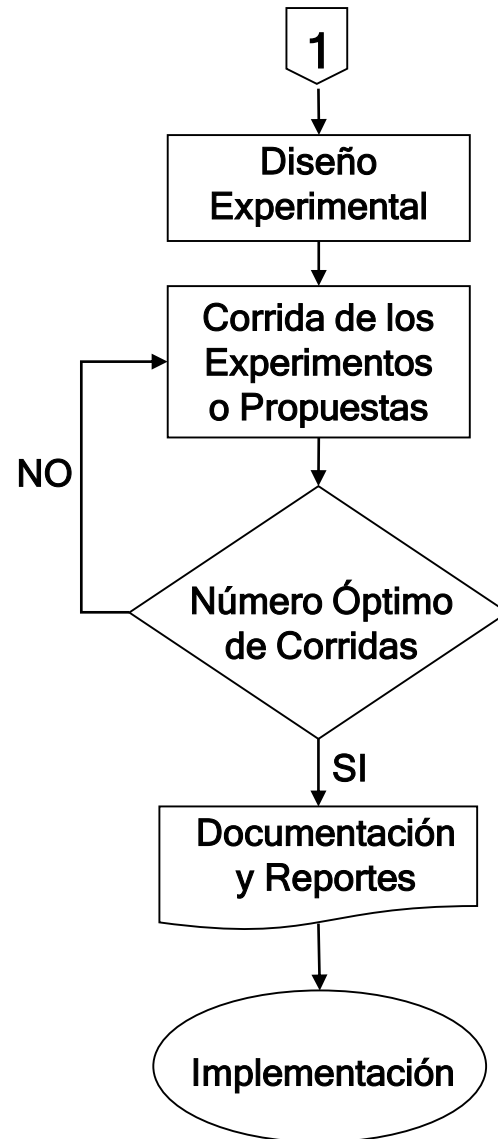
Sistemas, modelos y simulación



Pasos en un estudio de simulación (1)



Pasos en un estudio de simulación (2)



Introducción a la probabilidad

Presentación IntroProbabilidad.pdf

Tipos de modelos matemáticos

Modelo analítico: Podemos obtener una fórmula para la cantidad de interés. Difícil de derivar pero fácil de usar.

Ejemplos: modelo por EDO del crecimiento exponencial o logístico; cola de 1 servidor con tiempos exponenciales; ecuación de difusión.

Métodos numéricos: Usa iteraciones y/o aproximaciones.

Ejemplos: Runge-Kutta para EDO, elementos finitos para EDP, programación lineal y no lineal para optimización, solución de sistemas lineales para cadenas de Markov, etc.

Simulación estocástica: Implica incertidumbre/aleatoriedad.

Herramientas básicas: probabilidad y estadística.

Un modelo de simulación puede ser más detallado y realístico que uno analítico. Pero el analítico da más intuición del comportamiento del sistema, del impacto de decisiones y es más fácil de usar.

Tipos de modelos matemáticos

Modelos determinísticos: No hay aleatoriedad

Ejemplos: Ecuaciones diferenciales, programación lineal...

Modelos estocásticos: Comportamiento afectado por aleatoriedad.

Ejemplos: Cadenas de Markov (CM), sistema de colas, modelos de confiabilidad/riesgo, comportamiento humano/social, mecánica estadística...

Modelos estáticos: El “tiempo” no se incluye, o el sistema es independiente del tiempo.

Ejemplos: Confiabilidad de una red, distribución espacial de una población en un t fijo, estado estacionario de una CM...

Modelos dinámicos: Es de interés el comportamiento como función del tiempo.

Ejemplos: Dinámica de poblaciones, evolución de una CM...

Estructura básica de una simulación estocástica

1. Identificar una *variable aleatoria (v.a.)* X de interés y escribir un programa para simularla (puede depender de otras v.a. o parámetros determinísticos).
2. Generar una muestra *independiente e idénticamente distribuida (iid)* X_1, \dots, X_n con la misma distribución que X . Requiere de **generadores de números aleatorios**.
3. Estimar el valor esperado $E[X]$ (usando su media muestral \bar{X}) y estimar la precisión del estimador (usando un intervalo de confianza).

Números aleatorios

El núcleo de una simulación estocástica es la capacidad de generar ***números aleatorios***, los cuales representan el valor de una variable aleatoria con *distribución uniforme*.

Definición:

Los números aleatorios corresponden a una variable continua que toma un valor en el rango entre (0,1), donde cada valor posible tiene igual probabilidad de ocurrencia y es independiente; en otras palabras, cuentan con dos propiedades estadísticas:

- Uniformidad.
- Independencia.

Es decir, se busca que se generen X_n v.a's iid $\sim U(0,1)$.

Números aleatorios

Generación de números aleatorios

En un principio se generaban de manera manual: tirando dados, ruedas giratorias, barajas, etc. Ahora se utilizan *métodos computacionales* para generarlos.

Un generador de números aleatorios por computador utiliza una fórmula específica, completamente determinística. Por eso se conocen como ***números pseudoaleatorios***.

Números aleatorios

Propiedades de los generadores de números aleatorios

- Que estén uniformemente distribuidos
- No presenten auto-correlación y sean independientes
- Que no presenten patrones ni sesgos en la generación
- Se acerquen en su valor promedio al valor esperado
- Que no presenten una variación muy alta
- Que tengan un ciclo largo, antes de que el primer número generado se repita
- Que sea replicable el conjunto de aleatorios
- Que sea eficiente y rápido en la generación

Números aleatorios

Generadores congruentes

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

X_0 Semilla (con $0 \leq \text{semilla} < \text{norma}$)

a Constante multiplicadora

c Incremento

m Norma (también conocido como el residuo de la división).

Ejemplo en tablero: Generar la serie de números X_i y determinar su longitud máxima de período, para cuando

$$c = 2, \quad a = 13, \quad m = 5, \quad X_0 = 1$$

Números aleatorios

Generadores congruentes

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

X_0	Semilla (con $0 \leq \textit{semilla} < \textit{norma}$)
a	Constante multiplicadora
c	Incremento
m	Norma (también conocido como el residuo de la división).

Observaciones:

- Se generan a lo sumo m diferentes números aleatorios.
- El generador es cíclico (repite los números).
- La longitud de un generador aleatorio son los diferentes números aleatorios que se generan, antes de que se repita el ciclo.
- Ejemplo de un buen generador congruente:

$$m = 2^{32}, \quad a = 1.664.525, \quad c = 1.013.904.223$$

Números aleatorios

Generadores congruentes

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

X_0 Semilla (con $0 \leq \text{semilla} < \text{norma}$)

a Constante multiplicadora

c Incremento

m Norma (también conocido como el residuo de la división).

Generación de una v.a. $\sim U(0,1)$:

La secuencia de números pseudoaleatorios que “simulan” una variable aleatoria uniformemente distribuida entre 0 y 1 se obtiene dividiendo cada X_i por la norma m :

$$\frac{X_i}{m} \sim U(0,1)$$

Números aleatorios

Generadores congruentes

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

X_0	Semilla (con $0 \leq \textit{semilla} < \textit{norma}$)
a	Constante multiplicadora
c	Incremento
m	Norma (también conocido como el residuo de la división).

Ejercicio en R:

Con la función `random_uniform_congruential.R`, generar la serie de números aleatorios entre 0 y 1, y determinar su longitud máxima de período, para cuando:

$$c = 2, \quad a = 13, \quad m = 31, \quad X_0 = 1.$$

Números aleatorios

Generadores congruentes

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

X_0 Semilla (con $0 \leq \text{semilla} < \text{norma}$)

a Constante multiplicadora

c Incremento

m Norma (también conocido como el residuo de la división).

Observaciones:

Cuando el parámetro $c = 0$ esta ecuación se conoce como método congruencial multiplicativo.

Si $c \neq 0$ y $a = 1$ esta ecuación se conoce como método congruencial aditivo.

Números aleatorios

- En **R** la generación de una variable aleatoria con distribución uniforme $U(0,1)$ se realiza con el comando: `runif()`
- **Ejercicio en R:** Ejecutar el código `random_numbers.R`

Generación de variables aleatorias

1. Método de la transformada inversa para v.a.'s discretas
2. Método de la transformada inversa para v.a.'s continuas
3. Método del rechazo

Generación de variables aleatorias

1. Método de la transformada inversa para v.a.'s discretas
2. Método de la transformada inversa para v.a.'s continuas
3. Método del rechazo

Método de la transformada inversa para v.a. discretas

El método de la transformada inversa

Queremos generar una v.a discreta con función de probabilidad de masa p y CDF F . Es decir:

$$P(X = x_j) = p_j, \quad j = 0, 1, \dots, \quad \sum_j p_j = 1$$

Si generamos una v.a. $U \sim \text{Uniform}(0,1)$ y escribimos:

$$X = \begin{cases} x_0 & \text{If } U < p_0 \\ x_1 & \text{If } p_0 \leq U < p_0 + p_1 \\ \vdots & \\ x_j & \text{If } \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^j p_i \\ \vdots & \end{cases}$$

Tenemos que:

$$p\{X = x_j\} = p\left\{\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^j p_i\right\} = p_j$$

Luego X tiene la distribución deseada.

Método de la transformada inversa para v.a. discretas

El método de la transformada inversa

En otras palabras, después de generar un número aleatorio U , determinamos el valor de X encontrando el intervalo

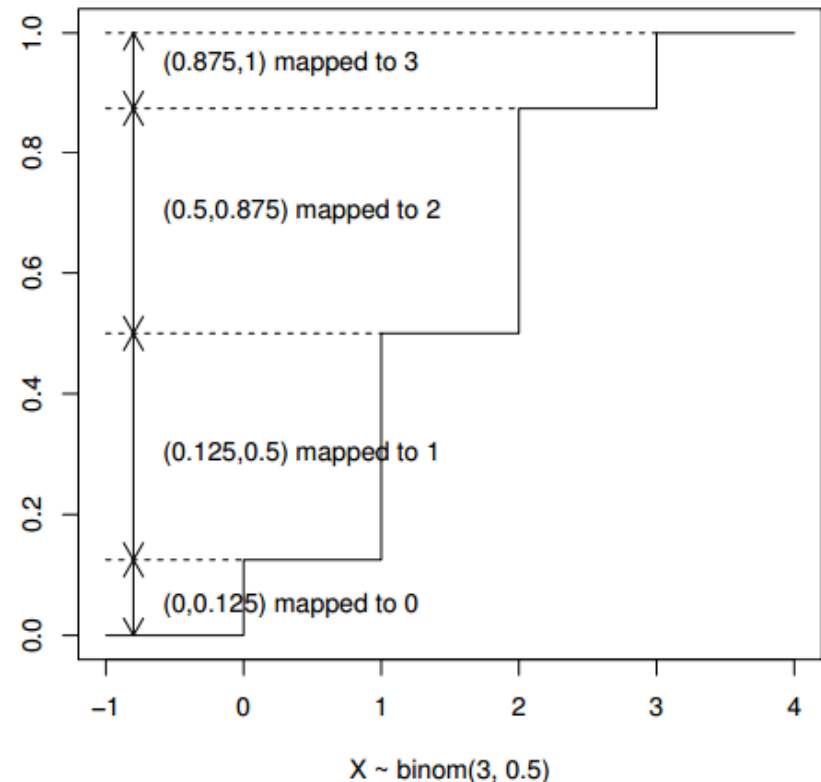
$$[F(x_{j-1}), F(x_j))$$

en el cual U se encuentra [o equivalentemente encontrando la inversa de $F(U)$].

Programa

```
# given U dist. as U(0,1)
X <- 0
while (F(X) < U)
{ X <- X + 1 }
```

simulating from a binom(3, 0.5) c.d.f.



Método de la transformada inversa para v.a. discretas

El método de la transformada inversa

Para simular una v.a. discreta, **R** provee la función:

```
sample(x, size, replace = FALSE, prob = NULL)
```

Las entradas son:

`x` A vector giving the possible values the rv can take;

`size` How many rv's to simulate;

`replace` Set this to TRUE to generate an iid sample;

`prob` A vector giving the probabilities of the values in `x`. If omitted then the values in `x` are assumed to be equally likely.

Método de la transformada inversa para v.a. discretas

Ejemplo: Distribución Binomial

Definición:

Una v.a. X se distribuye $\text{binom}(n, p)$ si es el número de “éxitos” en una serie de n experimentos independientes con dos posibles resultados: éxito o fracaso, siendo p la probabilidad de éxito en uno de tales experimentos. Tal experimento se conoce como *Experimento de Bernoulli*.

Ejemplo:

Se lanza 10 veces un dado no cargado de 6 caras. Nos interesa X , la v.a. que da el número de veces que sale el 4 en los 10 lanzamientos. Entonces:

$$X \sim \text{binom}(10, 1/6)$$

Método de la transformada inversa para v.a. discretas

Ejemplo: Distribución Binomial

Función de masa de probabilidad (pmf):

Se puede demostrar que la función de masa de probabilidad de $X \sim \text{binom}(n, p)$ es:

$$P(X = x) = p_X(x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

Simulación:

Se definen dos funciones (ver código `binom.cdf.R` en la siguiente diapositiva):

1. La función `binom.cdf` calcula la cdf F_X de X .
2. La función `cdf.sim` toma como primer argumento una función F que da la CDF de una v.a. entera no negativa. `cdf.sim` también usa el argumento ... para pasar parámetros a la función F desde la llamada de `cdf.sim`.

Para simular una v.a. $X \sim \text{binom}(n, p)$ se usa: `cdf.sim(binom.cdf, n, p)`

Método de la transformada inversa para v.a. discretas

Ejemplo: Distribución Binomial

Código:

```
binom.cdf <- function(x, n, p) {  
  Fx <- 0  
  for (i in 0:x) {  
    Fx <- Fx + choose(n, i)*p^i*(1-p)^(n-i)  
  }  
  return(Fx) }  
  
cdf.sim <- function(F, ...) {  
  X <- 0  
  U <- runif(1)  
  while (F(X, ...) < U) {  
    X <- X + 1  
  }  
  return(X) }  
  
n=10  
p=1/6  
cdf.sim(binom.cdf, n, p)
```

Método de la transformada inversa para v.a. discretas

Ejemplo: Distribución Binomial

Programa mejorado

```
# program
spuRs/resources/scripts/binom.sim.r
binom.sim <- function(n, p) {
  X <- 0
  px <- (1-p)^n
  Fx <- px
  U <- runif(1)
  while (Fx < U) {
    X <- X + 1
    px <- px*p/(1-p) * (n-X+1)/X
    Fx <- Fx + px
  }
  return(X)
}
```

Utiliza la siguiente fórmula de recursión para la pmf

$$p_X(x) = \frac{(n-x+1)p}{x(1-p)} p_X(x-1).$$

Método de la transformada inversa para v.a. discretas

Ejemplo: Distribución Binomial

Simulación a partir de un Experimento de Bernoulli:

```
X <- sum(runif(n) < p)
```

- Más rápido
- Más simple
- Utiliza más variables aleatorias uniformes para generar un solo valor de X

1. Método de la transformada inversa para v.a.'s discretas
2. Método de la transformada inversa para v.a.'s continuas
3. Método del rechazo

Método de la transformada inversa para v.a. continuas

Método de inversión para v.a. continuas

Supongan que nos dan $U \sim U(0,1)$ y queremos simular una v.a. continua X con CDF dada por F_X .

Definiendo $Y = F_X^{-1}(U)$ tenemos:

$$F_Y(y) = \mathbb{P}(Y \leq y) = \mathbb{P}(F_X^{-1}(U) \leq y) = \mathbb{P}(U \leq F_X(y)) = F_X(y).$$

En otras palabras, X y Y tienen la misma distribución.

Si podemos simular $U \sim U(0,1)$ podemos simular cualquier v.a. continua X para la cual se pueda calcular F_X^{-1} .

Método de la transformada inversa para v.a. continuas

Ejemplo: Distribución uniforme

$$X \sim U(1,3)$$

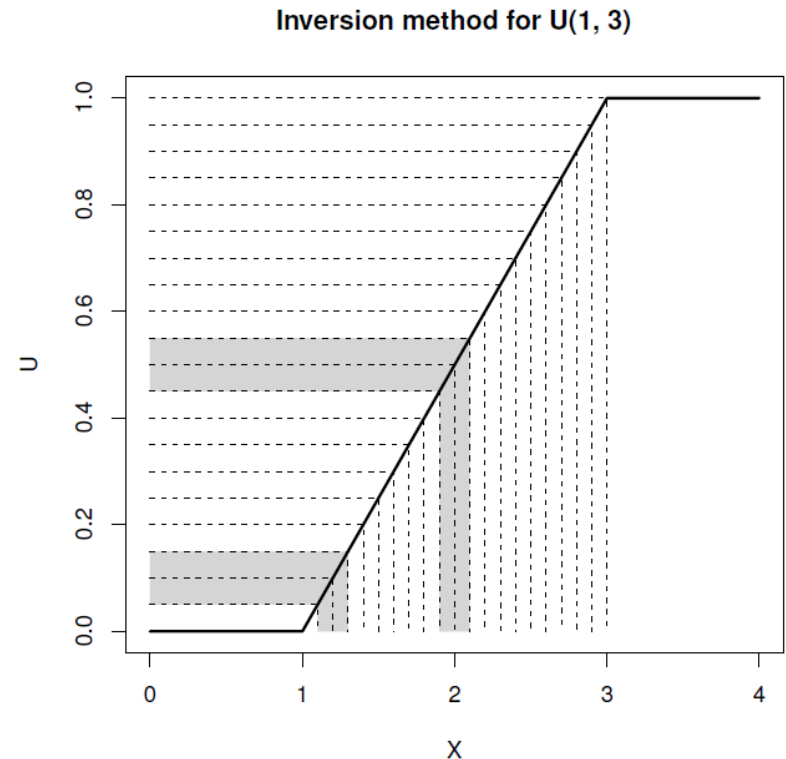
Observe que: $F_X(x) = \frac{x-1}{2}$ para $x \in (1,3)$.

Luego: $F_X^{-1}(y) = 2y + 1$, para $y \in (0,1)$.

El método de inversión nos dice que X se puede generar como:

$$X = 2U + 1$$

Con $U \sim U(0,1)$.



Método de la transformada inversa para v.a. continuas

Ejemplo: Distribución exponencial

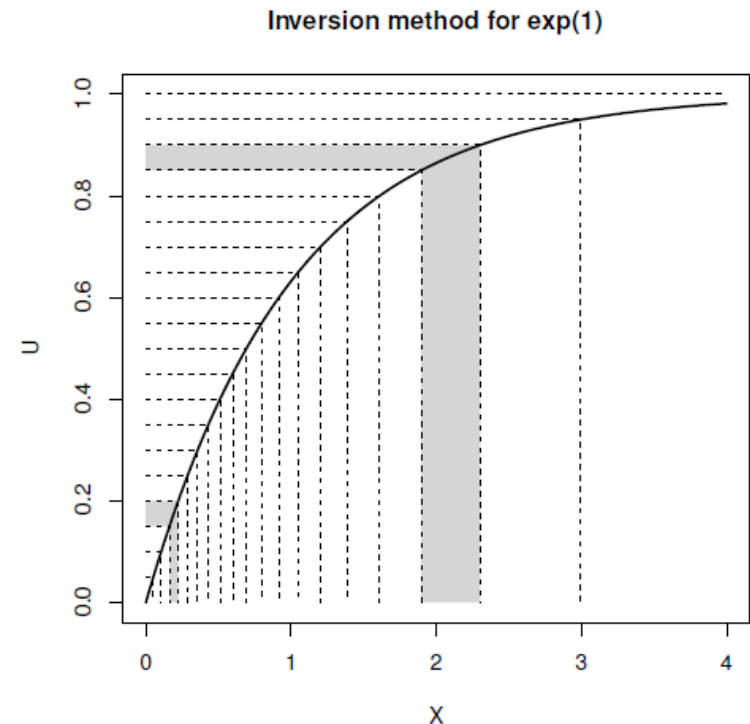
$$X \sim \text{Exp}(\lambda)$$

$$f_X(x) = \lambda e^{-\lambda x} \quad x > 0$$

$$F_X(x) = \begin{cases} 0, & x < 0 \\ 1 - e^{-\lambda x}, & x \geq 0 \end{cases}$$

Ejercicio:

- Halle $F_X^{-1}(y)$
- ¿Cómo se puede generar $X \sim \text{Exp}(\lambda)$?



Método de la transformada inversa para v.a. continuas

Ejemplo: Distribución exponencial

$$X \sim \text{Exp}(\lambda)$$

$$f_X(x) = \lambda e^{-\lambda x} \quad x > 0$$

$$F_X(x) = \begin{cases} 0, & x < 0 \\ 1 - e^{-\lambda x}, & x \geq 0 \end{cases}$$

Ejercicio:

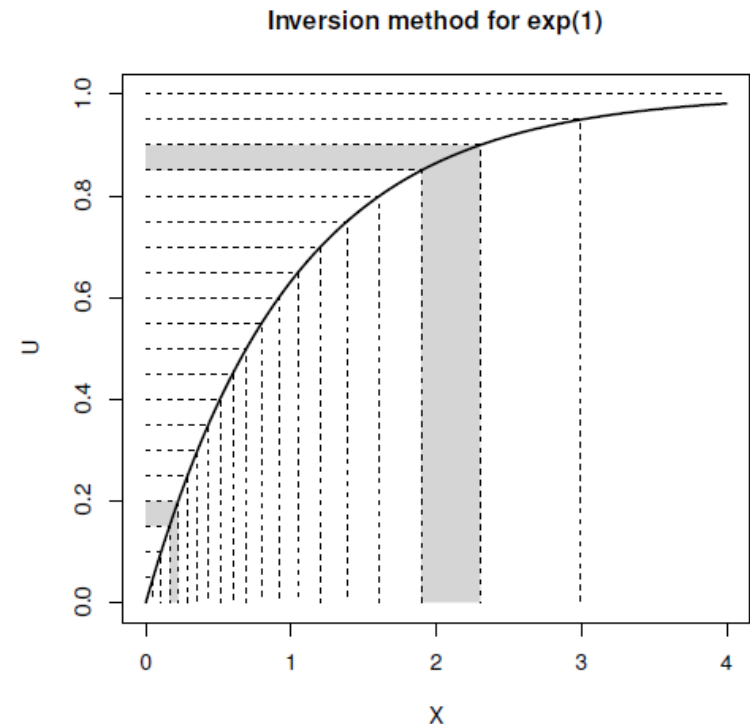
- $y = 1 - e^{-\lambda x} \rightarrow x = \frac{\ln(1-y)}{-\lambda} :$

$$F_X^{-1}(y) = -\frac{\ln(1-y)}{\lambda}$$

- Sea $Y \sim U(0,1)$. Entonces $X = -\frac{\ln(1-Y)}{\lambda}$ se distribuye $\text{Exp}(\lambda)$.

Como $1 - Y$ es también $U(0,1)$, X se puede generar como:

$$X = -\frac{\ln(U)}{\lambda}$$



Método de la transformada inversa para v.a. continuas

Ejemplo: Distribución exponencial

$$X \sim \text{Exp}(\lambda)$$

$$f_X(x) = \lambda e^{-\lambda x} \quad x > 0$$

$$F_X(x) = \begin{cases} 0, & x < 0 \\ 1 - e^{-\lambda x}, & x \geq 0 \end{cases}$$

Ejercicio:

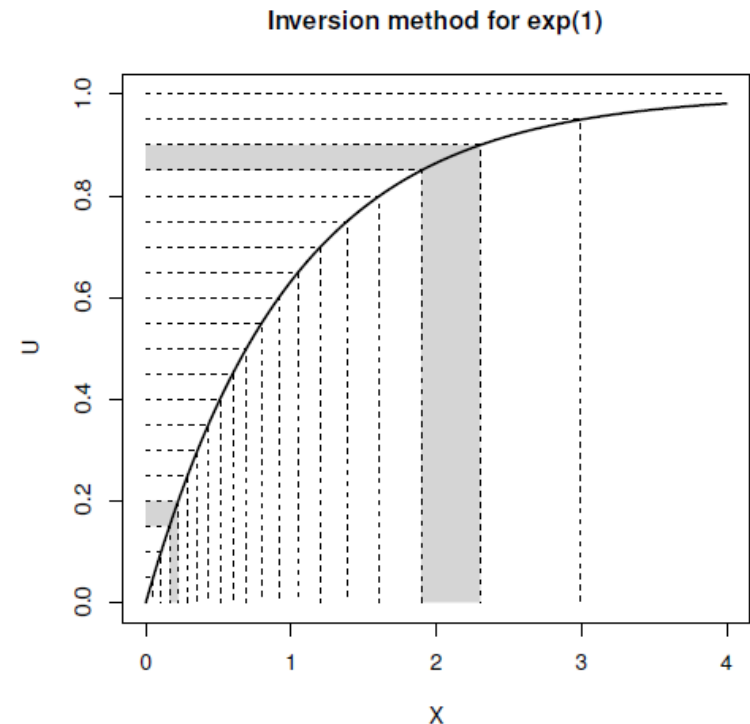
- $y = 1 - e^{-\lambda x} \rightarrow x = \frac{\ln(1-y)}{-\lambda} :$

$$F_X^{-1}(y) = -\frac{\ln(1-y)}{\lambda}$$

- Sea $Y \sim U(0,1)$. Entonces $X = -\frac{\ln(1-Y)}{\lambda}$ se distribuye $\text{Exp}(\lambda)$.

Como $1 - Y$ es también $U(0,1)$, X se puede generar como:

$$X = -\frac{\ln(U)}{\lambda}$$



Método de la transformada inversa para v.a. continuas

Desventajas del método de inversión:

1. Requiere calcular la inversa F^{-1} de la función de distribución, lo cual en muchos casos no es posible analíticamente, requiriendo su solución numérica.

Ejemplo: La distribución *gamma* con parámetros $\lambda, m > 0$ tiene *pdf*:

$$f(x) = \frac{\lambda^m x^{m-1} e^{-\lambda x}}{\Gamma(m)}, \quad x > 0$$

con $\Gamma(m)$ la función gamma. Su *CDF* es:

$$F(x) = \int_0^x \frac{\lambda^m t^{m-1} e^{-\lambda t}}{\Gamma(m)} dt = \frac{\gamma(m, \lambda x)}{\Gamma(m)}$$

con $\gamma(m, \lambda x)$ la función gamma incompleta.

$F^{-1}(x)$ se debe evaluar de forma numérica con algoritmos de búsqueda de raíces.

Método de la transformada inversa para v.a. continuas

Desventaja del método de inversión:

Requiere calcular la inversa F^{-1} de la función de distribución, lo cual en muchos casos no es posible analíticamente, requiriendo su solución numérica lo cual es “time-consuming”.

Ejemplo: La distribución *gamma* con parámetros $\lambda, m > 0$ tiene *pdf*:

$$f(x) = \frac{\lambda^m x^{m-1} e^{-\lambda x}}{\Gamma(m)}, \quad x > 0$$

con $\Gamma(m)$ la función gamma. Su *CDF* es:

$$F(x) = \int_0^x \frac{\lambda^m t^{m-1} e^{-\lambda t}}{\Gamma(m)} dt = \frac{\gamma(m, \lambda x)}{\Gamma(m)}$$

con $\gamma(m, \lambda x)$ la función gamma incompleta.

$F^{-1}(x)$ se debe evaluar numéricamente con algoritmos de búsqueda de raíces.

Método de la transformada inversa para v.a. continuas

2.2 General transformation methods

When a distribution with density f is linked in a relatively simple way to another distribution that is easy to simulate, this relationship can often be exploited to construct an algorithm to simulate variables from f .

Example 2.2. In Example 2.1, we saw how to generate an exponential random variable starting from a uniform. Now we illustrate some of the random variables that can be generated starting from an exponential distribution. If the X_i 's are iid $\mathcal{Exp}(1)$ random variables, then three standard distributions can be derived as

$$(2.1) \quad \begin{aligned} Y &= 2 \sum_{j=1}^{\nu} X_j \sim \chi_{2\nu}^2, & \nu &\in \mathbb{N}^*, \\ Y &= \beta \sum_{j=1}^a X_j \sim \mathcal{G}(a, \beta), & a &\in \mathbb{N}^*, \\ Y &= \frac{\sum_{j=1}^a X_j}{\sum_{j=1}^{a+b} X_j} \sim \mathcal{Be}(a, b), & a, b &\in \mathbb{N}^*, \end{aligned}$$

where $\mathbb{N}^* = \{1, 2, \dots\}$. For example, to generate χ_6^2 random variables, we could use the R code

```
> U=runif(3*10^4)
> U=matrix(data=U,nrow=3) #matrix for sums
> X=-log(U)                #uniform to exponential
> X=2* apply(X,2,sum)      #sum up to get chi squares
```

Método de la transformada inversa para v.a. continuas

Obviously, this is not nearly as efficient as calling `rchisq`, as can be checked by the R code

```
> system.time(test1());system.time(test2())
  user  system elapsed
0.104   0.000   0.107
  user  system elapsed
0.004   0.000   0.004
```

where `test1` corresponds to the R code above and `test2` to its substitution by `X=rchisq(10^4,df=6)`. ◀

Many other derivations of standard distributions are possible when taking advantage of existing probabilistic properties, as shown in Exercise 2.12.

⚡ These transformations are quite simple to use and hence will often be a favorite in our illustrations. However, there are limits to their usefulness, both in the scope of variables that can be generated that way (think, for instance, of a chi-squared distribution with a noneven number of degrees of freedom) and efficiency of generation. For any specific distribution, efficient algorithms have been developed. Thus, if R has a distribution built in, it is almost always worth using, as shown by Example 2.2. Moreover, the transformation method described above cannot reach all distributions; for example, we cannot get a standard normal.

1. Método de la transformada inversa para v.a.'s discretas
2. Método de la transformada inversa para v.a.'s continuas
3. Método del rechazo

Método del rechazo para v.a's continuas

Método del Rechazo: básico

Suponga una v.a. continua con *pdf* f con soporte en $[a, b]$ y acotada por arriba, es decir $f(x) < k$, $\forall x \in [a, b]$ y alguna constante k .

“Lanzamos” dardos uniformemente distribuidos bajo f . La coordenada X del dardo tiene la misma distribución que nuestra v.a.

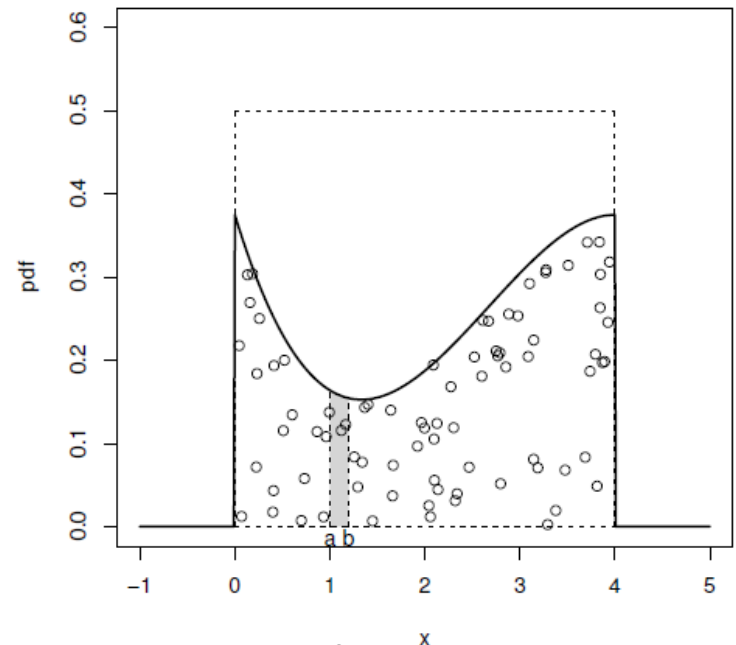


Fig. Puntos uniformemente distribuidos bajo f

Método del rechazo para v.a's continuas

Método del Rechazo: básico

Suponga una v.a. continua con pdf f con soporte en $[a, b]$ y acotada por arriba, es decir $f(x) < k$, $\forall x \in [a, b]$ y alguna constante k .

“Lanzamos” dardos uniformemente distribuidos bajo f . La coordenada X del dardo tiene la misma distribución que nuestra v.a.

Algoritmo:

- 1) Definir k , tal que $f(x) < k$, para todo $x \in [a, b]$
- 2) Generar un valor $U_1 \sim U[a, b]$
- 3) Generar un valor $U_2 \sim U[0, k]$ independiente de U_1
- 4) Entonces:
 - Si $U_2 \leq f(U_1)$, hacer $X = U_1$.
 - Si $U_2 > f(U_1)$, volver al paso 2.

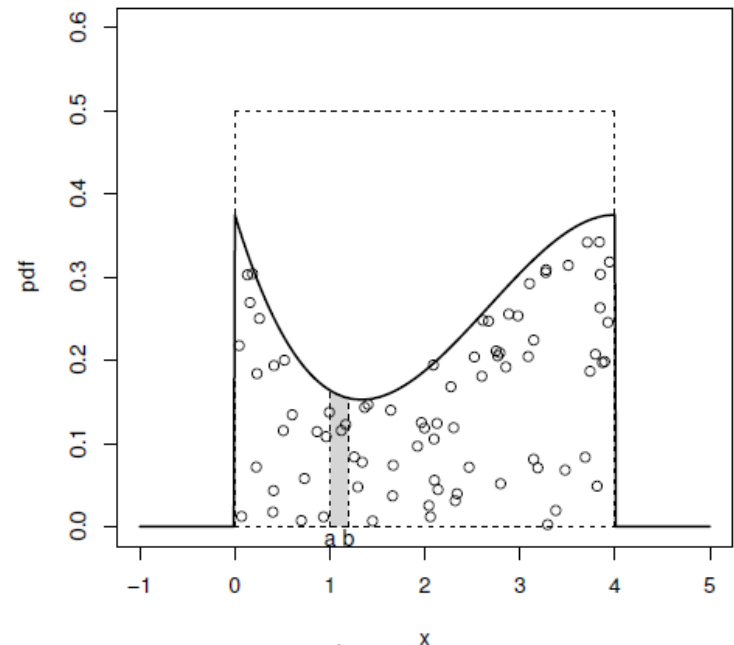


Fig. Puntos uniformemente distribuidos bajo f

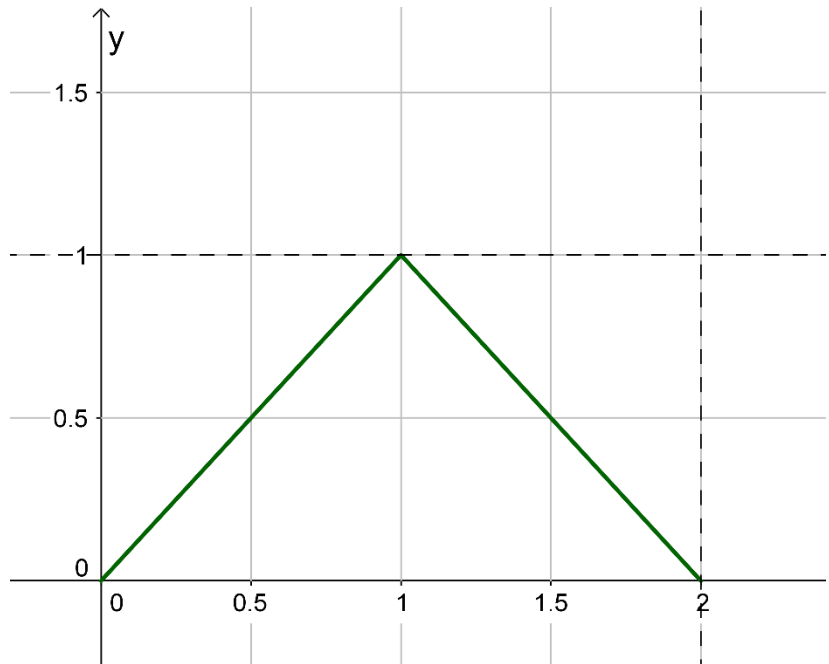
Método del rechazo para v.a's continuas

Ejemplo

Considere la v.a. X con *pdf* triangular definida por:

$$f_X(x) = \begin{cases} x & \text{if } 0 < x < 1; \\ (2 - x) & \text{if } 1 \leq x < 2; \\ 0 & \text{otherwise.} \end{cases}$$

Código: *Rechazo_Triangular.R*



Método del rechazo para v.a's continuas

Código:

```
# program spuRs/resources/scripts/rejecttriangle.r
rejectionK <- function(fx, a, b, K) {
  # simulates from the pdf fx using the rejection algorithm
  # assumes fx is 0 outside [a, b] and bounded by K
  # note that we exit the infinite loop using the return statement
  while (TRUE) {
    x <- runif(1, a, b)
    y <- runif(1, 0, K)
    if (y < fx(x)) return(x)
  }
}

fx<-function(x){
  # triangular density
  if ((0<x) && (x<1)) {
    return(x)
  } else if ((1<x) && (x<2)) {
    return(2-x)
  } else {
    return(0)
  }
}
```

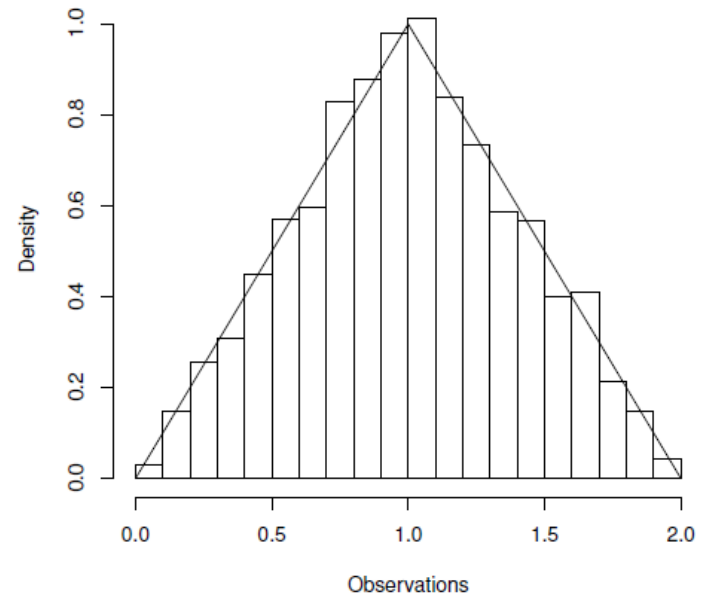
Método del rechazo para v.a's continuas

... continuación código:

```
fx<-function(x){  
  # triangular density  
  if ((0<x) && (x<1)) {  
    return(x)  
  } else if ((1<x) && (x<2)) {  
    return(2-x)  
  } else {  
    return(0)  
  }  
}  
  
# generate a sample  
set.seed(21)  
nreps <- 3000  
Observations <- rep(0, nreps)  
for(i in 1:nreps) {  
  Observations[i] <- rejectionK(fx, 0, 2,  
  1)  
}
```

```
# plot a scaled histogram of the sample  
and the density on top
```

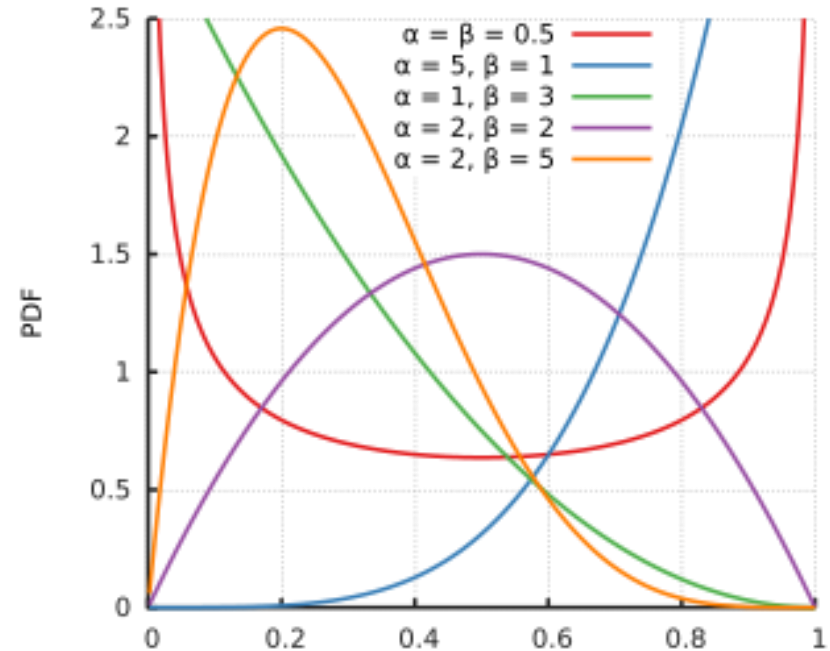
```
hist(Observations, breaks = seq(0, 2,  
by=0.1), freq = FALSE,  
ylim = c(0, 1.05), main = "")  
lines(c(0, 1, 2), c(0, 1, 0))
```



Método del rechazo para v.a's continuas

Ejercicio:

1. Averigüe sobre la distribución *beta* y su pdf.
2. Simule una v.a X distribuida **beta** con parámetros $\alpha = 2, \beta = 5$ utilizando el método del rechazo básico.
3. Compare el histograma generado con la pdf.



Método del rechazo para v.a's continuas

Método del Rechazo: general

- Queremos simular la v.a. X con distribución f .
- Se tiene un método eficiente para generar la v.a. Y con distribución g .
- Se genera Y de g y este valor generado se acepta con probabilidad proporcional a $f(Y)/g(Y)$.

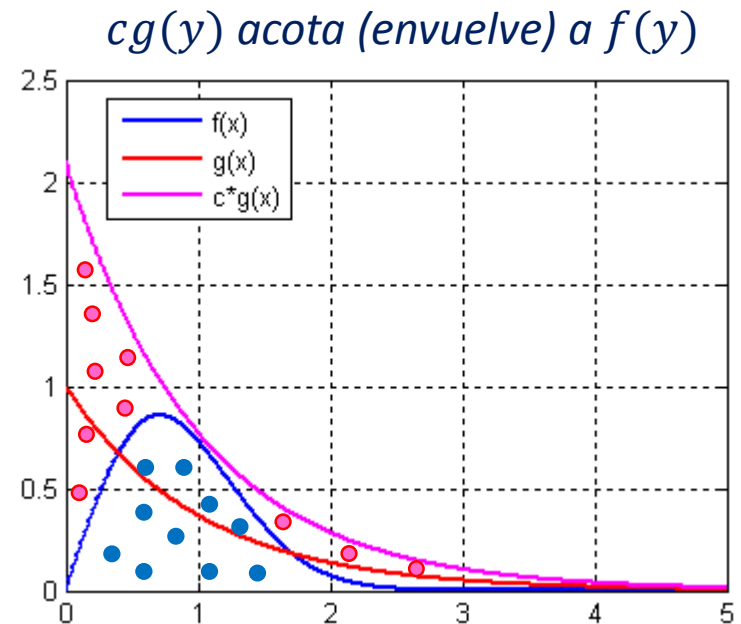
Método del rechazo para v.a's continuas

Método del Rechazo: general

- Queremos simular la v.a. X con distribución f .
- Se tiene un método eficiente para generar la v.a. Y con distribución g .
- Se genera Y de g y este valor generado se acepta con probabilidad proporcional a $f(Y)/g(Y)$.

Hay que hallar $c \geq 1$, una constante tal que:

$$\frac{f(y)}{g(y)} \leq c, \text{ para todo } y$$



Método del rechazo para v.a's continuas

Método del Rechazo: general

- Queremos simular la v.a. X con distribución f .
- Se tiene un método eficiente para generar la v.a. Y con distribución g .
- Se genera Y de g y este valor generado se acepta con probabilidad proporcional a $f(Y)/g(Y)$.

Hay que hallar $c \geq 1$, una constante tal que:

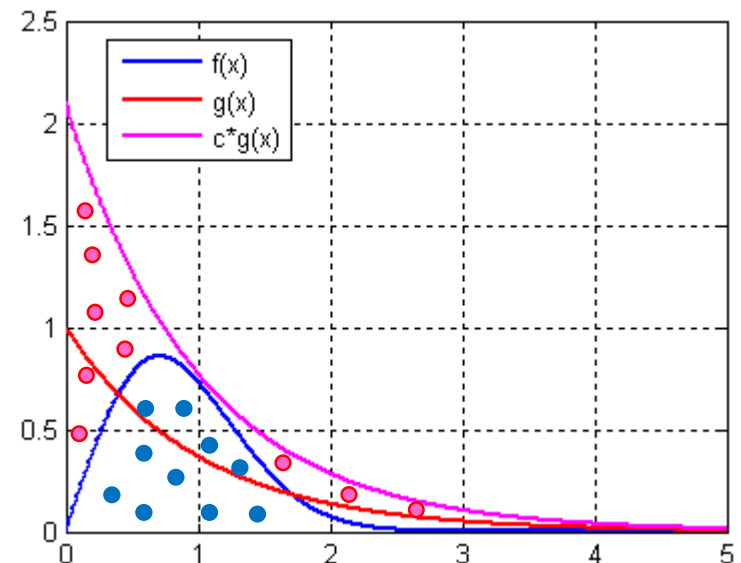
$$\frac{f(y)}{g(y)} \leq c, \text{ para todo } y$$

Algoritmo (en Jones et al, "Introduc. to R")

1. Generar X con densidad g
2. Generar $U \sim U(0, cg(X))$
3. Si $U \leq f(X)$, haga $Z = X$. De otra forma, regrese al paso 1.

Resultado: $Z \sim f$

$cg(y)$ acota (envuelve) a $f(y)$

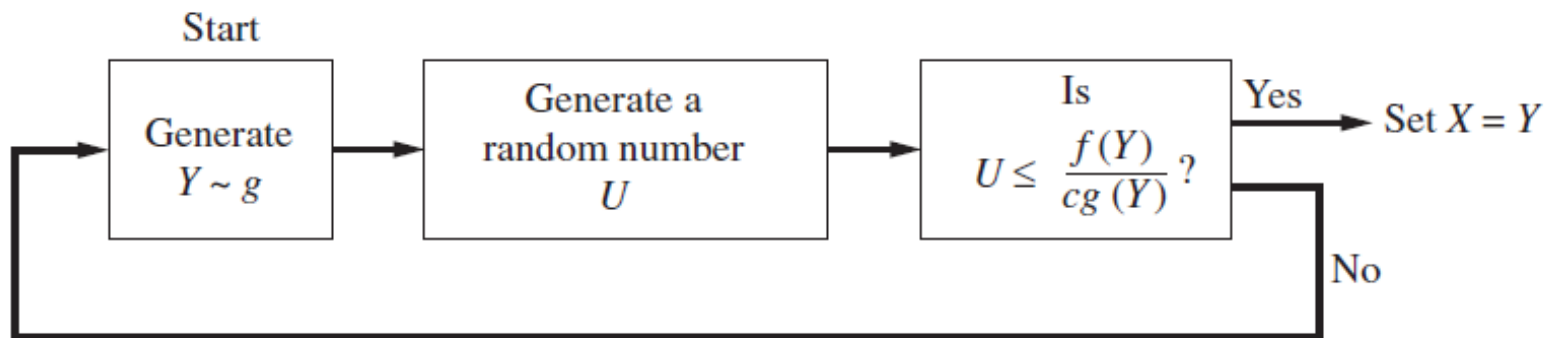


Método del rechazo para v.a's continuas

Método del Rechazo: general

Algoritmo (en Ross, "Simulation")

1. Generar Y con densidad g
2. Generar $U \sim U(0,1)$
3. Si $U \leq \frac{f(Y)}{cg(Y)}$, haga $X = Y$. De otra forma, regrese al paso 1.



Método del rechazo para v.a's continuas

Método del Rechazo: general

Eficiencia del método

- Se mide por el valor esperado del número de veces N necesarios para generar el número aleatorio Z .
- Como el área bajo la curva cg es c y el área bajo f es 1, la probabilidad de aceptar un candidato es $1/c$.
- N es una v.a. geométrica de parámetro $1/c$.
- Su valor esperado es $1/c$.
- Conclusión: Entre más cercano sea c a 1 más eficiente el método.

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

La distribución Gamma con parámetros $\lambda, m > 0$ (denotada $\text{Gamma}(\lambda, m)$) tiene *pdf*:

$$f(x) = \frac{\lambda^m x^{m-1} e^{-\lambda x}}{\Gamma(m)}, \quad x > 0$$

con $\Gamma(m)$ la función Gamma.

Como no hay fórmula explícita para la CDF o su inversa, entonces se simulará con el método del rechazo.

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

Para ilustrar el método se simulará el caso $X \sim \text{Gamma}\left(1, \frac{3}{2}\right)$ con pdf:

$$f(x) = \frac{x^{1/2} e^{-x}}{\sqrt{\pi}/2}, \quad x > 0$$

Su media es $3/2$, entonces para el método del rechazo se tomará una v.a. distribuida *exponencial* con media $3/2$ (tasa= $2/3$). Es decir:

$$g(x) = \frac{2}{3} e^{-2x/3}, \quad x > 0$$

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

Para ilustrar el método se simulará el caso $X \sim \text{Gamma}\left(1, \frac{3}{2}\right)$ con pdf:

$$f(x) = \frac{x^{1/2} e^{-x}}{\sqrt{\pi}/2}, \quad x > 0$$

Su media es $3/2$, entonces para el método del rechazo se tomará una v.a. distribuida *exponencial* con media $3/2$ (tasa= $2/3$). Es decir:

$$g(x) = \frac{2}{3} e^{-2x/3}, \quad x > 0$$

Para determinar la menor constante c tal que $cg(x) \geq f(x) \rightarrow f(x)/g(x) \leq c$, se buscará el máximo valor de:

$$\frac{f(x)}{g(x)} = \frac{3}{\sqrt{\pi}} x^{1/2} e^{-x/3}$$

$$f(x)/g(x)$$

Que se obtiene en el punto x para el cual la derivada de lo anterior se hace 0. El valor es (demostración en tablero):

$$x = 3/2$$

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

El valor máximo de $\frac{f(x)}{g(x)}$ es:

$$\max \frac{f(x)}{g(x)} = \frac{f(3/2)}{g(3/2)} = \frac{3}{\sqrt{\pi}} (3/2)^{1/2} e^{-(3/2)/3} = \frac{3^{3/2}}{\sqrt{2\pi e}} \approx 1.25732$$

Se hace $c = \max \frac{f(x)}{g(x)} = \frac{3^{3/2}}{\sqrt{2\pi e}} \approx 1.25732$, luego:

$$\frac{f(x)}{cg(x)} = \left(\frac{2e}{3}\right)^{1/2} x^{1/2} e^{-x/3}$$

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

El valor máximo de $\frac{f(x)}{g(x)}$ es:

$$\max \frac{f(x)}{g(x)} = \frac{f(3/2)}{g(3/2)} = \frac{3}{\sqrt{\pi}} (3/2)^{1/2} e^{-(3/2)/3} = \frac{3^{3/2}}{\sqrt{2\pi e}}$$

Se hace $c = \max \frac{f(x)}{g(x)} = \frac{3^{3/2}}{\sqrt{2\pi e}} \approx 1.25732$, luego:

$$\frac{f(x)}{cg(x)} = \left(\frac{2e}{3}\right)^{1/2} x^{1/2} e^{-x/3}$$

El algoritmo se lee:

Algoritmo

1. Generar $X \sim \text{Exp}(2/3)$.
2. Generar $Y \sim U(0, c \frac{2}{3} e^{-2X/3})$
3. Si $Y \leq \frac{X^{1/2} e^{-X}}{\sqrt{\pi}/2}$, hacer $Z = X$. De otra forma, regresar al paso 1.

$Z \sim f$

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

Solución para $\text{Gamma}(\lambda, m)$:

Se simula a partir de una exponencial de tasa $\frac{\lambda}{m}$, con densidad:

$$f(x) = \frac{\lambda}{m} e^{-(\lambda/m)x}$$

Se encuentra que el óptimo valor de c es:

$$c = m^m e^{-(m-1)} / \Gamma(m)$$

Código: `gamma.sim.R`

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

Código

```
# program spuRs/resources/scripts/gamma.sim.r

gamma.sim <- function(lambda, m) {
  # sim a gamma(lambda, m) rv using rejection with an exp envelope
  # assumes m > 1 and lambda > 0
  f <- function(x) lambda^m*x^(m-1)*exp(-lambda*x)/gamma(m)
  g <- function(x) lambda/m*exp(-lambda/m*x)
  c <- m^m*exp(1-m)/gamma(m)
  while (TRUE) {
    X <- rexp(1,rate=lambda/m) #X <- -log(runif(1))*m/lambda
    Y <- runif(1, 0, c*g(X))
    if (Y < f(X)) return(X)
  }
}

set.seed(1999)
n <- 10000
g <- rep(0, n)

for (i in 1:n) g[i] <- gamma.sim(1, 2)
hist(g, breaks=20, freq=F, xlab="x", ylab="pdf f(x)",
     main="+theoretical and simulated gamma(1, 2) density")
```

Método del rechazo para v.a's continuas

Ejemplo: Simulación de una v.a. Gamma

Comparación

