

# Simulación

Métodos generales para generación de variables aleatorias.

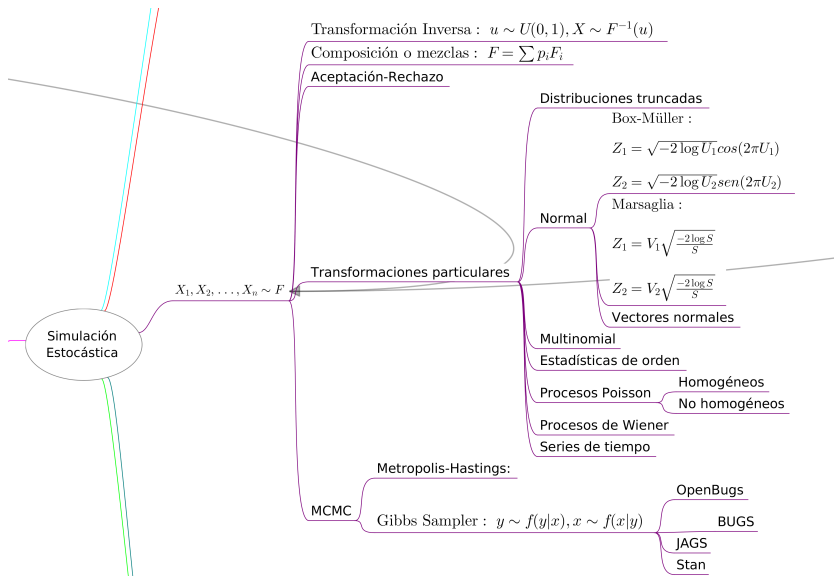
Jorge de la Vega Góngora

Departamento de Estadística,  
Instituto Tecnológico Autónomo de México

Semana 5



# Generación de variables aleatorias



# Métodos generales

# Método de la transformación inversa

El siguiente es uno de los teoremas que se estudia en probabilidad. A continuación se muestra aquí la versión más fácil de demostrar.

## Teorema de la Transformación inversa

Si  $X \sim F$  y  $F$  es *continua y estrictamente creciente*, entonces  $Y = F(X)$  es una variable aleatoria con distribución  $\mathcal{U}(0, 1)$ .

## Demostración.

Sea  $G$  la distribución de  $Y$ . Entonces si  $y \in (0, 1)$ ,

$$\begin{aligned} G(y) &= P(Y \leq y) = P(F(X) \leq y) \\ &= P(X \leq F^{-1}(y)) \\ &= F(F^{-1}(y)) = y \end{aligned}$$

Entonces  $G(y) = y$  en  $(0, 1)$ . Esta es la función de distribución uniforme. □

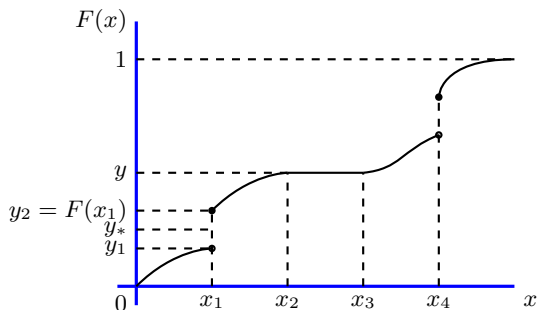
# Teorema de la función cuantíl I

generalización del Teorema integral de probabilidad

- ¿Qué pasa si  $X \sim F$  con  $F$  discontinua o no monótona? El teorema sigue aplicando de la misma manera, pero la prueba es un poco más elaborada.
- Para poder considerar el caso más general, necesitamos una definición más amplia de la inversa de una función. Si  $F$  es estrictamente creciente, la inversa está bien definida si se cumple la condición

$$F^{-1}(y) = x \iff F(x) = y.$$

Sin embargo, si  $F$  es constante en algún intervalo (como en el intervalo  $(x_2, x_3)$  en la figura), entonces la condición no define adecuadamente a la inversa. Necesitamos modificar la definición a la de una inversa generalizada.



# Teorema de la función cuantíl II

generalización del Teorema integral de probabilidad

## Función cuantíl

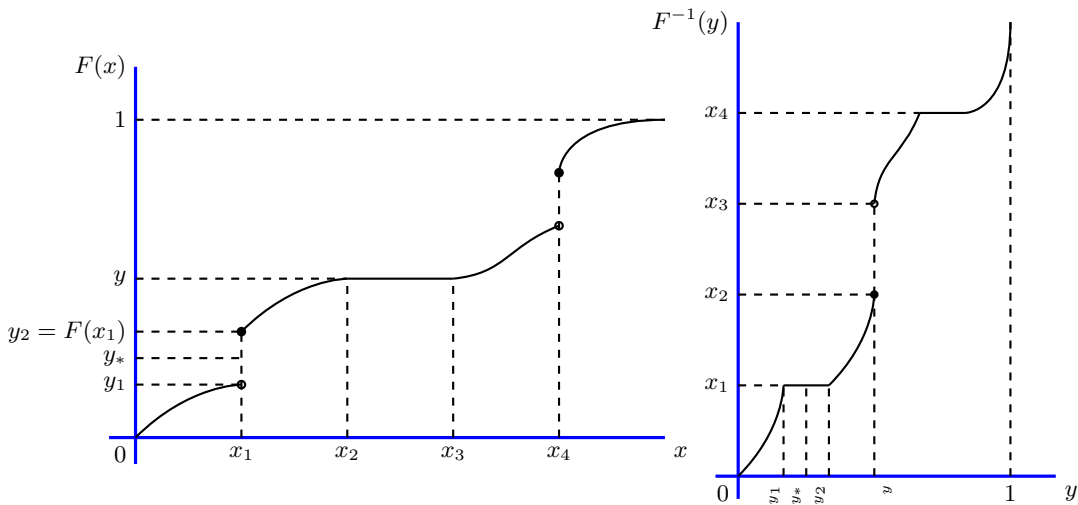
La inversa generalizada de  $F$  (o función cuantíl) se define como

$$F^{-1}(y) = \inf\{x | F(x) \geq y\}.$$

- De acuerdo a esta definición, en la figura anterior la observación  $y$  entonces está bien definida con esta inversa generalizada a ser  $F^{-1}(y) = x_2$ .
- Noten que para cualquier punto  $y_* \in [y_1, y_2]$ , se tiene  $F^{-1}(y_*) = \inf\{x | F(x) \geq y_*\} = x_1$ , ya que  $F$  es continua por la derecha y  $F(y_*)$  es mayor que cualquier  $y_*$  en ese intervalo.
- De la gráfica de la función inversa, es importante notar las siguientes características y convencerse de ellas:
  - La función cuantíl  $F^{-1}$  es no decreciente.
  - Saltos en la función  $F$  se traducen a segmentos planos o de constancia en  $F^{-1}$
  - Regiones planas o de constancia en  $F$  se traducen a saltos en la función  $F^{-1}$
  - $F$  es continua por la derecha, mientras que  $F^{-1}$  es continua por la izquierda.

# Teorema de la función cuantíl III

generalización del Teorema integral de probabilidad



# Teorema de la función cuantíl IV

generalización del Teorema integral de probabilidad

## Teorema de la función cuantíl

Sea  $F$  una función de distribución  $X$ . Si se define  $F^{-1}$  como la función cuantíl  $F^{-1}(y) = \inf\{x | F(x) \geq y\}$ ,  $0 < y < 1$  y  $U \sim \mathcal{U}(0, 1)$ , entonces la variable aleatoria  $X = F^{-1}(U)$  tiene función de distribución  $F$ .



# Teorema de la función cuantíl V

generalización del Teorema integral de probabilidad

## Demostración.

Para cualquier  $x$  tal que  $0 < F(x) < 1$  y cualquier  $u \in (0, 1)$  se cumple la siguiente relación:

$$F(x) \geq u \iff x \geq F^{-1}(u)$$

Para ver esto, supongan que  $x \geq F^{-1}(u) = \inf\{x : F(x) \geq u\}$ . Como  $F$  es no decreciente y continua por la derecha, el conjunto  $A_u = \{x : F(x) \geq u\}$  es un intervalo de la forma  $[x, \infty)$ . Por lo tanto,  $x$  debe satisfacer  $F(x) \geq u$ .

Conversamente, supongamos que  $F(x) \geq u$ . Entonces trivialmente  $x$  satisface la condición del conjunto  $x \geq \inf\{x : F(x) \geq u\}$  que es exactamente la definición de  $F^{-1}(u)$ . Entonces los eventos  $\{\omega | F^{-1}(U(\omega)) \leq x\}$  y  $\{\omega | U(\omega) \leq F(x)\}$  son iguales, y por lo tanto:

$$P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$$



El resultado anterior da un algoritmo que se puede descomponer en dos casos para obtener una variable aleatoria  $X \sim F$  a partir de una muestra de uniformes:

## Algoritmo de inversión

- Caso continuo:
  - 1 Genera  $u \sim \mathcal{U}(0, 1)$
  - 2 Obtener  $X \sim F^{-1}(u)$
- Caso discreto: Si  $x \in \{x_1, \dots, x_n\}$ , el conjunto de puntos de discontinuidad de  $F$  ordenados ( $n$  puede ser  $\infty$ ), entonces la transformación inversa es  $F^{-1}(u) = x_{(i)}$  donde  $F(x_{(i-1)}) < u \leq F(x_{(i)})$ .
  - 1 Genera  $u \sim \mathcal{U}(0, 1)$
  - 2 Tomar  $x = x_{(i)}$ , donde  $F(x_{(i-1)}) < u \leq F(x_{(i)})$ .

La solución del paso 2 puede ser difícil para algunas distribuciones. El capítulo III del libro de [libro de Luc Devroyé](#) desarrolla algunos métodos para implementar el algoritmo.

- a.  $X \sim \exp(\lambda)$ . Entonces  $X$  tiene función de densidad  $f(x) = \frac{1}{\lambda}e^{-x/\lambda}I(x > 0)$ ,  $\lambda > 0$ . La función de distribución es:

$$F(y) = \int_0^y \frac{1}{\lambda}e^{-x/\lambda} dx = \int_0^{y/\lambda} e^{-z} dz = 1 - e^{-y/\lambda}$$

Entonces  $X = F^{-1}(U) = -\lambda \log(1 - U) = -\lambda \log(U)$ . Ahora bien, podemos simplificar sustituyendo  $1 - U$  por  $U$  ya que  $U \sim \mathcal{U}(0, 1) \implies 1 - U \sim \mathcal{U}(0, 1)$ .

b. Si  $X \sim \mathcal{G}(\alpha, \beta)$ , entonces  $X$  tiene densidad  $f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}$ .

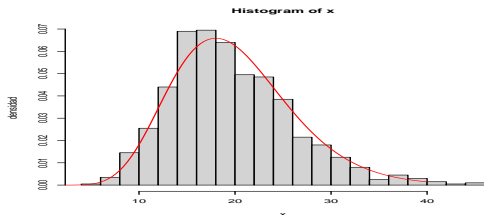
Noten que si  $Y_i \sim \exp\{\lambda\}$  y  $Y_1, \dots, Y_n$  son independientes, entonces  $X = \sum_{i=1}^n Y_i \sim \mathcal{G}(n, \lambda)$ . Esta distribución se conoce como la *distribución de Erlang*. Entonces generamos

$$X = -\lambda \sum_{i=1}^n \log(1 - u_i) = -\lambda \log\left(\prod_{i=1}^n u_i\right)$$

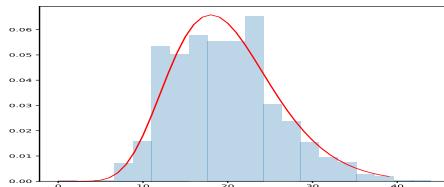
Para generar una muestra de una  $\mathcal{G}(10, 2)$ , usamos los siguientes códigos. Notar que en R se puede definir  $\beta$  como parámetro de escala (`shape`) o como tasa (`rate = 1/shape`). En python se usa `scale`. En el ejercicio se genera una muestra de tamaño  $n = 1000$ .

# Ejemplos II

```
x <- NULL; n <- 1000; alfa <- 10; beta <- 2
for(i in 1:n) x[i] <- -2*sum(log(runif(alfa)))
hist(x, prob = T, breaks=20, ylab = "densidad")
curve(dgamma(x, shape=10, scale = beta), from = 0, to = 40, add = T, col = "red")
```

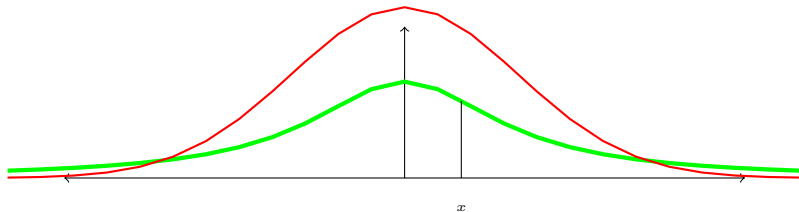


```
n = 1000
alfa = 10
beta = 2
x = np.arange(n)
for i in range(1,n):
    x[i] = -2*sum(np.log(np.random.uniform(size=alfa)))
_ = plt.hist(x,bins=20, alpha=0.3, density=True)
_ = plt.plot(range(0,40), st.gamma.pdf(range(0,40), a=alfa, scale=beta),'r')
plt.show()
```



c. Noten que  $\chi_{(n)}^2 \equiv \mathcal{G}(n/2, 2)$ .

d. **Distribución Cauchy.**  $X \sim Cauchy(0, 1)$  tiene densidad  $f(x) = \frac{c}{\pi(1+x^2)}$ .



Entonces la función de distribución es de la forma:

$$F(x) = \int_{-\infty}^x \frac{c}{\pi(1+v^2)} dv = \frac{2c}{\pi} \int_0^x \frac{1}{1+v^2} dv = \frac{c}{2} + \frac{c}{\pi} \tan^{-1}(x)$$

Entonces podemos tomar  $X = F^{-1}(u) = \tan(\pi(u - \frac{c}{2}))$ .

- e.  $X$  tiene una función de masa de probabilidad dada por:

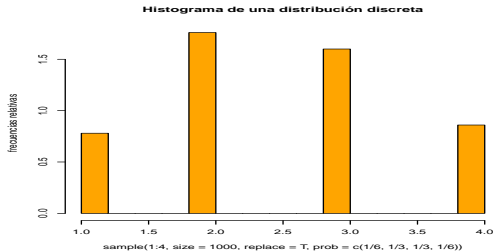
$$\begin{array}{c|cccc} X & 1 & 2 & 3 & 4 \\ \hline P(X=x) & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Entonces generamos  $u \sim \mathcal{U}(0, 1)$  si

$$u \in \begin{cases} [0, 1/6] \implies x = 1 \\ (1/6, 1/2] \implies x = 2 \\ (1/2, 5/6] \implies x = 3 \\ (5/6, 1] \implies x = 4 \end{cases}$$

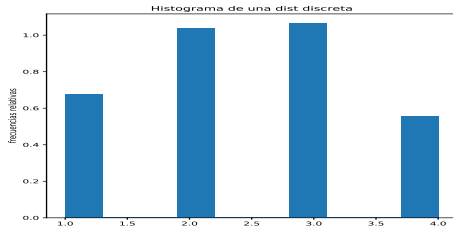
# Ejemplos II

```
hist(sample(1:4, size = 1000, replace = T,  
        prob = c(1/6,1/3,1/3,1/6)), col = 'orange',  
main = "Histograma de una distribución discreta",  
ylab = "frecuencias relativas", prob=T)
```



```
x = np.array([1, 2, 3, 4])  
probs = [1/6,1/3,1/3,1/6]
```

```
discreta = st.rv_discrete(values = (x, probs))  
_ = plt.hist(discreta.rvs(size=1000), density=True)  
plt.title(label='Histograma de una dist discreta')  
Text(0.5, 1.0, 'Histograma de una dist discreta')  
plt.ylabel(ylabel= 'frecuencias relativas')  
Text(0, 0.5, 'frecuencias relativas')  
plt.show()
```





# Ejemplos I

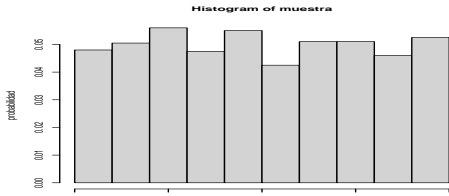
- f. Para una muestra de una uniforme discreta en  $\{1, \dots, n\}$  con  $P(Y = k) = \frac{1}{n}$ , generamos  $U \sim \mathcal{U}(0, 1)$  y define  $Y = [Un] + 1$ , donde  $f(x) = [x]$  es la función parte entera.

En este caso también se puede usar la función de R `sample` para generar la muestra deseada (se puede generar con reemplazo o sin reemplazo). En python se puede usar `numpy.random.choice`

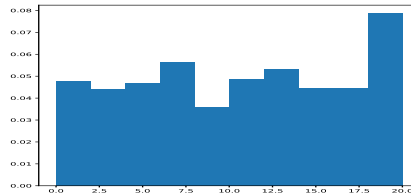
```
n <- 20
muestra <- sample(1:n, 1000, replace=T)
head(muestra)

[1] 15  6 19 19 14  6

hist(muestra, prob = T, ylab = "probabilidad")
```



```
n = 20
_ = plt.hist(np.random.choice(n+1, 1000), density=True)
plt.show()
```



## g. Distribución geométrica.

- La función de masa de probabilidad es  $f(x) = p(1-p)^x$  con  $x = 0, 1, 2, \dots$
- La función de distribución es  $F(x) = 1 - q^{x+1}$  donde  $q = 1 - p$ . Entonces, con el método de la transformación inversa necesitamos resolver para  $x$ :

$$1 - q^x < u \leq 1 - q^{x+1}$$

- Esta desigualdad se simplifica a  $x < \log(1-u)/\log(q) \leq x+1$ . La solución es  $x+1 = \lceil \frac{\log(1-u)}{\log(q)} \rceil$ <sup>1</sup>. Por ejemplo, con  $p = 1/4$

```
n <- 1000
p <- 0.25
u <- runif(n)
k <- ceiling(log(1-u)/log(1-p))-1 #se puede simplificar a floor(log(u)/log(1-p))
l <- floor(log(1-u)/log(1-p))
k[1:20]
[1] 0 1 1 1 0 0 6 0 6 1 14 0 0 3 8 1 8 2 1 4
l[1:20]
[1] 0 1 1 1 0 0 6 0 6 1 14 0 0 3 8 1 8 2 1 4
```

<sup>1</sup> $\lceil t \rceil$  es la función techo (el menor entero mayor o igual que  $t$ ). La función piso  $\lfloor t \rfloor$  devuelve el mayor entero que es menor o igual a  $t$ . Así,  $\lceil 3.5 \rceil = 4$  y  $\lfloor 3.5 \rfloor = 3$ .

## h. Distribución Poisson.

- La función de masa de probabilidad Poisson es de la forma  $f(x) = e^{-\lambda} \frac{\lambda^x}{x!}$ ,  $x = 0, 1, 2, 3, \dots$
- Para la distribución Poisson, la aplicación del método anterior es más complicado, porque no se tiene una fórmula explícita para el valor de  $x$  tal que  $F(x) < u \leq F(x+1)$ , ya que  $F$  es una suma de valores.
- El método para generar Poisson usa la fórmula recursiva:

$$f(x+1) = \frac{\lambda}{x+1} f(x), \quad F(x+1) = F(x) + f(x+1)$$

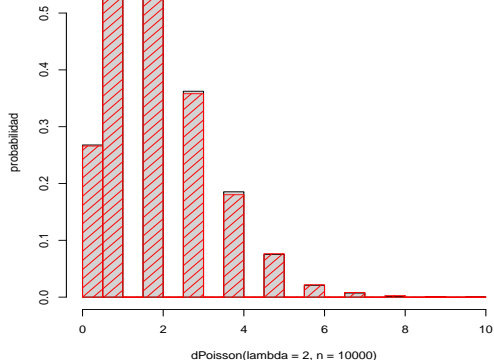
- Entonces, para generar una variable aleatoria Poisson, se genera  $u \sim \mathcal{U}(0, 1)$  y se busca en una tabla o vector construido con los valores de  $F(x)$  para encontrar el valor de  $x$  que es solución de  $F(x-1) < u \leq F(x)$  (Método de búsqueda).

# Ejemplos II

```
hist(dPoisson(lambda = 2, n = 10000), prob = T, ylab = "probabilidad")
hist(rpois(10000, lambda = 2), col = "red", density = 10,
     add = T, prob = T)
```

```
dPoisson <- function(lambda,n){
  # Función recursiva para generar muestras de la Poisson(lambda)
  # Primero se requiere generar un vector con los
  # valores de la distribución Poisson
  f <- Fn <- NULL
  f[1] <- exp(-lambda) #densidad en 0
  Fn[1] <- f[1] #distribución en 0
  #General el vector de valores de la distribución
  for (i in 2:(n+1)){
    f[i] <- lambda*f[i-1]/(i-1)
    Fn[i] <- Fn[i-1] + f[i]
  }
  u <- runif(n) # muestra de uniformes
  x <- NULL
  for(i in 1:n)x <- append(x,sum(Fn < u[i]))
  return(x)
}
```

Histogram of dPoisson(lambda = 2, n = 10000)



- 1 No siempre es fácil calcular  $F^{-1}$  en forma cerrada (e.g. distribución normal). En algunos casos se pueden aplicar métodos numéricos y encontrar soluciones aproximadas.
- 2 En el caso de las distribuciones discretas: en papel, los métodos son intuitivos, pero computacionalmente hay que usar algunos métodos de búsqueda, evaluación de funciones, etc. que pueden ser lentos. La eficiencia es importante para un buen método.
- 3 En relación a la eficiencia, hay que aplicar la siguiente regla: cuando hay más de un método, hay que usar el que sea menos costoso (computacionalmente).

## Ejemplo

Supongan que  $X_1 \sim F_1$  y  $X_2 \sim F_2$ . Entonces sabemos por estadísticas de orden que:

$$\begin{aligned}X_{(2)} &= \text{máx}\{X_1, X_2\} \sim F_1(x)F_2(x) \quad \text{y} \\X_{(1)} &= \text{mín}\{X_1, X_2\} \sim F_1(x) + F_2(x) - F_1(x)F_2(x)\end{aligned}$$

Por otra parte, consideren la distribución  $F(x) = x^2 I_{[0,1]}(x) = F_1(x)F_2(x)$  donde  $x \sim \mathcal{U}(0, 1)$ .

- con el método de inversión,  $X = \sqrt{u}$ .
- con el método usando el máximo,  $X = \text{máx}\{u_1, u_2\}$ .
- Ambos métodos son equivalentes en resultado, pero ¿Cuál es el método más rápido? Depende de la velocidad del generador uniforme, y de las funciones  $\sqrt{\cdot}$  y  $\text{máx}\{\cdot\}$ . Y en un caso se usa una  $u$  y en el otro, dos  $u$ 's.

Ejercicio: estimen 10 millones de números de esta manera y tomen los tiempos de ejecución.

# Método de composición o mezclas I

- A veces  $f$  o  $F$  se pueden escribir como una combinación lineal convexa de dos o más cdf's:

$$F = \sum_{i=1}^n p_i F_i, \quad p_i > 0, \quad \sum_{i=1}^n p_i = 1$$

- En aplicaciones financieras, de estudios de mercado y de problemas de clasificación, las mezclas son comunes.
- El problema a resolver es cómo generar observaciones de una mezcla de distribuciones.

## Muestreo por composición

Método de muestreo:

- 1 Muestrear un valor  $j$  tal que  $P(J = j) = p_j$ .
- 2 Obtener  $X$  con distribución  $F_j$ .

## Demostración.

$X$  tiene distribución:

$$F(x) = P(X \leq x) = \sum_{j=1}^n P(X \leq x | J = j) P(J = j) = \sum_{j=1}^n F_j(x) p_j$$



- Una variable aleatoria  $X$  es una mezcla continua si la distribución de  $X$  es de la forma

$$F_X(x) = \int_{-\infty}^{\infty} F_{X|Y=y}(x) f_Y(y) dy$$

para una familia  $X|Y = y$  indexada por los números reales  $y$  y función de peso  $f_Y(y)$  que es una densidad de probabilidad.



# Ejemplos

- a. Doble exponencial. La densidad es  $f(x) = \frac{1}{2}e^{-|x|}$ . Noten que  $f$  se puede escribir como

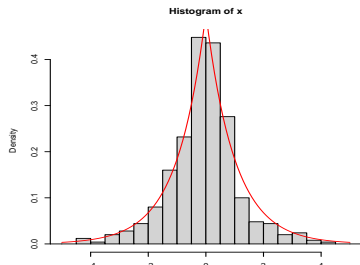
$$f(x) = \frac{1}{2}e^x I(x \leq 0) + \frac{1}{2}e^{-x} I(x > 0)$$

Entonces el algoritmo es:

- Generar  $u_1, u_2 \sim \mathcal{U}(0, 1)$
- Si  $u_1 \leq \frac{1}{2}$ , hacer  $X = \log(u_2)$ , en otro caso  $X = -\log(u_2)$ .

```
hist(x, xlim = c(-5,5), prob = T, breaks = 20)  
curve(0.5*exp(ifelse(x<0,1,-1)*x), from = -5, to = 5, add = T,col = "red")
```

```
u <- matrix(runif(1000),ncol=2)  
x <- ifelse(u[,1]<0.5,-1,1)*log(u[,2])
```

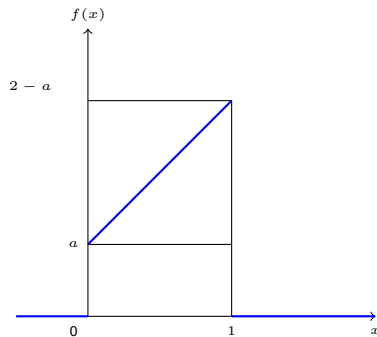


## b. Triangular o trapezoidal.

$f(x) = [a + 2(1 - a)x]I_{(0,1)}(x)$  para  $0 < a < 1$ .  $f$  se puede escribir como

$$f(x) = aI_{(0,1)}(x) + (1 - a)2xI_{(0,1)}(x) = af_1(x) + (1 - a)f_2(x)$$

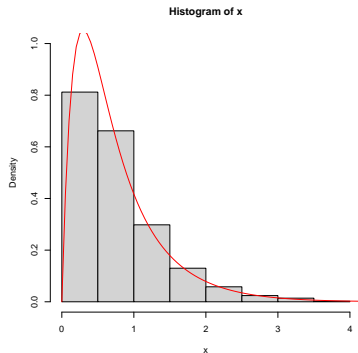
donde  $f_1(x)$  es una uniforme, y  $f_2(x)$  corresponde a  $\sqrt{U}$  o  $X = \max\{U_1, U_2\}$ .



- c. Mezcla de distribuciones gamma. Sean  $X_1 \sim \mathcal{G}(2, 2)$  y  $X_2 \sim \mathcal{G}(2, 4)$  independientes. Obtener una muestra de la mezcla  $F = 0.5F_{X_1} + 0.5F_{X_2}$ .

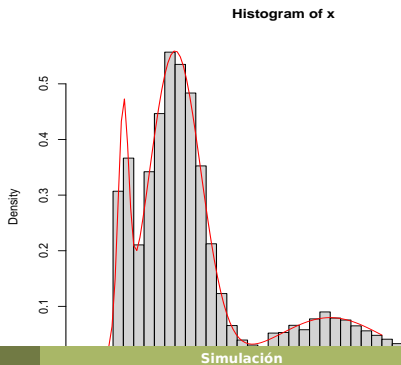
```
hist(x, prob = T, ylim=c(0,1))  
curve(0.5*dgamma(x,2,2)+0.5*dgamma(x,2,4),from=0,to=5,add=T,col="red")
```

```
n <- 1000  
x1 <- rgamma(n,2,2)  
x2 <- rgamma(n,2,4)  
u <- runif(n)  
k <- as.integer(u > 0.5)  
x <- k*x1 + (1-k)*x2
```



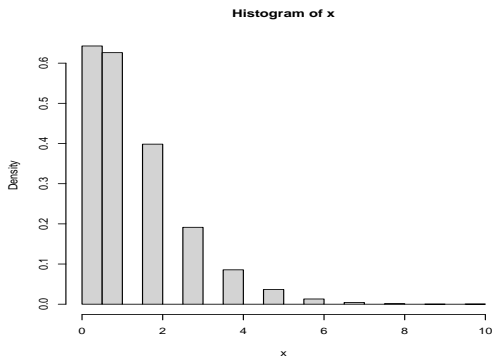
- d. Mezcla de distribuciones normales. Sean  $X_1 \sim \mathcal{N}(1, 0.1^2)$  y  $X_2 \sim \mathcal{N}(2, 0.5^2)$  y  $X_3 \sim \mathcal{N}(5, 1)$  independientes. Obtener una muestra de tamaño  $n = 10,000$  de la mezcla  $F = 0.1F_{X_1} + 0.7F_{X_2} + 0.2F_{X_3}$ .

```
n <- 10000
y <- sample(1:3, size = n, prob = c(0.1, 0.7, 0.2), replace = T)
x <- rnorm(n, mean = ifelse(y == 1, 1, ifelse(y == 2, 2, 5)),
          sd = ifelse(y == 1, 0.01, ifelse(y == 2, 0.5, 1)))
hist(x, prob = T, breaks = 50)
curve(0.1*dnorm(x, 1, sd = 0.1) + 0.7*dnorm(x, 2, sd = 0.5) + 0.2*dnorm(x, 5, sd = 1),
      from = -0, to = 6, add = T, col = "red")
```



- e. Mezcla Poisson-Gamma. En este caso, los pesos son la distribución gamma y por lo tanto la mezcla es continua: La distribución binomial negativa es una mezcla de distribuciones  $x|\lambda \sim \mathcal{P}(\lambda)$ , con  $\lambda \sim \mathcal{G}(k, \beta)$ . Entonces  $X \sim Nbin(k, p)$  donde  $p = \frac{\beta}{1+\beta}$

```
n <- 10000; k <- 4; beta <- 3
lambda <- rgamma(n, k, beta) #genera una lambda
x <- rpois(n, lambda) # genera una poisson
hist(x, prob = T)
```



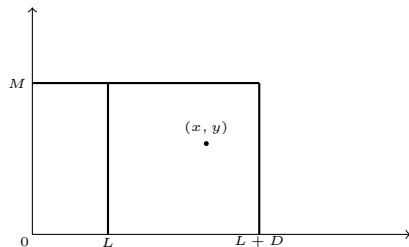
```
points(unifm(n), rlnorm(n, mean(n), beta/(4*beta)), col = "red", lwd = 2)
```

- Grandes  $p_j$ 's deben asociarse con métodos rápidos para producir una v.a.  $X \sim F_j$ . Y conversamente, si se usa un método lento para casos con  $p_j$ 's pequeñas, no habrá mucha pérdida de eficiencia.
- Las particiones de las clases o componentes no tienen porqué ser simétricas.
- Los modelos de mezclas, junto con algunos modelos bayesianos jerárquicos y modelos de cadenas de Markov son ejemplos de modelos con estructura jerárquica: hay varios niveles de aleatoriedad y la distribución de probabilidad de variables aleatorias en modelos superiores depende de los valores de variables aleatorias en los niveles inferiores. En estos casos, el proceso de simulación se realiza en pasos, siguiendo la estructura del modelo.

# Método de aceptación y rechazo (Von Neumann, 1951)

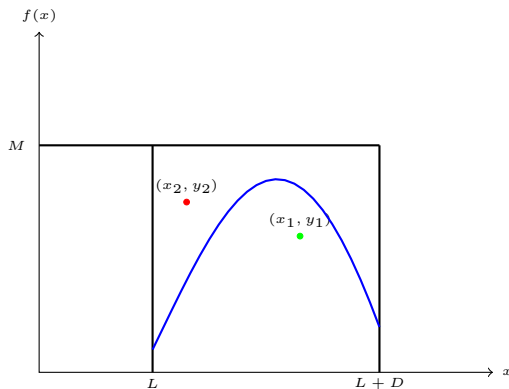
- Supongan que  $u_1, u_2$  son va iid  $\mathcal{U}(0, 1)$ .
- Definan:  $X = L + Du_1$ ,  $Y = Mu_2$ .
- Entonces  $(X, Y)$  es uniforme sobre el rectángulo  $[L, L + D] \times [0, M]$ :

$$\begin{aligned}P(X \leq x, Y \leq y) &= P(X \leq x)P(Y \leq y) \\&= P(L + DU_1 \leq x)P(MU_2 \leq y) \\&= P(U_1 \leq \frac{x - L}{D})P(U_2 \leq \frac{y}{M}) \\&= \frac{x - L}{D} \cdot \frac{y}{M}\end{aligned}$$



# Método de aceptación y rechazo (Von Neumann, 1951)

- Para una v.a. con soporte  $[L, L + D]$  y densidad acotada (mayorizada) por  $M$ , se puede utilizar una técnica de rechazo como sigue:
  1. Obtener  $(X, Y)$  como arriba
  2. Si  $Y \leq f(X)$ , usa  $X$ .
  3. Si  $Y > f(X)$ , rechaza  $(X, Y)$  y regresar a 1.





- La eficiencia del método depende de:
  - (a) la facilidad para hacer la comparación
  - (b) la proporción de rechazo del método, esto es:

$$P(\text{rechazo}) = 1 - \frac{\text{área de } f(x)}{\text{área del rectángulo}}$$

- Entonces, mientras más se ciña el rectángulo a la función, obtendremos mayor eficiencia.

# Método de aceptación-rechazo mejorado

- Supongan que la densidad objetivo a muestrear es  $f$ .
- En lugar de usar un rectángulo uniforme  $[L, L + D] \times [0, M]$ , se puede usar como “envoltura” una distribución  $g$  de la cual sea fácil muestrear.
- Podemos suponer que podemos encontrar una constante  $c$  tal que  $cg(x) \geq f(x) \quad \forall x$ . En este caso se dice que  $g$  *mayoriza* a  $f$ .
- Entonces el algoritmo mejorado queda como:

## Algoritmo mejorado de aceptación y rechazo

- 1 Muestrear  $Y \sim g$ ,
- 2 Muestrear  $u \sim \mathcal{U}(0, 1)$ .
- 3 Aceptar  $X = Y$  si  $ucg(x) \leq f(x) \iff u \leq \frac{f(x)}{cg(x)}$ , si no, rechaza  $Y$  y repite 1.

# Prueba del método de aceptación-rechazo

## Demostración.

Sea  $A = \left\{ \omega \mid U(\omega) \leq \frac{f(X(\omega))}{cg(X(\omega))} \right\}$  el evento que tiene los casos en donde se acepta la observación. Queremos probar que una variable aleatoria aceptada tiene la distribución objetivo cuando se da el evento  $A$ .

$$\begin{aligned} P(Y \leq x | A) &= \frac{P(Y \leq x, A)}{P(A)} \\ &= \frac{P(Y \leq x, U \leq \frac{f(Y)}{cg(Y)})}{P(U \leq \frac{f(Y)}{cg(Y)})} \\ &= \frac{\int_{-\infty}^x \int_0^{f(y)/cg(y)} du g(y) dy}{\int_{-\infty}^{\infty} \int_0^{f(y)/cg(y)} du g(y) dy} \\ &= \frac{\int_{-\infty}^x \frac{f(y)}{cg(y)} g(y) dy}{\int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y) dy} \\ &= \frac{\int_{-\infty}^x f(y) dy}{\int_{-\infty}^{\infty} f(y) dy} = \int_{-\infty}^x f(y) dy = P(X \leq x) = F_x(x) \end{aligned}$$



# Método de aceptación-rechazo mejorado

- En este caso, la probabilidad de aceptación es  $1/c$ .

## Demostración.

Considerar la probabilidad de que una iteración produce un valor aceptable:

$$\begin{aligned}P(\text{aceptación} | Y = y) &= P(Y = y, U \leq \frac{f(y)}{cg(y)}) \\&= P(Y = y)P(U \leq \frac{f(y)}{cg(y)} | Y = y) \\&= g(y) \frac{f(y)}{cg(y)} = \frac{f(y)}{c}\end{aligned}$$

Integrando sobre los posibles valores de  $y$ , obtenemos que  $P(\text{aceptación}) = 1/c$ . □

- El número de iteraciones necesarias para una aceptación en el algoritmo es una v.a.  $W \sim \text{geo}(1/c)$ . Por lo tanto el número promedio de iteraciones es  $c = E(W)$ .

- a. Generar una muestra de  $x \sim f(x) = 20x(1-x)^3 I_{(0,1)}(x) \equiv \mathcal{Be}(2, 4)$ . Usemos como función mayorizante un rectángulo  $g(x) \equiv \mathcal{U}(0, 1)$ . Para escoger la constante  $c$  mínima tal que  $\frac{f(x)}{g(x)} \leq c$ , usemos cálculo.

$$\frac{f(x)}{g(x)} = 20x(1-x)^3 = h(x)$$

$h'(x) = 20(1-x)^3 - 60x(1-x)^2 = 0 \implies x = 1/4$ . Entonces  $c = \frac{20}{4}(1 - 1/4)^3 = 2.109375$  y  $\frac{f(x)}{g(x)} \leq 2.109375$ . El algoritmo es entonces:

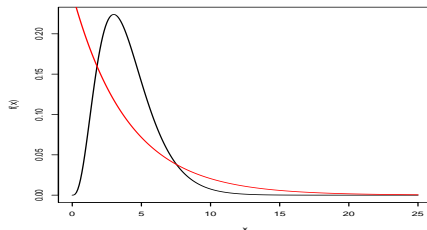
- Genera  $u_1, u_2 \sim \mathcal{U}(0, 1)$ .
- si  $u_2 \leq 9.48u_1(1-u_1)^3 \implies x = u_1$ , si no, regresa.

El número de veces que se realiza el paso 1 es  $c = 135/64 \approx 2.11$ .

## b. Envoltura exponencial para una Gamma

- Consideren muestrear de una distribución  $f(x) \sim \mathcal{G}(\alpha, 1)$ , usando una exponencial como función mayorizante.
- Usamos una  $\exp(\alpha)$  para envolver, con densidad  $g(y) = \frac{1}{\alpha} e^{-y/\alpha}$ . Tenemos que encontrar el valor de  $c$  que optimiza la  $g$  para cubrir a la  $f$ .

```
alfa <- 4
f <- function(x){dgamma(x,alfa,1)}
g <- function(x){dexp(x,1/alfa)}
x <- seq(0,25,0.1) #sucesión de valores de x
plot(x,f(x),type="l",lwd=2)
lines(x,g(x),col="red",lwd=2)
```



- Noten que  $h(x) = \frac{f(x)}{g(x)} = \alpha\Gamma(\alpha) \times \frac{x^{\alpha-1}e^{-x}}{e^{-x/\alpha}} = \alpha\Gamma(\alpha)x^{\alpha-1}e^{\frac{1-\alpha}{\alpha}x}$ . Necesitamos optimizar  $h(x)$ . Se puede hacer a mano o con la función `optimize` en R.

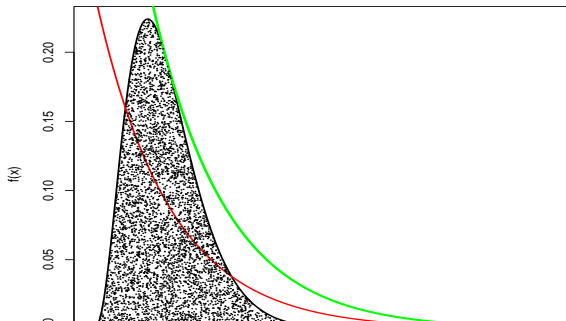
```
v <- optimize(f = function(x){f(x)/g(x)}, interval = c(0,25),
              maximum = T)
v
$maximum
[1] 3.999982

$objective
[1] 2.124248
c <- v$objective
```

- La siguiente gráfica muestra los puntos que caen debajo del área relevante, y la proporción del total de puntos generados.

# Ejemplos III

```
n <- 10000 #puntos a generar
plot(x,f(x),type="l",lwd=2)
lines(x, c*g(x), col="green", lwd=3) #exponencial mayorizada
lines(x, g(x), col="red",lwd=2) #exponencial original
u1 <- runif(n) #en principio u1 y u2 son independientes.
u2 <- runif(n)
y <- -alfa*log(1-u2)
indicadora <- u1 <= f(y)/(c*g(y)) #selecciona los puntos que se aceptan
w <- y[indicadora]
points(w,u1[indicadora]*c*g(w),pch=16,ce=0.3)
```



```
table(indicadora)/n
```

```
indicadora
FALSE TRUE
0.5248 0.4752
```

```
1/c #puntos aceptados
```

```
[1] 0.4707548
```



- c. Muestras de la distribución media-normal. Esta distribución tiene densidad:

$$f(x) = \frac{2}{\sqrt{2\pi}} \exp(-x^2/2) I(x \geq 0)$$

La función cobertura puede ser la distribución exponencial con parámetro  $1/\lambda$ . Para encontrar la constante  $c$  tal que  $f(x) \leq cg(x)$ , tenemos que

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

La función cuadrática dentro de la exponencial alcanza su máximo en  $x = \lambda$ . Entonces tenemos que  $f(x)/g(x) \leq c^*$ , donde

$$c^* = \frac{2}{\sqrt{2\pi}\lambda} e^{-\lambda^2/2+\lambda^2} = \sqrt{\frac{2}{2\pi\lambda^2}} e^{\lambda^2/2}$$

- La distribución de probabilidad de una suma de dos o más variables aleatorias  $S = X_1 + \cdots + X_n$  se llama una *convolución* de las distribuciones de las variables originales y se denota por:

$$f_1 * f_2 * \cdots * f_n$$

- El método de convolución se refiere a sumar variables aleatorias para obtener una nueva variable aleatoria con la distribución deseada.
- Lo que es importante entonces no es la función de distribución de la variable deseada, sino su relación con otras variables que se generan con mayor facilidad.

- 1 Distribución de Erlang. Esta distribución es un caso particular de la distribución  $\text{Gamma}(\alpha, \beta)$ , en donde el parámetro  $\alpha$  es natural. Hemos visto que una suma de exponenciales es Erlang:

$$\sum_{i=1}^n Y_i \sim \mathcal{G}(n, \beta)$$

donde  $Y_i \sim \exp(\beta)$ . Entonces la distribución de Erlang es una convolución.

- 2 Distribución Binomial. Sabemos que la suma de  $n$  Bernoullis con parámetro  $p$  independientes, son una binomial con parámetros  $n$  y  $p$ .
- 3 Si  $Y_i \sim \chi^2_{(1)}$  y  $Y_1, \dots, Y_n$  son iid, entonces  $\sum_{i=1}^n Y_i \sim \chi^2_{(n)}$ .
- 4  $X_1, \dots, X_n \sim \mathcal{G}(\alpha_i, \beta)$  e independientes  $\Rightarrow \sum_{i=1}^n X_i \sim \mathcal{G}(\sum_{i=1}^n \alpha_i, \beta)$
- 5 Suma de normales es normal.
- 6 La distribución Binomial Negativa  $\text{NegBin}(k, p)$  es la convolución de  $k$  iid geométricas con parámetro  $p$ .

- Si  $F$  es una función de distribución con densidad  $f$ , la densidad truncada en  $(a, b)$  es:

$$f^*(x) = \frac{f(x)}{F(b) - F(a)} I_{(a,b)}(x)$$

con distribución truncada

$$F^*(x) = \frac{F(x) - F(a)}{F(b) - F(a)} I_{(a,b)}(x) + I_{\{x \geq b\}}(x)$$

Entonces para generar una muestra de  $F^*$ , tenemos que recorrer la muestra de números uniformes al intervalo de interés:

- 1 Genera  $u \sim \mathcal{U}(0, 1)$ .
- 2 Define  $\nu = F(a) + (F(b) - F(a))u$ . Entonces  $\nu \sim \mathcal{U}(F(a), F(b))$ .
- 3 Define  $x = F^{-1}(\nu)$ .

# Generación de estadísticas de orden I

Para generar una estadística de orden  $X_{(i)}$  de una distribución  $F$ , podemos usar propiedades de la distribución beta:

## Algoritmo generación de estadísticas de orden

Sea  $X_1, \dots, X_n \sim F$  independientes.

- Si  $n$  es pequeña, hacer  $Y_i = X_{(i)}$  (ordena directamente la muestra).
- Si  $n$  es grande,
  - 1 genera  $\nu \sim Be(i, n - i + 1)$
  - 2 define  $X = F^{-1}(\nu)$

Recordar que si  $X \sim F$  entonces la  $i$ -ésima estadística de orden tiene la distribución dada por:

$$F^*(x_{(i)}) = \sum_{k=i}^n \binom{n}{k} F(x_{(i)})^k [1 - F(x_{(i)})]^{n-k}$$

Para la variable aleatoria  $X$  generada con el algoritmo indicado tenemos que:

$$\begin{aligned} F(x) &= P(X \leq x) = P(F^{-1}(\nu) \leq x) \\ &= P(\nu \leq F(x)) \\ &= \int_0^{F(x)} \frac{1}{B(i, n+i-1)} \nu^{i-1} (1-\nu)^{n-i+1-1} d\nu = * \end{aligned}$$

Noten que

$$B(i, n-i+1) = \frac{\Gamma(i)\Gamma(n-i+1)}{\Gamma(i+(n-i+1))} = \frac{\Gamma(i)\Gamma(n-i+1)}{\Gamma(n+1)} = \frac{(i-1)!(n-i)!}{n!}.$$

Entonces continuando con el desarrollo e integrando por partes:

$$\begin{aligned} * &= \int_0^{F(x)} \frac{n!}{(i-1)!(n-i)!} \nu^{i-1} (1-\nu)^{n-i} d\nu \\ &= \int_0^{F(x)} \binom{n}{i} i \nu^{i-1} (1-\nu)^{n-i} d\nu \\ &= \binom{n}{i} \int_0^{F(x)} i \nu^{i-1} (1-\nu)^{n-i} d\nu \\ &= \binom{n}{i} \left[ F(x)^i (1-F(x))^{n-i} + \int_0^{F(x)} \nu^i (n-i) (1-\nu)^{n-i-1} d\nu \right] \\ &= \dots \text{continuar usando la recurrencia.} \\ &= \sum_{k=i}^n \binom{n}{k} F(x)^k [1-F(x)]^{n-k} \end{aligned}$$