

Modelos y Simulación

Licenciatura en Ciencias de la Computación

Licenciatura en Matemática Aplicada

Dra. Patricia Kisbye

2024

Índice general

1. Conceptos básicos de probabilidad	7
1.1. Espacio de probabilidad	7
1.1.1. Espacio muestral	7
1.1.2. Axiomas de probabilidad	8
1.1.3. Probabilidad condicional	9
1.2. Variables aleatorias	11
1.2.1. Distribución conjunta	13
1.2.2. Distribución condicional	14
1.2.3. Convolución de distribuciones	16
1.2.4. Valor esperado	17
1.2.5. Varianza	18
1.2.6. Desigualdad de Chebyshev	19
1.2.7. Leyes de los grandes números	20
1.3. Distribuciones de probabilidad	20
1.3.1. Variables aleatorias discretas	21
1.3.2. Variables aleatorias continuas	24
2. Procesos de Poisson	31
2.1. El proceso de Poisson homogéneo	31
2.1.1. Distribución del número de llegadas $N(t)$	32
2.1.2. Proceso de Poisson homogéneo trasladado	33
2.1.3. Distribución del tiempo entre arribos	33
2.1.4. Distribución del tiempo de arribo	34
2.1.5. Superposición de procesos de Poisson homogéneos	35
2.1.6. Refinamiento de procesos de Poisson homogéneos	37
2.2. Procesos de Poisson no homogéneos	38
3. Números aleatorios	35
3.1. Introducción	35
3.2. Propiedades de un generador de números aleatorios	35

3.2.1.	Breve reseña histórica	36
3.3.	Principios generales	37
3.4.	Generadores congruenciales	38
3.4.1.	Generadores congruenciales lineales	39
3.4.2.	Generadores mixtos	40
3.4.3.	Generadores multiplicativos	41
3.4.4.	El problema de los hiperplanos	43
3.4.5.	Generadores congruenciales lineales combinados	43
3.4.6.	Otros generadores eficientes y portables	49
4.	El Método de Monte Carlo	51
4.1.	Estimación de integrales definidas	53
4.1.1.	Integración sobre $(0, 1)$	53
4.1.2.	Integración sobre un intervalo (a, b)	55
4.1.3.	Integración sobre $(0, \infty)$	56
4.2.	Estimación de integrales múltiples	57
4.2.1.	Estimación del valor de π	58
5.	Generación de variables aleatorias discretas	61
5.1.	Método de la transformada inversa	61
5.1.1.	Generación de una variable aleatoria uniforme discreta	64
5.1.2.	Cálculo de promedios	66
5.1.3.	Generación de una variable aleatoria geométrica	68
5.1.4.	Generación de variables Bernoulli	69
5.1.5.	Generación de una variable aleatoria Poisson	69
5.1.6.	Generación de una variable aleatoria binomial	71
5.2.	Método de aceptación y rechazo	72
5.3.	Método de composición	74
5.4.	Métodos alternativos	76
5.4.1.	El método del alias	76
5.4.2.	Método de la urna	80
6.	Generación de variables aleatorias continuas	83
6.1.	Introducción	83
6.2.	Método de la transformada inversa	83
6.2.1.	Simulación de una variable aleatoria exponencial	87
6.2.2.	Simulación de una variable aleatoria Poisson $X \sim \mathcal{P}(\lambda)$	87
6.2.3.	Simulación de una variable con distribución $\text{Gamma}(n, \lambda^{-1})$	88
6.3.	El método de aceptación y rechazo	90
6.4.	Simulación de variables aleatorias normales	94

6.4.1.	Por composición usando $ Z $	94
6.4.2.	Método polar	96
6.4.3.	Transformaciones de Box-Muller	97
6.4.4.	Método de razón entre uniformes	99
6.5.	Generación de un Proceso de Poisson	102
6.5.1.	Procesos de Poisson homogéneos	102
6.5.2.	Procesos de Poisson no homogéneos	103
6.6.	Método de aceptación y rechazo transformado	105
6.6.1.	Método de rechazo transformado con compresión	107

Capítulo 1

Conceptos básicos de probabilidad

1.1. Espacio de probabilidad

En lo que sigue daremos una noción intuitiva del concepto de *espacio de probabilidad* y los elementos que intervienen en su definición.

1.1.1. Espacio muestral

Dado un experimento, se llama **espacio muestral** al conjunto de resultados del experimento.

Ejemplo 1.1. En una carrera de 3 caballos a , b y c , se considera el orden de llegada a la meta y se representa el resultado con una terna. Así, la terna (b, c, a) indica que el caballo b llegó primero, c llegó segundo y a llegó tercero. Asumimos que no hay posibilidad de empate.

El espacio muestral de este experimento es el conjunto S formado por todos los resultados posibles:

$$S = \{(a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b), (c, b, a)\}.$$

Ejemplo 1.2. Si arrojamus una moneda dos veces, y denotamos por C si sale cara y por X si sale cruz, entonces, el espacio muestral formado por los resultados del experimento puede representarse por el conjunto

$$S = \{(C, C), (C, X), (X, C), (X, X)\}.$$

Ejemplo 1.3. En una urna hay tres bolas verdes y dos rojas. Se sacan dos bolas *simultáneamente*. El espacio muestral está formado por todos los conjuntos posibles de dos bolas. Si denotamos V_1, V_2 y V_3 a las bolas verdes y R_1, R_2 a las rojas, entonces el espacio muestral tiene $\binom{5}{2} = 10$ elementos:

$$S = \{\{V_1, V_2\}, \{V_1, V_3\}, \{V_1, R_1\}, \{V_1, R_2\}, \{V_2, V_3\}, \{V_2, R_1\}, \{V_2, R_2\}, \\ \{V_3, R_1\}, \{V_3, R_2\}, \{R_1, R_2\}\}.$$

Si en cambio el experimento consiste en sacar dos bolas *consecutivamente*, sin reposición, el espacio muestral consiste en 20 elementos, ya que deben considerarse todos los pares ordenados. Esto es, deben distinguirse los resultados (V_1, V_2) de (V_2, V_1) y así sucesivamente.

Cualquier subconjunto del espacio muestral es un **evento**. En el Ejemplo 1.1, el subconjunto

$$U = \{(b, c, a), (c, b, a)\}$$

es un evento, que se puede describir como el conjunto de los resultados en los que el caballo a sale último en la carrera.

Si A y B son eventos de un espacio muestral S , entonces también lo son su unión, su intersección y el complemento, que se denotan $A \cup B$, AB y A^c , respectivamente.

En particular, si A_1, A_2, \dots, A_n son eventos, también lo son:

- La unión de los eventos: $A_1 \cup A_2 \cup \dots \cup A_n$.
- La intersección de los eventos: $A_1 A_2 \dots A_n$.
- El espacio muestral: $S = A \cup A^c$.
- El conjunto vacío: $\emptyset = AA^c$.

Dos eventos A y B se dicen **mutuamente excluyentes** si $AB = \emptyset$. Los eventos A_1, A_2, \dots, A_n se dicen **mutuamente excluyentes dos a dos** si $A_i A_j = \emptyset$ para cada $i \neq j$.

1.1.2. Axiomas de probabilidad

Consideramos un espacio muestral S , y suponemos que existe una función P definida sobre el conjunto de eventos de S que satisface los siguientes axiomas:

Axioma 1: $0 \leq P(A) \leq 1$, para todo evento A .

Axioma 2: $P(S) = 1$.

Axioma 3: Si $A_1, A_2, \dots, A_n, \dots$, son mutuamente excluyentes dos a dos, entonces

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i), \quad n = 1, 2, \dots$$

Una tal función P se la denomina **probabilidad**, y $P(A)$ es la probabilidad del evento A . El Axioma 1 indica que la probabilidad es un número real entre 0 y 1. El Axioma 2 indica que la probabilidad de que ocurra algún resultado de S es 1. El Axioma 3 indica que si dos o más eventos son mutuamente excluyentes, entonces la probabilidad de que ocurra alguno de ellos es la suma de sus probabilidades.

De los axiomas podemos concluir además que:

- $P(A^c) = 1 - P(A)$, para todo evento A .
- $P(\emptyset) = 0$.
- $P(A \cup B) = P(A) + P(B) - P(AB)$, para cualquier par de eventos A y B .

Un **espacio de probabilidad** es un par (S, P) donde S es un espacio muestral y P es una probabilidad sobre S .

En teoría de Probabilidad, la familia de subconjuntos en la que está definida la función P puede no ser toda la familia de eventos, sino algunos de ellos, siempre que éstos constituyan una sigma-álgebra de subconjuntos Σ . En esta introducción asumiremos que Σ está conformado por todos los subconjuntos de S .

Ejemplo 1.4. Si S son los posibles resultados de la carrera de tres caballos, y se asigna a cada resultado la misma probabilidad:

$$P(\{(a, b, c)\}) = P(\{(b, a, c)\}) = \dots = \frac{1}{6}$$

entonces (S, P) es un espacio de probabilidad.

En particular, la probabilidad que el caballo a salga último es la probabilidad del evento $U = \{(b, c, a), (c, b, a)\}$, es decir:

$$P(U) = P(\{(b, c, a)\} \cup \{(c, b, a)\}) = P(\{(b, c, a)\}) + P(\{(c, b, a)\}) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}.$$

1.1.3. Probabilidad condicional

Retomando el Ejemplo 1.1, supongamos que el caballo a ha ganado la carrera. Con esta información, ¿cuál es la probabilidad de que el caballo c haya salido último?

El evento que el caballo a haya salido primero está dado por el siguiente subconjunto de resultados:

$$F = \{(a, b, c), (a, c, b)\}.$$

Asumiendo que todos los resultados son igualmente probables, se podría decir que *dado que el caballo a salió primero, la probabilidad que c salga último es $\frac{1}{2}$* .

Dados dos eventos A y F , y una probabilidad P , la **probabilidad condicional** de que ocurra A dado F se define como:

$$P(A | F) = \frac{P(AF)}{P(F)}$$

siempre que $P(F) > 0$.

Ejemplo 1.5. De una urna con 4 bolas rojas y 7 bolas azules se extraen dos bolas de manera consecutiva, sin reposición. ¿Cuál es la probabilidad que la segunda bola sea azul si la primera también lo es?

Para el cálculo de estas probabilidades es útil pensar a la urna como un vector $v = (v_1, \dots, v_{11})$, donde cada v_1, v_2, v_3 y v_4 representan las bolas rojas y las siete restantes una azul. El espacio muestral S consiste en todas las extracciones de dos bolas, por lo cual cada elemento de S puede representarse como un par ordenado (v_i, v_j) , y la probabilidad de cada uno de estos resultados puntuales es:

$$P(\{(v_i, v_j)\}) = \frac{1}{11 \cdot 10}.$$

Debemos calcular una probabilidad condicional $P(A | F)$, donde F es el evento donde la primera bola es azul y A es el evento que la segunda sea azul. Así:

$$P(A | F) = \frac{P(AF)}{P(F)}$$

El evento AF consiste en que ambas bolas sean azules, por lo cual

$$P(AF) = \frac{7 \cdot 6}{11 \cdot 10}.$$

Luego

$$P(A | F) = \frac{7 \cdot 6}{7 \cdot 10} = \frac{6}{10}.$$

Intuitivamente es un resultado razonable, ya que habiendo sacado una bola azul quedan 6 azules sobre un total de 10 bolas.

Dos eventos A y B se dicen **independientes** si $P(A | B) = P(A)$. En este caso se cumple:

$$P(AB) = P(A) \cdot P(B).$$

Notemos que para cualquier par de eventos A y F , se tiene que $A = AF \cup AF^c$. Dado que F y F^c son mutuamente excluyentes, entonces también lo son AF y AF^c . Por lo tanto

$$P(A) = P(AF) + P(AF^c).$$

Esto nos permite calcular la probabilidad de A como una suma ponderada de probabilidades condicionales:

$$\begin{aligned} P(A) &= P(A | F)P(F) + P(A | F^c)P(F^c) \\ &= P(A | F)P(F) + P(A | F^c)(1 - P(F)). \end{aligned} \quad (1.1)$$

En particular, también tenemos que $P(AF) = P(A | F)P(F)$ y $P(AF) = P(F | A)P(A)$. Entonces:

$$P(F | A) = \frac{P(AF)}{P(A)} = \frac{P(A | F) P(F)}{P(A)}. \quad (1.2)$$

La igualdad (1.2) se conoce como **Fórmula de Bayes**, y permite calcular la probabilidad condicional $P(F | A)$ (a posteriori) en términos de $P(F)$ (a priori). Puede generalizarse a un número finito de eventos. Esto es si F_1, F_2, \dots, F_n son eventos mutuamente excluyentes tales que $F_1 \cup F_2 \cup \dots \cup F_n = S$, entonces para cualquier evento A se tiene que:

$$\begin{aligned} P(A) &= P(AF_1) + P(AF_2) + \dots + P(AF_n) \\ &= P(A | F_1)P(F_1) + P(A | F_2)P(F_2) + \dots + P(A | F_n)P(F_n) \\ &= \sum_{i=1}^n P(A | F_i)P(F_i). \end{aligned}$$

Luego, la probabilidad condicional de que ocurra el evento F_j dado A es igual a:

$$P(F_j | A) = \frac{P(A | F_j)P(F_j)}{P(A)} = \frac{P(A | F_j)P(F_j)}{\sum_{i=1}^n P(A | F_i)P(F_i)}. \quad (1.3)$$

1.2. Variables aleatorias

Dado un espacio muestral (S, P) , una **variable aleatoria** es una función $X : S \mapsto \mathbb{R}$ tal que el conjunto $\{s \in S | X(s) \leq x\}$ es un evento sobre el que está definido P , para todo $x \in \mathbb{R}$.

Ejemplo 1.6. Si el experimento consiste en arrojar dos dados, la suma de los valores obtenidos es una variable aleatoria que toma los valores enteros entre 2 y 12.

Ejemplo 1.7. Si el experimento consiste en arrojar una moneda sucesivamente hasta que caiga cara, el número de veces que salió cruz es una variable aleatoria. Específicamente, el espacio muestral determinado por el experimento puede describirse como:

$$S = \{C, XC, XXC, XXXC, \dots\}$$

y la variable aleatoria toma valores en los enteros no negativos: $X : S \mapsto \mathbb{N} \cup 0$.

Ejemplo 1.8. También son ejemplos de variables aleatorias las que resultan de:

- contar el número de personas que ingresan a un local determinado cada día,
- medir el tiempo de servicio en un cajero automático,
- observar las tasas de interés en el mercado financiero.

Si X es una variable aleatoria sobre un espacio de probabilidad (S, P) , y x es un número real, entonces denotamos con $\{X \leq x\}$ al subconjunto de S :

$$\{X \leq x\} := \{s \in S \mid X(s) \leq x\}.$$

Una notación análoga denotará a los eventos $\{X = x\}$, $\{X > x\}$, $\{a < X < b\}$, y así siguiendo.

Dada una variable aleatoria X en un espacio de probabilidad S , se define la **función de distribución acumulada** como:

$$F(x) = P(X \leq x) := P(\{s \in S \mid X(s) \leq x\}). \quad (1.4)$$

La función F tiene dominio en los números reales y toma valores en el intervalo $[0, 1]$. En particular cumple las siguientes propiedades:

- F es no decreciente,
- para todo $x \in \mathbb{R}$ se cumple que $0 \leq F(x) \leq 1$,
- F es continua a derecha.

Diremos que una variable aleatoria es **discreta** si toma sólo un número finito o numerable de valores. En este caso, se define la **función de probabilidad de masa** por

$$p(x) = P(X = x).$$

Si la variable X toma valores en un conjunto finito $\{x_1, x_2, \dots, x_N\}$, entonces se cumple que

$$\sum_{i=1}^N p(x_i) = 1,$$

y si toma una cantidad infinita numerable de valores x_1, x_2, \dots , entonces

$$\sum_{i=1}^{\infty} p(x_i) = 1.$$

Ejemplo 1.9. Si una variable aleatoria toma dos valores, 1 y 2, y $p(1) = 0.3$, entonces $p(2) = 0.7$.

Ejemplo 1.10. Si la probabilidad de que una moneda salga cara es $\frac{1}{3}$, y las tiradas de moneda son eventos independientes, entonces la probabilidad de realizar n tiradas hasta obtener la primera cara es:

$$p(n) = P(\{n-1 \text{ tiradas cruz y una cara}\}) = \left(\frac{2}{3}\right)^{n-1} \frac{1}{3}, \quad n = 1, 2, \dots$$

y se tiene que

$$\sum_{n=1}^{\infty} p(n) = \frac{1}{3} \left(\frac{1}{1 - \frac{2}{3}} \right) = 1.$$

Una variable aleatoria se dice que es una variable **(absolutamente) continua** si existe una función f tal que para todo subconjunto $C \subseteq \mathbb{R}$ se cumple:

$$P(X \in C) = \int_C f(x) dx.$$

La función f se denomina **función de densidad de probabilidad** de la variable aleatoria X .

Por ejemplo, si $C = \{X < 2\}$ y f es la función de densidad de X , entonces

$$P(X \leq 2) = \int_{-\infty}^2 f(x) dx,$$

o si $C = \{-2 < X < 5\}$, entonces

$$P(-2 < X \leq 5) = \int_{-2}^5 f(x) dx.$$

Notemos además que si X es continua, entonces $P(X = a) = \int_a^a f(t) dt$, y por lo tanto $P(X = a) = 0$. En particular, f y la función de distribución acumulada F se relacionan por:

$$F(a) = P(X \leq a) = \int_{-\infty}^a f(x) dx.$$

Derivando con respecto a a tenemos que $F'(a) = f(a)$, para todo $a \in \mathbb{R}$ tal que F es derivable en a .

Además, si $\epsilon > 0$, entonces

$$P(a - \epsilon/2 \leq X \leq a + \epsilon/2) = \int_{a-\epsilon/2}^{a+\epsilon/2} f(x) dx,$$

es decir que para valores de ϵ cercanos a 0 esta probabilidad es aproximadamente $f(a) \cdot \epsilon$. De esta manera $f(a)$ da una medida de la probabilidad de que la variable aleatoria tome valores cercanos a a .

1.2.1. Distribución conjunta

Si X e Y son variables aleatorias sobre un espacio de probabilidad (S, P) , se llama **función de distribución acumulada conjunta** de X e Y a la función $F : \mathbb{R} \times \mathbb{R} \mapsto [0, 1]$ dada por

$$F(a, b) = P(X \leq a, Y \leq b).$$

En particular, si X e Y son variables aleatorias discretas se define la **función de masa de probabilidad conjunta** de X e Y como:

$$p(a, b) = P(X = a, Y = b).$$

Dos variables aleatorias X e Y se dicen **conjuntamente continuas** si existe una función f llamada **función de densidad conjunta** tal que

$$P(X \in C, Y \in D) = \int \int_{x \in C, y \in D} f(x, y) dx dy.$$

Si F es función de distribución conjunta de X e Y , entonces pueden calcularse las distribuciones de probabilidad de X e Y a partir de F , también llamadas **distribuciones marginales** F_X y F_Y . Esto es:

$$F_X(a) = P(X \leq a) = F(a, \infty), \quad F_Y(b) = P(Y \leq b) = F(\infty, b).$$

Si X e Y son variables aleatorias discretas con función de masa conjunta p , entonces las funciones de probabilidad de masa marginales de X e Y están dadas respectivamente por:

$$p_X(a) = \sum_b p(a, b), \quad p_Y(b) = \sum_a p(a, b).$$

Aquí los subíndices de la sumatoria b y a toman todos los valores posibles en el rango de Y y X , respectivamente.

Si X e Y son conjuntamente continuas con función de densidad conjunta f , las distribuciones marginales están dadas por:

$$F_X(a) = \int_{-\infty}^a \int_{-\infty}^{\infty} f(x, y) dy dx = \int_{-\infty}^a f_X(x) dx.$$

$$F_Y(b) = \int_{-\infty}^{\infty} \int_{-\infty}^b f(x, y) dy dx = \int_{-\infty}^b f_Y(y) dy.$$

Notemos que las correspondientes densidades marginales f_X y f_Y están dadas por:

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y) dy \quad \text{y} \quad f_Y(y) = \int_{-\infty}^{\infty} f(x, y) dx.$$

Así, a partir de la función de distribución conjunta es posible obtener las distribuciones marginales de X e Y . La recíproca no es cierta en general.

1.2.2. Distribución condicional

Si X e Y son variables aleatorias discretas, entonces la **probabilidad de masa condicional** $p_{X|Y}$ se define como

$$p_{X|Y}(x | y) = P(X = x | Y = y).$$

En particular, tenemos que:

$$p_{X|Y}(x | y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{p(x, y)}{p_Y(y)}.$$

Si X e Y son conjuntamente continuas, se define la **función de densidad condicional** como:

$$f_{X|Y}(x | y) = \frac{f(x, y)}{f_Y(y)}.$$

En particular, la probabilidad condicional $P(X \leq x | Y = y)$ está dada por

$$P(X \leq a | Y = y) = \int_{-\infty}^a \frac{f(x, y)}{f_Y(y)} dx = \int_{-\infty}^a \frac{f(x, y)}{\int_{-\infty}^{\infty} f(s, y) ds} dx.$$

En particular tenemos una versión de la fórmula de Bayes que relaciona las distribuciones condicionales y marginales para dos variables aleatorias X e Y :

$$p_{X|Y}(x | y) = \frac{p_{Y|X}(y | x)p_X(x)}{p_Y(y)}, \quad f_{X|Y}(x | y) = \frac{f_{Y|X}(y | x)f_X(x)}{f_Y(y)}.$$

Ejemplo 1.11. Consideremos X e Y variables aleatorias continuas con función de densidad conjunta

$$f_{X,Y}(x, y) = \begin{cases} \frac{e^{-x/y} e^{-y}}{y} & x \geq 0, 0 < y < \infty \\ 0 & \text{en otro caso.} \end{cases}$$

Entonces la densidad de Y está dada por:

$$f_Y(y) = \int_{-\infty}^{\infty} f(x, y) dx = \int_0^{\infty} \frac{e^{-x/y} e^{-y}}{y} dx = e^{-y}, \quad 0 < y < \infty,$$

y $f_Y(y) = 0$ en otro caso. Luego la función de densidad condicional de X dado Y está dada por:

$$f_{X|Y}(x | y) = \begin{cases} \frac{e^{-x/y}}{y} & 0 < x < \infty, 0 < y < \infty \\ 0 & \text{en otro caso.} \end{cases}$$

Luego la probabilidad condicional de X dado Y se calcula como:

$$P(X \leq a | Y = y) = \int_0^a \frac{e^{-x/y}}{y} dx = 1 - e^{-a/y}, \quad 0 < a < \infty, 0 < y < \infty.$$

Dos variables aleatorias X e Y son **independientes** si para todo C, D subconjuntos de \mathbb{R} se cumple que

$$P(X \in C, Y \in D) = P(X \in C) \cdot P(Y \in D).$$

Esto es, si los conjuntos $\{X \in C\}$ y $\{Y \in D\}$ son eventos independientes.

En tal caso, si X e Y son variables aleatorias discretas se cumple que $p(a, b) = p_X(a) \cdot p_Y(b)$, y si X e Y son variables aleatorias conjuntamente continuas se cumple que $f(x, y) = f_X(x) \cdot f_Y(y)$. Por lo tanto, si X e Y son independientes, la distribución conjunta se obtiene a partir de las distribuciones de X e Y .

1.2.3. Convolución de distribuciones

Dadas dos variables aleatorias X e Y sobre un espacio de probabilidad (S, P) consideremos la distribución de la suma de estas variables, $X + Y$. Así, si X e Y son variables discretas, entonces $X + Y$ también es discreta y se tiene que $X + Y$ toma un valor a cada vez que X e Y toman valores x y $a - x$, y recíprocamente. Por lo tanto:

$$P(X + Y = a) = \sum_x P(X = x, Y = a - x) = \sum_x p(x, a - x),$$

donde $p(x, y)$ denota la probabilidad conjunta de X e Y . Si X e Y son conjuntamente continuas, entonces $X + Y \leq a$ cada vez que X toma un valor x e Y es menor o igual a $a - x$. Luego

$$P(X + Y \leq a) = \int_{-\infty}^{\infty} \int_{-\infty}^{a-x} f_{X,Y}(x, y) dy dx = \int_{-\infty}^{\infty} \int_{-\infty}^a f_{X,Y}(x, y - x) dy dx,$$

con $f_{X,Y}$ la densidad conjunta de X e Y .

Si las variables son independientes, entonces para el caso discreto resulta que la probabilidad de masa de la variable aleatoria $X + Y$ está dada por:

$$P(X + Y = a) = \sum_x p_X(x) p_Y(a - x).$$

Para el caso continuo tenemos que

$$P(X + Y \leq a) = \int_{-\infty}^{\infty} \int_{-\infty}^a f_X(x) f_Y(y - x) dy dx = \int_{-\infty}^a \int_{-\infty}^{\infty} f_X(x) f_Y(y - x) dx dy,$$

de donde vemos que la densidad de $X + Y$ está dada por $\int_{-\infty}^{\infty} f_X(x) f_Y(y - x) dx$. Aquí p_X , p_Y , f_X y f_Y son las probabilidades de masa y densidades marginales respectivamente.

Esto da lugar al concepto de **convolución** de funciones de probabilidad. En particular, para las probabilidades de masa la convolución $p_X * p_Y(a)$ se define por:

$$p_X * p_Y(a) = \sum_x p_X(x) p_Y(a - x).$$

Para las funciones de densidad, la convolución $f_X * f_Y(a)$ se define por:

$$f_X * f_Y(a) = \int_{-\infty}^{\infty} f_X(x) f_Y(a - x) dx.$$

Así, si X e Y son variables aleatorias discretas independientes, entonces la probabilidad de masa de $X + Y$ está dada por la convolución $p_X * p_Y$. Si son continuas e independientes, la densidad de $X + Y$ está dada por la convolución de las densidades marginales, $f_X * f_Y$.

Ejemplo 1.12. Consideremos X e Y dos variables aleatorias independientes con densidades f y g respectivamente, dadas por:

$$f(x) = \begin{cases} e^{-x} & x > 0 \\ 0 & x \leq 0 \end{cases} \quad g(x) = \begin{cases} 3e^{-3x} & x > 0 \\ 0 & x \leq 0. \end{cases}$$

Como $f(x) = 0$ para valores de x negativos, y $g(x - y) = 0$ si $y > x$, tenemos que para $x > 0$:

$$\begin{aligned} (f * g)(x) &= \int_{-\infty}^{\infty} f(y) g(x - y) dy = \int_0^x f(y) g(x - y) dy \\ &= \int_0^x e^{-y} 3e^{-3(x-y)} dy \\ &= 3e^{-3x} \int_0^x e^{2y} dy \\ &= \frac{3}{2} e^{-3x} (e^{2x} - 1) = \frac{3}{2} (e^{-x} - e^{-3x}). \end{aligned}$$

Para $x \leq 0$ se verifica $(f * g)(x) = 0$. Así, para $a > 0$ tenemos que:

$$P(X+Y \leq a) = \int_0^a \frac{3}{2} (e^{-x} - e^{-3x}) dx = \frac{3}{2} \left(1 - e^{-a} - \frac{1}{3} (1 - e^{-3a}) \right) = \frac{1}{2} (2 - 3e^{-a} + e^{-3a}).$$

1.2.4. Valor esperado

Dada una variable aleatoria discreta X que toma valores x_i , $i = 1, 2, \dots, n, \dots$, se llama **valor esperado** o **esperanza matemática** a la cantidad (si existe)

$$E[X] = \sum_i x_i P(X = x_i).$$

Si X es una variable aleatoria continua, su valor esperado se define por el valor de la integral (si existe):

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx.$$

Es importante notar que el valor esperado no es necesariamente un valor posible de X .

Ejemplo 1.13. Si X toma los valores 1, 2, 3 y 4, con $p(1) = p(2) = 0.3$, $p(3) = p(4) = 0.2$, entonces

$$E[X] = 1 \cdot 0.3 + 2 \cdot 0.3 + 3 \cdot 0.2 + 4 \cdot 0.2 = 2.3.$$

Sean X e Y dos variables aleatorias sobre un mismo espacio de probabilidad S . Entonces se cumplen las siguientes propiedades:

a) Si $g : \mathbb{R} \mapsto \mathbb{R}$, entonces $g(X)$ es una variable aleatoria y

$$\begin{aligned} E[g(X)] &= \sum_i g(x_i) p(x_i), & (\text{si } X \text{ es discreta}), \\ E[g(X)] &= \int_{-\infty}^{\infty} g(x) f(x) dx, & (\text{si } X \text{ es continua}). \end{aligned}$$

En particular, tomando $g(x) = ax + b$ se deduce que

$$E[aX + b] = aE[X] + b.$$

b) El valor esperado es un operador lineal. Esto es,

$$E[X + Y] = E[X] + E[Y].$$

1.2.5. Varianza

La **varianza** es una medida de la dispersión de X en torno a su valor esperado $E[X] = \mu$, y está definido por:

$$\text{Var}(X) = E[(X - \mu)^2].$$

Notemos que

$$\text{Var}(X) = E[(X^2 - 2X\mu + \mu^2)] = E[X^2] - 2\mu E[X] + \mu^2 = E[X^2] - \mu^2.$$

La varianza es siempre un número positivo a menos que la variable aleatoria sea siempre constante. Es importante notar que si a y b son números reales, entonces

$$\text{Var}(aX + b) = a^2 \text{Var}(X).$$

La varianza no verifica la condición de linealidad. En efecto, si $E[X] = \mu$, $E[Y] = \theta$, entonces

$$\begin{aligned} \text{Var}(X + Y) &= E[((X + Y) - (\mu + \theta))^2] \\ &= E[(X - \mu)^2 + (Y - \theta)^2 + 2(X - \mu)(Y - \theta)] \\ &= E[(X - \mu)^2] + E[(Y - \theta)^2] + 2E[(X - \mu)(Y - \theta)] \\ &= \text{Var}(X) + \text{Var}(Y) + 2E[(X - \mu)(Y - \theta)]. \end{aligned}$$

Si X e Y son variables aleatorias, se define la **covarianza** de X e Y por:

$$\text{cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)].$$

De esta manera,

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{cov}(X, Y).$$

Si X e Y son independientes entonces $\text{cov}(X, Y) = 0$ (la recíproca en general no es cierta). Esto puede verse observando que $\text{cov}(X, Y) = E[XY] - E[X]E[Y]$. Luego, si X e Y son independientes y conjuntamente continuas con densidad conjunta f , entonces $f(x, y) = f_X(x)f_Y(y)$ y por lo tanto:

$$E[XY] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xy f(x, y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f_X(x) y f_Y(y) dx dy = E[X]E[Y].$$

Si X e Y son discretas con probabilidad de masa conjunta p , entonces $p(x, y) = p_X(x)p_Y(y)$ y

$$E[XY] = \sum_i \sum_j x_i y_j p(x_i, y_j) = \sum_i \sum_j x_i p_X(x_i) y_j p_Y(y_j) = E[X]E[Y].$$

Por lo tanto, si X e Y son variables aleatorias independientes, se cumple que

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y).$$

Otra medida de dispersión de una variable aleatoria es la **desviación estándar** $\sigma(X)$, definida por

$$\sigma(X) = \sqrt{\text{Var}(X)}.$$

Tiene la ventaja de mantener las mismas unidades (distancia, longitud, tiempo, etc.) que X y $E[X]$. A su vez, se define la **correlación** de dos variables aleatorias X e Y por:

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma(X) \cdot \sigma(Y)}.$$

Se cumple que $\rho(X, Y)$ es un número entre -1 , y 1 y da una medida normalizada de la covarianza entre dos variables aleatorias.

1.2.6. Desigualdad de Chebyshev

Si X toma sólo valores no negativos y $a > 0$, entonces se cumple la desigualdad

$$P(X \geq a) \leq \frac{E[X]}{a}. \quad (1.5)$$

Para probar (1.5) consideramos la variable aleatoria Y dada por

$$Y = \begin{cases} a & \text{si } X \geq a \\ 0 & \text{en caso contrario.} \end{cases}$$

Dado que $Y \leq X$, lo mismo vale para sus valores esperados: $E[Y] \leq E[X]$. Luego,

$$a P(X \geq a) = E[Y] \leq E[X],$$

de donde se deduce el resultado. De esta propiedad para variables aleatorias no negativas puede derivarse la Desigualdad de Chebyshev.

Teorema 1.1 (Desigualdad de Chebyshev). Si X es variable aleatoria con media μ y varianza σ^2 , entonces para $k > 0$

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

Para la demostración, basta considerar la variable aleatoria

$$Y = \frac{|X - \mu|^2}{\sigma^2},$$

que toma valores positivos y su valor esperado es 1: $E[Y] = 1$. Aplicando la desigualdad (1.5) para un $k > 0$, tenemos que:

$$P(Y \geq k^2) \leq \frac{1}{k^2}.$$

Dado que los eventos $\{Y \geq k^2\}$ y $\{\frac{|X - \mu|}{\sigma} \geq k\}$ son iguales, se sigue el resultado.

Así por ejemplo, la probabilidad que los valores de una variable aleatoria estén a una distancia menor a 2 desviaciones estándar del valor esperado es mayor a $1 - 0.25 = 0.75$.

1.2.7. Leyes de los grandes números

Dos variables aleatorias se dicen **idénticamente distribuidas** si tienen una misma función de distribución acumulada.

Los siguientes resultados teóricos enuncian que si se tiene una sucesión de variables aleatorias independientes, idénticamente distribuidas, con valor esperado μ , entonces los promedios de estas variables *convergen*, en algún sentido de convergencia, al valor μ .

Específicamente, si $X_1, X_2, \dots, X_n, \dots$ son variables aleatorias independientes e idénticamente distribuidas, con media μ , se cumplen las siguientes propiedades:

■ **Ley débil de los grandes números:**

$$P\left(\left|\frac{X_1 + X_2 + \dots + X_n}{n} - \mu\right| > \epsilon\right) \rightarrow 0 \quad n \rightarrow \infty.$$

■ **Ley fuerte de los grandes números:**

Con probabilidad 1 se cumple que:

$$\lim_{n \rightarrow \infty} \frac{X_1 + X_2 + \dots + X_n}{n} = \mu.$$

1.3. Distribuciones de probabilidad

En la literatura se puede encontrar un gran número de distribuciones de probabilidad teóricas. En esta sección consideraremos algunas de estas distribuciones que luego utilizaremos para modelado y simulación.

1.3.1. Variables aleatorias discretas

Distribución uniforme discreta: $U\{1, n\}$

Se dice que una variable aleatoria tiene **distribución uniforme** si todos sus valores son equiprobables. Con $U\{1, n\}$ denotaremos a la variable aleatoria que toma valores en el conjunto $\{1, 2, \dots, n\}$, todos con la misma probabilidad $\frac{1}{n}$.

$$p(i) = \frac{1}{n}, \quad i = 1, 2, \dots, n.$$

El valor esperado y la varianza están dados por:

$$E[X] = \frac{n+1}{2}, \quad \text{Var}(X) = \frac{n^2-1}{12}.$$

Distribución de Bernoulli: $B(p)$

Una variable aleatoria que toma dos valores con probabilidad p (éxito) y $1-p$ (fracaso), se dice **de Bernoulli**. La distribución de Bernoulli teórica se corresponde con la variable aleatoria que toma los valores 1 y 0:

$$X = \begin{cases} 1 & \text{con prob. } p \\ 0 & \text{con prob. } 1-p. \end{cases}$$

Su valor esperado y varianza están dados por:

$$E[X] = p, \quad \text{Var}(X) = p \cdot (1-p).$$

Distribución Binomial $B(n, p)$

Si consideramos un experimento que consiste en n ensayos independientes, cada uno con probabilidad p de éxito, entonces el número de éxitos tiene una distribución binomial de parámetros n y p . El rango de la variable aleatoria es el conjunto $\{0, 1, 2, \dots, n\}$, y la función de masa de probabilidad está dada por:

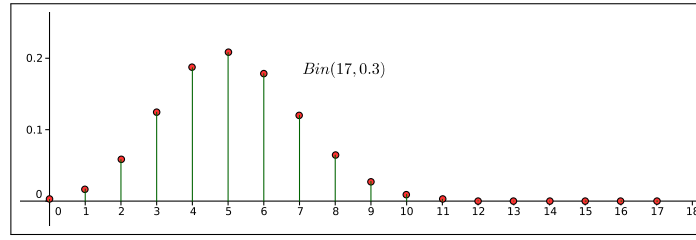
$$p(i) = P(X = i) = \binom{n}{i} p^i (1-p)^{n-i}, \quad 0 \leq i \leq n.$$

Más adelante resultará útil la siguiente fórmula recursiva para las probabilidades de masa:

$$p(0) = (1-p)^n, \quad p(i+1) = \frac{n-i}{i+1} \frac{p}{(1-p)} p(i), \quad 0 \leq i \leq n-1.$$

El valor esperado y la varianza están dados por:

$$E[X] = np, \quad \text{Var}(X) = np(1-p).$$

Figura 1.1: Distribución binomial. $\text{Bin}(17, 0.3)$ **Distribución de Poisson: $\mathcal{P}(\lambda)$**

Una variable aleatoria se dice que es de **Poisson con parámetro λ** si toma valores en $\mathbb{N} \cup 0$ con probabilidad de masa

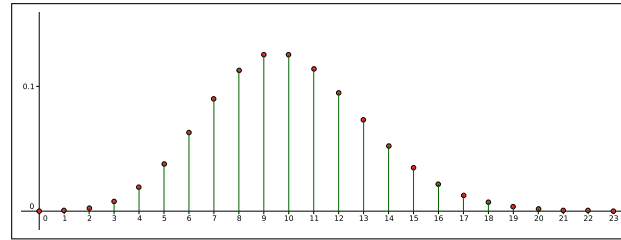
$$p(i) = P(X = i) = e^{-\lambda} \frac{\lambda^i}{i!}, \quad i \geq 0.$$

Una fórmula recursiva para estas probabilidades está dada por:

$$p(0) = e^{-\lambda}, \quad p(i+1) = \frac{\lambda}{i+1} p(i), \quad i \geq 0.$$

El valor esperado y la varianza están dados por:

$$E[X] = \lambda, \quad \text{Var}(X) = \lambda.$$

Figura 1.2: Distribución de Poisson. $\lambda = 10$ **Distribución Geométrica: $\text{Geom}(p)$**

Dada una sucesión de ensayos independientes con probabilidad p de éxito, la variable aleatoria geométrica cuenta el número de ensayos independientes hasta obtener el primer éxito. Su rango es el conjunto de números naturales, \mathbb{N} .

La función de probabilidad de masa está dada por:

$$p(n) = P(X = n) = p(1-p)^{n-1}, \quad n \geq 1.$$

El valor esperado y la varianza están dados por:

$$E[X] = \frac{1}{p}, \quad \text{Var}(X) = \frac{1-p}{p^2}$$

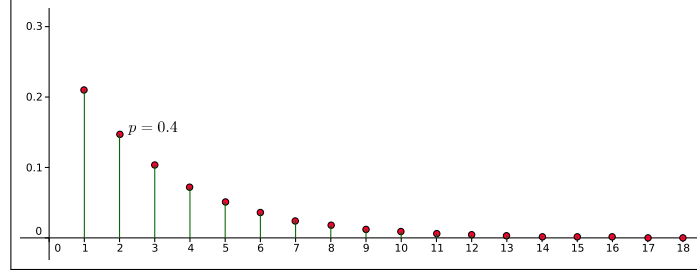


Figura 1.3: Distribución geométrica. $p = 0.4$

Distribución Binomial Negativa o Pascal: $\text{Bn}(r, p)$

Esta distribución se corresponde con la variable aleatoria que mide el número de ensayos independientes con probabilidad de éxito p , hasta obtener r éxitos. La variable toma valores en el intervalo $\{r, r+1, r+2, \dots\} = \{n \in \mathbb{N} \mid n \geq r\}$. La función de probabilidad de masa está dada por

$$P(X = n) = \binom{n-1}{r-1} p^r (1-p)^{n-r}, \quad n \geq r.$$

El valor esperado y la varianza están dados por:

$$E[X] = \frac{r}{p}, \quad \text{Var}(X) = \frac{r(1-p)}{p^2}.$$

Distribución Hipergeométrica $H(n, N, M)$

Esta distribución se corresponde con la variable aleatoria que mide el número de éxitos en una muestra de tamaño n extraída de un conjunto de $N+M$ elementos, donde un éxito equivale a extraer un elemento del subconjunto de cardinal N .

El rango de esta distribución es $\{0, 1, 2, \dots, n\}$. La función de probabilidad de masa está dada por:

$$p(i) = P(X = i) = \frac{\binom{N}{i} \binom{M}{n-i}}{\binom{N+M}{n}}.$$

La probabilidad $p(i)$ es 0 si $i > n$ o $n-i > M$. El valor esperado y la varianza están dadas por:

$$E[X] = \frac{nN}{N+M}, \quad \text{Var}(X) = \frac{nNM}{(N+M)^2} \left(1 - \frac{n-1}{N+M-1}\right).$$

1.3.2. Variables aleatorias continuas

Denotaremos con I_A a la **función indicadora del conjunto** A , dada por

$$I_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}.$$

Distribución uniforme $U(a, b)$

Una variable aleatoria continua X se dice uniformemente distribuida en (a, b) si su función de densidad está dada por

$$f(x) = \frac{1}{b-a} \mathbb{I}_{(a,b)}(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & \text{c.c.} \end{cases}$$

Su función de distribución acumulada está dada por:

$$F(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b \end{cases},$$

y la varianza y su valor esperado son:

$$E[X] = \frac{a+b}{2}, \quad \text{Var}(X) = \frac{1}{12}(b-a)^2.$$

Distribución Normal $N(\mu, \sigma)$

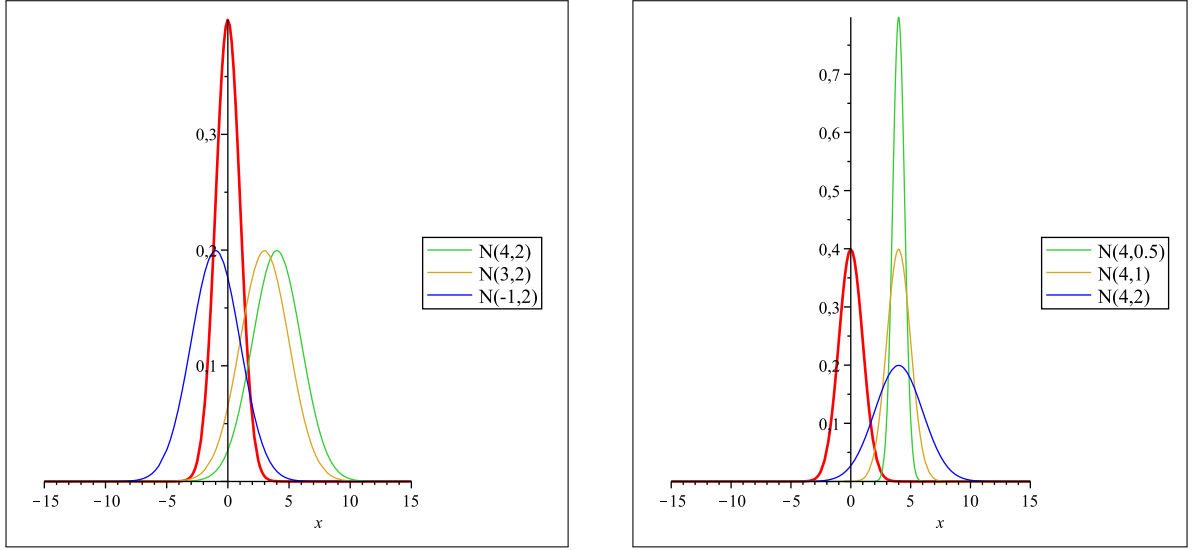
Una variable aleatoria continua X se dice normalmente distribuida con media μ y varianza σ^2 si su función de densidad de probabilidad está dada por

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad x \in \mathbb{R}.$$

En tal caso usamos la notación $X \sim N(\mu, \sigma)$. Si $Z \sim N(0, 1)$ se dice que su distribución es **normal estándar**. La función de densidad es entonces:

$$f_Z(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in \mathbb{R}.$$

La Figura 1.3.2 muestra distintas distribuciones normales, comparadas con la distribución normal estándar. En el primer gráfico la varianza es constante y en el segundo es constante la media.

Figura 1.4: Ejemplos de distribución normal, variando μ y σ

Si $X \sim N(\mu, \sigma)$, entonces $\frac{X-\mu}{\sigma}$ tiene distribución normal estándar. Denotemos con Φ a la función de distribución acumulada de una variable aleatoria normal estándar, $Z \sim N(0, 1)$:

$$\Phi(x) = P(Z \leq x) = \frac{1}{2\pi} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt. \quad (1.6)$$

La integral $\Phi(x)$ dada en (1.6) no tiene una fórmula cerrada, por lo cual es usual utilizar valores tabulados para el cálculo o interpolación de valores de Φ . De la expresión para la función de densidad, se observa que es una función par, esto es, $f(x) = f(-x)$. Luego se cumple que

$$\Phi(x) = 1 - \Phi(-x), \quad \text{para todo } x \in \mathbb{R}.$$

Si X es normal con media μ y varianza σ^2 , entonces su función de distribución acumulada puede expresarse en términos de Φ :

$$F(x) = P(X \leq x) = P\left(\frac{X - \mu}{\sigma} \leq \frac{x - \mu}{\sigma}\right) = \Phi\left(\frac{x - \mu}{\sigma}\right).$$

Algunos valores importantes a recordar son las probabilidades de que los valores de una variable aleatoria normal se distribuyan alrededor de la media, a una distancia menor de $k\sigma$, para ciertos valores de k . Para $k = 1, 2, 3$ tenemos:

$$P(|X - \mu| < \sigma) \simeq 68 \%, \quad P(|X - \mu| < 2\sigma) \simeq 95 \%, \quad P(|X - \mu| < 3\sigma) \simeq 99.7 \%.$$

Si $Z \sim N(0, 1)$ y α es un número entre 0 y 1, se suele denotar z_α al número real tal que

$$P(Z > z_\alpha) = \alpha.$$

Así por ejemplo,

$$z_{0.05} = 1.64, \quad z_{0.025} = 1.96, \quad z_{0.01} = 2.33, \quad z_{0.005} = 2.58.$$

Debido a la simetría de la densidad de la normal estándar, estos valores indican en particular que:

$$P(-1.64 \leq Z \leq 1.64) = 90 \%, \quad P(-2.58 \leq Z \leq 2.58) = 99 \%,$$

y resultados similares para el 95 % y el 98 %.

Un resultado importante en la teoría de probabilidad es el llamado Teorema Central del Límite. Este teorema establece que la suma de n variables aleatorias independientes, igualmente distribuidas, todas con media μ y varianza σ^2 , tiene una distribución aproximadamente normal, con media $n\mu$ y varianza $n\sigma^2$. Más precisamente:

Teorema 1.2 (Teorema Central del límite). Sean X_1, X_2, \dots , variables aleatorias independientes igualmente distribuidas con media μ y varianza σ^2 . Entonces

$$\lim_{n \rightarrow \infty} P\left(\frac{X_1 + X_2 + \dots + X_n - n\mu}{\sigma\sqrt{n}} < x\right) = \Phi(x).$$

Este teorema resultará útil para estimaciones estadísticas de intervalos de confianza a partir de una muestra de tamaño n .

Distribución exponencial $\mathcal{E}(\lambda)$

Una variable aleatoria X tiene distribución exponencial con parámetro λ , $\lambda > 0$, si su función de densidad está dada por

$$f_\lambda(x) = \lambda e^{-\lambda x} \mathbb{I}_{(0, \infty)}(x).$$

Su valor esperado y varianza están dados por:

$$E[X] = \frac{1}{\lambda}, \quad \text{Var}(X) = \frac{1}{\lambda^2}.$$

Si $X \sim \mathcal{E}(\lambda)$, entonces $cX \sim \mathcal{E}(\frac{1}{c}\lambda)$.

Se dice que una variable aleatoria X tiene **falta de memoria** si

$$P(X > s + t \mid X > s) = P(X > t),$$

para todo $s, t \in \mathbb{R}$. En efecto, si X tiene distribución exponencial con parámetro λ , entonces para cada t se cumple:

$$P(X > t) = 1 - (1 - e^{-\lambda t}) = e^{-\lambda t}.$$

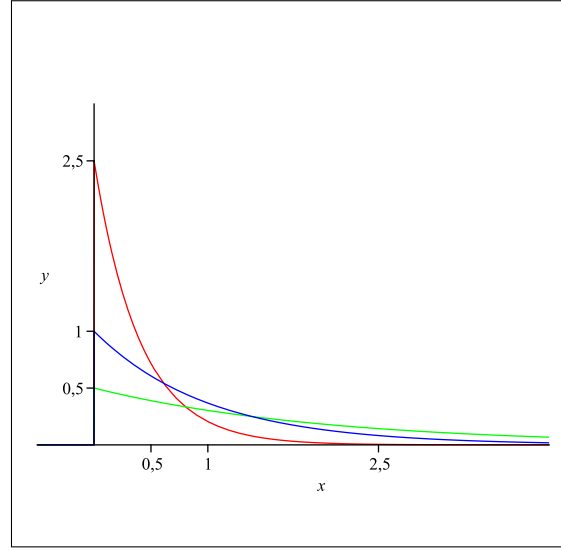


Figura 1.5: Función de densidad de variables exponenciales

Luego,

$$\begin{aligned}
 P(X > s + t \mid X > s) &= \frac{P(X > s + t, X > s)}{P(X > s)} = \frac{P(X > s + t)}{P(X > s)} \\
 &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda s}} = e^{-\lambda t} \\
 &= P(X > t).
 \end{aligned}$$

Las variables aleatorias con distribución exponencial son las únicas variables aleatorias continuas con la propiedad de falta de memoria. El análogo en el caso discreto son las variables aleatorias geométricas.

Sean X_1, X_2, \dots, X_n son variables aleatorias independientes con función de distribución acumulada F_1, F_2, \dots, F_n , y sea M el mínimo entre estas variables:

$$M = \min_{1 \leq i \leq n} \{X_1, X_2, \dots, X_n\}.$$

Entonces M es una variable aleatoria, y su función de distribución acumulada cumple que:

$$1 - F_M(x) = P(M > x) = (1 - F_1(x)) \cdot (1 - F_2(x)) \cdots (1 - F_n(x)).$$

En particular, si las variables aleatorias tienen distribución exponencial:

$$X_i \sim \mathcal{E}(\lambda_i),$$

entonces su distribución satisface:

$$1 - F_M(x) = e^{-\lambda_1 x} \cdot e^{-\lambda_2 x} \cdots e^{-\lambda_n x} = e^{-(\sum_i \lambda_i) x}.$$

Por lo tanto, la distribución del **mínimo entre n variables aleatorias exponenciales independientes** es exponencial:

$$M \sim \mathcal{E}(\lambda_1 + \lambda_2 + \cdots + \lambda_n).$$

Distribución Lognormal $LN(\mu, \sigma)$

Una variable aleatoria continua X se dice **lognormal** si su logaritmo $\ln(X)$ tiene distribución normal. Denotaremos $X \sim LN(\mu, \sigma)$ si el logaritmo de X tiene media μ y desviación estándar σ . En tal caso su función de densidad está dada por:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-(\log(x)-\mu)^2/(2\sigma^2)} \cdot \mathbb{I}_{(0,\infty)}(x).$$

Su valor esperado y varianza están dados por:

$$E[X] = e^{\mu+\sigma^2/2}, \quad Var(X) = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1).$$

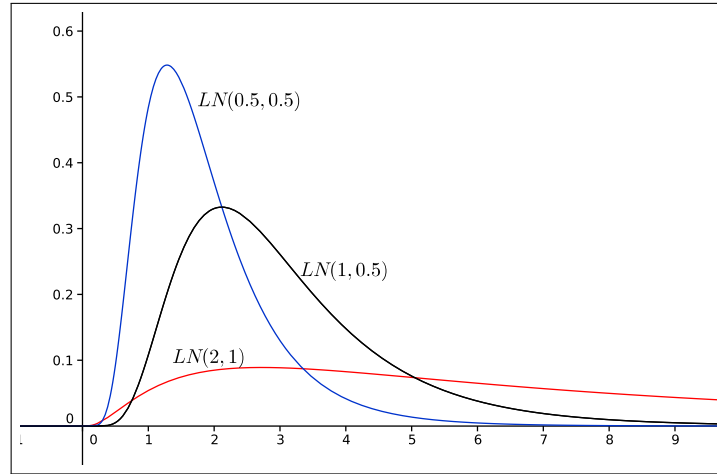


Figura 1.6: Distribución lognormal

Distribución Gamma $\Gamma(\alpha, \beta)$

Una variable aleatoria Gamma con parámetros α y β tiene una función de densidad dada por:

$$f(x) = \frac{1}{\Gamma(\alpha)} \beta^{-\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}, \quad x > 0.$$

En la Figura 1.7 se muestran los gráficos de $\Gamma(\alpha, 1)$, para $\alpha = 0.5, 1, 2$ y 3 . Notar de la definición de f que $\alpha = 1$ corresponde a la distribución exponencial $\mathcal{E}(1)$. Su valor esperado y varianza están dados por:

$$E[X] = \alpha\beta, \quad Var(X) = \alpha\beta^2.$$

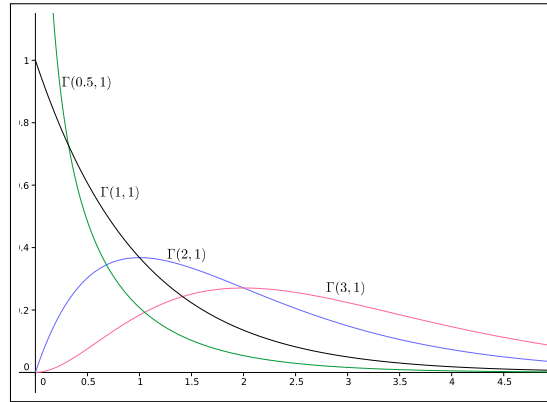


Figura 1.7: Distribuciones Gamma

Distribución Weibull (α, β)

Una variable aleatoria continua Weibull con parámetros α y β , con $\alpha > 0$ y $\beta > 0$, tiene función de densidad dada por:

$$f(x) = \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} \mathbb{I}_{(0, \infty)}(x).$$

En la Figura 1.8 se muestran gráficos para $\beta = 1$ y $\alpha = 0.5, 1, 2$ y 3 . Su valor esperado y varianza son:

$$E[X] = \frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right) \quad \text{Var}(X) = \frac{\beta^2}{\alpha} \left[2\Gamma\left(\frac{2}{\alpha}\right) - \frac{1}{\alpha} \left(\Gamma\left(\frac{1}{\alpha}\right) \right)^2 \right].$$

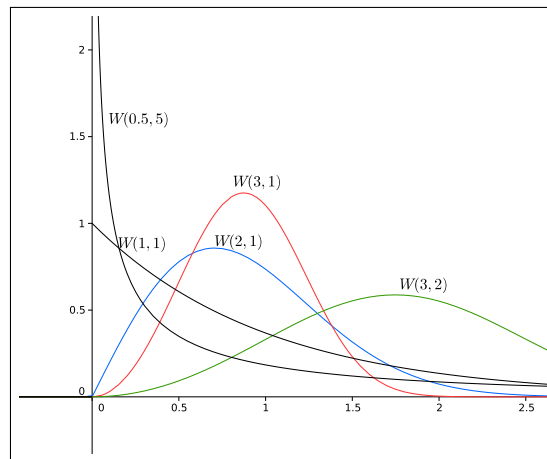


Figura 1.8: Distribución Weibull

Capítulo 2

Procesos de Poisson

Denotamos con I un subconjunto de números reales. I puede ser un intervalo real, o los números naturales, o cualquier otro subconjunto. Dado el espacio de probabilidad (S, P) , un **proceso estocástico** X es una familia de variables aleatorias indexada por el conjunto I . Si I es un intervalo real entonces el proceso estocástico se dice **continuo**. Si I es un subconjunto de \mathbb{Z} el proceso se dice **discreto**. Es decir, para cada $t \in I$, $X(t)$ es una variable aleatoria. En un proceso estocástico, la variable t suele representar una variable temporal o espacial.

2.1. El proceso de Poisson homogéneo

Definición 2.1. Un proceso estocástico continuo $\{N(t), t \geq 0\}$ es un **proceso de Poisson homogéneo de intensidad λ** , para un $\lambda > 0$, si cumple las siguientes propiedades:

- a) $N(0) = 0$
- b) para cada $n \geq 1$ y cada partición $0 \leq t_0 < t_1 < \dots < t_n$ se tiene que $N(t_0), N(t_1) - N(t_0), \dots, N(t_n) - N(t_{n-1})$ son variables aleatorias independientes.
- c) Para cada $t \geq 0, s > 0$, se cumple que la distribución de $N(t + s) - N(t)$ y $N(s)$ están igualmente distribuidas.
- d) $\lim_{h \rightarrow 0} \frac{P(N(h) = 1)}{h} = \lambda,$
- e) $\lim_{h \rightarrow 0} \frac{P(N(h) \geq 2)}{h} = 0.$

Un proceso de Poisson puede pensarse como el proceso de contar el número de arribos o llegadas ocurridos hasta el tiempo t , sabiendo que la tasa de llegada por unidad de tiempo es λ . Con esta analogía, las propiedades anteriores significan, de manera intuitiva, que:

- a) Al momento $t = 0$ no se contabiliza ningún arribo.

- b) **Incrementos independientes:** Si se consideran dos o más intervalos de tiempo no solapados entre sí, el número de arribos que ocurre en uno y otro intervalo son variables aleatorias independientes.
- c) **Incrementos estacionarios:** La distribución del número de llegadas que ocurre en un período de tiempo depende sólo del tiempo transcurrido y no de la ubicación en el tiempo de este período. Esta propiedad es la que determina que el proceso sea homogéneo.
- d) Las últimas dos propiedades indican que en un intervalo pequeño de tiempo la probabilidad que ocurra una llegada es proporcional a la longitud del intervalo, con constante de **intensidad** igual a la tasa de arribos λ . Además la probabilidad de que lleguen dos o más, simultáneamente, tiende a ser nula cuando el intervalo de tiempo es reducido.

2.1.1. Distribución del número de llegadas $N(t)$

Para cada t , la variable aleatoria $N(t)$ tiene una distribución de Poisson con media λt . Una forma intuitiva de ver esta propiedad del proceso de Poisson es la siguiente.

Consideremos un intervalo de longitud $t > 0$, subdividido en n intervalos de longitud $\frac{t}{n}$.

$$t_0 = 0 < t_1 < t_2 < \dots < t_n = t, \quad t_i = \frac{i}{n} t.$$

Dado que el proceso de Poisson tiene incrementos independientes y para un n suficientemente grande la probabilidad de que ocurra más de un evento es nula, el número de llegadas en un subintervalo $(t_{i-1}, t_i]$ es una Bernoulli con $p_n = \frac{\lambda t}{n}$:

$$P(N(t_i) - N(t_{i-1}) = 1) \simeq \lambda \cdot \frac{t}{n}.$$

Así, el número de llegadas en el intervalo $[0, t]$ es una suma de n variables aleatorias Bernoulli independientes con el mismo parámetro p_n , y por lo tanto $N(t)$ se aproxima a una variable aleatoria con distribución binomial $B(n, \frac{\lambda t}{n})$.

Ahora bien, para n grande y con $n \cdot p_n$ tendiendo a una constante (en este caso λ), estas binomiales convergen a una distribución de Poisson con parámetro $n \cdot \frac{\lambda t}{n} = \lambda \cdot t$.

Por último, vemos que por la condición de estacionariedad la variable $N(t+s) - N(t)$ tiene distribución de Poisson con media λs .

Ejemplo 2.1. Juan recibe mensajes de texto a partir de las 10:00 hs de la mañana a razón de 10 mensajes por hora de acuerdo a un proceso de Poisson homogéneo. Calcular la probabilidad de que Juan haya recibido exactamente 18 mensajes para el mediodía y 70 mensajes para las 17:00 hs.

Denotamos con $N(t)$ al proceso de llegada de mensajes fijando $t = 0$ a las 10 de la mañana y midiendo el tiempo en horas. Entonces queremos determinar:

$$P(N(2) = 18, N(7) = 70).$$

Los eventos $\{N(2) = 18\}$ y $\{N(7) = 70\}$ no son independientes ya que involucran intervalos de tiempo solapados. Pero podemos reescribir la probabilidad deseada como

$$P(N(2) = 18, N(7) - N(2) = 70 - 18) = P(N(2) = 18, N(7) - N(2) = 52).$$

De esta manera podemos usar la propiedad de independencia de las variables $N(2)$ y $N(7) - N(2)$ y luego que $N(7) - N(2)$ y $N(5)$ tienen la misma distribución. Por lo tanto:

$$\begin{aligned} P(N(2) = 18, N(7) - N(2) = 52) &= P(N(2) = 18) \cdot P(N(7) - N(2) = 52) \\ &= P(N(2) = 18) \cdot P(N(5) = 52) \\ &= \left(\frac{e^{-2 \cdot 10} (2 \cdot 10)^{18}}{18!} \right) \left(\frac{e^{-5 \cdot 10} (5 \cdot 10)^{52}}{52!} \right) \\ &= 0.0045 \end{aligned}$$

2.1.2. Proceso de Poisson homogéneo trasladado

Dado un proceso de Poisson homogéneo $N(t)$, $t \geq 0$, con parámetro λ , podemos considerar el proceso de eventos desde un valor fijo $t_0 > 0$. Esto es, el proceso

$$\tilde{N}(t) = N(t + t_0) - N(t_0), \quad t \geq 0.$$

El proceso $\tilde{N}(t)$ es el proceso $N(t)$ con su origen trasladado a t_0 . Notemos que $\tilde{N}(t)$ verifica $\tilde{N}(0) = N(t_0) - N(t_0) = 0$. Además hereda de $N(t)$ la propiedad de tener incrementos estacionarios e independientes como así también se verifica que la tasa de arribos es λ . Luego $\tilde{N}(t)$ es un proceso de Poisson homogéneo con tasa de llegada λ .

Ejemplo 2.2. Consideremos en el Ejemplo 2.1 el proceso $\tilde{N}(t)$ que cuenta los mensajes de texto recibidos por Juan a partir de las 12 hs. Entonces $\tilde{N}(t)$ viene dado por

$$\tilde{N}(t) = N(t + 2) - N(2) = N(t + 2) - 18, \quad t \geq 0,$$

donde el tiempo de origen $t = 0$ corresponde ahora a las 12:00 hs.

2.1.3. Distribución del tiempo entre arribos

Llamamos X_1 al tiempo transcurrido hasta el primer evento, y X_j el tiempo transcurrido entre el $j - 1$ -ésimo y el j -ésimo evento, para cada $j > 1$. Veremos que cada X_i , $i \geq 1$, es una variable aleatoria con distribución exponencial de media $\frac{1}{\lambda}$, es decir con distribución $\mathcal{E}(\lambda)$, y que para todo n las variables X_1, X_2, \dots, X_n son independientes.

Para el tiempo hasta el primer arribo tenemos que:

$$P(X_1 > t) = P(N(t) = 0) = e^{-\lambda t},$$

luego

$$P(X_1 \leq t) = 1 - e^{-\lambda t}.$$

Para X_2 calculamos la probabilidad condicional $P(X_2 > t \mid X_1 = s)$:

$$\begin{aligned} P(X_2 > t \mid X_1 = s) &= P(0 \text{ eventos en } (s, s+t] \mid X_1 = s) \\ &= P(0 \text{ eventos en } (s, s+t]) \\ &= P(N(t+s) - N(s) = 0) \\ &= e^{-\lambda t} \end{aligned}$$

Ahora, como

$$P(X_2 > t) = \int_{-\infty}^{\infty} P(X_2 > t \mid X_1 = s) f_{X_1}(s) ds$$

y $P(X_2 > t \mid X_1 = s)$ no depende de s , resulta:

$$P(X_2 > t) = e^{-\lambda t} \int_{-\infty}^{\infty} f_{X_1}(s) ds = e^{-\lambda t}.$$

Por lo tanto, $X_2 \sim \mathcal{E}(\lambda)$ y es independiente de X_1 .

Analizamos ahora la variable aleatoria X_j . Sea $s = s_1 + \dots + s_{j-1}$: tiempo hasta el evento $j - 1$. Entonces:

$$\begin{aligned} P(0 \text{ eventos en } (s, s+t] \mid X_1 = s_1, \dots, X_{j-1} = s_{j-1}) \\ &= P(N(t+s) - N(s) = 0 \mid X_1 = s_1, \dots, X_{j-1} = s_{j-1}) \\ &= P(0 \text{ eventos en } (s, s+t]) \\ &= e^{-\lambda t} \end{aligned}$$

Nuevamente $P(X_j > t \mid X_1 = s_1, \dots, X_{j-1} = s_{j-1})$ es independiente de los valores s_1, s_2, \dots, s_{j-1} , por lo cual X_j también resulta de distribución exponencial e independiente de X_1, X_2, \dots, X_{j-1} .

Así, para cualquier n las variables aleatorias X_1, X_2, \dots, X_n son variables aleatorias independientes, igualmente distribuidas, con distribución exponencial con media $\frac{1}{\lambda}$.

$$X_i \sim \mathcal{E}(\lambda), \quad i = 1, 2, \dots$$

2.1.4. Distribución del tiempo de arribo

Las variables aleatorias X_1, X_2, \dots, X_n describen los tiempos *entre arribos sucesivos*, y ya hemos visto que son variables aleatorias exponenciales, independientes, $X_j \sim \mathcal{E}(\lambda)$. Ahora bien, si denotamos con S_n a la variable:

$$S_n = \sum_{j=1}^n X_j,$$

entonces S_n representa el *tiempo de arribo* o de llegada del n -ésimo evento. Analizaremos la distribución de estas variables.

Sea F_n la función de distribución acumulada de S_n . Notemos que los siguientes eventos son iguales:

$$\{S_n \leq t\} = \{N(t) \geq n\}.$$

Por lo tanto,

$$F_n(t) = P(S_n \leq t) = P(N(t) \geq n) = \sum_{j=n}^{\infty} P(N(t) = j) = \sum_{j=n}^{\infty} e^{-\lambda t} \frac{(\lambda t)^j}{j!}.$$

Luego la función de densidad de S_n está dada por:

$$\begin{aligned} f_n(t) = \frac{d}{dt} F_n(t) &= \sum_{j=n}^{\infty} (-\lambda) e^{-\lambda t} \frac{(\lambda t)^j}{j!} + \sum_{j=n}^{\infty} e^{-\lambda t} \frac{j \lambda (\lambda t)^{j-1}}{j!} \\ &= -\sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^j}{j!} + \sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^{j-1}}{(j-1)!} \\ &= \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!} \end{aligned}$$

Así vemos que f_n es la función de densidad de una **variable aleatoria Gamma** con parámetros $(n, \beta = \frac{1}{\lambda})$. Esto es,

$$S_n \sim \text{Gamma}(n, \frac{1}{\lambda}).$$

2.1.5. Superposición de procesos de Poisson homogéneos

Consideremos ahora $N_1(t), N_2(t), \dots, N_n(t)$, para $t \geq 0$, n procesos de Poisson homogéneos independientes entre sí, con tasas $\lambda_1, \lambda_2, \dots, \lambda_n$, respectivamente. Cada uno de ellos consiste en el conteo de una sucesión de eventos con una cierta tasa de arribos constante. La superposición o suma de estos procesos de Poisson es el proceso estocástico $M(t)$ dado por

$$M(t) = N_1(t) + N_2(t) + \dots + N_n(t), t \geq 0.$$

Este proceso también resulta ser un proceso de Poisson homogéneo, y la tasa de arribos correspondiente es

$$\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n.$$

No desarrollaremos la prueba aquí, aunque no es difícil comprobar que $M(t)$ cumple las propiedades dadas en la Definición 2.1. Para determinar la tasa λ de llegada, es suficiente comprobarlo para $n = 2$, es decir la superposición o suma de dos procesos de Poisson independientes. Luego podemos proceder por inducción en el número de procesos involucrados. En efecto, dado que

$N_1(t)$ y $N_2(t)$ son variables aleatorias independientes con distribución de Poisson de parámetros $\lambda_1 \cdot t$ y $\lambda_2 \cdot t$, entonces $N_1(t) + N_2(t)$ tiene una distribución de Poisson con parámetro $(\lambda_1 + \lambda_2) \cdot t$. Procediendo por inducción, se concluye que $M(t) = N_1(t) + \dots + N_n(t)$ tiene distribución de Poisson con parámetro $\lambda \cdot t = (\lambda_1 + \lambda_2 + \dots + \lambda_n) \cdot t$.

En este punto podemos analizar la respuesta a la siguiente pregunta: Dados estos n procesos, ¿cuál es la probabilidad que el primer evento que ocurra sea del proceso $N_k(t)$?

Para responder a esta pregunta usaremos la siguiente notación: Para cada proceso $N_j(t)$ denotaremos con $X_1^{(j)}$ al tiempo transcurrido hasta el primer arribo. Tendremos entonces n variables aleatorias exponenciales, $X_1^{(1)} \sim \mathcal{E}(\lambda_1)$, $X_1^{(2)} \sim \mathcal{E}(\lambda_2)$, \dots , $X_1^{(n)} \sim \mathcal{E}(\lambda_n)$, independientes entre sí. Querriamos determinar para cada k cuál es la probabilidad de que el mínimo entre estas variables aleatorias sea alcanzado por la variable $X_1^{(k)}$ o equivalentemente, que $X_1^{(k)}$ tome un valor menor o igual a las restantes $n - 1$ variables aleatorias. Tenemos que

$$\begin{aligned} P\left(\min\{X_1^{(1)}, \dots, X_1^{(n)}\} = X_1^{(k)}\right) &= P\left(X_1^{(1)} \geq X_1^{(k)}, X_1^{(2)} \geq X_1^{(k)}, \dots, X_1^{(n)} \geq X_1^{(k)}\right) \\ &= P\left(\min\{X_1^{(1)}, \dots, X_1^{(k-1)}, X_1^{(k+1)}, \dots, X_1^{(n)}\} \geq X_1^{(k)}\right). \end{aligned}$$

Ahora bien, el mínimo entre las $n - 1$ exponenciales quitando $X_1^{(k)}$, es una variable aleatoria Y con distribución exponencial de parámetro $\lambda_1 + \lambda_2 + \dots + \lambda_{k-1} + \lambda_{k+1} + \dots + \lambda_n$ y es independiente de X_k . Así, llamando f_{X_k} a la densidad de $X_1^{(k)}$, f_Y a la densidad de Y y f_{Y, X_k} a la densidad conjunta, tenemos que lo anterior es igual a

$$\begin{aligned} P(Y \geq X_1^{(k)}) &= \int_{-\infty}^{\infty} \int_{-\infty}^y f_{Y, X_k}(y, x) dx dy \\ &= \int_0^{\infty} f_Y(y) \int_0^y f_{X_k}(x) dx dy \\ &= \int_0^{\infty} \lambda_Y e^{-\lambda_Y y} (1 - e^{-\lambda_k y}) dy \\ &= 1 - \lambda_Y \cdot \frac{1}{\lambda_Y + \lambda_k} = \\ &= \frac{\lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_n} \end{aligned}$$

Concluimos entonces que la probabilidad de que el mínimo entre n variables aleatorias exponenciales independientes, X_1, X_2, \dots, X_n , sea la variable $X_1^{(k)}$ es proporcional a λ_k . Específicamente está dado por

$$P\left(\min\{X_1^{(1)}, \dots, X_1^{(n)}\} = X_1^{(k)}\right) = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_n}.$$

Este valor es también la probabilidad de que el primer evento que ocurra de los n procesos de Poisson provenga del proceso $N_k(t)$ con tasa de llegada λ_k .

Ejemplo 2.3. Supongamos que en una estación de tren llegan tres líneas de trenes: Naranja, Amarilla y Verde. Los arribos de estos trenes constituyen cada uno un proceso de Poisson homogéneo con tasas de llegada de un tren cada 15 minutos, un tren cada 10 minutos y un tren cada 20 minutos, respectivamente. Un pasajero llega a la estación de tren y puede tomar cualquiera de estos trenes para ir a su destino.

- ¿Cuál es el tiempo mínimo promedio que debe esperar hasta que llegue el primer tren?
- ¿Cuál es la probabilidad de que el primer tren que llegue sea de la línea Naranja?

Para resolver este problema, recordamos que podemos situar el origen de los procesos al momento de que el pasajero llega al tren. Denotamos X_N , X_A y X_V las variables aleatorias que representan el tiempo de arribo del primer tren de la línea Naranja, Amarilla y Verde, respectivamente. Estas variables aleatorias tienen distribución exponencial con parámetro $\lambda_N = \frac{1}{15}$ para el caso de X_N , $\lambda_A = \frac{1}{10}$ para el caso de X_A y $\lambda_V = \frac{1}{20}$ para X_V . Si consideramos el proceso dado por la superposición de los tres procesos de arribo, tenemos que la tasa de llegada λ es la suma de las tasas de arribo de cada una de las líneas, esto es:

$$\lambda = \frac{1}{10} + \frac{1}{15} + \frac{1}{20} = \frac{13}{60},$$

es decir, 13 trenes cada 60 minutos. Dado que $\frac{1}{\lambda}$ es el valor esperado del tiempo de arribo del primer tren, tenemos que el tiempo promedio de espera es $\frac{60}{13} = 4.615$ minutos.

Para determinar la probabilidad de que el primer tren sea de la línea Naranja calculamos:

$$P(\text{primer tren de la línea Naranja}) = \frac{\frac{1}{15}}{\frac{1}{10} + \frac{1}{15} + \frac{1}{20}} = \frac{4}{13} = 0.31.$$

2.1.6. Refinamiento de procesos de Poisson homogéneos

Relacionado con la superposición está el concepto de refinamiento (thinning) de un proceso de Poisson homogéneo. En este caso se tiene un proceso de conteo $M(t)$, $t \geq 0$, con tasa de llegada λ , e independientemente del proceso $M(t)$ cada evento se clasifica del tipo k con una cierta probabilidad p_k , para $1 \leq k \leq n$ y $p_1 + p_2 + \dots + p_n = 1$. Entonces pueden definirse n procesos de conteo $N_1(t)$, $N_2(t)$, \dots , $N_n(t)$, donde $N_j(t)$ es el número de eventos del tipo j hasta el tiempo t .

Podemos observar que para cada j , $1 \leq j \leq n$, $N_j(t)$ cumple las propiedades de un proceso de Poisson homogéneo. La propiedad de tener incrementos independientes y estacionarios es heredada del proceso $M(t)$. En este caso se verifica que $N_j(t)$ es un proceso de Poisson homogéneo con tasa de arribos $\lambda \cdot p_j$. En efecto, notemos que en un intervalo de longitud h , con h muy pequeño, puede ocurrir uno o ningún evento del proceso $M(t)$. Veamos cuál es la probabilidad de que en este intervalo ocurra un evento del tipo j :

$$P(N_j(t+h) - N_j(t) = 1) = P(M(t+h) - M(t) = 1 \text{ y el evento sea del tipo } j),$$

y como la clasificación es independiente del proceso, esto es igual a

$$P(M(t+h) - M(t) = 1) \cdot p_j = (\lambda h) \cdot p_j = (\lambda \cdot p_j) h.$$

Luego la tasa de llegada del proceso N_j es $\lambda \cdot p_j$.

Ejemplo 2.4. Supongamos que en una carretera pasan vehículos por un cierto punto de acuerdo a un proceso de Poisson homogéneo, con una tasa de 200 vehículos por hora. Hay una probabilidad del 20 % que un vehículo sea un camión, un 70 % que sea un automóvil y un 10 % que sea una moto. Luego se pueden definir tres procesos de conteo: $N_1(t)$ el conteo de camiones, $N_2(t)$ el de automóviles y $N_3(t)$ el de motos. Se cumple que $N_1(t)$ es un proceso de Poisson homogéneo con tasa de llegada $\lambda_1 = 200 \cdot 0.20 = 40$ camiones por hora, $N_2(t)$ tiene tasa de llegada $\lambda_2 = 200 \cdot 0.7 = 140$ automóviles por hora y $N_3(t)$ tiene tasa de llegada $\lambda_3 = 200 \cdot 0.1 = 20$ motos por hora.

Se puede probar además que los n procesos $N_1(t), N_2(t), \dots, N_n(t)$ resultan independientes entre sí. Notemos entonces que con esta propiedad de independencia se puede ver que la superposición de estos n procesos produce el proceso de Poisson homogéneo original $M(t)$.

2.2. Procesos de Poisson no homogéneos

Los procesos de Poisson homogéneos asumen que la tasa de arribos en distintos intervalos de tiempo sólo depende de la longitud de ese período de tiempo. Por ejemplo, si se quisiera modelar el número de llegadas de clientes a un banco se estaría suponiendo que en cualquier hora del día la tasa de llegadas es la misma. Si en cambio se quiere tomar la hipótesis de que el promedio de llegadas varía en distintas horas del día es conveniente introducir una función del tiempo para modelar la tasa de arribos.

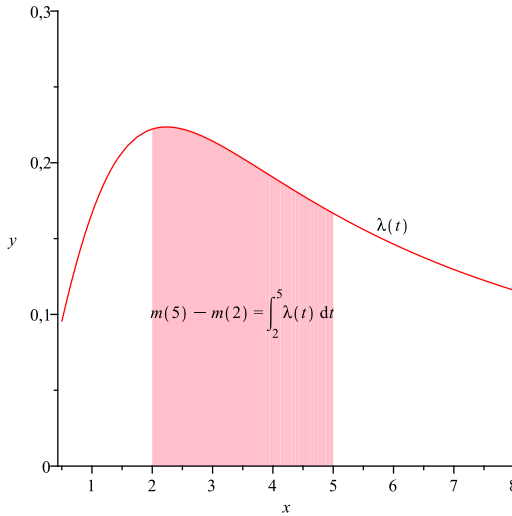
Definición 2.2. Un proceso $N(t), t \geq 0$ es un **proceso de Poisson no homogéneo** con función de intensidad $\lambda(t), t \geq 0$, si:

- a) $N(0) = 0$
- b) para cada $n \geq 1$ y cada partición $0 < t_0 < t_1 < \dots < t_n$ se tiene que $N(t_0), N(t_1) - N(t_0), \dots, N(t_n) - N(t_{n-1})$ son variables aleatorias independientes.
- c) $\lim_{h \rightarrow 0} \frac{P(N(t+h) - N(t) = 1)}{h} = \lambda(t),$
- d) $\lim_{h \rightarrow 0} \frac{P(N(t+h) - N(t) \geq 2)}{h} = 0.$

La función **valor medio del proceso** mide la intensidad media del número de llegadas en un intervalo. Está dada por:

$$m(t) = \int_0^t \lambda(s) ds$$

Notemos que en este caso los incrementos son independientes pero no estacionarios, ya que la distribución de $N(t+s) - N(t)$ dependerá de la función de intensidad λ en el período $(t, t+s]$. Por otra parte si $\lambda(t) = \lambda$, constante, entonces $m(t) = \lambda \cdot t$ y es el caso del proceso de Poisson homogéneo.



En particular, se tiene que para cada $t \geq 0$ y $s > 0$, el número de llegadas en el intervalo $(t, t+s]$ es una variable aleatoria Poisson con media $m(t, t+s) = m(t+s) - m(t)$:

$$m(t, t+s) = m(t+s) - m(t) = \int_t^{t+s} \lambda(x) dx.$$

Es decir:

$$P(N(t+s) - N(t) = j) = e^{-m(t,t+s)} \cdot \frac{(m(t,t+s))^j}{j!}.$$

Ejemplo 2.5. Los clientes llegan a una tienda de acuerdo a un proceso de Poisson no homogéneo con función de intensidad

$$\lambda(t) = \begin{cases} 2t & 0 \leq t < 1 \\ 2 & 1 \leq t < 2 \\ 4-t & 2 \leq t \leq 4, \end{cases}$$

donde t se mide en horas. ¿Cuál es la probabilidad de que lleguen dos clientes en las dos primeras horas y tres clientes en las dos horas siguientes?

En este caso se pide calcular:

$$P(N(2) = 2, N(4) - N(2) = 3).$$

Dado que los incrementos son independientes, se puede calcular separadamente $P(N(2) = 2)$ y $P(N(4) - N(2) = 3)$. Para las dos primeras horas tenemos que la intensidad media es:

$$m(0, 2) = \int_0^1 2s \, ds + \int_1^2 2 \, ds = 3,$$

y para las siguientes dos horas es:

$$m(2, 4) = \int_2^4 4 - s \, ds = 2.$$

Por lo tanto:

$$\begin{aligned} P(N(2) = 2) &= e^{-3} \frac{3^2}{2!} = \frac{9}{2e^3} \simeq 0.224 \\ P(N(4) - N(2) = 3) &= e^{-2} \frac{2^3}{3!} = \frac{8}{6e^2} \simeq 0.18. \end{aligned}$$

Así, la probabilidad de que lleguen dos clientes en las primeras dos horas y tres clientes en las dos siguientes es:

$$P(N(2) = 2) \cdot P(N(4) - N(2) = 3) = \frac{6}{e^5} \simeq 0.04.$$

El siguiente resultado será útil para la simulación de procesos de Poisson no homogéneos.

Proposición 2.1. Sea $N(t)$ el número de eventos ocurridos hasta el tiempo t en un proceso de Poisson homogéneo con intensidad λ . Supongamos que en tiempo t un evento es contado con probabilidad $p(t)$, independientemente de lo ocurrido hasta ese instante. Entonces el proceso de conteo de estos eventos $M(t)$ es un proceso de Poisson no homogéneo con intensidad $\lambda(t) = \lambda \cdot p(t)$.

Para comprobar que cumple las propiedades de un proceso de Poisson no homogéneo, notemos que $M(0) = 0$ ya que $N(0) = 0$. Los incrementos son independientes por ser una propiedad de $N(t)$ y porque $p(t)$ es independiente de lo ocurrido hasta el tiempo t . Si $(t, t + h)$ es un intervalo de tiempo pequeño, entonces:

$$P(M(t+h) - M(t) = 1) = P(N(t+h) - N(t) = 1 \text{ y este evento sea contado}) \simeq (\lambda h) \cdot p(t).$$

Por último, la probabilidad de que ocurran dos o más eventos en un período de amplitud h tiende a 0 para un h pequeño ya que también es una propiedad del proceso $N(t)$.

En particular, si $N(t)$ es un proceso de Poisson no homogéneo con función de intensidad $\lambda(t)$ y $\lambda \in \mathbb{R}$ es una constante tal que

$$\lambda(t) \leq \lambda,$$

para todo t , entonces $N(t)$ puede verse como el proceso de contar eventos de un proceso de Poisson homogéneo con intensidad λ donde los eventos son contados con probabilidad

$$p(t) = \frac{\lambda(t)}{\lambda}.$$

Capítulo 3

Números aleatorios

3.1. Introducción

En simulación estocástica los generadores de números (pseudo)aleatorios con distribución uniforme en el intervalo $[0, 1]$ son empleados para diversos usos:

- en forma directa, es decir, porque se desea obtener valores uniformemente distribuidos en $[0, 1]$,
- para generar muestras de otras variables aleatorias, con distribuciones discretas o continuas,
- para generar muestras de un conjunto de variables aleatorias dependientes, como por ejemplo procesos estocásticos y distribuciones multivariadas.

La ejecución de una simulación está fuertemente correlacionada con el generador de variables aleatorias uniformes utilizado, por lo cual es importante garantizar las buenas propiedades del generador. Dado que los valores de la variable son generados a partir de una secuencia de pasos o algoritmo, sería más preciso referirse a **generadores de números pseudoaleatorios**.

3.2. Propiedades de un generador de números aleatorios

Por generador de números pseudoaleatorios entenderemos un algoritmo capaz de producir secuencias de números:

$$u_1, u_2, \dots, u_N,$$

que sean realizaciones de muestras de tamaño N de una variable aleatoria uniforme $U \sim \mathcal{U}(0, 1)$. Para ello debemos acordar qué entendemos por una *muestra* de esta variable.

Definición 3.1. Una N -upla de variables aleatorias (U_1, U_2, \dots, U_N) es una muestra de tamaño N de una variable aleatoria uniforme en $(0, 1)$ si cumple:

1. Para cada $i = 1, \dots, N$ y cada $u \in \mathbb{R}$ vale

$$\mathbb{P}(U_i \leq u) = \begin{cases} 0 & \text{si } u < 0 \\ u & \text{si } 0 \leq u \leq 1 \\ 1 & \text{si } u > 1. \end{cases}$$

2. Para cada k , $2 \leq k \leq N$ y cada k -upla (i_1, i_2, \dots, i_k) con $1 \leq i_1 < \dots < i_k \leq N$ vale que

$$\mathbb{P}(U_{i_1} \leq u_1, \dots, U_{i_k} \leq u_k) = \mathbb{P}(U_{i_1} \leq u_1) \dots \mathbb{P}(U_{i_k} \leq u_k),$$

cualesquiera sean u_1, \dots, u_k ,

Esta definición significa que no sólo las secuencias de números ($k = 1$) deben estar uniformemente distribuidas en $(0, 1)$, sino que además los pares de números generados (u_{i_1}, u_{i_2}) deben distribuirse uniformemente en un cuadrado de lado 1, las ternas $(u_{i_1}, u_{i_2}, u_{i_3})$ en un cubo de lado 1, y así siguiendo.

Además de satisfacer las propiedades (1) y (2), un buen generador debería satisfacer, en la mejor medida posible, las dos siguientes propiedades:

- a) **repetibilidad y portabilidad**, y
- b) **velocidad computacional**.

Por **repetibilidad** se entiende que, si en ocasiones repetidas se dan los mismos parámetros, el generador debe producir la misma sucesión. Esta propiedad garantiza que los resultados de una simulación sean confiables.

Portabilidad significa que, sobre las mismas condiciones de definición, la sucesión sea la misma, independientemente del lenguaje computacional usado para implementar el algoritmo de generación, y de la computadora utilizada. Esta propiedad suele ser difícil de alcanzar, pero aún así es un aspecto deseable.

La **velocidad computacional** está estrictamente ligada a la precisión deseada en los resultados finales de la simulación para la cual el generador es utilizado. Cuanto más rápido sea el generador, más resultados serán obtenidos en el mismo tiempo de uso del computador.

De hecho, toda vez que se realiza una simulación, se acepta una solución de compromiso entre los requerimientos expuestos anteriormente, pero siempre los prioritarios debieran ser las propiedades (1) y (2) de la Definición 3.1.

3.2.1. Breve reseña histórica

Los primeros científicos y personas interesadas en obtener secuencias de números aleatorios usaron procedimientos físicos. Existe una larga historia de experimentos de simulación basados

en tiradas de monedas o dados, ruletas, que aún hoy son usados en juegos tales como bingos, casinos, loterías, etc. Sin embargo estos procedimientos carecen de la propiedad de repetibilidad.

Probablemente uno de los primeros trabajos serios en generación de números aleatorios fue la tabla de Tippett (1927), que consistía de 10.400 números de cuatro dígitos a partir de 41.600 dígitos aleatorios tomados de datos censales. Lamentablemente no satisfacían ni (1) ni (2) de la Definición 3.1. La primera máquina generadora de números aleatorios fue usada en 1939 por Kendall y Babington-Smith con la cual produjeron una tabla de 100.000 dígitos aleatorios agrupados de a cuatro, formando 25.000 números de cuatro dígitos. También en 1955 la RAND Corporation utilizó extensamente una tabla de 1.000.000 de dígitos aleatorios que fue obtenida a partir de una ruleta electrónica especialmente diseñada. Si bien estos métodos recibieron cierta aceptación, por un lado por satisfacer ciertos tests y/o porque funcionaban en la práctica, tienen la desventaja de no cumplir la condición de repetibilidad. Por lo tanto sólo pueden ser usados en situaciones donde la repetibilidad no es lo buscado, por ejemplo en loterías; o en ciertas aplicaciones estadísticas, como en muestreos aleatorios de poblaciones no muy grandes.

3.3. Principios generales

Al definir un generador, hay ciertos **principios generales** que deben satisfacerse, además de las condiciones que ya hemos marcado:

- P1)** La secuencia generada debe ser intuitivamente aleatoria.
- P2)** Esa aleatoriedad debe ser establecida teóricamente o, al menos, debe pasar ciertos tests de aleatoriedad. La aleatoriedad de una secuencia jamás debe ser asumida sin esas verificaciones.
- P3)** Debe conocerse algo sobre las propiedades teóricas del generador.

El principio **P1** es ciertamente razonable. No debería ser posible poder anticipar cuál es el siguiente número en una secuencia si ésta es aleatoria. El segundo principio (**P2**) se refiere a que deben satisfacerse tests estadísticos en relación a las condiciones (1) y (2) de la Definición 3.1. La tercera premisa, (**P3**), es para poder garantizar de manera teórica la condición de repetibilidad del generador.

Estos principios generales indican que los generadores ad hoc deben ser evitados. Es decir, no cualquier algoritmo que genere números *arbitrariamente* entre 0 y 1 debe considerarse un buen generador.

Ejemplo 3.1. Uno de los primeros trabajos que sugieren un método bien definido de generación de una secuencia determinística intentando imitar una secuencia aleatoria, fue de von Neumann. Este método conocido como "mid square" puede ser escrito de la siguiente manera:

1. Sea X_0 un número entero de 4 dígitos decimales (puede ser que los dígitos de más a la izquierda sean 0). Hacer $i = 0$.
2. Calcular X_i^2 . El resultado es un número de 8 dígitos: $X_i^2 = d_8d_7d_6d_5d_4d_3d_2d_1$, con d_j en el conjunto $\{0, 1, \dots, 9\}$, $0 \leq j \leq 7$.
3. Definir $X_{i+1} = d_6d_5d_4d_3$, esto es, los cuatro dígitos centrales o "middle".
4. Hacer $i = i + 1$ y seguir en 2.

```
def vonNeumann(u):
    u = (u**2 // 100) % 10000
    return u
```

Este método es un ejemplo de lo que llamaríamos un **mal generador de números aleatorios**. Veamos por qué.

Aunque ciertas secuencias obtenidas sean bien aleatorias y se repitan sólo después de un número bien grande de términos, sucede que no se conocen bien las propiedades del generador. En particular, pareciera que las secuencias que se obtienen dependen fuertemente del valor inicial X_0 , o semilla. Los siguientes ejemplos muestran algunas secuencias obtenidas a partir de este algoritmo, donde el primer valor es la semilla dada:

- a) **4010** 801 6416 1650 7225 2006 240 576 3317 24 5 0 0...
- b) **2100** 4100 8100 6100 2100 4100 8100 6100 2100 4100 8100...
- c) **3792** 3792 3792 3792 3792 3792...
- d) **1234** 5227 3215 3362 3030 1809 2724 4201 6484 422 1780...

En (a) puede verse que la secuencia degenera en el valor 0 luego de algunas iteraciones, en (b) la secuencia alterna entre solo cuatro valores y en (c) los valores se repiten paso tras paso.

3.4. Generadores congruenciales

Es claro que con una computadora no es posible generar cualquier número real, en particular si posee infinitas cifras decimales. Los generadores que analizaremos a continuación producen en realidad una secuencia de números enteros

$$y_1, y_2, \dots, y_N, \quad y_j \in \{0, 1, \dots, M-1\},$$

para un cierto entero positivo M "grande", y a partir de esta muestra se considera la sucesión de números en $[0, 1)$ como

$$u_1 = \frac{y_1}{M}, u_2 = \frac{y_2}{M}, \dots, u_N = \frac{y_N}{M}, \dots$$

Nos ocuparemos entonces de generar secuencias de números enteros:

$$y_n = f(y_{n-1}) \quad \text{mód } M,$$

para algún M entero positivo, y comenzando de un valor inicial (semilla) y_0 .

3.4.1. Generadores congruenciales lineales

Definición 3.2. Sea M un entero positivo, $M \geq 2$. Una sucesión $y_1, y_2, \dots, y_n, \dots$ con valores en $\{0, 1, \dots, M-1\}$ se dice **generada por el generador congruencial lineal con parámetros a, c y M y semilla y_0** si

$$y_n = (ay_{n-1} + c) \quad \text{mód } M, \quad n \geq 1,$$

donde a, c e y_0 son enteros del conjunto $\{0, \dots, M-1\}$.

En la terminología usual a se dice un **multiplicador**, c es el **incremento** y M es el **módulo**. Si $c \neq 0$ el generador se dice **mixto** y si $c = 0$ se dice **multiplicativo**.

```
def randMixto(a, c, M, u) :
    return (a*u+c) % M

def randMulti(a, M, u) :
    return (a*u) % M
```

Ejemplo 3.2. La secuencia

$$0, 1, 6, 15, 12, 13, 2, 11, 8, 9, \dots$$

fue generada por un generador congruencial lineal. El lector, ¿podría decir cual será el próximo número (entre 0 y 15)?

Ejemplo 3.3. La secuencia

$$1, 12, 1, 12, 1, 12, 1, 12, 1, 12, \dots$$

ha sido generada por otro generador congruencial. ¿Podría decirse que genera una secuencia intuitivamente aleatoria?

Los ejemplos anteriores muestran que no siempre una secuencia generada por un generador congruencial lineal resulta intuitivamente aleatoria. En principio pareciera que cuanto más números distintos aparecen en la secuencia hasta obtener un valor repetido, más impredecible será el próximo número. Aunque esto no es del todo cierto, ya que una secuencia del tipo $1, 2, 3, \dots, M-1$, es totalmente predecible.

Es claro que la secuencia puede tener a lo sumo M números diferentes, y que si un número se repite entonces también se repite la secuencia que sigue a ese número. Entonces, si K es el menor número tal que

$$y_n = y_{n+K}, \quad \text{para todo } n \geq N_0,$$

para algún N_0 , diremos que K es el **período** de la secuencia y_0, y_1, \dots . Claramente, $K \leq M$.

Pregunta: Ahora, ¿cómo escoger a , c , M y y_0 para obtener las secuencias con mayor período K posible?

Aún consiguiendo un período K grande, es importante elegir a , c y la semilla y_0 de modo que las secuencias tengan un comportamiento lo más aleatorio posible. Por ejemplo, ¿qué ocurre si elegimos M arbitrario, $a = c = 1$ e $y_0 = 0$? En este caso obtenemos la secuencia

$$0, 1, 2, 3, \dots, M-1,$$

que es de período completo (esto es, $K = M$), pero no es siquiera intuitivamente aleatoria.

Para el caso de generadores congruenciales lineales existen varios resultados teóricos que dan pautas de cómo obtener secuencias con períodos grandes. La aleatoriedad debe ser testeada aparte, esto es, ninguno de estos resultados garantizan que se satisfagan los tests de aleatoriedad.

3.4.2. Generadores mixtos

Algo a notar es que, si el generador es multiplicativo ($c = 0$), entonces un buen generador no debería alcanzar nunca el valor 0, de lo contrario la secuencia degeneraría en una sucesión infinita de ceros. Por lo tanto, para obtener un período máximo, esto es $K = M$, necesariamente debe ser un generador mixto.

Teorema 3.1. Consideremos una secuencia dada por el generador:

$$y_{i+1} = a y_i + c \pmod{M}, \quad c \neq 0.$$

Entonces la secuencia tiene período M si y sólo si se cumplen todas las siguientes condiciones:

- El máximo común divisor entre c y M es 1: $(c, M) = 1$.
- $a \equiv 1 \pmod{p}$, para cualquier factor primo p de M .
- Si 4 divide a M , entonces $a \equiv 1 \pmod{4}$.

La demostración puede encontrarse en [?]. Damos un ejemplo y sugerimos al lector pensar en otros casos que respondan a este teorema.

Ejemplo 3.4. Consideremos $M = 16$, $c = 3$, $a = 5$. Notemos que si M es una potencia de 2, es relativamente fácil encontrar valores para c y a .

$$y_{n+1} = 5y_n + 3 \pmod{16}.$$

Entonces:

- Se cumple que $(c, M) = (3, 16) = 1$.
- Se cumple que $5 \equiv 1 \pmod{2}$, donde 2 es el único primo que divide a M .
- 4 divide a 16, y $5 \equiv 1 \pmod{4}$.

Si se cumplen todas las hipótesis, debería obtenerse una secuencia máxima. Tomamos $y_0 = 0$.

$$y_1 = 5 \cdot 0 + 3 \pmod{16}, \quad y_1 = 3,$$

$$y_2 = 5 \cdot 3 + 3 \pmod{16}, \quad y_2 = 2,$$

siguiendo el razonamiento la secuencia continúa:

$$\dots \quad 13 \quad 4 \quad 7 \quad 6 \quad 1 \quad 8 \quad 11 \quad 10 \quad 5 \quad 12 \quad 15 \quad 14 \quad 9 \quad 0 \quad 3 \quad 2 \dots$$

Notemos en particular que, si $M = 2^n$, entonces a debe ser de la forma $4m + 1$ y c debe ser impar. Una ventaja de tomar M igual a una potencia de 2, digamos 2^n , es que computacionalmente tomar módulo equivale a considerar los últimos n bits de la representación.

Ejemplo 3.5. En [?] se presenta el siguiente ejemplo de un modelo de generador provisto por bibliotecas de ANSI C, y que responde a un tipo de generador lineal congruencial mixto. En este caso se toma:

$$a = 1103515245, \quad c = 12345, \quad M = 2^{32}.$$

Así en este caso el período de la secuencia es $K = 2^{32} = 4\,294\,967\,296$.

3.4.3. Generadores multiplicativos

Una desventaja de un generador mixto es que para cada valor generado se efectúa una suma, operación que no se realiza para un generador multiplicativo. Si bien en este caso el período de la secuencia puede ser a lo sumo $M - 1$, es preferible usar este tipo de generadores para ganar en velocidad y costo operacional.

Veamos entonces qué condiciones debe tener un generador multiplicativo para poder obtener un período máximo, $K = M - 1$. Antes de presentar un resultado relacionado con estos generadores, definimos el concepto de **raíz primitiva** de un número natural.

Definición 3.3. Sea M un número natural. Se dice que a es una **raíz primitiva** de M si

$$a^{(M-1)/p} \not\equiv 1 \pmod{M}$$

para cualquier factor primo p de $M - 1$.

Con esta definición, enunciamos el siguiente resultado:

Teorema 3.2. Para un generador multiplicativo

$$y_{i+1} = a y_i \pmod{M},$$

el período K de la secuencia verifica las siguientes tres propiedades:

- Si $K = M - 1$ entonces M es primo.
- Si M es primo, entonces K divide a $M - 1$.
- $K = M - 1$ si y sólo si a es raíz primitiva de M y M es primo.

Entonces el Teorema 3.2 nos da pautas para encontrar generadores multiplicativos con período máximo: determinar un número primo M y una raíz primitiva a de M . Cuanto mayor sea M mayor será el período de la secuencia.

Ejemplo 3.6. Si tomamos $M = 19$, entonces $a = 2$ es una raíz primitiva de M . Para ver esto, notemos que $M - 1 = 18$ es divisible por los primos 2 y 3. Como $a^{18/2} = 2^9$, $a^{18/3} = 2^6$, y 19 no divide a $2^9 - 1$ ni a $2^6 - 1$, entonces 2 es raíz primitiva de $M = 19$.

Si tomamos como semilla $y_0 = 1$ tenemos la secuencia:

1 2 4 8 16 13 7 14 9 18 17 15 11 3 6 12 5 10 1...

que tiene período 19.

Notemos en el Ejemplo 3.6, que los primeros elementos de la secuencia no parecen tan aleatorios. Esto se debe a que la raíz primitiva elegida, 2, es muy pequeña. Luego es conveniente tomar una raíz primitiva más grande para superar más rápidamente el valor de M . Afortunadamente, existen resultados que nos permiten encontrar otras raíces primitivas a partir de una conocida. En particular, si a es raíz primitiva y $(m, M - 1) = 1$, entonces a^m es raíz primitiva.

Entonces, si en el Ejemplo 3.6 tomamos $a = 2^5 = 13 \pmod{19}$, la secuencia obtenida es:

1 13 17 12 4 14 11 10 16 18 6 2 7 15 5 8 9 3 1 ...

Ejemplo 3.7. En [?] se menciona el generador congruencial multiplicativo con

$$a = 7^5 = 16807 \quad M = 2^{31} - 1 = 2\,147\,483\,647.$$

En este caso, M es un **primo de Mersenne**, es decir, de la forma $2^k - 1$. La factorización en primos de $M - 1$ está dada por:

$$M - 1 = 2^{31} - 2 = 2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331.$$

Dado que 7 es raíz primitiva de M y $(5, M - 1) = 1$, entonces $7^5 = 16807$ es raíz primitiva, y la secuencia generada tiene período:

$$K = M - 1 = 2\,147\,483\,646.$$

3.4.4. El problema de los hiperplanos

Una de las **desventajas** de cualquier generador congruencial lineal, sea mixto o multiplicativo es que no se cumple tan satisfactoriamente la condición (2) de la Definición 3.1.

Para el caso de los generadores congruenciales lineales, ocurre lo siguiente. Si la secuencia producida es:

$$y_0, y_1, y_2, \dots, y_n, \dots,$$

está demostrado que los puntos $(y_j, y_{j+1}, \dots, y_{j+k-1})$, $j = 0, 1, 2, \dots$ están ubicados en no más de

$$(k!M)^{1/k} = (k!)^{1/k} \sqrt[k]{M}$$

hiperplanos paralelos en \mathbb{R}^k . ¿Qué nos dice esto? Para un k suficientemente grande, los puntos quedan ubicados en una cantidad finita de hiperplanos, y por lo tanto existen "frangas" en las que no cae ninguna k -upla. La Figura 3.1 ilustra esta situación para pares ordenados ($k = 2$) en el caso de algunos generadores congruenciales lineales.

Un mal generador y ejemplo de este problema es **RANDU**. Este es un generador del tipo congruencial lineal multiplicativo ($c = 0$), con $M = 2^{31}$ y $a = 2^{16} + 3$:

$$y_n = 65539 \cdot y_{n-1} \pmod{2\,147\,483\,648}.$$

Las ternas generadas por RANDU se ubican en 15 planos paralelos, dentro del cubo de lado 1. Figura 3.2. Cabe mencionar que este generador fue implementado en computadoras de IBM por mucho tiempo, y difundido a otros sistemas.

El período de la secuencia para $y_0 = 1234567890$ es $K = 268.435.466$ y para $y_0 = 1$ es 536.870.912.

3.4.5. Generadores congruenciales lineales combinados

Una forma de producir *mayor aleatoriedad* en una secuencia, y evitar la desventaja de los hiperplanos mencionada en la sección anterior, es utilizar más de un generador congruencial lineal y combinarlos entre ellos. Por combinar entendemos sumarlos o restarlos, y en general se recomienda la resta.

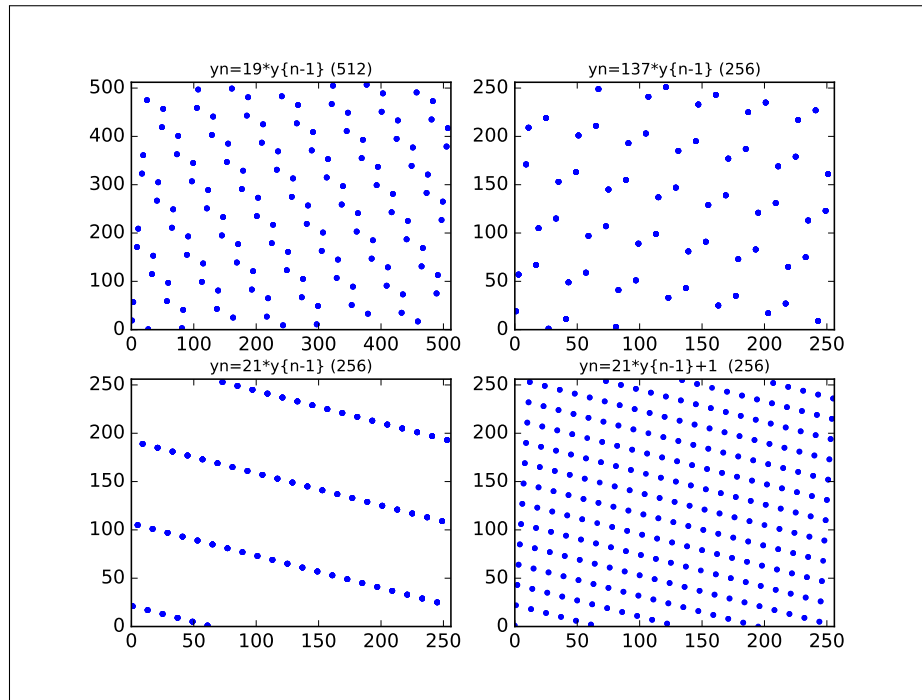


Figura 3.1: Generadores congruenciales. Distribución de pares (y_i, y_{i+1})

Ahora bien, ¿es cierto que si sumamos dos variables aleatorias uniformes obtenemos otra distribución uniforme? La respuesta es NO, y podemos verlo claramente arrojando dos dados. Supongamos que sus caras son igualmente probables, entonces las sumas no lo son: el 7 es más probable que el 2, por ejemplo:

$$2 = 1 + 1 \qquad 3 = 1 + 2 = 2 + 1 \qquad 4 = 1 + 3 = 2 + 2 = 3 + 1$$

Sin embargo, si sumamos las caras y consideramos el resultado módulo 6, entonces, vemos que:

$$\begin{aligned} 1 &= 1 + 6 = 2 + 5 = 3 + 4 = 4 + 3 = 5 + 2 = 6 + 1 && \text{mód } (6) \\ 2 &= 2 + 6 = 3 + 5 = 4 + 4 = 5 + 3 = 6 + 2 = 1 + 1 && \text{mód } (6) \\ 3 &= 3 + 6 = 4 + 5 = 5 + 4 = 6 + 3 = 2 + 1 = 1 + 2 && \text{mód } (6) \\ 4 &= 4 + 6 = 5 + 5 = 1 + 3 = 2 + 2 = 3 + 1 = 6 + 4 && \text{mód } (6) \\ 5 &= 5 + 6 = 6 + 5 = 1 + 4 = 2 + 3 = 3 + 2 = 4 + 1 && \text{mód } (6) \\ 0 &= 6 + 6 = 1 + 5 = 2 + 4 = 3 + 3 = 4 + 2 = 5 + 1 && \text{mód } (6) \end{aligned}$$

Esto es, ahora cada suma tiene la misma probabilidad de ocurrir, y por lo tanto **la suma módulo 6** tiene distribución uniforme.

Ejercicio 3.1. ¿Qué pasaría si uno de los dados tuviera 5 caras, y consideráramos las sumas módulo 6?, ¿y si se consideran módulo 5?

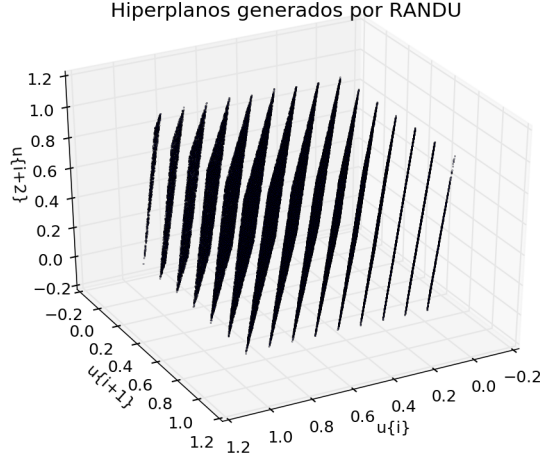


Figura 3.2: Generador RANDU

Este resultado se generaliza en el siguiente teorema:

Teorema 3.3. Sean W_1, W_2, \dots, W_n variables aleatorias discretas independientes, tales que $W_1 \sim U(\{0, d-1\})$ para cierto $d \geq 1$. Entonces

$$W = \left(\sum_{j=1}^n W_j \right) \text{ mód } d$$

es una variable aleatoria uniforme discreta en $\{0, d-1\}$.

Notemos que sólo se requiere que una de las variables aleatorias sea uniforme en $\{0, d-1\}$. La demostración puede encontrarse en [?] y es la siguiente. Es suficiente considerar $n = 2$, ya que $W_2 + \dots + W_n$ es una variable aleatoria discreta. Entonces, para cada k , $0 \leq k \leq d-1$

$$P(W_1 + W_2 \equiv k \text{ mód } d) = P(W_1 + W_2 = k + m \cdot d, \text{ para algún } m).$$

Pero para cada $m \in \mathbb{Z}$,

$$P(W_1 + W_2 = k + m \cdot d) = \sum_{i=0}^{d-1} P(W_1 = i) P(W_2 = k - i + m \cdot d) = \frac{1}{d} \sum_{i=0}^{d-1} P(W_2 = k - i + m \cdot d).$$

Sumando sobre todos los valores de m posibles, resulta

$$P(W_1 + W_2 \equiv k \text{ mód } d) = \frac{1}{d}.$$

Con este resultado vemos que si en particular se suman dos o más generadores congruenciales, tomando módulo uno de los módulos de ellos se obtiene un nuevo generador. ¿Cuál es la ventaja de hacer esto? Recordemos que los generadores congruenciales tienen un determinado período. Si sumamos dos generadores con un mismo período K , entonces K será el período de la suma. Pero si los períodos son diferentes, entonces la suma tiene un período mayor.

Este resultado se basa en el siguiente teorema, y su demostración puede verse en [?].

Teorema 3.4. Consideremos una familia de N generadores, donde para cada j , $j = 1, 2, \dots, N$, el generador j tiene período K_j y evoluciona de acuerdo a una ley:

$$y_{n,j} = f_j(y_{n-1,j}), \quad n \geq 1, \quad y_0 = \text{semilla del generador } j.$$

Entonces el período K de la secuencia

$$s_n = (y_{n,1}, y_{n,2}, \dots, y_{n,N}), \quad n \geq 1,$$

es igual al mínimo común múltiplo de K_1, K_2, \dots, K_N .

Así, los Teoremas 3.3 y 3.4 constituyen la base teórica que nos permite obtener nuevos generadores congruenciales de v.a. uniformes a partir de la suma o resta de dos o más generadores. Además, con una buena elección de los períodos de los generadores se podrá garantizar un período mucho más largo para la combinación.

Ejemplo 3.8. Un generador de estas características es el propuesto en [?]. Se consideran los generadores:

$$x_n = 40014x_{n-1} \quad \text{mód } 2^{31} - 85$$

$$y_n = 40692y_{n-1} \quad \text{mód } 2^{31} - 249$$

Los períodos de estos generadores tienen un solo 2 como factor común:

$$2^{31} - 86 = 2 \cdot 3 \cdot 7 \cdot 631 \cdot 81031, \quad 2^{31} - 250 = 2 \cdot 19 \cdot 31 \cdot 1019 \cdot 1789,$$

por lo que el período de la secuencia $(x_n - y_n) \text{ mód } M$ (para cualquiera de los módulos M) es del orden del producto de los dos períodos dividido 2. En este caso,

$$K \approx 2^{61} = 2\,305\,843\,009\,213\,693\,952 \sim 2.3 \times 10^{18}.$$

La Figura 3.3 ilustra una muestra de aproximadamente 5000 puntos (U_i, U_{i+1}) para cada uno de los tres generadores, con $U_i < 0.001$.

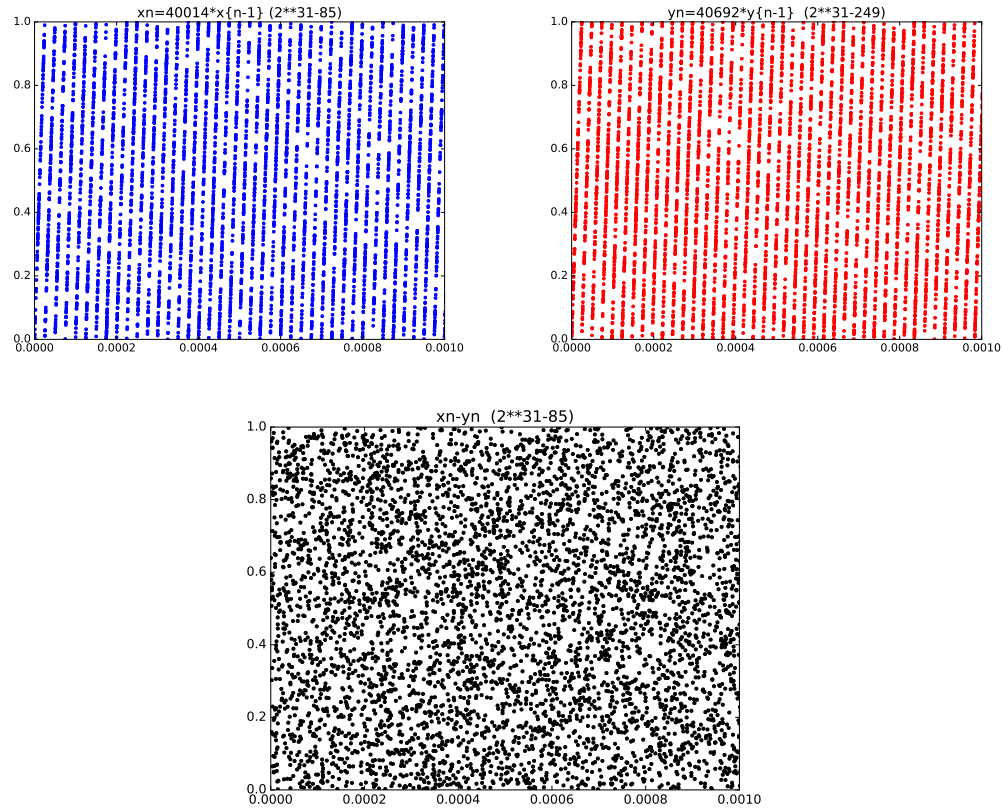


Figura 3.3: Generador del Ejemplo 3.8.

3.4.6. Otros generadores eficientes y portables

Existen otras formas de mejorar la eficiencia de los generadores construidos a partir de un generador congruencial, como son del tipo Fibonacci, resta con préstamo, suma con acarreo, y otros tantos. En [?] se presenta una extensa lista de generadores candidatos a ser combinados para obtener aún mejores generadores. Algunos son los siguientes:

Módulo	Secuencia	Período
2^{32}	$x_n = 69069 x_{n-1} + \text{impar}$	2^{32}
2^{32}	$x_n = x_{n-1} * x_{n-2}$	2^{31}
2^{32}	$x_n = x_{n-1} + x_{n-2} + C$	2^{58}
$2^{31} - 69$	$x_n = x_{n-3} - x_{n-1}$	2^{62}
$2^{32} - 18$	$x_n = x_{n-2} - x_{n-3} - C$	2^{95}

La constante C indica un 0 o un 1 según corresponda, para acarreos o préstamos.

Por último, señalamos que en la biblioteca Python se implementa la rutina **Mersenne-twister**, bastante más compleja que los generadores que hemos visto. Invitamos al lector a investigar sobre este generador en particular.

Capítulo 4

El Método de Monte Carlo

El **método de Monte Carlo** es un procedimiento general para seleccionar muestras aleatorias de una población utilizando números aleatorios.

La denominación Monte Carlo fue popularizado por los científicos Stanislaw Ulam, Enrico Fermi, John von Neumann, y Nicholas Metropolis, entre otros, quienes ya trabajaban sobre **muestreo estadístico**. Su nombre hace referencia al Casino de Montecarlo en Mónaco.

Este método se utiliza para calcular numéricamente expresiones matemáticamente complejas y difíciles de evaluar con exactitud, o que no pueden resolverse analíticamente. En nuestro caso analizaremos el cálculo o estimación de integrales definidas, y aproximaciones al valor de π .

El método de Monte Carlo se basa en dos resultados fundamentales de la Teoría de la Probabilidad:

1. La **Ley Fuerte de los Grandes Números**: Si $X_1, X_2, \dots, X_n, \dots$ son variables aleatorias independientes, idénticamente distribuidas, con media μ , entonces:

$$P\left(\lim_{n \rightarrow \infty} \frac{X_1 + X_2 + \dots + X_n}{n} = \mu\right) = 1, \quad (4.1)$$

o equivalentemente con probabilidad 1

$$\lim_{n \rightarrow \infty} \frac{X_1 + X_2 + \dots + X_n}{n} = \mu.$$

2. Si X es una variable aleatoria absolutamente continua, con función de densidad f , y $g : \mathbb{R} \mapsto \mathbb{R}$ es una función, entonces $g \circ X$ es una variable aleatoria y su valor esperado está dado por

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) dx.$$

Supongamos entonces que se quiere determinar un cierto número θ , desconocido, y que se sabe que este valor puede calcularse como

$$\theta = E[g(X)]$$

para cierta variable aleatoria X que posee una distribución \mathcal{F} . En algunos casos puede ocurrir que, por la naturaleza de la función g , o de la función de densidad de X , resulta muy difícil o imposible determinar $E[g(X)]$. El método de Monte Carlo propone encontrar un **valor estimado** de θ , donde *estimado* significa que hay una alta probabilidad de obtener un valor muy cercano a la verdadera solución. Para esto se considera una secuencia X_1, X_2, \dots de variables aleatorias, independientes, todas con la misma distribución de X , es decir, $X_i \sim \mathcal{F}$ para todo $i \geq 1$. Entonces $g(X_1), g(X_2), \dots, g(X_n), \dots$ son todas variables aleatorias independientes, con media igual $\theta = E[g(X)]$, y por la Ley Fuerte de los Grandes Números se tiene que:

$$\theta = \lim_{n \rightarrow \infty} \frac{g(X_1) + g(X_2) + \dots + g(X_n)}{n},$$

con probabilidad 1.

Notemos que el lado izquierdo de la igualdad es un número, mientras que lo que está a la derecha de la igualdad es el límite de una variable aleatoria. Por esto se indica que la probabilidad de que este límite sea igual a θ es 1.

También es importante aclarar que la Ley Fuerte de los Grandes Números no determina que existe una cota para el error

$$\left| \theta - \frac{g(x_1) + g(x_2) + \dots + g(x_n)}{n} \right|$$

para cualquier realización de las X_i :

$$X_1 = x_1, \quad X_2 = x_2, \quad X_3 = x_3, \quad \dots \quad X_n = x_n,$$

y un n suficientemente grande; sino que establece que para **casi** todas las realizaciones ocurre que

$$\lim_{n \rightarrow \infty} \frac{g(x_1) + g(x_2) + \dots + g(x_n)}{n} = \mu.$$

Dicho de una manera más coloquial, es prácticamente improbable que una realización de las X_i 's no cumpla que el límite de sus promedios

$$\frac{g(x_1) + g(x_2) + \dots + g(x_n)}{n}$$

se aproxime a θ a medida que n tiende a infinito.

4.1. Estimación de integrales definidas

4.1.1. Integración sobre $(0, 1)$

Una de las aplicaciones del método de Monte Carlo es facilitar el cálculo de integrales definidas. En realidad no se determina el valor exacto, sino que se utiliza para **estimar** el valor de la integral principalmente en casos que el cálculo analítico no es posible.

Veamos primero el caso en que se desee calcular el valor θ de una integral definida en el intervalo $[0, 1]$:

$$\theta = \int_0^1 g(x) dx. \quad (4.2)$$

Si ahora consideramos una variable aleatoria uniforme U , $U \sim \mathcal{U}(0, 1)$, entonces la función de densidad de U es

$$f(x) = I_{(0,1)}(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{c.c} \end{cases},$$

y por lo tanto (4.2) se puede escribir como un valor esperado:

$$\theta = \int_{-\infty}^{\infty} g(x) f(x) dx = \int_0^1 g(x) f(x) dx = E[g(U)].$$

Ahora, por la Ley Fuerte de los grandes números podemos considerar una sucesión de N variables aleatorias uniformes $U_i \sim \mathcal{U}(0, 1)$, independientes, y aproximar el valor θ con el límite de promedios:

$$\lim_{N \rightarrow \infty} \frac{g(U_1) + g(U_2) + \cdots + g(u_N)}{N} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N g(U_i) \simeq \theta.$$

En la práctica, esta integral puede aproximarse con una muestra de tamaño N suficientemente grande: $U_1 = u_1, U_2 = u_2, \dots, U_N = u_N$ y estimar θ con:

$$\theta = \frac{1}{N} \sum_{i=1}^N g(u_i).$$

Ejemplo 4.1. Para estimar el valor de la integral

$$\int_0^1 (1 - x^2)^{3/2} dx,$$

consideramos una realización de N variables aleatorias uniformes independientes:

$$U_1 = u_1, \quad U_2 = u_2, \quad \dots \quad U_N = u_N$$

y aproximamos el valor de la integral por

$$\int_0^1 (1 - x^2)^{3/2} dx \sim \frac{1}{N} \sum_{i=1}^N (1 - u_i^2)^{3/2}.$$

```

from random import random

def g(u): ##función a integrar en (0,1)
    return (1 - u ** 2) ** (1.5)
##estimación de la integral de g con Nsim simulaciones
def MonteCarlo(g, Nsim):
    Integral = 0
    for _ in range(Nsim):
        Integral += g(random())
    return Integral/Nsim

```

El valor exacto de esta integral es $\frac{3\pi}{16}$, aproximadamente 0.5890486226. (Ver Figura 4.1 y Cuadro 4.1).

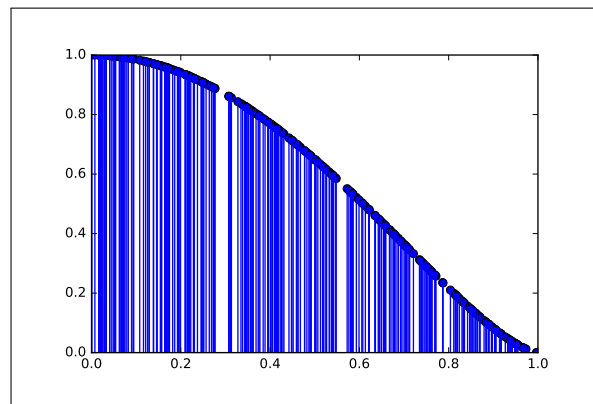


Figura 4.1: $N = 300$, Valor estimado=0.6067846103

# Simulaciones	Valor estimado
1000	0.5788611891947463
10000	0.5863982284155244
100000	0.5906798799945882
1000000	0.5882139637408895

Tabla 4.1: Estimaciones de $\int_0^1 (1 - x^2)^{3/2} dx$

4.1.2. Integración sobre un intervalo (a, b)

Para estimar el valor de una integral definida, sobre un intervalo (a, b) , con a y b reales, se aplica un cambio de variables para transformarla en una integral entre 0 y 1. Esto es, si

$$\theta = \int_a^b g(x) dx,$$

con $a < b$, entonces definimos la variable y :

$$y = \frac{x - a}{b - a}, \quad dy = \frac{1}{b - a} dx$$

y así el valor de θ puede calcularse como:

$$\int_a^b g(x) dx = \int_0^1 g(a + (b - a)y)(b - a) dy = \int_0^1 h(y) dy.$$

donde

$$h(y) = g(a + (b - a)y)(b - a), \quad y \in (0, 1).$$

Ejemplo 4.2. Para estimar el valor de la integral

$$\int_{-1}^1 e^{x+x^2} dx,$$

realizamos un cambio de variable:

$$y = \frac{x + 1}{2}, \quad dy = \frac{1}{2} dx.$$

Luego

$$\int_{-1}^1 e^{x+x^2} dx = \int_0^1 e^{(2y-1)+(2y-1)^2} 2 dy$$

Una estimación con N simulaciones estará dada por el valor de una expresión:

$$\frac{2}{N} \sum_{i=1}^N e^{(2u_i-1)+(2u_i-1)^2}$$

donde u_1, u_2, \dots, u_n es una realización de las variables aleatorias U_1, U_2, \dots, U_N , todas uniformes en $(0, 1)$ e independientes entre sí.

Notar que si $U \sim U(0, 1)$, entonces $2U - 1 \sim U(-1, 1)$.

Función a integrar

```
def funciong(x):
    return exp(x ** 2 + x)
```

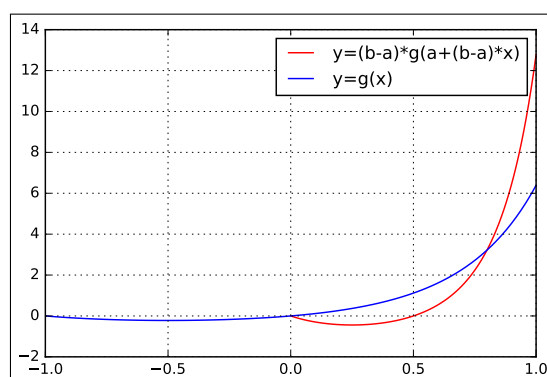
```

##Estima la integral de funciong entre a y b con Nsim simulaciones
def IntegralMonteCarlo(funciong, a, b, Nsim):
    Integral = 0
    for _ in range(Nsim):
        Integral += g(a + (b-a) * random())
    return Integral * (b-a)/Nsim

```

# simulaciones	valor estimado
1000	3.60200514128
10000	3.59901204683
100000	3.56130047278
1000000	3.59370623674
10000000	3.58717156846

Tabla 4.2: Estimaciones de la integral

Figura 4.2: Gráfico de $g(x) = e^{x+x^2}$ con y sin cambio de variables

4.1.3. Integración sobre $(0, \infty)$

En el caso de la estimación de una integral en el intervalo $(0, \infty)$:

$$\theta = \int_0^{\infty} g(x) dx,$$

también se aplica un cambio de variables, transformando biyectivamente el intervalo $(0, \infty)$ en $(0, 1)$. Un cambio de variables posible es el siguiente:

$$y = \frac{1}{x+1}, \quad dy = -\frac{1}{(x+1)^2} dx = -y^2 dx.$$

Luego se tiene que:

$$\int_0^\infty g(x) dx = - \int_1^0 \frac{g(\frac{1}{y} - 1)}{y^2} dy = \int_0^1 \frac{g(\frac{1}{y} - 1)}{y^2} dy = \int_0^1 h(y) dy,$$

con

$$h(y) = \frac{1}{y^2} g\left(\frac{1}{y} - 1\right).$$

Ejemplo 4.3. Para estimar la siguiente integral:

$$\int_0^\infty \cos(x) e^{-x} dx$$

se aplica el método de Monte Carlo para la estimación de la integral con el cambio de variables propuesto:

$$\int_0^1 \frac{\cos(\frac{1}{x} - 1) e^{-(\frac{1}{x} - 1)}}{x^2} dx \sim \theta \sim \frac{1}{N} \sum_{i=1}^N \frac{\cos(\frac{1}{u_i} - 1) e^{-(\frac{1}{u_i} - 1)}}{u_i^2}.$$

# simulaciones	valor estimado
1000	0.540652067791
10000	0.503039650709
100000	0.4991965288
1000000	0.499299312179

Tabla 4.3: Estimaciones de la integral

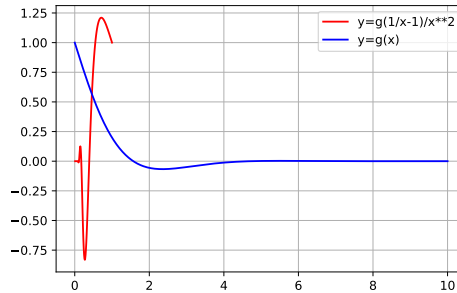


Figura 4.3: Gráfico de $g(x) = \cos(x) e^{-x}$, con y sin cambio de variables

4.2. Estimación de integrales múltiples

El método de Monte Carlo para el cálculo de integrales en una variable no es muy eficiente comparado con otros métodos numéricos que convergen más rápidamente al valor de la integral.

Sin embargo, para la estimación de integrales múltiples este método cobra mayor importancia ya que computacionalmente es menos costoso.

Nuevamente, una integral múltiple de una función en varias variables definida en un hiper-cubo de lado 1 puede estimarse con el método de Monte Carlo.

Para calcular la cantidad

$$\theta = \int_0^1 \cdots \int_0^1 g(x_1, \dots, x_l) dx_1 \dots dx_l$$

utilizamos el hecho que

$$\theta = E[g(U_1, \dots, U_l)]$$

con U_1, \dots, U_l independientes y uniformes en $(0, 1)$. Esto es así porque su distribución conjunta está dada por:

$$f(x_1, x_2, \dots, x_l) = \mathbb{I}_{(0,1) \times (0,1) \times \dots \times (0,1)}(x_1, x_2, \dots, x_l),$$

y entonces

$$\theta = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(x_1, \dots, x_l) f(x_1, \dots, x_l) dx_1 \dots dx_l.$$

Si se tienen N muestras independientes de estas l variables,

$$(U_1^1, \dots, U_l^1), \quad (U_1^2, \dots, U_l^2), \quad \dots \quad (U_1^N, \dots, U_l^N)$$

podemos estimar el valor de θ como

$$\theta \sim \frac{1}{N} \sum_{i=1}^N g(U_1^i, \dots, U_l^i)$$

4.2.1. Estimación del valor de π

Una aplicación de Monte Carlo en su uso para la estimación de integrales múltiples, es el cálculo estimado del valor de π . Recordemos que el área de un círculo de radio r es $\pi \cdot r^2$. Si tomamos $r = 1$, entonces π está dado por el valor de una integral:

$$\pi = \int_{-1}^1 \int_{-1}^1 \mathbb{I}_{\{x^2+y^2 < 1\}}(x, y) dx dy.$$

Si X e Y son v.a. independientes, uniformes en $(-1, 1)$, ambas con densidad

$$f_X(x) = f_Y(x) = \frac{1}{2} \cdot \mathbb{I}_{(-1,1)}(x),$$

entonces su densidad conjunta es igual al producto de sus densidades:

$$f(x, y) = f_X(x) \cdot f_Y(y) = \frac{1}{4} \cdot \mathbb{I}_{(-1,1) \times (-1,1)}(x, y).$$

En particular, (X, Y) resulta un vector aleatorio con distribución uniforme en $(-1, 1) \times (-1, 1)$, y tenemos que

$$\pi = \int_{-1}^1 \int_{-1}^1 \mathbb{I}_{\{x^2+y^2 < 1\}}(x, y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} 4 \cdot \mathbb{I}_{\{x^2+y^2 < 1\}}(x, y) f(x, y) dx dy.$$

Entonces $\frac{\pi}{4} = E[g(X, Y)]$ donde $g(x, y) = \mathbb{I}_{\{x^2+y^2 < 1\}}(x, y)$. Así, para estimar π podemos generar secuencias de pares (X_i, Y_i) , $i \geq 1$, donde X_i e Y_i son variables aleatorias uniformes en $(-1, 1)$, y luego estimar el valor de π como:

$$4 \cdot \frac{1}{N} \sum_{j=1}^N \mathbb{I}_{x^2+y^2 \leq 1}(x_i, y_i).$$

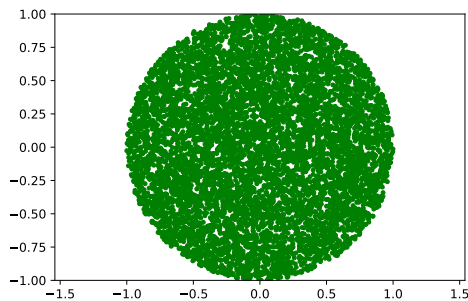
En otras palabras, π será estimado por la proporción de pares (X, Y) que caigan dentro del círculo de radio 1, multiplicado por 4.

Notemos que si $U_1, U_2 \sim U(0, 1)$, entonces

$$X = 2U_1 - 1 \quad Y = 2U_2 - 1$$

verifican $X, Y \sim U(-1, 1)$.

```
def valorPi(Nsim):
    enCirculo = 0.
    for _ in range(Nsim):
        u = 2 * random() - 1
        v = 2 * random() - 1
        if u ** 2 + v ** 2 <= 1:
            enCirculo += 1
    return 4 * enCirculo/Nsim
valorPi(Nsim)
```



# Simulaciones	Valor estimado
1000	3.16
10000	3.1216
100000	3.14292
1000000	3.141056
10000000	3.1420524

Capítulo 5

Generación de variables aleatorias discretas

Existen distintos métodos para generar variables aleatorias discretas a partir de un generador de números aleatorios. En particular veremos los siguientes:

1. Método de la **Transformada Inversa**.
2. Método de **aceptación y rechazo**, o simplemente **método de rechazo**.

Cada uno de estos métodos puede a su vez ser mejorado u optimizado según cuál es la variable aleatoria en particular que se desea simular. En todos los casos se asume la existencia de un buen generador de números aleatorios, es decir, de valores de una variable aleatoria uniforme en $(0, 1)$.

5.1. Método de la transformada inversa

Consideremos una variable aleatoria discreta X , con función de probabilidad de masa dada por:

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

donde los valores $\{x_n\}$ de la variable están ordenados en forma creciente. Esto es, si $i < j$ entonces $x_i < x_j$.

La función de distribución acumulada de X , F_X , satisface:

$$F_X(x) = \begin{cases} 0 & x < x_0 \\ p_0 & x_0 \leq x < x_1 \\ p_0 + p_1 & x_1 \leq x < x_2 \\ \vdots & \vdots \\ p_0 + p_1 + \cdots + p_{n-1} & x_{n-1} \leq x < x_n \\ \dots & \dots \end{cases}.$$

El método de la transformada inversa lleva este nombre porque se basa en la inversa de la función de distribución acumulada F_X . Es claro que si X es una variable aleatoria discreta, entonces F_X no es inyectiva y por lo tanto tampoco es inversible, pero sí establece una correspondencia biunívoca entre ciertos subintervalos de $[0, 1)$ y los valores de la variable aleatoria X a través de la función de distribución F_X . La correspondencia es la siguiente:

$$\begin{aligned} x_0 &\longrightarrow I_0 = [0, p_0) \\ x_1 &\longrightarrow I_1 = [p_0, p_0 + p_1) \\ x_2 &\longrightarrow I_2 = [p_0 + p_1, p_0 + p_1 + p_2) \\ &\dots \quad \dots \end{aligned}$$

Notemos que estos subintervalos del intervalo $[0, 1)$ son disjuntos, su unión es el intervalo $[0, 1)$ y la longitud de I_j es p_j . Por otro lado, si consideramos una variable aleatoria con distribución uniforme $U \sim U(0, 1)$, entonces la probabilidad que U tome valores en el intervalo I_j es p_j :

$$P(p_0 + \cdots + p_{j-1} \leq U < p_0 + p_1 + p_2 + \cdots + p_j) = p_j.$$

El método de la transformada inversa propone generar valores de X generando una variable aleatoria uniforme, y según a qué intervalo pertenece U es el valor de X que se genera. Es decir: si $U \in [0, p_0)$, se genera x_0 . Si $U \in [p_0, p_0 + p_1)$, se genera x_1 , y así siguiendo.

El algoritmo general para una variable aleatoria discreta que toma un número finito de valores es como el siguiente:

```
# x: vector de valores posibles de X
# p: vector de probabilidades
def discretaX(p, x):
    U = random()
    i, F = 0, p[0]
    while U >= F:
        i += 1; F += p[i]
    return x[i]
```

Ejemplo 5.1. Si consideramos la variable aleatoria X que toma valores en el conjunto $\{1, 2, 3, 4\}$, con las probabilidades:

$$p_1 = 0.20, \quad p_2 = 0.15, \quad p_3 = 0.25, \quad p_4 = 0.40,$$

el algoritmo de transformada inversa para generar valores de X es como sigue:

```
def generaX1():
    U = random()
    if U < 0.20: return 1
    elif U < 0.35: return 2
    elif U < 0.60: return 3
    else: return 4
```

Una primera mejora que puede hacerse a este algoritmo, y que en general vale para cualquier generador de variables aleatorias discretas, es ordenar las probabilidades de mayor a menor manteniendo la correspondencia entre estas probabilidades y los valores a generar. Esto reduce el número de comparaciones puesto que cada condicional ($U < F_X$) es más probable que sea aceptado cuanto mayor sea F_X . Así, en el Ejemplo 5.1, las probabilidades se ordenan de la siguiente forma:

$$p_4 = 0.40, \quad p_3 = 0.25, \quad p_1 = 0.20, \quad p_2 = 0.15,$$

y el algoritmo modificado es como sigue:

```
def generaX2():
    U = random()
    if U < 0.40: return 4
    elif U < 0.65: return 3
    elif U < 0.85: return 1
    else: return 2
```

Notemos que el número esperado de comparaciones en el primer caso es

$$1 \cdot 0.20 + 2 \cdot 0.15 + 3 \cdot 0.65 = 2.45$$

mientras que en el segundo es:

$$1 \cdot 0.40 + 2 \cdot 0.25 + 3 \cdot 0.35 = 1.95$$

Dado que $2.45/1.95 \simeq 1.26$ se tiene que el primer algoritmo (`generaX1`) tarda en promedio un 26 % más de tiempo que el segundo (`generaX2`) en lo que se refiere al tiempo en comparaciones.

5.1.1. Generación de una variable aleatoria uniforme discreta

Si X es una variable con distribución uniforme discreta en $\{1, \dots, n\}$ entonces $p_1 = p_2 = \dots = p_n = \frac{1}{n}$. La aplicación del método de la transformada inversa conduce al siguiente algoritmo:

```
def udiscreta(n):
    U = random()
    x = 1; F = 1/n
    while U >= F:
        F += 1 / n
        x += 1
    return x
```

Este algoritmo recorre los subintervalos de longitud $\frac{1}{n}$ hasta encontrar aquel donde se encuentra el valor de U . El algoritmo arrojará el valor $X = j$ si y sólo si U se encuentra en el j -ésimo subintervalo, es decir que se verifica:

$$\frac{j-1}{n} \leq U < \frac{j}{n}$$

Ahora bien, esto implica que nU pertenece al intervalo $[j-1, j)$, y por lo tanto la parte entera inferior de nU , $\lfloor nU \rfloor$, satisface

$$\lfloor nU \rfloor + 1 = j.$$

Así, el algoritmo puede reescribirse como:

```
def udiscreta(n):
    U = random()
    return int(n * U) + 1
```

Este algoritmo puede extenderse al caso de generación de una variable aleatoria discreta uniforme con valores enteros en el intervalo $[m, k]$, esto es, $U \sim U[m, k]$. Notemos que X toma $k - m + 1$ valores:

$$m, \quad m+1, \quad m+2, \quad \dots \quad k = m + (k - m).$$

Por lo tanto es suficiente generar una variable uniforme con valores en $[1, k - m + 1]$ y sumarle $m - 1$ a cada valor.

```
def udiscreta(m, k):
    U = random()
    return int(U * (k - m + 1)) + m
```

Generación de una permutación aleatoria de un conjunto de cardinal N

Una aplicación de la generación de variables aleatorias con distribución uniforme discreta es el de generar **permutaciones aleatorias** en un conjunto de cardinal N . El número de permutaciones de un conjunto de N elementos es $N!$, y el objetivo es poder generar permutaciones **equiprobables**, es decir, cada una con probabilidad $\frac{1}{N!}$ de ocurrencia.

Consideramos un ordenamiento de los elementos de un conjunto A , de cardinal N :

$$(a_0, a_1, \dots, a_{N-1})$$

Una primera idea, pero errónea, es intercambiar a_0 con cualquier elemento de A elegido con distribución uniforme, luego intercambiar a_1 con otro elegido uniformemente en A , y así siguiendo. Si bien se obtiene una permutación de A , ocurre que este procedimiento no genera todas las permutaciones con la misma probabilidad. Un algoritmo que ejecute estos pasos tiene N^{N-1} secuencias diferentes de ejecución, pero

$$\frac{N^{N-1}}{N!}$$

no es un número entero, salvo para $N = 2$. Por lo tanto, para $N > 2$ no es posible que todas las permutaciones se obtengan con la misma probabilidad.

En cambio, si se intercambia a_0 con algún elemento de $\{a_0, \dots, a_{N-1}\}$ elegido aleatoriamente, luego a_1 con alguno de $\{a_1, \dots, a_{N-1}\}$ y así siguiendo hasta intercambiar a_{N-2} con un elemento de $\{a_{N-2}, a_{N-1}\}$, entonces el método genera todas las permutaciones con igual probabilidad:

```
def permutacion(a): #a=[a[0], a[1], ..., a[N-1]]
    N = len(a)
    for j in range(N-1):
        indice = int((N-j) * random()) + j
        a[j], a[indice] = a[indice], a[j]
    return a
```

Notemos que en los sucesivos pasos se generan uniformes en los intervalos de enteros $[0, N-1]$, $[1, N-1]$, \dots , $[N-2, N-1]$. Si se recorre el arreglo en sentido inverso, comenzando por intercambiar a_{N-1} en lugar de a_0 , entonces el algoritmo requiere generar uniformes en $[0, N-1]$, $[0, N-2]$, \dots , $[0, 1]$, que se reducen a calcular una parte entera inferior:

```
def permutacion(a): #a=[a[0],a[1],...,a[N-1]]
    N = len(a)
    for j in range(N-1,0,-1):
        indice = int((j+1) * random())
        a[j], a[indice] = a[indice], a[j]
```

En ciertos casos de muestreo se requiere obtener un subconjunto aleatorio de cierto conjunto de individuos. Es decir, dado un conjunto de N elementos, obtener un subconjunto de r elementos, con $r < N$, pero elegidos aleatoriamente.

Si el ciclo del algoritmo anterior se ejecuta para $j = N - 1, N - 2, \dots, N - r$, los últimos r elementos del vector permutado forman un subconjunto aleatorio de cardinal r , y en consecuencia los restantes forman un subconjunto aleatorio de tamaño $N - r$. Así, para mayor eficiencia, si $r < N/2$ conviene ejecutar el algoritmo r veces y tomar los últimos r elementos, y de lo contrario conviene ejecutarlo $N - r$ veces, y tomar los r primeros.

```
## Devuelve un subconjunto aleatorio de A de r elementos
def subcAleatorio(r,A):
    N = len(A)
    for j in range(N-1, N-1-r, -1):
        indice = int((j+1) * random())
        A[j], A[indice] = A[indice], A[j]
    return A[N-r:]
```

Un **método alternativo** para obtener permutaciones aleatorias de un conjunto de cardinal N consiste en generar N números aleatorios: $u_0, u_1, u_2, \dots, u_{N-1}$, y luego ordenarlos, por ejemplo, de menor a mayor:

$$u_{i_0} < u_{i_1} < u_{i_2} < \dots < u_{i_{N-1}}.$$

Así, los índices de los números ya ordenados forman una permutación del conjunto $\{1, 2, \dots, N\}$:

$$(i_0, i_1, i_2, \dots, i_{N-1}).$$

Sin embargo este método tiene la desventaja de requerir $O(N \log(N))$ comparaciones.

5.1.2. Cálculo de promedios

Recordemos que el Método de Monte Carlo se basa en dos resultados teóricos fundamentales: la Ley Fuerte de los Grandes Números, y la posibilidad de calcular el valor esperado

$E[g(X)]$ a partir de la función de densidad o de probabilidad de masa, según corresponda, de la variable aleatoria X .

Supongamos que se quiere calcular un promedio de una gran cantidad de valores:

$$\bar{a} = \frac{1}{N} \sum_{i=1}^N a_i = \frac{a_1 + a_2 + \cdots + a_N}{N},$$

donde $N \gg 1$, y que por la gran cantidad de términos o por la complejidad de éstos resulta muy complicado calcular la suma. Notemos que si X es una variable aleatoria uniforme discreta en $[1, N]$, y g es una función tal que $g(i) = a_i$, entonces el valor que se desea calcular es justamente $E[g(X)]$:

$$\bar{a} = E[g(X)].$$

Luego, por la Ley Fuerte de los grandes números, se tiene que si $X_1, X_2, \dots, X_n, \dots$ son variables aleatorias independientes, uniformes en $[1, N]$, entonces

$$\frac{1}{n}(g(X_1) + g(X_2) + \cdots + g(X_n)) \simeq \frac{1}{n}(a_{X_1} + a_{X_2} + \cdots + a_{X_n}) = \bar{a}.$$

Así el valor de \bar{a} se puede estimar como un promedio de N términos de la sumatoria elegidos al azar:

$$\bar{a} \sim \frac{1}{M} \sum_{j=1}^M a_{i_j}.$$

Ejemplo 5.2. Se requiere calcular el promedio

$$S = \sum_{i=1}^{10000} e^{\frac{1}{i}}.$$

Luego tomamos $g(i) = \exp(1/i)$, y estimamos $E[g(X)]$ para $X \sim U[1, 10000]$. Notemos que

$$S = 10000 \cdot \frac{1}{10000} \sum_{i=1}^{10000} e^{\frac{1}{i}} = 10000 \cdot E[g(X)].$$

Usando Monte Carlo, se puede estimar el valor de $\frac{S}{10000}$ con 100 simulaciones de la siguiente forma:

- Generar 100 valores de U , $U = u_i$, $1 \leq i \leq 100$, con distribución uniforme en $[1, 10000]$.
- Calcular para cada uno $\exp(1/u_i)$.
- Sumarlos y dividir por 100.

Como estos pasos llevan a una aproximación de $S/10000$, la estimación de S se obtiene multiplicando por 10000.

$$S = \sum_{i=1}^{10000} e^{\frac{1}{i}} \sim 10000 \cdot \frac{1}{100} \sum_{i=1}^{100} e^{\frac{1}{u_i}}.$$

```
Suma=0
Nsim=100
for _ in range(Nsim):
    U = int(random() * 10000) + 1
    Suma += exp(1 / U)
Suma = Suma / Nsim * 10000
```

5.1.3. Generación de una variable aleatoria geométrica

Una variable aleatoria geométrica con probabilidad de éxito p tiene una probabilidad de masa dada por:

$$p_i = P(X = i) = pq^{i-1}, \quad i \geq 1, \quad q = (1 - p),$$

y la función de distribución acumulada cumple:

$$F_X(j - 1) = P(X \leq j - 1) = 1 - P(X > j - 1) = 1 - q^{j-1}.$$

El método de la transformada inversa asigna entonces el valor $X = j$ si la variable aleatoria uniforme U satisface $U \in [1 - q^{j-1}, 1 - q^j]$. En este caso ocurre que:

$$q^j < 1 - U \leq q^{j-1}.$$

Dado que $0 < q < 1$, las potencias de q son decrecientes. Por lo tanto la propiedad anterior equivale a encontrar el menor j tal que $q^j < 1 - U$. Luego:

$$X = \min\{j : q^j < 1 - U\}.$$

Para determinar j , aplicamos el logaritmo, y así:

$$X = j \quad \text{si y sólo si} \quad j = \min\{k \mid k \log(q) < \log(1 - U)\}.$$

o equivalentemente y usando que $\log(q) < 0$:

$$X = j \quad \text{si y sólo si} \quad j = \min\{k \mid k > \frac{\log(1 - U)}{\log(q)}\} = \left\lceil \frac{\log(1 - U)}{\log(q)} \right\rceil + 1$$

Así, el algoritmo para generar valores de una variable aleatoria con distribución geométrica $X \sim \text{Geom}(p)$ es el siguiente:

```
def geometrica(p):
    U = random()
    return int(log(1-U)/log(1-p))+1
```

5.1.4. Generación de variables Bernoulli

Un método sencillo de generar valores de una Bernoulli $B(p)$ es:

```
def Bernoulli(p):
    U=random()
    if U < p: return 1
    else:      return 0
```

Ahora bien, si se quiere obtener una secuencia de valores de N variables Bernoulli, independientes, puede optimizarse el algoritmo anterior utilizando la generación de una variable geométrica. Notemos que $X \sim \text{Geom}(p)$ es la variable que mide el número de ensayos independientes de una variable Bernoulli $B(p)$ hasta obtener un éxito. Entonces, si por ejemplo se genera el valor $X = 5$, esto es equivalente a generar 5 valores consecutivos de una variable $B(p)$:

0 0 0 0 1.

Así, si se quieren generar N valores de variables aleatorias $B(p)$, independientes, es suficiente generar valores de $X \sim \text{Geom}(p)$, x_1, x_2, \dots, x_k , hasta que $x_1 + x_2 + \dots + x_k \geq N$, y así se obtiene la secuencia de Bernoulli:

$\underbrace{0 \quad 0 \quad \dots \quad 1}_{(x_1-1) \text{ ceros y un } 1} \quad \underbrace{0 \quad 0 \quad \dots \quad 1}_{(x_2-1) \text{ ceros y un } 1} \dots \quad \underbrace{0 \quad 0 \quad \dots \quad 1}_{(x_k-1) \text{ ceros y un } 1}.$

```
##devuelve una lista de N Bernoullis B(p)
```

```
def NBernoullis(N,p):
    Bernoullis = [0] * N
    j = geometrica(p)-1
    while j < N:
        Bernoullis[j] = 1
    %      j+ = geometrica(p)
    return Bernoullis
```

5.1.5. Generación de una variable aleatoria Poisson

La función de probabilidad de masa de una variable aleatoria Poisson de razón λ está dada por:

$$p_i = P(X = i) = e^{-\lambda} \frac{\lambda^i}{i!}, \quad i = 0, 1, 2, \dots$$

Notemos que las probabilidades cumplen una relación de recurrencia:

$$p_0 = e^{-\lambda}, \quad p_{i+1} = e^{-\lambda} \frac{\lambda^{i+1}}{(i+1)!} = p_i \frac{\lambda}{i+1}.$$

Así, el algoritmo obtenido por el método de la transformada inversa se puede escribir:

```
def Poisson(lamda):
    U = random()
    i = 0; p = exp(-lamda)
    F = p
    while U >= F:
        i += 1
        p *= lamda / i
        F = F + p
    return i
```

Una observación con respecto a este algoritmo, es que chequea desde el valor 0 en adelante, pero p_0 no es la probabilidad mayor. Se podría optimizar si el algoritmo recorre primero las probabilidades más grandes. Además, el algoritmo realiza $n + 1$ comparaciones para generar el valor n . Dado que el valor esperado de la variable es λ , entonces el número de comparaciones es, en promedio, $\lambda + 1$. Así, a mayor valor de λ mayor será el número de comparaciones que realice el algoritmo.

Una forma de mejorar este algoritmo es comenzar por analizar el valor más probable. El valor máximo de las probabilidades es $p_{\lfloor \lambda \rfloor}$, es decir, que los valores cercanos a λ son los más probables. Así, se puede optimizar el número de comparaciones buscando a partir del valor

$$I = \lfloor \lambda \rfloor,$$

y luego de manera ascendente o descendente según el valor de la variable uniforme U generada:

```
def Poisson(lamda):
    p = exp(-lamda); F = p
    for j in range(1, int(lamda) + 1):
        p *= lamda / j
        F += p
    U = random()
    if U >= F:
        j = int(lamda) + 1
        while U >= F:
            p *= lamda / j; F += p
            j += 1
        return j - 1
    else:
```

```

j = int(lamda)
while U < F:
    F -= p;    p *= j/lamda
    j -= 1
return j+1

```

De esta manera, el promedio de búsquedas se reduce a

$$1 + E[|X - \lambda|].$$

En particular, si $\lambda \gg 1$, la distribución se aproxima a una normal de media y varianza λ , $N(\lambda, \sqrt{\lambda})$, por lo cual $\frac{X-\lambda}{\sqrt{\lambda}}$ se aproxima a una normal estándar $Z \sim N(0, 1)$.

Por lo tanto, el promedio de búsquedas, $1 + E[|X - \lambda|]$, puede estimarse como:

$$\begin{aligned} 1 + \sqrt{\lambda} E \left[\frac{|X - \lambda|}{\sqrt{\lambda}} \right] &= 1 + \sqrt{\lambda} E[|Z|] \\ &= 1 + 0.798\sqrt{\lambda}, \end{aligned}$$

que mejora el promedio en la versión anterior: $1 + \lambda$.

5.1.6. Generación de una variable aleatoria binomial

La generación de una variable aleatoria binomial responde a un caso similar al de una variable Poisson. Si $X \sim B(n, p)$, entonces la función de masa de probabilidad es:

$$p_i = P(X = i) = \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}, \quad i = 0, 1, \dots, n.$$

En este caso, la fórmula recursiva para las probabilidades está dada por:

$$p_0 = (1-p)^n, \quad p_{i+1} = \frac{n-i}{i+1} \frac{p}{1-p} p_i, \quad 0 \leq i < n.$$

Recordamos además que el valor esperado y la varianza están dados por

$$E[X] = np \quad \text{Var}[X] = np(1-p).$$

Si se aplica directamente el método de transformada inversa, el algoritmo resulta:

```

def Binomial(n, p):
    c = p / (1 - p)
    prob = (1 - p) ** n

```

```

F = prob;      i=0
U = random()
while U >= F:
    prob *= c * (n-i) / (i+1)
    F += prob
    i += 1
return i

```

Al igual que con la generación de variables aleatorias Poisson, el inconveniente de este algoritmo es que al chequear desde 0 hasta generar el valor correspondiente, el número de comparaciones es 1 más que el valor generado. Por lo tanto, el promedio de comparaciones es $E[X] + 1 = np + 1$.

Además de la optimización análoga al método de Poisson, una alternativa que podría mejorar el número de comparaciones es considerar el valor de p . Notemos que si $p > 1/2$ entonces $1 - p < \frac{1}{2}$, y por lo tanto es conveniente que el algoritmo genere $Y \sim B(n, 1 - p)$, que lleva menor número de comparaciones, y devolver el valor $X = n - Y$.

En cuanto a mejorar el algoritmo de manera análoga que Poisson, en este caso el valor más probable corresponde a

$$i = \lfloor np \rfloor,$$

y el promedio de comparaciones, para n grande resulta del orden de

$$1 + \sqrt{np(1-p)} \approx 0.798.$$

5.2. Método de aceptación y rechazo

El **método de aceptación y rechazo** o método de rechazo para generar una variable aleatoria X , supone el conocimiento de un método para la generación de otra variable aleatoria Y que cumpla con las siguientes propiedades:

- Si $P(X = x_j) > 0$, entonces $P(Y = x_j) > 0$, para todo x_j en el rango de X .
- Existe una constante positiva c tal que:

$$\frac{P(X = x_j)}{P(Y = y_j)} \leq c,$$

para todos los x_j tal que $P(X = x_j) > 0$.

Si denotamos $p_j = P(X = x_j)$ y $q_j = P(Y = x_j)$, entonces de la segunda propiedad vemos que:

$$\sum_{j \geq 1} p_j \leq c \cdot \sum_{j \geq 1} q_j \leq c$$

Como el miembro izquierdo de la desigualdad es 1, se tiene que la constante c es siempre mayor o igual a 1. Además la igualdad se daría sólo en el caso en que X e Y tienen la misma distribución de probabilidad, en cuyo caso ya se tendría un método para generar X . Asumimos entonces que $c > 1$, y por lo tanto $\frac{1}{c} < 1$.

```

1 Simular Y
2 U = random()
3 if U < p(Y) / (c * q(Y)) :
4     return Y    ## aceptación: X=Y
5 else:
6     volver a 1 ## rechazo

```

digamos que $Y = x_3$

$$\left(\frac{U < \frac{p(X=x_3)}{c \cdot p(Y=x_3)}}{c \cdot p(Y=x_3)} \right)$$

Como puede verse de la descripción del algoritmo, se trata de una iteración de pasos que se repite en caso de un rechazo y finaliza en una aceptación. Existe en esto una cierta analogía con una variable geométrica, en el sentido que se continúa el ciclo o ejecución de los condicionales (ensayos) hasta que se obtiene un éxito: aceptar el valor de Y .

En efecto, en cada paso, la probabilidad de generar algún valor de X es la probabilidad de aceptar el valor de Y en esa iteración, y esto está dado por:

$$\begin{aligned}
 P(\text{aceptar } Y) &= P\left(\{Y = y_1, U < \frac{p_1}{c q_1}\} \cup \{Y = y_2, U < \frac{p_2}{c q_2}\} \cup \dots\right) \\
 &= P\left(\bigcup_{j \geq 1} \{Y = y_j, U < \frac{p_j}{c q_j}\}\right).
 \end{aligned}$$

Notemos que esta unión es disjunta, y además U e Y son independientes. Por lo tanto resulta:

$$P(\text{aceptar } Y) = \sum_{j \geq 1} P(Y = y_j) \cdot P(U \leq \frac{p_j}{c q_j}) = \sum_{j \geq 1} q_j \cdot \frac{p_j}{c q_j} = \frac{1}{c}.$$

Ahora bien, un valor particular x_j de la variable X será generado sí y sólo sí es generado en alguna iteración:

$$\begin{aligned}
 P(\text{generar } x_j) &= \sum_{k \geq 1} P(\text{generar } x_j \text{ en la iteración } k) \\
 &= \sum_{k \geq 1} P(\text{rechazar } Y \text{ } (k-1) \text{ veces y aceptar } Y = x_j \text{ en la iteración } k) \\
 &= \sum_{k \geq 1} \left(1 - \frac{1}{c}\right)^{k-1} P(Y = x_j, U \leq \frac{p_j}{c q_j}) \\
 &= \sum_{k \geq 1} \left(1 - \frac{1}{c}\right)^{k-1} q_j \frac{p_j}{c q_j} \\
 &= p_j.
 \end{aligned}$$

Por lo tanto, el algoritmo simula una variable aleatoria con la distribución deseada. Además, el número de iteraciones del algoritmo hasta aceptar el valor de Y es una variable aleatoria

geométrica con probabilidad de éxito $\frac{1}{c}$ y de fracaso $1 - \frac{1}{c}$. Por lo tanto, cuanto menor sea c más eficiente será el algoritmo.

Ejemplo 5.3. Sea X una variable aleatoria con valores en $\{1, 2, \dots, 10\}$ y probabilidades 0.11, 0.12, 0.09, 0.08, 0.12, 0.10, 0.09, 0.09, 0.10, 0.10. Si se generan valores con el método de la transformada inversa, optimizando de modo que los primeros condicionales tengan mayor probabilidad de ser aceptados, el número esperado de iteraciones (comparaciones) será:

$$(1 + 2) \cdot 0.12 + 3 \cdot 0.11 + (4 + 5 + 6) \cdot 0.10 + (7 + 8) \cdot 0.09 + 9 \cdot 0.17 = 5.07.$$

Por otro lado, si se genera esta variable rechazando con una variable aleatoria $Y \sim U[1, 10]$, tenemos que $P(Y = j) = 0.1$, y por lo tanto

$$P(X = j) \leq 1.2 \cdot P(Y = j), \quad 1 \leq j \leq 10.$$

Luego el valor esperado del número de iteraciones hasta generar un valor de X es igual al valor esperado de una variable aleatoria geométrica con $p = \frac{1}{1.2}$, que es $c = 1.2$. Es decir, que requiere la generación de una variable uniforme y una comparación por cada ciclo.

5.3. Método de composición

El método de composición permite generar una variable aleatoria X con función de probabilidad de masa dada por:

$$P(X = j) = \alpha p_j + (1 - \alpha) q_j \quad j = 0, 1, \dots,$$

donde α es algún número entre 0 y 1.

Entonces, si se tienen métodos eficientes para generar valores de dos variables aleatorias X_1 y X_2 , con funciones de probabilidad de masa

$$P(X_1 = x_j) = p_j, \quad P(X_2 = x_j) = q_j, \quad j = 1, \dots,$$

entonces estos métodos pueden ser utilizados para generar valores de X :

$$X = \begin{cases} X_1 & \text{con probabilidad } \alpha \\ X_2 & \text{con probabilidad } 1 - \alpha \end{cases}$$

El algoritmo sería como el que sigue:

```
U = random()
if U < alfa:
    simular X1
```

```

    return X1
else:
    simular X2
    return X2

```

Notemos que:

$$\begin{aligned}
 P(\text{generar } x_j) &= P(U < \alpha, X_1 = x_j) + P(\alpha \leq U < 1, X_2 = x_j) \\
 &= P(U < \alpha) P(X_1 = x_j) + P(\alpha \leq U < 1) P(X_2 = x_j) \\
 &= \alpha p_j + (1 - \alpha) q_j.
 \end{aligned}$$

Así la probabilidad de generar $X = x_j$ es la probabilidad deseada.

Ejemplo 5.4. Si X es la variable aleatoria que toma los valores del 1 al 5 con probabilidad 0.05, y los valores del 6 al 10 con probabilidad 0.15, entonces X toma algún valor entre 1 y 5 con probabilidad 0.25, y un valor entre 6 y 10 con probabilidad 0.75. Por otro lado, los valores del 1 al 5 son equiprobables y los del 6 al 10 también. Luego se pueden elegir X_1 y X_2 uniformes discretas en $[1, 5]$ y $[6, 10]$ respectivamente, y generar X con el siguiente algoritmo:

```

U = random()
V = random()
if U < 0.75:
    return int(5 * V) + 6
else:
    return int(5 * V) + 1

```

También se pueden considerar las variables aleatorias $Y_1 \sim U[1, 10]$ e $Y_2 \sim U[6, 10]$, de modo que los valores de 1 a 5 sean generados con Y_1 y los restantes puedan ser generados con Y_1 o con Y_2 . Esto es:

$$P(Y_1 = j) = 0.1, \quad 1 \leq j \leq 10, \quad P(Y_2 = j) = 0.2, \quad 6 \leq j \leq 10.$$

Dado que X toma algún valor entre 1 y 5 con probabilidad 0.25, y estos son la mitad de los valores que toma Y_1 , se da una ponderación $\alpha = 0.5$ a Y_1 , y la misma ponderación a Y_2 . Así, en este caso el algoritmo es el siguiente:

```

U = random()
V = random()
If U < 0.5:
    return int(10 * V) + 1
else:
    return int(5 * V) + 6

```

En general, si se tienen n variables aleatorias con funciones de distribución acumulada F_1, F_2, \dots, F_n , y $\alpha_1, \alpha_2, \dots, \alpha_n$ son números positivos tales que

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 1,$$

entonces la función de distribución acumulada dada por

$$F(x) = \alpha_1 F_1(x) + \alpha_2 F_2(x) + \dots + \alpha_n F_n(x),$$

se llama una **mezcla** o **composición** de las funciones de distribución F_1, F_2, \dots, F_n .

Una variable aleatoria X con función de distribución F puede generarse a través de la simulación de una variable discreta I que toma valores en $\{1, 2, \dots, n\}$ con $P(I = j) = \alpha_j$, y una segunda simulación de una variable con distribución F_j , si $I = j$. Esta forma de generar X se llama **método de composición**.

5.4. Métodos alternativos

5.4.1. El método del alias

Una aplicación del método de composición es el llamado **método del alias**. Este método sirve para generar variables aleatorias que toman un número finito de valores, digamos, $1, 2, \dots, n$.

El método requiere construir en primer lugar los llamados **alias**, pero una vez completada esta etapa la simulación en sí es muy simple. La distribución de la variable X , F_X , se describe como una composición equiprobable de $n - 1$ variables aleatorias:

$$F_X(x) = \frac{1}{n-1} F_1(x) + \frac{1}{n-1} F_2(x) + \dots + \frac{1}{n-1} F_{n-1}(x),$$

y $F_j(x)$ es la distribución de una variable X_j que toma a lo sumo dos valores posibles.

Para definir esta composición o mezcla, se utiliza el siguiente resultado:

Proposición 5.1. Sea $P(X = j) = p_j$, $1 \leq j \leq n$ una función de probabilidad de masa, con $n > 1$. Entonces:

- a) Existe j , $1 \leq j \leq n$, tal que $p_j < \frac{1}{n-1}$.
- b) Dado j como en a), existe $i \neq j$ tal que $p_i + p_j \geq \frac{1}{n-1}$.

La demostración es sencilla. Para ver a), notemos que si no fuera cierto entonces la suma de las probabilidades cumpliría

$$\sum_{j=1}^n p_j \geq n \cdot \frac{1}{n-1} > 1,$$

lo cual es absurdo.

Por otro lado, dado j como en a), debe existir i que cumpla b). De lo contrario, se tendría que:

$$1 - p_j = \sum_{i \neq j} p_i < (n-1) \left(\frac{1}{n-1} - p_j \right) = 1 - (n-1)p_j.$$

Esto es, $p_j > (n-1)p_j$, lo cual sólo es válido si $n = 1$.

El procedimiento es como sigue. En primer lugar, notemos que si se multiplican todas las probabilidades por $n-1$, entonces su suma es igual a $n-1$:

$$(n-1)p_1, \quad (n-1)p_2, \quad \dots, \quad (n-1)p_n. \quad (5.1)$$

La idea es distribuir estos $n-1$ valores dados en (5.1) como probabilidades de masa de $n-1$ variables aleatorias, de modo que estas variables tomen a lo sumo dos valores distintos. Por ejemplo, si se tuviera una variable aleatoria X que toma valores en $\{1, 2, 3\}$ con probabilidades de masa:

$$p_1 = 0.3, \quad p_2 = 0.6, \quad p_3 = 0.1,$$

al mutiplicar por 2 tenemos los valores:

$$2p_1 = 0.6, \quad 2p_2 = 1.2, \quad 2p_3 = 0.2.$$

El objetivo es descomponer la suma de estos tres valores ($= 2$) como la suma de las probabilidades de dos variables aleatorias X_1 y X_2 . El valor más pequeño es 0.2 que corresponde al valor 3. Entonces consideramos una variable X_1 que tome el valor 3 con probabilidad 0.2. Como debe tomar dos valores, el segundo valor debe tener probabilidad 0.8.

Dado que $2p_2 = 1.2 = 0.8 + 0.4$, asignamos a X_1 el valor 2 con probabilidad 0.8. Así formamos una primer variable aleatoria X_1 que tome los valores 3 y 2 con probabilidades 0.2 y 0.8, respectivamente.

Ahora, como $1.2 = 0.8 + 0.4$, entonces la segunda variable a definir tomará los valores 2 y 1 con probabilidades 0.4 y 0.6:

$$X_1 = \begin{cases} 2 & p = 0.8 \\ 3 & q = 0.2 \end{cases} \quad X_2 = \begin{cases} 1 & p = 0.6 \\ 2 & q = 0.4 \end{cases}$$

y ahora, para cada $x = 1, 2, 3$ se cumple que:

$$P(X = x) = \frac{1}{2} P(X_1 = x) + \frac{1}{2} P(X_2 = x).$$

En un caso general, se eligen j e i como en la Proposición 5.1, y se define

$$X_1 = \begin{cases} j & \text{con probabilidad } (n-1)p_j \\ i & \text{con probabilidad } 1 - (n-1)p_j \end{cases}.$$

Notemos que X_1 toma todo el peso del valor j , y parte del peso del valor i . Digamos que de la lista de valores en (5.1) se han utilizado p_j y una "parte" de p_i , de modo que suman 1. Si se restan estos dos valores en (5.1), la suma total es igual a $n - 2$.

Es decir, si dividimos por $n - 2$ se tendría nuevamente una variable aleatoria \tilde{X} con valores en $\{1, 2, \dots, n\} - \{j\}$, que tiene probabilidades

$$P(\tilde{X} = k) = \frac{1}{n-2} (n-1) p_k, \quad k \neq j, i,$$

y $P(\tilde{X} = i)$ es tal que la suma de las probabilidades es 1.

Ahora se eligen nuevos índices j e i con las propiedades dadas en a) y b) de la Proposición 5.1, respectivamente, y se construye una variable X_2 , cambiando n por $n - 1$. El procedimiento sigue hasta que sólo quedan a lo sumo dos valores por generar.

Notemos que en cada paso, la variable X_k definida toma todo el peso de su correspondiente valor j , por lo cual este valor ya no será generado por las siguientes variables X_k . En consecuencia, el algoritmo de construcción del alias tiene $n - 1$ pasos.

Ilustramos estos pasos con un ejemplo:

Ejemplo 5.5. Por ejemplo, si X es la v. a. que toma valores en $\{1, 2, 3, 4\}$ con

$$p_1 = \frac{7}{16}, \quad p_2 = \frac{1}{4}, \quad p_3 = \frac{1}{8}, \quad p_4 = \frac{3}{16}.$$

Multiplicando estas probabilidades por $n - 1 = 3$, obtenemos los siguientes valores tabulados en la Tabla 5.1:

j	1	2	3	4	
$p_j \times (3)$	$\frac{21}{16}$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{9}{16}$	Suma=3

Tabla 5.1: Método del alias - Paso 1

La Proposición 5.1 nos dice que alguno de los valores de la tabla es menor que 1, y que para este valor hay otro cuya suma excede a 1. En general, se sugiere tomar el menor y mayor valor de la tabla respectivamente. En este caso: $j = 3$ e $i = 1$.

Consideramos entonces la variable aleatoria:

$$X_1 = \begin{cases} 3 & \text{con probabilidad } 3 \cdot p_3 = \frac{3}{8} \\ 1 & \text{con probabilidad } 1 - \frac{3}{8} = \frac{5}{8}. \end{cases}$$

Así, restando las probabilidades que toma X_1 para los valores 3 y 1, la tabla resulta como en la Tabla 5.2: Notemos que los valores de la segunda fila de la Table 5.2 suman 2, y al menos un

j	1	2	3	4	
$p_j \times 3$	$\frac{21}{16}$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{9}{16}$	Suma=2
	$\frac{11}{16}$	$\frac{3}{4}$		$\frac{9}{16}$	

Tabla 5.2: Método del alias - Paso 2

elemento es menor que 1 y la suma de este con algún otro elemento supera a 1. Podemos elegir en este caso $j = 4$ e $i = 2$, y definir la variable:

$$X_2 = \begin{cases} 4 & \text{con probabilidad } \frac{9}{16} \\ 2 & \text{con probabilidad } 1 - \frac{9}{16} = \frac{7}{16} \end{cases}.$$

Repitiendo el paso anterior, la tabla resulta como en el Cuadro 5.3:

j	1	2	3	4	
$p_j \times 3$	$\frac{21}{16}$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{9}{16}$	Suma=1
	$\frac{11}{16}$	$\frac{3}{4}$		$\frac{9}{16}$	
	$\frac{11}{16}$	$\frac{5}{16}$			

Tabla 5.3: Método del alias - Paso 3

y en este último paso directamente definimos

$$X_3 = \begin{cases} 1 & \text{con probabilidad } \frac{11}{16} \\ 2 & \text{con probabilidad } \frac{5}{16} \end{cases}.$$

De esta forma, la distribución F de la variable aleatoria original X se puede escribir como:

$$F_X(x) = \frac{1}{3} F_{X_1}(x) + \frac{1}{3} F_{X_2}(x) + \frac{1}{3} F_{X_3}(x),$$

y el algoritmo por el método del alias para simular valores de X será como el siguiente:

```
def aliasX():
    I=int (random() * 3)
    V = random()
    if I == 0:
        if V < 5/8:    return 1
        else:         return 3
    elif I == 1:
```

```

    if V < 9/16: return 4
    else:       return 2
else: ##I == 2
    if V < 11/16: return 1
    else:       return 2

```

Ejercicio 5.1. Comprobar que el algoritmo anterior genera los valores del 1 al 4 con las probabilidades deseadas.

El nombre de **alias** proviene del hecho que es posible definir las variables X_k de modo que $P(X_k = k)$ sea positiva, para $1 \leq k < n$. En ese caso, X_k toma el valor k y posiblemente otro valor que se denomina su "alias".

También puede optimizarse este algoritmo generando una única variable uniforme $U \sim U(0, 1)$, y tomar $I = \lfloor (n-1)U \rfloor$, y $V = (n-1)U - I$.

5.4.2. Método de la urna

Otro método simple, pero quizás menos eficiente en cuanto al uso de memoria, es el **método de la urna**. Dada una variable aleatoria X , que toma un número finito de valores, digamos $\{1, 2, \dots, n\}$, llamamos $p_j = P(X = j)$. El método consiste en considerar un valor $k \in \mathbb{N}$ tal que $k p_j$ sea entero, para todo j , $1 \leq j \leq n$.

Ahora se considera un arreglo A de k posiciones, y se almacena cada valor i en $k p_i$ posiciones del arreglo.

El algoritmo simplemente selecciona una posición al azar del arreglo y devuelve el valor en dicha posición.

```

def urnaX():
    I = int(random() * k)
    return A[I]

```

Ejemplo 5.6. Si X toma los valores $\{1, 2, 3\}$ con probabilidades

$$p_1 = 0.24, \quad p_2 = 0.46, \quad p_3 = 0.30,$$

se puede tomar $k = 100$. Así, se define un arreglo A con 100 posiciones, en las cuales se almacena el 1 en 24 posiciones, el 2 en 46 posiciones y el 3 en 30 posiciones.

Una mejora de este método, en cuanto al lugar de almacenamiento, es considerar las posiciones de los décimos, centésimos, milésimos, etc., por separado.

Así, en el Ejemplo 5.6, los décimos en las probabilidades están dados por 2, 4 y 3, y los centésimos están dados por 4, 6 y 0.

Luego se consideran dos arreglos:

1. A_1 : de $2 + 4 + 3 = 9$ posiciones, con 2 posiciones con el valor 1, 4 posiciones con valor 4 y 3 posiciones con el valor 3.
2. A_2 : de $4 + 6 + 0 = 10$ posiciones, con 4 posiciones con el valor 1, 6 con el valor 2 y ninguna con el valor 3.

El arreglo A_1 se le da peso $\frac{90}{100} = 0.9$, y al arreglo A_2 se le da un peso de $\frac{10}{100} = 0.10$, y el algoritmo es como sigue:

```
def urnaXmejorada():  
    U = random()  
    if U < 0.9:  
        I = int(random() * 9)  
        return A1[I]  
    else:  
        I=int(random() * 10)  
        return A2[I]
```

Capítulo 6

Generación de variables aleatorias continuas

6.1. Introducción

Se dice que una variable aleatoria X es continua, o más propiamente, absolutamente continua, si existe $f : \mathbb{R} \mapsto \mathbb{R}$ tal que

$$F(x) := P(X \leq x) = \int_{-\infty}^x f(t) dt.$$

Al igual que para las variables aleatorias discretas, analizaremos dos métodos de generación para variables continuas, con las respectivas mejoras en cada caso particular. Estos métodos también se denominan:

- Método de la transformada inversa.
- Método de aceptación y rechazo.

6.2. Método de la transformada inversa

La generación o simulación de valores de una variable aleatoria por el método de la transformada inversa requiere conocer la función de distribución acumulada de la variable X a simular. Si X es (absolutamente) continua y f es su función de densidad, la función de distribución acumulada:

$$F(x) = \int_{-\infty}^x f(t) dt,$$

cumple la propiedad de ser continua, no decreciente, y:

$$\lim_{x \rightarrow -\infty} F(x) = 0, \quad \lim_{x \rightarrow \infty} F(x) = 1.$$

En esta sección nos referiremos a variables aleatorias X cuya función de distribución sea monótona creciente sobre el conjunto $F^{-1}(0, 1) = \{x \mid 0 < F(x) < 1\}$. De esta manera, la función F tiene inversa sobre el intervalo $(0, 1)$. Para los casos en que no se cumple esta propiedad también es posible definir métodos de simulación de las variables, pero las demostraciones de su validez son un tanto más complejas.

Proposición 6.1. Sea $U \sim \mathcal{U}(0, 1)$ una variable aleatoria. Para cualquier función de distribución continua F , monótona creciente en $F^{-1}(0, 1)$, la variable aleatoria X definida por:

$$X = F^{-1}(U)$$

tiene distribución F .

Notemos que U toma valores en el intervalo $(0, 1)$, excluyendo los puntos 0 y 1. La propiedad de monotonía de F asegura que F tiene inversa en $(0, 1)$. Así, si $X = F^{-1}(U)$, su función de distribución está dada por:

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

Como F es creciente, entonces $a \leq b$ si y sólo si $F(a) \leq F(b)$. En particular, los conjuntos:

$$\{F^{-1}(U) \leq x\} \quad \text{y} \quad \{U \leq F(x)\}$$

son iguales. Así resulta:

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

Luego X tiene distribución F .

Esta proposición da un método en principio simple para simular una variable aleatoria continua con función de distribución acumulada F :

```
def Tinversa():
    U=random()
    return G(U) # G = F-1
```

Sin embargo, pueden existir algunas complicaciones. Por ejemplo, que la inversa de F involucre funciones computacionalmente costosas ($\sqrt[n]{f(x)}$, $\log(x)$, etc.), o que no pueda ser calculada explícitamente (por ejemplo la distribución de una variable normal, o de una gamma).

Aparecen entonces otras estrategias para generar estas variables, como por ejemplo expresando a F como la distribución del mínimo o del máximo de variables aleatorias independientes, o de la suma de variables independientes, o distribuciones condicionales u otros casos particulares.

Ejemplo 6.1. Escribir un método para generar el valor de una variable aleatoria X con función de densidad

$$f(x) = -\frac{x}{2} + 1, \quad 0 \leq x \leq 2.$$

$$F(x) = \begin{cases} 0 & x \leq 0 \\ -\frac{x^2}{4} + x & 0 < x < 2 \\ 1 & x \geq 2 \end{cases}$$

F resulta entonces monótona creciente en el intervalo $(0, 2)$.

Para simular X debemos determinar la inversa de F en $(0, 1)$. Es decir, para cada $u \in (0, 1)$, encontrar el valor x tal que

$$-\frac{x^2}{4} + x = u.$$

Esta ecuación tiene dos soluciones para x :

$$\begin{cases} x = 2 + 2\sqrt{1-u} \\ x = 2 - 2\sqrt{1-u} \end{cases} \quad \text{o}$$

pero sólo la segunda pertenece al intervalo $(0, 2)$. Luego el algoritmo de simulación de esta variable por el método de la transformada inversa es:

```
def inversaX():
    U = 1 - random()
    return 2 - 2 * sqrt(U)
```

También, utilizando que U y $1 - U$ están igualmente distribuidas, vale el algoritmo:

```
def inversaX():
    U = random()
    return 2 - 2 * sqrt(U)
```

Ejemplo 6.2. Escribir un método para simular una variable aleatoria Y con función de densidad

$$f(x) = \begin{cases} 0.25 & 0 \leq x \leq 2 \\ 0.5 & 2 < x < 3 \\ 0 & \text{en otro caso.} \end{cases}$$

Para esta variable aleatoria la función de distribución acumulada está dada por:

$$F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x}{4} & 0 \leq x \leq 2 \\ \frac{x-1}{2} & 2 < x < 3 \\ 1 & x \geq 3 \end{cases}$$

Notemos que Y toma valores en el intervalo $(0, 3)$, y que $F(2) = 0.5$. Luego el algoritmo de simulación de la variable es:

```
def inversaY():
    U = random()
    if U < 0.5:
        return 4*U
    else:
        return 2*U+1
```

Ejemplo 6.3. Consideremos X_1, X_2, \dots, X_n n variables aleatorias independientes, con funciones de distribución acumulada F_1, F_2, \dots, F_n , respectivamente:

$$F_i(a) = P(X_i \leq a), \quad i = 1, 2, \dots, n.$$

Llamamos X a la variable aleatoria que es el máximo de las X_i , $1 \leq i \leq n$:

$$X = \max\{X_1, X_2, \dots, X_n\}.$$

Entonces la función de distribución acumulada de X está dada por:

$$F_X(a) = F_1(a) \cdot F_2(a) \dots F_n(a).$$

En particular, si X_1, X_2, \dots, X_n son uniformes en $(0, 1)$, independientes entre sí, tenemos que $F_i(x) = x$, $0 \leq x \leq 1$ e $1 \leq i \leq n$. Luego el máximo de las X_i tiene una distribución dada por:

$$F_X(x) = x^n, \quad 0 \leq x \leq 1.$$

Así, si se desea simular una variable aleatoria con distribución $F_X(x) = x^n$, en lugar de tomar una raíz n -ésima también es posible generar n uniformes y tomar su máximo. ¿Es más eficiente?

```
def raizn(n):
    Maximo = 0.
    for _ in range(n):
        Maximo = max(Maximo, random())
    return Maximo
```

6.2.1. Simulación de una variable aleatoria exponencial

Si X es una variable aleatoria con distribución exponencial con parámetro $\lambda = 1$, $X \sim \mathcal{E}(1)$, entonces su función de distribución acumulada está dada por:

$$F(x) = \begin{cases} 1 - e^{-x} & x > 0 \\ 0 & x \leq 0. \end{cases}$$

Luego la inversa de F sobre $(0, 1)$ está dada por:

$$F^{-1}(u) = -\ln(1 - u), \quad u \in (0, 1).$$

Así, el algoritmo de simulación para $X \sim \mathcal{E}(1)$ es:

```
def exponencial():
    U = 1-random()
    return -log(1-U)
```

Notemos además que si X es exponencial con parámetro 1, esto es, $X \sim \mathcal{E}(1)$, entonces $Y = \frac{1}{\lambda}X$ es exponencial con media $\frac{1}{\lambda}$: $Y \sim \mathcal{E}(\lambda)$. Luego el método por transformada inversa para simular valores de $Y \sim \mathcal{E}(\lambda)$ es:

```
def exponencial(lamda):
    U = 1-random()
    return -log(U)/lamda
```

6.2.2. Simulación de una variable aleatoria Poisson $X \sim \mathcal{P}(\lambda)$

Sea $N(t)$, $t \geq 0$, un proceso de Poisson homogéneo con intensidad λ . Entonces se cumple que:

- Los tiempos de llegada entre eventos son variables aleatorias exponenciales de parámetro λ , es decir, media $\frac{1}{\lambda}$.
- El número de eventos en un intervalo de tiempo de longitud t es una variable aleatoria Poisson de media $\lambda \cdot t$.

En particular $N(1)$ es una variable aleatoria Poisson de media λ que indica el número de arribos hasta el tiempo $t = 1$, y los tiempos entre arribos en el intervalo $[0, 1]$ son exponenciales de parámetro λ .

Por lo tanto, si se simulan variables aleatorias exponenciales X_1, X_2, \dots , con $X_i \sim \mathcal{E}(\lambda)$ para $i \geq 1$, hasta que $X_1 + X_2 + \dots + X_n \leq 1$ y $X_1 + X_2 + \dots + X_n + X_{n+1} > 1$, entonces n representa el número de arribos hasta $t = 1$. Esto es:

$$N(1) = \max\{n \mid X_1 + X_2 + \dots + X_n \leq 1\}$$

Si se simula cada exponencial X_i con $-\frac{1}{\lambda} \ln(1 - U_i)$, con $U_i \sim \mathcal{U}(0, 1)$, tenemos que

$$\begin{aligned} N(1) &= \max\{n \mid X_1 + X_2 + \dots + X_n \leq 1\} \\ &= \max\{n \mid -\frac{1}{\lambda} (\ln(1 - U_1) + \ln(1 - U_2) + \dots + \ln(1 - U_n)) \leq 1\} \\ &= \max\{n \mid -\frac{1}{\lambda} (\ln((1 - U_1) \cdot (1 - U_2) \cdots (1 - U_n))) \leq 1\} \\ &= \max\{n \mid \ln((1 - U_1) \cdot (1 - U_2) \cdots (1 - U_n)) \geq -\lambda\} \\ &= \max\{n \mid (1 - U_1) \cdot (1 - U_2) \cdots (1 - U_n) \geq e^{-\lambda}\} \end{aligned}$$

Luego:

$$N(1) = \min\{n \mid (1 - U_1) \cdot (1 - U_2) \cdots (1 - U_n) < e^{-\lambda}\} - 1$$

```
def Poisson_con_exp(lamda):
    X = 0
    Producto = 1- random()
    cota = exp(-lamda)
    while Producto >= cota:
        Producto *= 1 - random()
        X += 1
    return X
```

6.2.3. Simulación de una variable con distribución $\text{Gamma}(n, \lambda^{-1})$

Hemos visto que la suma de n variables aleatorias exponenciales, independientes, con parámetro λ , es una variable aleatoria con distribución $\text{Gamma}(n, \lambda^{-1})$. Esta propiedad nos permite dar un algoritmo de simulación de una variable Gamma a través de la generación de n exponenciales independientes con el mismo parámetro. Notemos que si U_1, U_2, \dots, U_n son valores de una variable uniforme $U \sim \mathcal{U}(0, 1)$, entonces:

$$\begin{aligned} X &= -\frac{1}{\lambda} \ln(1 - U_1) - \frac{1}{\lambda} \ln(1 - U_2) \cdots - \frac{1}{\lambda} \ln(1 - U_n) \\ &= -\frac{1}{\lambda} \ln((1 - U_1) \cdot (1 - U_2) \cdots (1 - U_n)) \end{aligned}$$

De esta forma, un algoritmo para la simulación de una variable $\text{Gamma}(n, \lambda^{-1})$ es como sigue:

```
def Gamma(n, lamda):
    'Simula una gamma con parámetros n y 1/lamda'
    U = 1
    for _ in range(n):
        U *= 1-random()
    return -log(U) / lamda
```

Notemos que este método requiere la simulación de n variables uniformes y calcula un único logaritmo, mientras que para generar n exponenciales es necesario calcular n logaritmos. Veamos cómo puede utilizarse la generación de una $Gamma(n, \lambda^{-1})$ para generar n exponenciales independientes de parámetro λ y restringiendo el número de logaritmos a calcular.

Teorema 6.1. Si $X, Y \sim \mathcal{E}(\lambda)$, independientes, entonces

$$f_{X|X+Y}(x | t) = \frac{1}{t} \mathbb{I}_{(0,t)}(x),$$

es decir, X condicional a $X + Y = t$ es uniforme en $(0, t)$.

Demostración. Notemos que la función de densidad conjunta de X y $X + Y$ cumple:

$$\begin{aligned} f_{X|X+Y}(x | t) &= \frac{f_{X,X+Y}(x, t)}{f_{X+Y}(t)} = \frac{f_{X+Y|X}(t, x) f_X(x)}{f_{X+Y}(t)} = \frac{f_Y(t - x) f_X(x)}{f_{X+Y}(t)} \\ &= \frac{e^{-\lambda(t-x)} e^{-\lambda x}}{e^{-\lambda t}} \frac{1}{t} \cdot \mathbb{I}_{[0,t]}(x) = \frac{1}{t} \cdot \mathbb{I}_{[0,t]}(x). \end{aligned}$$

La demostración de este teorema y del caso general para n exponenciales está disponible en el material complementario. \square

Luego, para generar X, Y exponenciales independientes, de parámetro λ podemos aplicar el siguiente algoritmo:

```
def DosExp(lamda):
    V1, V2 = 1-random(), 1-random()
    t = -log(V1 * V2) / lamda
    U = random()
    X = t * U
    Y = t - X
    return X, Y
```

Este algoritmo requiere el cálculo de un único logaritmo y la generación de tres uniformes.

En el caso general, para simular n exponenciales a partir de una $X \sim \text{Gamma}(n, \lambda^{-1})$ se requiere calcular un único logaritmo y $n - 1$ uniformes adicionales: $V_1, V_2, \dots, V_{n-1}, V_j \sim U(0, 1)$. En primer lugar se ordenan los valores de las V_i :

$$V_{i_1} < V_{i_2} < \dots < V_{i_{n-1}}.$$

Si $X = t$, entonces el intervalo $(0, t)$ se divide en n subintervalos consecutivos, no superpuestos, de longitud:

$$tV_{i_1}, \quad t(V_{i_2} - V_{i_1}), \quad t(V_{i_{n-1}} - V_{i_{n-2}}), \quad t(1 - V_{i_{n-1}}).$$

Las longitudes de cada uno de estos intervalos son variables aleatorias con distribución exponencial $\mathcal{E}(\lambda)$ e independientes entre sí. El algoritmo resulta entonces:

```
def Nexponenciales(n, lamda):
    t = 1
    for _ in range(n): t *= random()
    t = -log(t)/lamda
    unif = random.uniform(0, 1, n-1)
    unif.sort()
    exponenciales = [unif[0]*t]
    for i in range(n-2):
        exponenciales.append((unif[i+1]-unif[i])*t)
    exponenciales.append((1-unif[n-2])*t)
    return exponenciales
```

6.3. El método de aceptación y rechazo

Supongamos que se quiere generar una variable aleatoria X con función de densidad f :

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t) dt,$$

y que se tiene un método para generar otra variable Y , con densidad g , tal que

$$\frac{f(y)}{g(y)} \leq c, \quad \text{para todo } y \in \mathbb{R} \text{ tal que } f(y) \neq 0.$$

El **método de rechazo** para generar X a partir de Y tiene el siguiente algoritmo:

```
def Aceptacion_Rechazo_X():
    while 1:
```

```

Simular Y
U = random()
if U < f(Y) / (c * g(Y)) :
    return Y

```

De manera análoga al caso del método para simular variables aleatorias discretas, se puede probar que:

- a) La variable aleatoria generada por el algoritmo `Aceptacion_Rechazo_X()` tiene densidad f .
- b) El número de iteraciones del algoritmo es una variable aleatoria geométrica con media c .

Entonces, para cada variable X que se desee generar rechazando contra una variable Y , es necesario determinar una cota c para el cociente $\frac{f(x)}{g(x)}$, válida para todo $x \in \mathbb{R}$. Para determinar este valor es útil recordar algunos resultados del Análisis Matemático.

En primer lugar, considerar la función

$$h(x) = \frac{f(x)}{g(x)}, \quad x \text{ tal que } f(x) \neq 0,$$

y para esta función determinar:

- a) Puntos críticos: Esto es, $f'(x) = 0$ o $f'(x)$ no existe.
- b) Analizar cuáles de estos puntos corresponden a máximos locales.
- c) Evaluar h en los extremos de su dominio, si es acotado, o los límites correspondientes.
- d) Elegir una cota a partir de los valores obtenidos en (b) y (c).

Ejemplo 6.4. Se quiere generar una variable aleatoria X con función de densidad:

$$f(x) = 20x(1-x)^3, \quad 0 < x < 1.$$

En particular, X tiene distribución $Beta(2, 4)$:

$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \mathbb{I}_{(0,1)}(x)$	Variable $Beta(\alpha, \beta)$
--	--------------------------------

Dado que $f(x) \neq 0$ en el intervalo $(0, 1)$, se podría utilizar el método de rechazo generando una variable $Y \sim \mathcal{U}(0, 1)$. Tenemos entonces que

$$f_X(x) = 20x(1-x)^3, \quad f_Y(x) = 1, \quad x \in (0, 1),$$

y $h(x) = \frac{f_X(x)}{f_Y(x)} = f_X(x)$.

Los pasos para analizar una cota de h son los siguientes:

$$\begin{aligned}h(x) &= 20x(1-x)^3, & 0 < x < 1 \\h'(x) &= 20(1-x)^2 \cdot (1-4x)\end{aligned}$$

- Puntos críticos en $(0, 1)$: $x = 1/4$.
- $x = 1/4$ es el único punto crítico, luego es un máximo de h .
- $h(1/4) = f(1/4) = 135/64$ es el valor máximo de h

$$c = \frac{135}{64} = 2.109375.$$

Además,

$$\frac{f(x)}{c \cdot g(x)} = 20 \cdot \frac{64}{135} x(1-x)^3 \sim 9.481481 \cdot x(1-x)^3.$$

Luego, el algoritmo para generar X rechazando con una distribución uniforme es:

```
def Beta_2_4():
    while 1:
        Y = random()
        U = random()
        if U < 9.481481 * Y * (1-Y) ** 3:
            return Y
```

Este método requiere un número promedio de ciclos del orden de $c = \frac{135}{64} \approx 2.11$.

Ejemplo 6.5. Analicemos cómo simular una variable aleatoria X , con densidad $\text{Gamma}(\frac{3}{2}, 1)$:

$$f(x) = \begin{cases} Kx^{1/2}e^{-x} & x > 0 \\ 0 & \text{c.c.} \end{cases}, \quad K = \frac{1}{\Gamma(\frac{3}{2})} = \frac{2}{\sqrt{\pi}}.$$

En general, si X es una Gamma con función de densidad dada por

$$g(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta x} x^{\alpha-1}, \quad x > 0,$$

entonces su valor esperado es

$$E[X] = \frac{\alpha}{\beta}.$$

En este caso, $\alpha = \frac{3}{2}$ y $\beta = 1$, por lo cual $E[X] = \frac{3}{2}$. Dado que una variable exponencial tiene el mismo rango que X , una posibilidad entonces es utilizar el método de rechazo con una variable exponencial $Y \sim \mathcal{E}(\lambda)$ que tenga igual media que X , es decir, $Y \sim \mathcal{E}(\frac{2}{3})$.

El método de rechazo implica calcular la constante c tal que $\frac{f_X(x)}{f_Y(x)} \leq c$, si $x > 0$. Esto es, debemos hallar una cota superior para la función:

$$h(x) = \frac{Kx^{1/2}e^{-x}}{\frac{2}{3}e^{-\frac{2}{3}x}} = K\frac{3}{2}x^{1/2}e^{-\frac{1}{3}x}, \quad x > 0.$$

Tenemos que $h'(x) = 0$ si y sólo si

$$\left(\frac{1}{2} - \frac{1}{3}x\right)x^{-\frac{1}{2}}e^{-\frac{1}{3}x} = 0$$

Dado que $\lim_{x \rightarrow \infty} h(x) = 0$ y $\lim_{x \rightarrow 0^+} h(x) = 0$, el valor máximo estará dado en algún punto crítico. En este caso, el valor de x tal que

$$\frac{1}{2} - \frac{1}{3}x = 0, \quad x = \frac{3}{2}.$$

Entonces el valor máximo alcanzado por la función h es:

$$h(3/2) = 3 \left(\frac{3}{2\pi e} \right)^{1/2}.$$

Este valor puede ser utilizado como la cota c del algoritmo. Luego, como Ahora, como

$$\frac{f(x)}{cg(x)} = \sqrt{\frac{2e}{3}} x^{1/2} e^{-x/3} \sim 1.34717$$

Así el algoritmo resulta:

```
def Gamma () :
    while 1:
        U = 1-random()
        Y = - log(U) * 1.5
        V = random()
        if V < 1.3471 * Y ** 0.5 * exp(-Y / 3) :
            return Y
```

Podríamos preguntarnos en el Ejemplo 6.5 si es razonable rechazar con una exponencial de igual media que la Gamma a generar. Para ello, deberíamos encontrar una cota de:

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

y determinar el valor de λ para la cual la cota es mínima. Notemos que:

$$\frac{f(x)}{\lambda e^{-\lambda x}} = \frac{Kx^{1/2}e^{-(1-\lambda)x}}{\lambda},$$

y esta función es no acotada si $\lambda \geq 1$. Luego corresponde analizar los casos en que $0 < \lambda < 1$.

Analizando puntos críticos, podemos ver que el punto de máximo está en:

$$x = \frac{1}{2(1-\lambda)}, \quad 0 < \lambda < 1.$$

y el valor máximo es igual a

$$c_\lambda = \frac{K}{\lambda} (2(1-\lambda))^{-1/2} e^{-1/2}.$$

El valor máximo c_λ será mínimo si $\lambda(1-\lambda)^{1/2}$ es máximo, y esto ocurre si $\lambda = \frac{2}{3}$. Por lo tanto $\lambda = \frac{2}{3}$ minimiza el valor de la cota c .

6.4. Simulación de variables aleatorias normales

Las variables aleatorias normales tienen un uso extensivo, por lo cual es importante obtener un buen método para simularlas. En primer lugar, recordemos que si $Z \sim N(0, 1)$, entonces

$$X = \sigma \cdot Z + \mu \sim N(\mu, \sigma).$$

Por lo tanto es suficiente tener un buen método para generar variables normales estándar.

El **método de la transformada inversa** no es una opción, puesto que la función de distribución acumulada:

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt$$

no tiene una expresión cerrada y por lo tanto no se conoce una forma explícita de su inversa.

6.4.1. Por composición usando $|Z|$

Una observación es que la función de densidad de Z es par, por lo cual puede intentarse un método para generar $|Z|$ y usar luego un método de composición. Más precisamente, consideramos $U \sim \mathcal{U}(0, 1)$ y X la variable aleatoria dada por

$$X = \begin{cases} |Z| & \text{si } U < 0.5 \\ -|Z| & \text{si } U \geq 0.5, \end{cases}.$$

Entonces se cumple que:

$$\begin{aligned} P(X \leq a) &= P(U < 0.5, |Z| \leq a) + P(U > 0.5, -|Z| \leq a) \\ &= \frac{1}{2} P(|Z| \leq a) + \frac{1}{2} P(-|Z| \leq a). \end{aligned}$$

Con un análisis de casos $a \geq 0$ y $a < 0$ podemos ver que $P(X \leq a) = P(Z \leq a)$, es decir que X tiene distribución normal.

Así Z es la mezcla o composición de las variables $|Z|$ y $-|Z|$, ambas con ponderación $1/2$:

$$F_Z(x) = 0.5 \cdot F_{|Z|}(x) + 0.5 \cdot F_{-|Z|}(x),$$

Luego, conociendo un algoritmo para la generación de $|Z|$ tenemos el siguiente método para simular Z :

```
def Normal_composicion():
    Generar |Z|
    if random() < 0.5:
        return |Z|
    else:
        return -|Z|
```

Es necesario entonces un método para generar $|Z|$. Su densidad está dada por:

$$f_{|Z|}(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} e^{-x^2/2} & x > 0 \\ 0 & c.c. \end{cases}$$

Una posibilidad es utilizar el **método de rechazo** con una exponencial, por ejemplo, $X = \mathcal{E}(1)$. Denotamos g a la densidad de X . Así, debemos encontrar una cota para:

$$\frac{f_{|Z|}(x)}{g(x)} = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2} + x}.$$

Esta función alcanza un máximo en el punto en que el exponente es máximo, lo cual ocurre en $x = 1$. El valor máximo y por consiguiente la cota mínima c está dada por:

$$c = \sqrt{\frac{2e}{\pi}} \approx 1.32$$

Para describir el algoritmo, calculamos:

$$\frac{f(x)}{c g(x)} = \exp \left\{ -\frac{(x-1)^2}{2} \right\}.$$

Luego el método de rechazo simula $Y_1 \sim \mathcal{E}(1)$, U uniforme, e itera hasta que:

$$U \leq \exp \left\{ -\frac{(Y_1-1)^2}{2} \right\},$$

es decir:

$$\log(U) \leq -\frac{(Y_1-1)^2}{2} \quad \text{o equivalentemente} \quad -\log(U) \geq \frac{(Y_1-1)^2}{2}.$$

Dado que $Y_2 = -\log(U)$ genera una exponencial $\mathcal{E}(1)$, el algoritmo por el método de rechazo para generar Z se traduce en:

```
def Normal_rechazo(mu, sigma):
    while True:
        Y1 = -log(random())
        Y2 = -log(random())
        if Y2 >= (Y1-1) ** 2 / 2:
            if random() < 0.5:
                return Y1 * sigma + mu
            return -Y1 * sigma + mu
```

Recordemos que una variable con distribución exponencial posee la propiedad de **falta de memoria**:

$$P(X > s + t \mid X > t) = P(X > s).$$

Es decir, condicional a que la variable X sea mayor a t , la cantidad que excede a este valor también tiene distribución exponencial. En particular, en el paso del algoritmo anterior en que se acepta el valor de Y_1 , ocurre que Y_2 supera a $(Y_1 - 1)^2/2$ y por lo tanto la cantidad que excede:

$$Y_2 - (Y_1 - 1)^2/2$$

tiene distribución exponencial $\mathcal{E}(1)$.

De esta manera, el método de rechazo para generar $Z \sim N(0, 1)$ permite generar también una exponencial. Esta exponencial podría ser utilizada en pasos sucesivos como el primer valor de Y_1 , y así disminuir en 1 el número de exponenciales generadas. Como el método tiene un promedio de $c \sim 1.32$ iteraciones, el número de exponenciales necesarias será del orden de

$$2 \cdot 1.32 - 1 = 1.64.$$

6.4.2. Método polar

Otro método eficiente para generar variables normales es el **método polar**. El nombre se refiere al uso de coordenadas polares para referenciar puntos del plano:

$$x = r \cos \theta, \quad y = r \sin \theta, \quad r \geq 0, \quad 0 \leq \theta < 2\pi. \quad (6.1)$$

Consideramos dos variables aleatorias X e Y , normales estándar, independientes. Dado que toman cualquier valor real, entonces el par (X, Y) denota a algún punto del plano. Las coordenadas polares correspondientes a este punto, R y Θ , verifican:

$$R^2 = X^2 + Y^2, \quad \Theta = \arctan \frac{Y}{X}.$$

El objetivo es analizar la distribución de las variables R^2 y Θ y simular estas variables para obtener dos normales estándar.

La función de densidad conjunta de X e Y está dada por:

$$f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}.$$

La transformación de coordenadas (d, θ) a (x, y) está dada por:

$$d = x^2 + y^2, \quad \theta = \arctan \frac{y}{x},$$

y por lo tanto la matriz jacobiana de esta transformación es:

$$\begin{pmatrix} \frac{\partial}{\partial x}(x^2 + y^2) & \frac{\partial}{\partial y}(x^2 + y^2) \\ \frac{\partial}{\partial x}(\arctan \frac{y}{x}) & \frac{\partial}{\partial y}(\arctan \frac{y}{x}) \end{pmatrix} = \begin{pmatrix} 2x & 2y \\ -\frac{y}{x^2+y^2} & \frac{x}{x^2+y^2} \end{pmatrix}$$

con determinante jacobiano igual a 2. Por lo tanto, la densidad conjunta de R^2 y Θ satisface:

$$f_{R^2, \Theta}(d, \theta) = \frac{1}{2} f_{X,Y}(x, y) = \frac{1}{4\pi} e^{-d/2}.$$

Así, observando que θ toma valores en $[0, 2\pi)$ y $d > 0$, formalmente resulta:

$$f_{R^2, \Theta}(d, \theta) = \underbrace{\frac{1}{2\pi} \mathbb{I}_{[0, 2\pi)}(\theta)}_{\Theta \sim U(0, 2\pi)} \cdot \underbrace{\frac{1}{2} e^{-d/2} \mathbb{I}_{[0, \infty)}(d)}_{R^2 \sim \mathcal{E}(\frac{1}{2})}.$$

Es decir, R^2 y Θ resultan ser variables aleatorias independientes, con distribución exponencial $\mathcal{E}(\frac{1}{2})$ y uniforme $U(0, 2\pi)$, respectivamente. El **Método polar** genera dos variables normales estándar, y el algoritmo es como sigue. Corresponde a la función `normal.gauss()` de Python.

```
def MetodoPolar():
    Rcuadrado = -2 * log(1 - random())
    Theta = 2 * Pi * random()
    X = sqrt(Rcuadrado) * cos(Theta)
    Y = sqrt(Rcuadrado) * sen(Theta)
    return (X * sigma + mu, Y * sigma + mu)
```

6.4.3. Transformaciones de Box-Muller

Las transformaciones:

$$\begin{cases} X = \sqrt{-2 \log(U_1)} \cos(2\pi U_2) \\ Y = \sqrt{-2 \log(U_1)} \sin(2\pi U_2) \end{cases} \quad (6.2)$$

se denominan **transformaciones de Box-Muller**. Una desventaja de estas transformaciones es que requieren el cálculo de dos funciones trigonométricas: seno y coseno. Para mejorar este paso, notemos que si generamos uniformemente puntos en el círculo unitario, y (V_1, V_2) son las coordenadas de un punto aleatorio en este círculo, entonces para cada r , $0 < r < 1$ se cumple que:

$$P(V_1^2 + V_2^2 \leq r) = P(\sqrt{V_1^2 + V_2^2} \leq \sqrt{r}) = \frac{\pi r}{\pi} = r,$$

y para α , $0 \leq \alpha < 2\pi$,

$$P(0 < \arctan \frac{V_2}{V_1} < \alpha) = \frac{\alpha/2}{\pi} = \frac{1}{2\pi} \alpha.$$

Así, las variables aleatorias $S^2 = V_1^2 + V_2^2$ y $\Theta = \arctan \frac{V_2}{V_1}$ resultan uniformemente distribuidas en $(0, 1)$ y $(0, 2\pi)$ respectivamente. Además por propiedades geométricas no es difícil ver que son independientes:

$$P(S^2 \leq r, \Theta < \alpha) = r \cdot \frac{\alpha}{2\pi}.$$

Así tenemos que:

$$\cos \Theta = \frac{V_1}{\sqrt{V_1^2 + V_2^2}} = \frac{V_1}{\sqrt{S^2}}, \quad \sin \Theta = \frac{V_2}{\sqrt{V_1^2 + V_2^2}} = \frac{V_2}{\sqrt{S^2}}. \quad (6.3)$$

En resumen, para generar $\cos \Theta$ y $\sin \Theta$ debemos:

1. Generar un punto aleatorio (V_1, V_2) del círculo unitario.
2. Calcular $\cos \Theta$ y $\sin \Theta$ usando (6.3).

La generación de puntos aleatorios en el círculo se reduce a generar pares de puntos (V_1, V_2) en el cuadrado $[-1, 1] \times [-1, 1]$ hasta obtener uno en el círculo unitario. Las **transformaciones de Box-Muller** se reescriben entonces como:

$$X = \sqrt{-2 \log U} \frac{V_1}{\sqrt{(V_1^2 + V_2^2)}} \quad (6.4)$$

$$Y = \sqrt{-2 \log U} \frac{V_2}{\sqrt{(V_1^2 + V_2^2)}} \quad (6.5)$$

Ahora, como $S^2 = V_1^2 + V_2^2$ es uniforme en $(0, 1)$ y es independiente de Θ , puede ser utilizado como U en (6.4) y (6.5), resultando las ecuaciones:

$$X = \sqrt{-2 \log U} \cdot \frac{V_1}{\sqrt{U}} = V_1 \cdot \sqrt{\frac{-2 \log U}{U}} \quad (6.6)$$

$$Y = \sqrt{-2 \log U} \cdot \frac{V_2}{\sqrt{U}} = V_2 \cdot \sqrt{\frac{-2 \log U}{U}} \quad (6.7)$$

Finalmente, el método polar con las transformaciones (6.6) para simular dos variables normales resulta:

```

def Polar_Box_Muller(mu, sigma):
    #Generar un punto aleatorio en el círculo unitario.
    while True:
        V1, V2 = 2 * random()-1, 2 * random()-1
        if V1 ** 2 + V2 ** 2 <= 1:
            S = V1 ** 2 + V2 ** 2
            X = V1 * sqrt(-2 * log(S) / S)
            Y = V2 * sqrt(-2 * log(S) / S)
            return (X * sigma + mu, Y * sigma + mu)

```

6.4.4. Método de razón entre uniformes

El método de razón entre uniformes se aplica en la generación de variables aleatorias continuas. Para ello, dada X una variable aleatoria continua con función de densidad f se considera el conjunto del plano C_f dado por:

$$C_f = \left\{ (u, v) \mid 0 < u < \sqrt{f(v/u)} \right\}.$$

Kinderman y Monahan¹ demostraron el siguiente resultado:

Teorema 6.2. Si U y V son variables aleatorias continuas tales que (U, V) está uniformemente distribuida en C_f entonces la variable aleatoria $X = \frac{V}{U}$ tiene distribución f .

Para ver esto, consideremos U y V como en el teorema. Su distribución conjunta está dada por

$$f_{U,V}(u, v) = \frac{1}{|C_f|} \cdot \mathbb{I}_{C_f}(u, v),$$

donde \mathbb{I}_{C_f} denota la función indicadora del conjunto C_f . Sea $T : C_f \mapsto \mathbb{R}^2$ dada por:

$$T(u, v) = \left(\frac{v}{u}, u \right) = (x, y).$$

T es biyectiva sobre su imagen y el valor absoluto del jacobiano de esta transformación es:

$$\left| \det \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \right| = \left| \det \begin{pmatrix} -\frac{v}{u^2} & \frac{1}{u} \\ 1 & 0 \end{pmatrix} \right| = \left| -\frac{1}{u} \right| = \frac{1}{y}.$$

Luego, si $T(U, V) = (X, Y)$, la densidad conjunta de X e Y está dada por

$$f_{X,Y}(x, y) = y \cdot f_{U,V}(T^{-1}(x, y)) = \begin{cases} \frac{1}{|C_f|} y & \text{si } 0 < y < \sqrt{f(x)} \\ 0 & \text{en otro caso,} \end{cases}$$

¹A. J. KINDERMAN AND J. F. MONAHAN, *Computer generation of random variables using de ratio of uniform deviates*. ACM Transactions on Mathematical Software, Vol 3, No. 3, September 1977, Pages 257-260

y así la densidad marginal de $X = V/U$ es:

$$f_X(x) = \int_0^{\sqrt{f(x)}} \frac{1}{|C_f|} y dy = \frac{1}{2|C_f|} f(x).$$

En particular, para que f_X sea densidad debe ser $|C_f| = \frac{1}{2}$, y queda demostrado que $f_X(x) = f(x)$.

De esta manera, para generar una variable aleatoria X con densidad f puede seguirse el siguiente algoritmo:

1. Generar un vector aleatorio (U, V) uniformemente en un rectángulo $(0, c) \times (a, b)$ que contenga a C_f .
2. Si $U^2 < f(V/U)$ devolver $Z = \frac{V}{U}$, sino volver al paso 1.

Para el caso de Z con distribución normal estándar, el conjunto C_f está dado por

$$C_f = \{(u, v) \mid 0 < u < \frac{1}{\sqrt[4]{2\pi}} e^{-\frac{v^2}{4u^2}}\}.$$

Para simplificar la escritura, llamamos $C = \sqrt[4]{2\pi}$. Así, un par (u, v) pertenece a C_f si y sólo si

$$\ln(u \cdot C) < -\frac{v^2}{4u^2} \quad \text{si y sólo si} \quad 0 \leq v^2 < -4u^2 \ln(u \cdot C), \quad (6.8)$$

La condición (6.8) requiere que $0 < u \leq 1/C$ porque de lo contrario $\ln(u \cdot C) > 0$ o el logaritmo no está definido. Notemos que la función:

$$G(u) = -4u^2 \ln(u \cdot C), \quad 0 < u \leq 1/C$$

alcanza un máximo en el valor u tal que $G'(u) = 0$. Dado que:

$$G'(u) = -4(2u \ln(u \cdot C) + u) = -4u \cdot (2 \ln(u \cdot C) + 1),$$

el máximo se alcanza en $u = e^{-0.5}/C$ y el valor máximo es

$$G(e^{-0.5}/C) = -4 \frac{1}{eC^2} (-0.5) = \frac{2}{eC^2}.$$

Así la ecuación (6.8) implica que $|v| < \frac{2}{C\sqrt{2e}}$. Sea $b = \frac{2}{C\sqrt{2e}}$. Entonces todo par (u, v) en el conjunto C_f satisface que $0 < C \cdot u < 1$ y $-b < v < b$, o lo que es lo mismo C_f está comprendido dentro del rectángulo:

$$R = [0, \frac{1}{C}] \times [-b, b].$$

Así, para generar pares (U, V) uniformemente en C_f podemos generar una secuencia de pares (U, V) uniformes en R hasta que cumplan la condición (6.8). Equivalentemente, generamos pares (U_1, U_2) uniformes en $[0, 1) \times [0, 1)$ y tomamos $U = U_1/C$, $V = 2b \cdot (U_2 - 0.5)$. Luego chequeamos (6.8) observando que

$$Z = \frac{V}{U} = \frac{\frac{4}{C \cdot (2e)^{1/2}} (U_2 - 0.5)}{\frac{U_1}{C}} = \frac{4}{(2e)^{1/2}} \frac{U_2 - 0.5}{U_1},$$

y también que

$$U^2 < f(Z) \quad \text{es equivalente a} \quad U_1^2 < e^{-Z^2/2} \quad \text{o} \quad Z^2 < -4 \ln(U_1).$$

Los pasos son entonces:

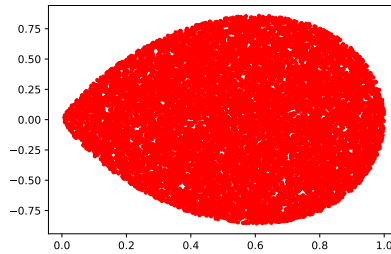
1. Generar $U_1, U_2 \sim \mathcal{U}(0, 1)$.
2. Calcular $Z = (2e)^{-1/2} \frac{U_2 - 0.5}{U_1}$.
3. Si $Z^2 < -4 \ln(U_1)$ devolver Z . Si no volver al paso 1.

El siguiente código es la implementación en Python del método de razón de uniformes para la función `random.normalvariate(mu, sigma)`.

```
from math import exp
NV_MAGICCONST = 4 * exp(-0.5) / sqrt(2.0)
def normalvariate(mu, sigma):
    while 1:
        u1 = random()
        u2 = 1.0 - random()
        z = NV_MAGICCONST * (u1 - 0.5) / u2
        zz = z * z / 4.0
        if zz <= -log(u2):
            break
    return mu + z * sigma
```

La probabilidad de aceptación en este algoritmo es el cociente entre el área de C_f y el área de R :

$$\frac{\text{Área de } C_f}{\text{Área de } R} = \frac{0.5}{\frac{1}{\sqrt{2\pi}} \frac{4}{\sqrt{2e}}} = \frac{\sqrt{\pi e}}{4} = 0.7306$$

Figura 6.1: Región C_f para la distribución normal estándar

6.5. Generación de un Proceso de Poisson

6.5.1. Procesos de Poisson homogéneos

En un proceso de Poisson homogéneo de razón λ , los tiempos de llegada entre eventos sucesivos son exponenciales con parámetro λ . Así, para generar los primeros n eventos de un Proceso de Poisson homogéneo, generamos exponenciales $X_i \sim \mathcal{E}(\lambda)$, $1 \leq i \leq n$:

- Primer evento: al tiempo X_1 .
- j -ésimo evento: al tiempo $X_1 + X_2 + \dots + X_j$, para $1 \leq j \leq n$.

Para generar los eventos en las primeras T unidades de tiempo, generamos eventos hasta que $j + 1$ excede a T . El siguiente algoritmo devuelve NT al número de eventos que ocurren hasta el tiempo T . El i -ésimo elemento de *Eventos* (*Eventos*[$i - 1$]) indica el tiempo en que ocurre el evento i , $1 \leq i \leq NT$:

```
def eventosPoisson(lamda, T):
    t = 0
    NT = 0
    Eventos = []
    while t < T:
        U = 1 - random()
        t += - log(U) / lamda
        if t <= T:
            NT += 1
            Eventos.append(t)
    return NT, Eventos
```

Recordemos que a partir de una variable $\text{Gamma}(n, \lambda^{-1})$ y $n - 1$ uniformes, es posible generar n exponenciales. Para ello, ordenamos las $n - 1$ uniformes en forma creciente y los valores de estas uniformes indican los tiempos de arribo.

Por otro lado, se puede probar el siguiente resultado:

Proposición 6.2. Dado $N(T)$, la distribución de los tiempos de arribo en un Proceso de Poisson homogéneo de intensidad λ es uniforme en $(0, T)$.

Demostración. En el material complementario. □

Por lo tanto, un método alternativo para generar los tiempos de arribo hasta el tiempo T consiste en:

- Generar una variable aleatoria Poisson de media λT , y tomar $n = N(T)$.
- Generar n variables aleatorias uniformes U_1, U_2, \dots, U_n .
- Ordenarlas: $U_{i_1} < U_{i_2} < \dots < U_{i_n}$.
- Los tiempos de arribo son: $TU_{i_1}, TU_{i_2}, \dots, TU_{i_n}$.

Si bien este algoritmo posee la ventaja de no generar una secuencia de exponenciales, requiere realizar un ordenamiento de $n = N(T)$ números.

6.5.2. Procesos de Poisson no homogéneos

Consideremos un proceso de Poisson no homogéneo con función de intensidad $\lambda(t)$, y sea λ tal que

$$\lambda(t) \leq \lambda.$$

En la Proposición 2.1 hemos visto que si $M(t)$ es un proceso homogéneo con intensidad λ , y consideramos $N(t)$ el proceso que cuenta los eventos de $M(t)$ con probabilidad $\frac{\lambda(t)}{\lambda}$, entonces $N(t)$ resulta un proceso de Poisson no homogéneo con intensidad $\lambda(t)$. Esta propiedad nos permite proponer el siguiente algoritmo de generación de un proceso de Poisson no homogéneo con función de intensidad $\lambda(t)$:

```
def Poisson_no_homogeneo_adelgazamiento(T):
    'Devuelve el número de eventos NT y los tiempos en Eventos'
    'lamda_t(t): intensidad, lamda_t(t) <= lamda'
    NT = 0
    Eventos = []
    U = 1 - random()
    t = -log(U) / lamda
    while t <= T:
        V = random()
```

```

if V < lamda_t(t) / lamda:
    NT += 1
    Eventos.append(t)
    t += -log(1 - random()) / lamda
return NT, Eventos

```

Nuevamente, NT finaliza con el número de eventos hasta el tiempo T , es decir es $N(T)$, y $Eventos[0], Eventos[1], \dots, Eventos[NT - 1]$ son los tiempos de cada evento.

Notemos que el algoritmo es más eficiente cuánto más cerca esté λ de $\lambda(t)$, ya que en ese caso $\lambda(t)/\lambda$ es próximo a 1. Una mejora de este algoritmo es particionar el intervalo $(0, T)$ en subintervalos, y aplicar el algoritmo anterior en cada uno de ellos con un valor λ_i adecuado. Esto es, considerar k intervalos consecutivos $[t_{i-1}, t_i)$, $1 \leq i \leq k$ tales que:

$$0 = t_0 < t_1 < \dots < t_k = T,$$

y valores $\lambda_1, \lambda_2, \dots, \lambda_k$ que cumplan

$$\lambda(s) \leq \lambda_i, \quad s \in [t_{i-1}, t_i).$$

Ejemplo 6.6. Consideremos un proceso de Poisson no homogéneo con función de intensidad

$$\lambda(t) = 2t + 1, \quad 0 \leq t \leq 6.$$

Una posibilidad es implementar el algoritmo recién visto a través de la generación de un proceso de Poisson homogéneo con intensidad constante $\lambda = 13$. Pero también podemos subdividir el intervalo $[0, 6]$ en tres subintervalos:

$$I_1 = [0, 2], \quad I_2 = (2, 4], \quad I_3 = (4, 6],$$

y acotar $\lambda(t)$ con $\lambda_1 = 5$, $\lambda_2 = 9$ y $\lambda_3 = 13$ respectivamente.

Notemos que por la propiedad de falta de memoria de la exponencial, si $t \in I_i$ y $t - \frac{\ln(1-U)}{\lambda_i} \in I_j$ con $j > i$, entonces $\frac{\lambda_i}{\lambda_j} \cdot (t - t_i)$ es exponencial con parámetro λ_j . Esta propiedad resulta útil para aprovechar la última exponencial generada en cada subintervalo.

Así, para este ejemplo en particular, el algoritmo resulta:

```

def Poisson_adelgazamiento_mejorado(T):
    interv = [2, 4, 6] #T<=6
    lamda = [5, 9, 13]
    j = 0 #recorre subintervalos.
    t = -log(1 - random()) / lamda[j]
    NT = 0

```

```

Eventos = []
while t <= T:
    if t <= interv[j]:
        V = random()
        if V < (2 * t + 1) / lamda[j]:
            NT += 1
            Eventos.append(t)
            t += -log(1 - random()) / lamda[j]
        else: #t > interv[j]
            t = interv[j] + (t - interv[j]) * lamda[j] / lamda[j + 1]
            j += 1
return NT, Eventos

```

6.6. Método de aceptación y rechazo transformado

El método de aceptación y rechazo para simular una variable aleatoria X con densidad f se basa en simular una variable aleatoria Y con densidad g , y aceptar el valor Y generado con probabilidad $\frac{f(Y)}{cg(Y)}$ para cierto $c > 1$. El **método de rechazo transformado** se basa en la misma idea pero utiliza la inversa de la distribución acumulada de Y en lugar de la densidad.

Si g es la función de densidad de Y , entonces su función de distribución acumulada está dada por:

$$G(x) = \int_{-\infty}^x g(t) dt, \quad x \in \mathbb{R}.$$

G toma valores en el intervalo $[0, 1]$ y al igual que para el método de la transformada inversa asumiremos además que su inversa está definida sobre $(0, 1)$ y la denotaremos H :

$$H(u) = G^{-1}(u), \quad u \in (0, 1).$$

Hemos visto además que si U es una variable aleatoria con distribución uniforme en $(0, 1)$ entonces $H(U)$ tiene igual distribución que Y . Por lo tanto el método de aceptación y rechazo podría describirse con los siguientes pasos:

```

def Aceptacion_Rechazo_X():
    while 1:
        U = random()
        V = random()
        if V < f(H(U)) / (c * g(H(U))):
            return H(U)

```

Ahora bien, g es la densidad de Y , y por lo tanto $g(x) = G'(x)$. Además, por ser H la inversa de G tenemos que $G(H(u)) = (G \circ H)(u) = u$ y por lo tanto

$$(G \circ H)'(u) = 1 \quad (6.9)$$

Pero

$$(G \circ H)'(u) = G'(H(u)) \cdot H'(u) = g(H(u)) \cdot H'(u). \quad (6.10)$$

De (6.9) y (6.10) se sigue que

$$g(H(u)) = \frac{1}{H'(u)}.$$

Por lo tanto el método de aceptación y rechazo utilizando la transformada inversa resulta:

```
def Aceptacion_Rechazo_Transformado():
    while 1:
        U = random()
        V = random()
        if V < f(H(U)) * Hprima(U) / c: ## Hprima = derivada de H
            return H(U)
```

Ejemplo 6.7. Para ilustrar la aplicación del método consideremos nuevamente la simulación de la variable $\text{Gamma}(\frac{3}{2}, 1)$ como en el Ejemplo 6.5. Así, si se simula la variable X rechazando con una variable aleatoria exponencial $Y \sim \mathcal{E}(2/3)$ tenemos:

$$f(x) = \frac{2}{\sqrt{\pi}} x^{1/2} e^{-x}, \quad g(x) = \frac{2}{3} e^{-\frac{2}{3}x}, \quad x > 0$$

y la constante c dada por:

$$c = 3\sqrt{\frac{3}{2\pi e}}.$$

La función de distribución acumulada de Y es $G(x) = 1 - e^{-\frac{2}{3}x}$ para $x > 0$. Su inversa $H : (0, 1) \mapsto \mathbb{R}^{>0}$ y su derivada están dadas por:

$$H(u) = -\frac{3}{2} \ln(1 - u), \quad H'(u) = \frac{3}{2(1 - u)}.$$

Para el método de aceptación y rechazo transformado es necesario calcular $f(H(u)) \cdot H'(u)$, para $0 < u < 1$:

$$\begin{aligned} f(H(u)) \cdot H'(u) &= \frac{2}{\sqrt{\pi}} \left(-\frac{3}{2} \ln(1 - u) \right)^{1/2} e^{\frac{3}{2} \ln(1 - u)} \frac{3}{2(1 - u)} \\ &= 3\sqrt{\frac{3}{2\pi}} (-\ln(1 - u))^{1/2} (1 - u)^{3/2} \frac{1}{1 - u} \\ &= 3\sqrt{\frac{3}{2\pi}} (-\ln(1 - u) (1 - u))^{1/2}. \end{aligned}$$

Dividiendo por la constante c resulta:

$$\frac{f(H(u)) \cdot H'(u)}{3\sqrt{\frac{3}{2\pi e}}} = \frac{1}{3\sqrt{\frac{3}{2\pi e}}} \cdot 3\sqrt{\frac{3}{2\pi}} (-\ln(1-u)(1-u))^{1/2} = e^{1/2} (-\ln(1-u)(1-u))^{1/2}.$$

Así la implementación del método de aceptación y rechazo transformado resulta:

```
exp_un_medio = 1.6487212 ## exp(0.5)
def Gamma():
    while True:
        U = 1 - random()
        V = random()
        if V < exp_un_medio * sqrt(-log(U) * U):
            return -1.5 * log(U)
```

6.6.1. Método de rechazo transformado con compresión

Una particularidad del método de rechazo transformado es que la función

$$\frac{1}{c} \cdot f(H(u)) \cdot H'(u), \quad 0 < u < 1 \quad (6.11)$$

tiene su gráfico en el cuadrado unitario $[0, 1] \times [0, 1]$. El algoritmo consiste en generar puntos aleatorios (U, V) en este cuadrado; si el punto queda por debajo del gráfico de la función se devuelve el valor $H(U)$ y de lo contrario se repite el proceso. Ahora bien, si la variable Y utilizada para rechazar tiene una densidad próxima a la densidad de X , entonces por debajo del gráfico de la función en (6.11) se podrá ubicar un rectángulo con un área más o menos grande. Por ejemplo, para el caso del Ejemplo 6.7, el gráfico de la función dada en (6.11) y un posible rectángulo bajo el gráfico es como en la Figura 6.6.1.

Notemos que si en el algoritmo un punto (U, V) cae en el rectángulo entonces el valor $H(U)$ será necesariamente aceptado, y por lo tanto no es necesario computar $f(H(U)) H'(U)$. De esta manera, si el rectángulo es de la forma $[u_1, u_2] \times [0, v_1]$ el algoritmo general puede modificarse como:

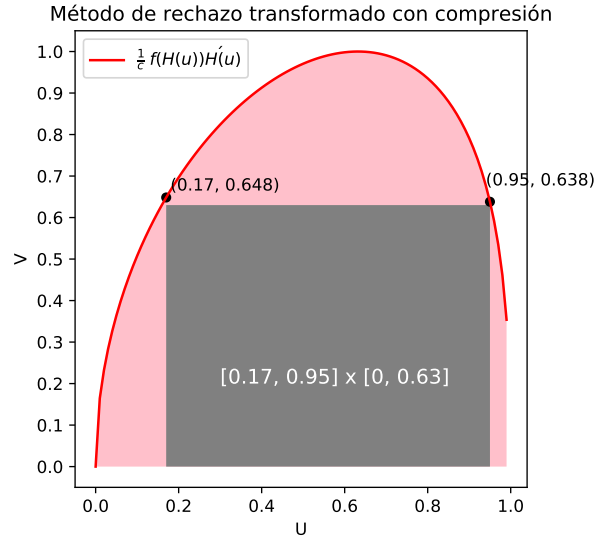


Figura 6.2: $\frac{1}{c} f(H(u)) \cdot H'(u) = e^{1/2} (-(1-u) \ln(1-u))^{1/2}$

```
def Rechazo_con_compresion():
    while True:
        U = random()
        V = random()
        if u_1 <= U <= u_2 and V <= v_1:
            return H(U)
        if V < f(H(U)) * Hprima(U) / c:
            return H(U)
```

Así, si bien el número esperado de iteraciones del algoritmo para devolver un valor sigue siendo c , se logra reducir el número de evaluaciones $f(H(U)) H'(U)$ en una cantidad proporcional al área del rectángulo, siendo ahora:

$$c \cdot (1 - (u_2 - u_1) \cdot v_1).$$

En el Ejemplo 6.7, la constante c es aproximadamente $c \simeq 1.2573168$. Tomando $u_1 = 0.17$, $u_2 = 0.95$, $v_1 = 0.63$ como se muestra en la Figura 6.6.1, el número de evaluaciones promedio se reduce a:

$$1.2573167 \cdot (1 - 0.78 \cdot 0.63) \simeq 0.64.$$

En otras palabras, si para generar 100 números el algoritmo requiere realizar unas 126 iteraciones, aproximadamente caen en el rectángulo unos 62 ($\sim 126 \cdot 0.78 \cdot 0.63$) puntos y por lo tanto sólo en 64 de las 126 iteraciones será necesario evaluar $f(H(U)) \cdot H'(U)$.

En el artículo de Wolfgang Hörmann² se describe la aplicación de este método para el caso particular de la generación de una variable normal y de una variable Poisson, tomando como H un representante de una familia especial de funciones de distribución de variables continuas. Dado que una Poisson se trata de una variable discreta, se utiliza como densidad f el histograma normalizado, esto es, $f(x) = P(X = \lfloor x \rfloor)$ para $x \in \mathbb{R}$. El lenguaje Python implementa la generación de $X \sim \mathcal{P}(\lambda)$, para $\lambda \geq 10$ basándose en el trabajo de Hörman, mientras que para $\lambda < 10$ aplica suma de exponenciales como hemos descripto en la Sección 6.2.2.

²Wolfgang Hörman, *New Generators of Normal and Poisson Deviates Based on Transformed Rejection Method*. Department of Applied Statistics and Data Processing. Wirtschaftsuniversität Wien. Preprint Series. Preprint 4. (1992)