

SIMULACIÓN DE MONTE CARLO



Daniel Rigou

2021 (VERSION: 1.0)

*He deals the cards to find the answer
The sacred geometry of chance
The hidden law of a probable outcome
The numbers lead a dance*

*Shape of my heart
Sting*

CONTENIDO

Introducción	4
Descripción general del método	5
Definición de simulación de Monte Carlo	6
Generación de números aleatorios uniformes	7
Método del cuadrado medio	8
Método congruencial mixto	8
Características de la distribución uniforme	10
Generación de variables aleatorias discretas	12
Generación de variables aleatorias continuas	16
Método de la transformada inversa	17
Método de composición	22
Generación de números aleatorios normales	24
Ejemplo #1: Estimación del peso de cilindros	25
Intervalo de confianza y número de iteraciones	27
Ejemplo #2: Estimación de π – Intervalos de confianza para variable binomiales	29
Ejemplo #3: Comparación de alternativas de financiación	31
Tipos de modelos de simulación y métodos de resolución	35
Modelos de simulación estáticos	35
Modelos de simulación dinámicos	35
Ejemplo #4: Simulación regenerativa	39
Ejemplo #5: Modelo de stock aleatorio	41
Generación de números aleatorios en Excel®	45
Generación de números aleatorios en “R”	46
Bibliografía	47
Anexo I: Código “R” para el Ejemplo #1	48
Anexo II: Código “R” para el Ejemplo #2	49
Anexo III: Código “R” para el Ejemplo #3	50
Anexo IV: Código “R” para el Ejemplo #5	52
Anexo V: Tabla de números aleatorios uniformes	54
Anexo VI: Tabla de números aleatorios normales	55

INTRODUCCIÓN

La simulación numérica de sistemas complejos en los que intervienen procesos aleatorios, ha demostrado ser una herramienta eficaz para resolver aquellos casos en los que desarrollar modelos algebraicos resulta dificultoso o incluso imposible. El método que abordaremos aquí está orientado a desarrollar modelos que combinan un conjunto de variables aleatorias con diferentes distribuciones de probabilidad. Estos modelos son de utilidad en diversos campos: economía y finanzas, gestión de proyectos de inversión, gestión de negocios, aplicaciones en sistemas físicos, químicos e incluso biológicos. El método de Monte Carlo toma ventaja de la capacidad de cálculo de las computadoras digitales para resolver problemas complejos allí donde la formulación analítica del problema o la experimentación con el sistema real no es posible. La capacidad de cálculo es imprescindible porque el método se basa en la ejecución repetida de ensayos o iteraciones; en cada iteración se seleccionan valores para las variables aleatorias del problema y se resuelve el modelo matemático con esos valores. Los resultados numéricos obtenidos para las variables de salida del modelo son luego tratados con procedimientos estadísticos para su correcta interpretación.

Los modelos de simulación de Monte Carlo son modelos descriptivos; es decir que los resultados refieren al comportamiento o desempeño de un sistema bajo determinadas condiciones de operación. Esta técnica nos permite evaluar los méritos de diferentes cursos de acción mediante la experimentación con un modelo matemático que representa una situación real sobre la que deben tomarse decisiones. En cambio, los modelos de optimización dan por respuesta la mejor estrategia para operar un sistema. Un ejemplo de este tipo de modelos son los modelos de programación lineal que dan como resultado una acción óptima.

La simulación es un proceso por medio del cual se realizan experimentos sobre un modelo estocástico que reproduce el comportamiento dinámico del sistema real. Como se trata de un experimento estadístico desarrollado en una computadora digital, los resultados deben interpretarse como tales; es decir mediante pruebas estadísticas adecuadas.

El método de simulación tiene varios nombres: simulación aleatoria, simulación de Monte Carlo, método de Monte Carlo, entre otros. El tipo de simulación que abordaremos aquí es diferente a la simulación de procesos o a la simulación de reservorios de petróleo y gas.

La forma actual de esta técnica fue desarrollada en el laboratorio de Los Álamos (Estados Unidos) durante la Segunda Guerra Mundial para resolver problemas probabilísticos vinculados con la difusión de neutrones en un material fisionable y estudiar ecuaciones diferenciales que modelan fenómenos físicos. Estos primeros desarrollos fueron hechos por Nicholas Metropolis, Stanislaw Ulam¹ y John von Neumann. Se atribuye a Nicholas Metropolis haber sugerido el nombre de Monte Carlo para esta técnica, inspirado en el famoso casino del principado de Mónaco². La posibilidad de desarrollar este tipo de modelos se realimentó en el hecho que, por esos años, se completó el desarrollo de ENIAC (*Electronic Numerical Integrator And Computer*) una de las primeras computadoras digitales que estaba construida con casi 18000 tubos de vacío y ocupaba una superficie de 167 metros cuadrados.

¹ Nicholas Metropolis; Stanislaw Ulam. *The Monte Carlo Method*. Journal of the American Statistical Association. Vol 44, Nº 247. (Sep. 1949), pp 335-341.

² Nicholas Metropolis. *The Beginning of the Monte Carlo Method*. Los Alamos Science. Special Issue 1987, pp 125-130.

Tanto mediante aplicativos especialmente diseñados para simulación como mediante lenguajes de propósito general, el desarrollo de modelos de simulación de Monte Carlo es relativamente sencillo y sobre todo sumamente flexible.

Descripción general del método

Supongamos que queremos estudiar la variable $x: f(v_1; v_2; \dots v_n)$ donde las variables v_i son variables aleatorias. Supongamos además que la interrelación entre las variables, dada por la función f es compleja. Podemos incluso pensar a la función f , en sentido amplio, como un algoritmo complejo que vincula las variables v_i entre sí y con parámetros que definen el caso en estudio. La forma, en que dados los valores de las v_i se obtiene el valor de x , es perfectamente conocida. Admitamos que conocemos también las características de las variables aleatorias v_i ; son los datos de entrada del modelo que hemos podido estimar mediante, por ejemplo, un relevamiento, o a través de la comparación con casos conocidos estudiados anteriormente, o por algún otro método de estimación.

El método de simulación de Monte Carlo, se implementa en un programa de computación de la siguiente manera:

1. Generar un valor para cada una de las variables aleatorias del modelo (las v_i).
2. Resolver el modelo con los valores obtenidos en el paso 1. Es decir obtener un valor para $x: f(v_1; v_2; \dots v_n)$ con los valores simulados de las variables v_i .
3. Repetir los pasos 1 y 2 un número suficiente de veces (n).
4. Analizar estadísticamente los valores obtenidos para x .

Naturalmente que los valores generados en el paso 1 para las variables aleatorias, tienen que finalmente responder a las características que se habían estimado para cada una de ellas. Por ejemplo, si determinada variable del modelo se estima que es normal de media \bar{u} y desvío σ , el algoritmo tiene que generar en cada paso un valor que provenga de esa distribución de probabilidad. Es decir que completadas las n iteraciones del modelo, los valores utilizados para esa variable debe tener una distribución normal de media \bar{u} y desvío σ .

Finalizado el experimento, dispondremos de n valores para la variable resultado (x). Con estos n valores podremos estimar la naturaleza de esta variable aleatoria; por ejemplo, la distribución de densidad de probabilidad y sus parámetros. Por simplicidad nos hemos referido a una sola variable resultado, pero pueden ser varias.

Es evidente que dentro de una computadora nada puede ocurrir en forma aleatoria. Sin embargo, para el desarrollo del método debemos disponer de un generador de números aleatorios para poder simular las variables aleatorias v_i que intervienen en el modelo. Desarrollaremos luego en detalle este asunto, digamos por el momento que lo que seremos capaces de realizar dentro del programa de computadora, es un algoritmo que generará una serie de números que serán indistinguibles de una serie de números al azar. Estas series son llamadas series de números pseudo-aleatorios. Como explicaremos luego solamente es necesario un generador de números al azar uniformes 0-1; con estos números puede derivarse prácticamente cualquier distribución de probabilidad teórica o empírica. Es por esta razón que casi todos los paquetes de software de propósito general traen embebida alguna función para generar números pseudo-aleatorios uniformes 0-1.

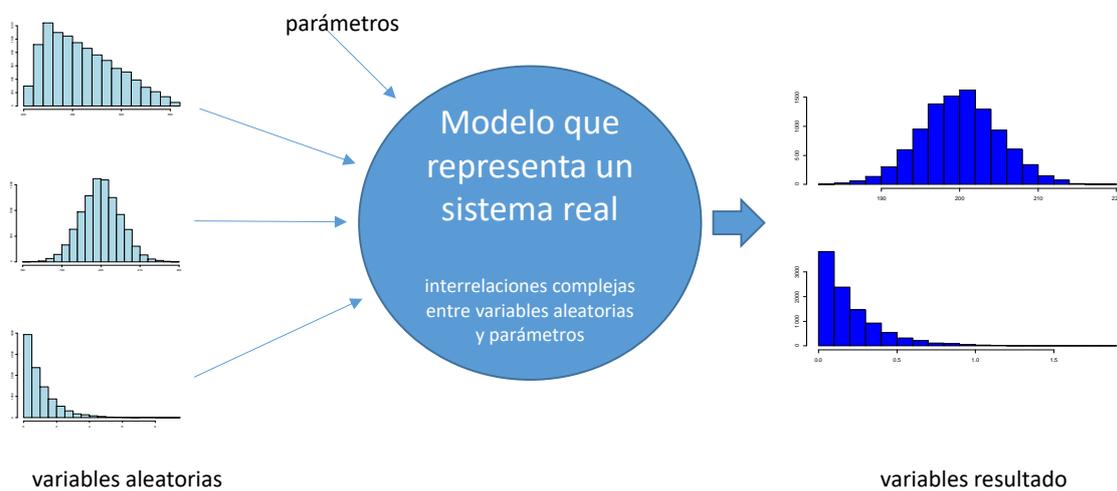
Definición de Simulación de Monte Carlo

Precisamos lo desarrollado hasta aquí con una definición de Simulación de Montecarlo bastante difundida y aceptada³:

Simulación es una técnica numérica para realizar experimentos en una computadora digital. Estos experimentos involucran ciertos tipos de modelos matemáticos y lógicos que describen el comportamiento de sistemas de negocios, económicos, sociales, biológicos, físicos o químicos a través de largos períodos de tiempo.

La definición destaca que se trata de una “técnica numérica”. Es por ello que para la variable (o las variables) resultado obtendremos una serie de valores numéricos sujetos a error experimental; el equivalente a las observaciones en un experimento de laboratorio. Por esta razón suele decirse que la simulación de Monte Carlo es un recurso extremo o último recurso. Si se dispone de un modelo matemático robusto para la resolución de un caso, es preferible utilizar el modelo matemático ya que tiene un mayor grado de precisión.

Otro elemento importante de la definición es que destaca que el campo de aplicación es amplio. Esto nos indica el nivel de flexibilidad que tiene este tipo de modelos y constituye una de las ventajas de los modelos de simulación de Monte Carlo: su aplicabilidad a sistemas con un alto grado de complejidad y aleatoriedad.



³ Thomas H. Naylor, Joseph L. Balintfy, Donald S. Burdick, and Kong Chu. *Computer Simulation Techniques*. John Wiley & Sons, New York, 1966.

GENERACIÓN DE NÚMEROS ALEATORIOS UNIFORMES

En los modelos de simulación existe la necesidad de generar números aleatorios. En particular se requiere poder generar números distribuidos uniformemente entre 0 y 1 ya que a partir de esta distribución uniforme 0-1 (o rectangular 0-1) pueden generarse números aleatorios de cualquier distribución de probabilidad teórica o empírica.

La forma más estricta de garantizar la aleatoriedad es mediante dispositivos mecánicos; por ejemplo ruletas, bolilleros, dados, un mazo de cartas o monedas. Sin embargo esta forma de generar números al azar no es práctica para modelos implementados en una computadora. Existen tablas de números al azar creadas mediante equipamiento físico. La más famosa es la tabla de un millón de números al azar creada por la *Rand Corporation*. La tabla, publicada por primera vez en 1955, fue hecha conectando una rueda de ruleta a una computadora, y filtrando luego los pulsos producidos por este dispositivo mecánico⁴. El uso de estas tablas supone disponer los números aleatorios almacenados en la computadora y utilizarlos como datos de entrada del modelo. Otros generadores físicos utilizan el decaimiento radiactivo o el ruido atmosférico como fuente de aleatoriedad.

Una alternativa a estos números al azar *verdaderos* son los números *pseudo-aleatorios*. Estos números son producidos por algoritmos y, si el algoritmo está bien calibrado, resultan indistinguibles de números verdaderamente aleatorios. Esto quiere decir que superan pruebas estadísticas de aleatoriedad (por ejemplo la prueba χ^2). Este tipo de números si bien son de naturaleza determinística, pueden generarse a alta velocidad y de modo eficiente mediante una computadora. En general los algoritmos funcionan a partir de un valor inicial llamado semilla; sobre este valor semilla se aplica una serie de operaciones matemáticas para obtener un nuevo valor aleatorio. Luego se utiliza el valor obtenido para generar el siguiente número aleatorio. El proceso se repite para la cantidad de números aleatorios requeridos. El valor semilla asegura la posibilidad de repetir la secuencia; esto es importante en el desarrollo y calibración de modelos de simulación ya que utilizando la misma semilla se pueden repetir las mismas corridas y depurar el modelo con mayor facilidad.

Los generadores de números al azar deben cubrir ciertos requisitos:

- Los números deben ser uniformemente distribuidos y estadísticamente independientes.
- El algoritmo debe ser rápido ya que los modelos de simulación requieren varios miles de iteraciones.
- Los generadores necesariamente repiten la secuencia de números. Se necesita por tanto que la secuencia sea lo más larga posible; típicamente de más de un millón de dígitos.
- Debe asegurarse que no se produzca degeneración. Una secuencia de números con degeneración supone que algunos dígitos, (o todos), a partir de determinado punto se repiten en ciclos cortos. Es decir que debe garantizarse la aleatoriedad y no recurrencia en cada dígito.
- Deben requerir poco almacenamiento en la computadora.

⁴ El libro con los antecedentes puede descargarse en:
https://www.rand.org/pubs/monograph_reports/MR1418.html

No es el objetivo de este texto desarrollar en profundidad los algoritmos para generar números aleatorios uniformes; sin embargo, se describirán dos de los más conocidos.

Método del cuadrado medio (*middle-square method*)

Este método se encuentra entre los primeros formulados para generar números pseudo-aleatorios. Lo propuso John Von Newman y consiste en:

- A partir de un número arbitrario, elevarlo al cuadrado y tomar el número formado por las cifras centrales del mismo.
- Estas cifras centrales serán el nuevo número aleatorio que a su vez será elevado al cuadrado repitiendo el proceso.

Las principales deficiencias de este método son que los ciclos son relativamente cortos, no es posible prever la longitud del ciclo; por otra parte si se genera un valor igual a cero, todos los sucesivos valores son también cero. El método es difícil de analizar y no siempre ha producido buenos resultados. Sin embargo se han utilizado series de 750000 números satisfactoriamente en el laboratorio de Los Álamos.

El siguiente ejemplo⁵ genera una secuencia de números aleatorios de cuatro dígitos a partir de una valor semilla = 5134

x_i	x_i^2	Dígitos centrales	Número aleatorio (r)
5134	26357956	3579	0.3579
3579	12809241	8092	0.8092
8092	65480464	4804	0.4804
4804	23078416	0784	0.0784
0784	614656	6146	0.6146

Método congruencial mixto

La idea básica de este tipo de métodos fue introducida por Derrick Henry Lehmer⁶. Del modelo originalmente propuesto por él se crearon numerosas variantes.

La relación de recurrencia es:

$$X_{n+1} = (a \cdot X_n + c) \bmod m$$

Donde:

X_0 : es la semilla

a : es el multiplicador ($a > 0$)

c : es la constante aditiva ($c > 0$)

m : es el módulo ($m > X_0$; $m > a$; $m > c$)

⁵ Tomado de: Fernando Salvador. *TP Simulación* (mimio). Centro de Estudiantes de Ingeniería "La Línea Recta". Buenos Aires 1992.

⁶ D. H. Lehmer. *Mathematical methods in large-scale computing units*. Annals Comp. Laboratory Harvard Univ. 26 (1951), pp. 141-146

La expresión **mod** indica que X_{n+1} es el residuo de la división entre $(a \cdot X_n + c)$ y m ⁷.

La selección de los parámetros debe ajustarse a ciertas reglas para que el generador tenga período completo; es decir que el período sea igual a m . Esta selección está también vinculada al número de bits que tiene la palabra completa de la computadora que se está utilizando.

Es deseable que el módulo sea el número primo más grande posible y que a su vez sea menor que la base utilizada por el sistema (binario, hexadecimal etc.) elevado al número de bits que tiene la palabra de la computadora.

Lehmer propuso como parámetros $a = 23$; $m = 10^8 + 1$ con los que produjo secuencias satisfactorias de más de 5 millones de números con 8 posiciones decimales en las computadoras utilizadas a mediados del Siglo XX⁸.

Actividad 1⁹

Utilice los siguientes parámetros en la ecuación de recurrencia del método congruencial mixto y verifique que, al contrario de lo que inicialmente puede pensarse, el período es menor que el módulo.

$$\begin{aligned}X_0 &= 21 \\a &= 4 \\c &= 1 \\m &= 100\end{aligned}$$

Este ejemplo muestra que una inadecuada selección de los parámetros puede conducir a una serie de números aleatorios con período menor que el módulo.

Los números obtenidos pueden tomarse como una sucesión de números aleatorios dividiéndolos por el módulo ($m = 100$).

⁷ El operador *mod* es equivalente a la función +RESIDUO(a,b) en Excel®; y al operador a%%b en R.

⁸ Hull, T.E. and A.R. Dobell. *Random Number Generators*. SIAM Review 4.3 (1962) 230-254.

⁹ Tomado de: Fernando Salvador. *TP Simulación* (mimio). Centro de Estudiantes de Ingeniería "La Línea Recta". Buenos Aires 1992.

Actividad 2

Genere 1000 números aleatorios uniformes 0-1 con la ecuación de recurrencia del método congruencial mixto. Utilice los parámetros sugeridos por Lehmer:

- $a=23$
- $c = 1$
- $m = 10^8 + 1$

y el valor semilla $X_0 = 51911487$

1. Clasifique los valores obtenidos en diez rangos iguales y realice un histograma.
2. Calcule el valor del estadístico χ^2 para los diez rangos.
3. Para un nivel de significación $\alpha = 0,05$ determine si puede o no rechazar la hipótesis que los números simulados provienen de una distribución uniforme 0-1¹⁰.

Genere 1000 números aleatorios uniformes 0-1 empleando el generador del software de su preferencia, y realice nuevamente las actividades indicadas los puntos 1,2 y 3.

Antes de desarrollar cómo a partir de un generador de números aleatorios uniformes (0,1), es posible obtener números aleatorios de cualquier distribución de probabilidad; es conveniente estudiar las principales características de la distribución uniforme o rectangular.

Características de la distribución uniforme

La función de densidad de la distribución uniforme de probabilidad en (a,b) es:

$$f(x) = \begin{cases} \frac{1}{(b-a)} & : \text{si } a \leq x \leq b \\ 0 & : \text{para otro valor de } x \end{cases}$$

La función acumulada es:

$$F(x) = \begin{cases} 0 & : \text{si } x < a \\ \frac{(x-a)}{(b-a)} & : \text{si } a \leq x \leq b \\ 1 & : \text{si } x > b \end{cases}$$

¹⁰ Condición de rechazo $\chi^2 > \chi_{v,(1-\alpha)}^2$. Para una descripción detallada del ensayo de hipótesis ver: Roberto Mariano García. *Inferencia estadística y diseño de experimentos*. 1ª ed. Buenos Aires, Eudeba 2004; (página. 519).

La esperanza es:

$$E(x) = \int_a^b f(x) \cdot x \cdot dx = \int_a^b \frac{x}{(b-a)} dx = \frac{1}{(b-a)} \frac{x^2}{2} = \frac{b^2 - a^2}{2(b-a)} = \frac{(b+a)}{2}$$

La varianza es:

$$Var(x) = E[(x - \mu)^2] = E(x^2) - \mu^2$$

$$Var(x) = \int_a^b \frac{x^2}{(b-a)} dx - \mu^2 = \frac{1}{3} \frac{(b^3 - a^3)}{(b-a)} - \left(\frac{(b+a)}{2}\right)^2$$

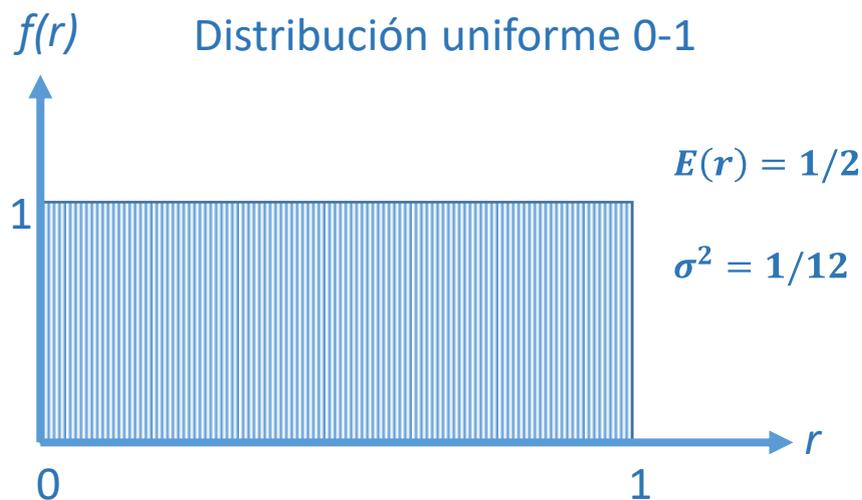
$$Var(x) = \frac{4b^3 - 4a^3 - 3b^3 - 6ab^2 - 3ba^2 + 3ab^2 + 6ba^2 + 3a^3}{12(b-a)}$$

$$Var(x) = \frac{b^3 - a^3 - 3ab^2 + 3ba^2}{12(b-a)} = \frac{(b-a)^3}{12(b-a)} = \frac{(b-a)^2}{12}$$

Para la distribución uniforme 0-1 $a = 0$ y $b = 1$; resultan entonces:

$$E(x) = \frac{(b+a)}{2} = \frac{1}{2}$$

$$Var(x) = \frac{(b-a)^2}{12} = \frac{1}{12}$$



GENERACIÓN DE VARIABLES ALEATORIAS DISCRETAS

Si se conocen las probabilidades de los resultados posibles de una variable estocástica discreta, es posible simularlos asignando proporcionalmente los números aleatorios uniformes a las frecuencias de cada uno de los resultados posibles.

Por ejemplo, supongamos que deseamos simular el resultado de revolver una moneda, y que la frecuencia de ocurrencia del evento *cara* es 50% y del evento *ceca* es 50%. Si se utiliza un generador de números aleatorios uniformes 0-1, es posible asignar el resultado *cara* cada vez que el número generado sea menor que 0.5 y asignar el resultado *ceca* cada vez que el número aleatorio sea mayor o igual a 0.5. Repetido el experimento una cantidad grande de veces, se obtendrá una distribución de los resultados *cara* y *ceca* que será indistinguible de los resultados que se pueden obtener del experimento físico real de revolver una moneda (equilibrada). Siendo r un número aleatorio uniforme 0-1, podemos formalizar el experimento de la siguiente manera:

$$\text{si: } 0 \leq r < 0.5 \rightarrow \textit{cara}$$

$$\text{si: } 0.5 \leq r \leq 1 \rightarrow \textit{ceca}$$

Debemos notar aquí que la división en 0.5 se realiza por simplicidad. Cualquier otra forma de segmentar la variable r que respete la frecuencia de aparición del resultado *cara* y del resultado *ceca* es también válida. Por ejemplo:

$$\text{si: } 0 \leq r < 0.25 \rightarrow \textit{cara}$$

$$\text{si: } 0.25 \leq r < 0.75 \rightarrow \textit{ceca}$$

$$\text{si: } 0.75 \leq r \leq 1 \rightarrow \textit{cara}$$

En esta segmentación, la frecuencia de aparición del resultado *cara* será 50%, porque números aleatorios entre 0 y 0.25, y entre 0.75 y 1 aparecerán el 50% de las veces; y también será 50% la frecuencia de aparición del resultado *ceca*, porque números aleatorios entre 0.25 y 0.75 aparecerán el 50% de las veces. Es evidente que esta segunda forma de segmentar la variable r es más compleja y no ofrece ninguna ventaja a simplemente segmentar en 0.5.

Un ejemplo algo más difícil es el de simular el resultado de arrojar repetidamente un dado de seis caras. Si el dado está equilibrado, podemos realizar la simulación en una computadora segmentando los resultados de los números aleatorios r de la siguiente manera:

$$\text{si: } 0 \leq r < 1/6 \rightarrow 1$$

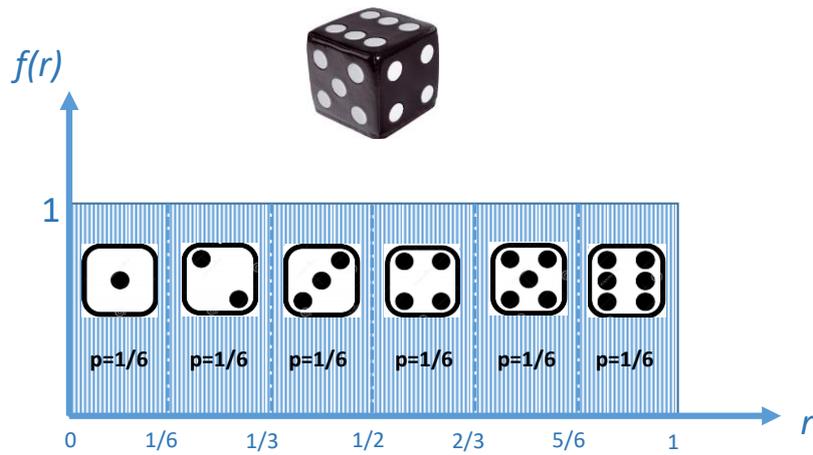
$$\text{si: } 1/6 \leq r < 1/3 \rightarrow 2$$

$$\text{si: } 1/3 \leq r < 1/2 \rightarrow 3$$

$$\text{si: } 1/2 \leq r < 2/3 \rightarrow 4$$

$$\text{si: } 2/3 \leq r < 5/6 \rightarrow 5$$

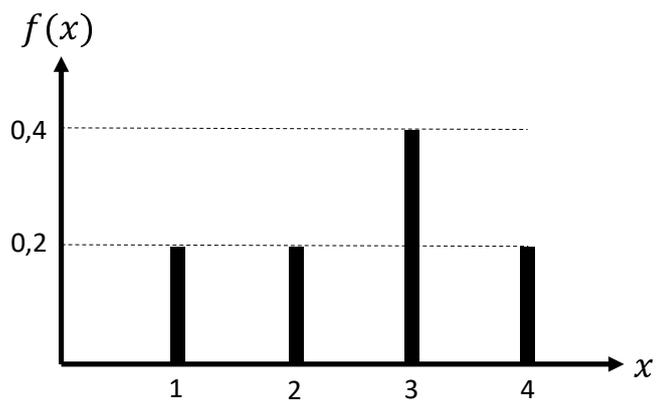
$$\text{si: } 5/6 \leq r \leq 1 \rightarrow 6$$



De este modo cada uno de los seis resultados posibles tendrá una probabilidad de ocurrencia de $1/6$. Los resultados obtenidos en la computadora, serán indistinguibles del experimento real de arrojar repetidas veces un dado equilibrado de seis caras.

Supongamos ahora que deseamos generar números aleatorios provenientes de una distribución discreta de probabilidades como la siguiente:

x	$f(x)$
1	0.2
2	0.2
3	0.4
4	0.2



Es posible simular la variable x asignando proporcionalmente los números aleatorios uniformes r a las frecuencias de cada uno de los 4 resultados posibles.

$$\text{si: } 0 \leq r < 0.2 \rightarrow x = 1$$

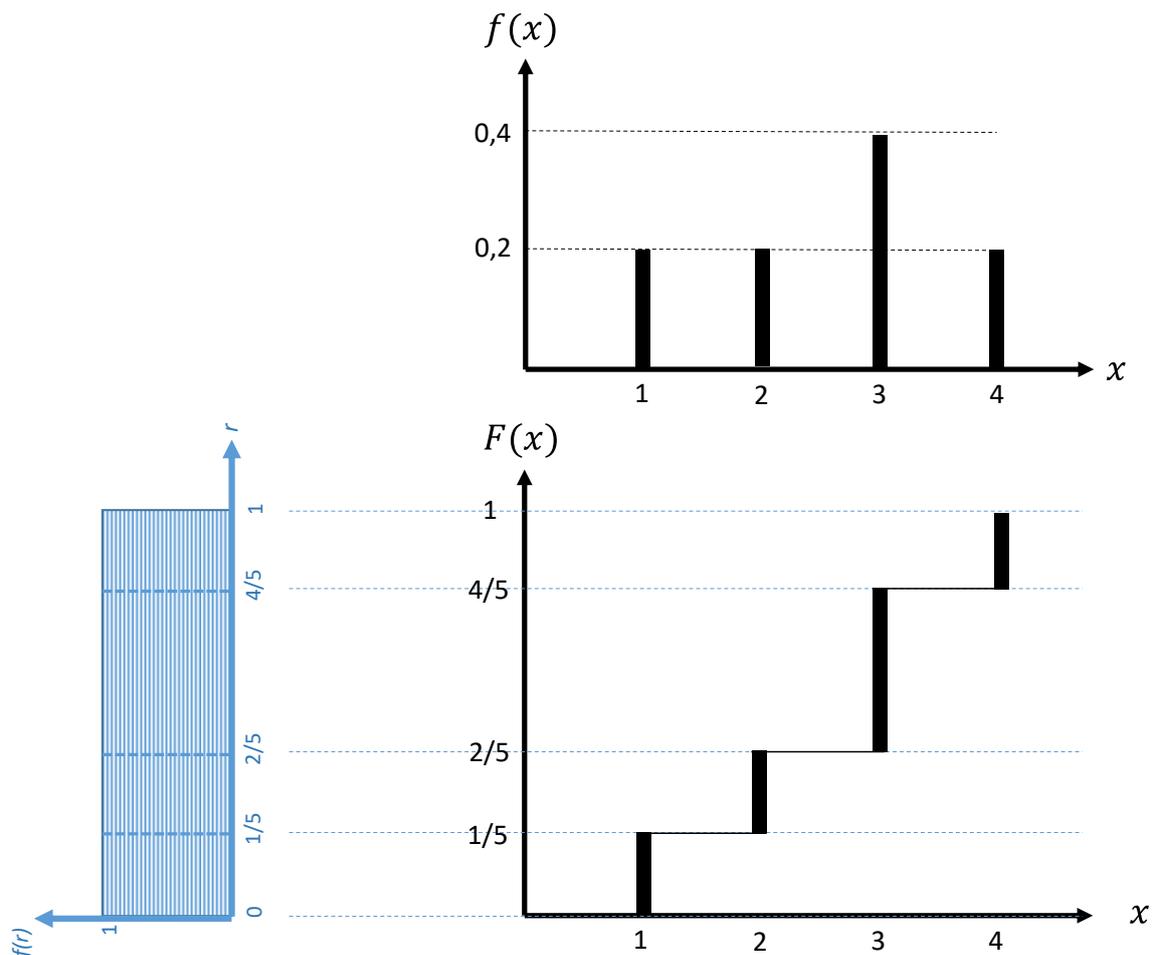
$$\text{si: } 0.2 \leq r < 0.4 \rightarrow x = 2$$

$$\text{si: } 0.4 \leq r < 0.8 \rightarrow x = 3$$

$$\text{si: } 0.8 \leq r \leq 1 \rightarrow x = 4$$

De este modo, la variable x adoptará el valor 1 el 20% de las veces, el valor 2 el 20% de las veces, el valor 3 el 40% de las veces y el valor 4 el restante 20% de las veces. Los valores de la variable x así obtenidos estarán ajustados a la distribución discreta de probabilidades dada $f(x)$.

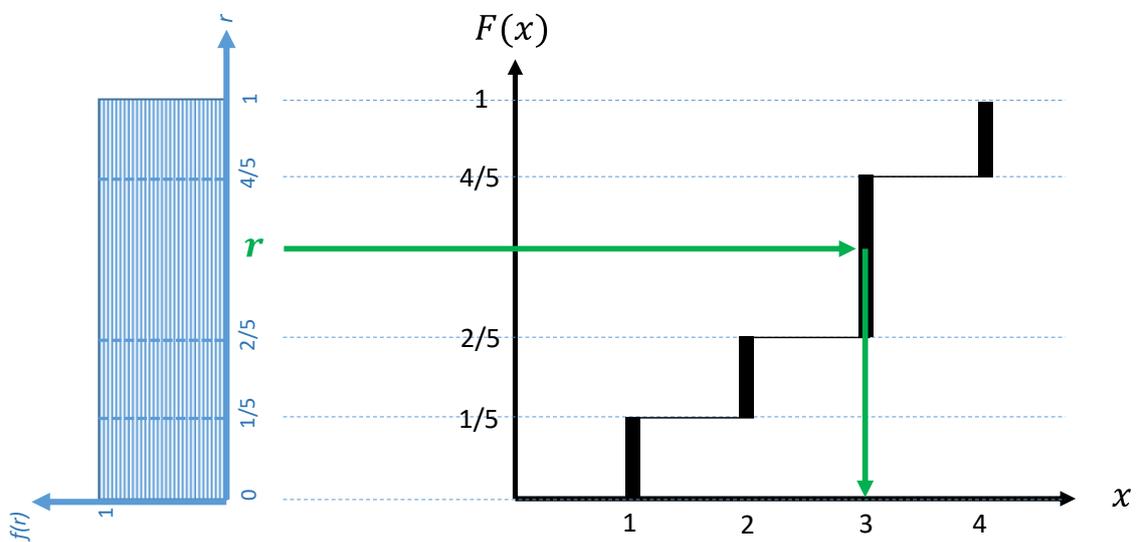
En definitiva, si se conocen las probabilidades de los resultados posibles de una variable discreta, se los puede simular asignando proporcionalmente los números aleatorios uniformes a las frecuencias de cada uno de los resultados. Este procedimiento se puede sistematizar acumulando las probabilidades de la variable aleatoria para obtener, de este modo, los segmentos de la variable aleatoria uniforme. El siguiente gráfico explica lo anterior:



Conocida la probabilidad de cada uno de los posibles valores de la variable x , se obtiene la probabilidad acumulada hasta ese valor $F(x_i)$. Para cada valor generado de la variable aleatoria uniforme 0-1 (r) se selecciona un valor de la variable que se desea simular (x_i) de modo que:

$$F(x_{i-1}) \leq r < F(x_i)$$

La interpretación gráfica consiste en ubicar el valor de r en el eje de ordenadas de $F(x)$ y trazar una horizontal hasta intersectar el gráfico de la función; y desde la intersección, una vertical hasta el eje de abscisas para obtener el valor de la variable x .



Simulación de 5 valores de la variable aleatoria x

r	Rango de $F(x)$	x
0.25593	$1/5 - 2/5$	2
0.44276	$2/5 - 4/5$	3
0.22820	$1/5 - 2/5$	2
0.17200	$0 - 1/5$	1
0.88627	$4/5 - 1$	4

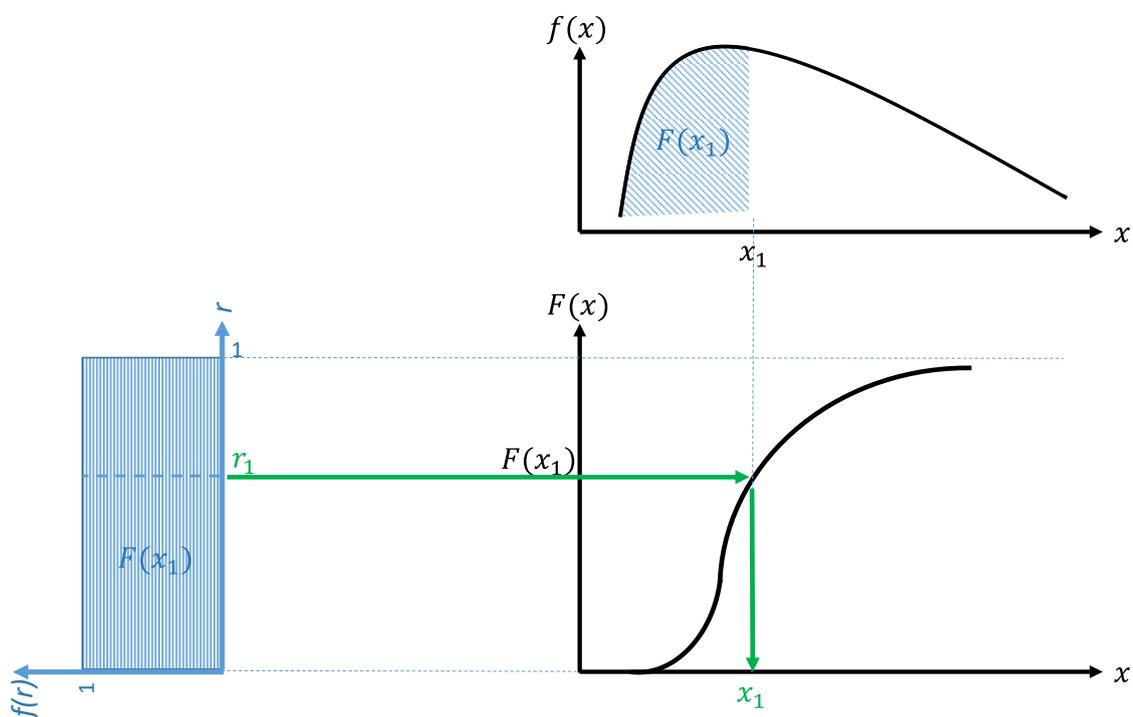
GENERACIÓN DE VARIABLES ALEATORIAS CONTINUAS

En el caso de las distribuciones de probabilidad discretas, hemos visto que para simular un valor de la variable aleatoria, se selecciona un valor de la distribución uniforme 0-1 (r) y se elige el rango de $F(x)$ que contiene ese valor de r . Si imaginamos que una función continua es equivalente a una función discreta con infinitos rangos, la función acumulada $F(x)$ es una función continua en la que los escalones se reducen a puntos. Podemos entonces vincular los valores de la distribución uniforme 0-1 (r) y los valores de la variable aleatoria a simular (x) mediante la expresión:

$$r = F(x)$$

Por ejemplo, en el gráfico siguiente se puede observar que valores menores a r_1 se obtendrán en el generador de números uniformes 0-1 con una frecuencia r_1 . También se puede ver que la distribución de probabilidad de la variable a simular nos muestra que valores de la variable a simular menores a x_1 deberían obtenerse con una frecuencia $F(x_1)$.

Es fácil ver que lo anterior puede generalizarse haciendo $r_i = F(x_i)$. Es decir asociando los valores de la distribución aleatoria 0-1 con los valores de la acumulada de la función de distribución de probabilidad $f(x)$.



Método de la Transformada Inversa.

El razonamiento expuesto se sintetiza en el llamado Método de la Transformada Inversa. El método utiliza la distribución acumulada de la variable aleatoria $F(x)$, que está definida en el intervalo 0-1 para igualarlo con el valor de los números aleatorios uniformes 0-1 (r).

$$r = F(x)$$

Para generar el valor de x , debe despejarse y expresarse en función de r . Entonces:

$$x = F^{-1}(r)$$

El método requiere, por un lado que la distribución de probabilidad de la variable a simular sea integrable y por otro que pueda encontrarse la transformada inversa.

Ejemplo: Distribución exponencial

Aplicaremos a continuación el Método de la Transformada Inversa para generar números al azar que provengan de la distribución de probabilidad siguiente:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & : \text{si } x \geq 0 \\ 0 & : \text{si } x < 0 \end{cases}$$

La distribución acumulada es:

$$F(x) = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}$$

$$\text{Como } r = F(x) \rightarrow r = 1 - e^{-\lambda x} \rightarrow e^{-\lambda x} = (1 - r)$$

Dado que r sigue una distribución uniforme 0-1; $(1 - r)$ también, entonces:

$$e^{-\lambda x} = r \rightarrow x = -\frac{1}{\lambda} \ln r$$

Para completar el ejemplo, generamos 5 números aleatorios para una función exponencial de parámetro $\lambda = 2$

r	$x = -\frac{1}{2} \ln r$
0.55087	0.298
0.16822	0.891
0.45547	0.393
0.90286	0.051
0.21031	0.780

Ejemplo: Distribución uniforme

Mediante el Método de la Transformada inversa, desarrollaremos la expresión que nos permite generar números aleatorios uniformes en el intervalo (a, b) .

$$f(x) = \begin{cases} 0 & : \text{si } x < a \\ \frac{1}{(b-a)} & : \text{si } a \leq x \leq b \\ 0 & : \text{si } x > b \end{cases}$$

$$F(x) = \int_a^x \frac{1}{(b-a)} dx = \left. \frac{x}{(b-a)} \right|_a^x = \frac{x}{(b-a)} - \frac{a}{(b-a)} = \frac{(x-a)}{(b-a)}$$

$$r = F(x) = \frac{(x-a)}{(b-a)} \rightarrow x = a + r(b-a)$$

Generamos 5 números aleatorios para la distribución uniforme (2,6).

r	$x = a + r(b-a)$
0.13674	2.55
0.73707	4.95
0.19763	2.79
0.22501	2.90
0.36787	3.47

Ejemplo: Simular 5 valores de una variable aleatoria cuya función de distribución está dada por¹¹:

$$f(x) = \begin{cases} 0 & : \text{si } x < 1 \\ 3(x-1)^2 & : \text{si } 1 \leq x \leq 2 \\ 0 & : \text{si } x > 2 \end{cases}$$

¹¹ Tomado de: Fernando Salvador. *TP Simulación* (mimio). Centro de Estudiantes de Ingeniería "La Línea Recta". Buenos Aires 1992.

$$F(x) = \int_1^x 3(x-1)^2 dx = (x-1)^3 \Big|_1^x = (x-1)^3$$

$$r = (x-1)^3 \rightarrow x = 1 + \sqrt[3]{r}$$

r	$x = 1 + \sqrt[3]{r}$
0.25975	1.64
0.51805	1.80
0.84757	1.95
0.94508	1.98
0.28698	1.66

Ejemplo: Simular 5 valores de una variable aleatoria cuya distribución está dada por:

$$f(x) = \begin{cases} x & \text{si } 0 \leq x \leq 1 \\ 1/2 & \text{si } 1 < x \leq 2 \end{cases}$$

$$F(x) = \int_0^1 x dx = \frac{x^2}{2} \quad \text{si } 0 \leq x \leq 1$$

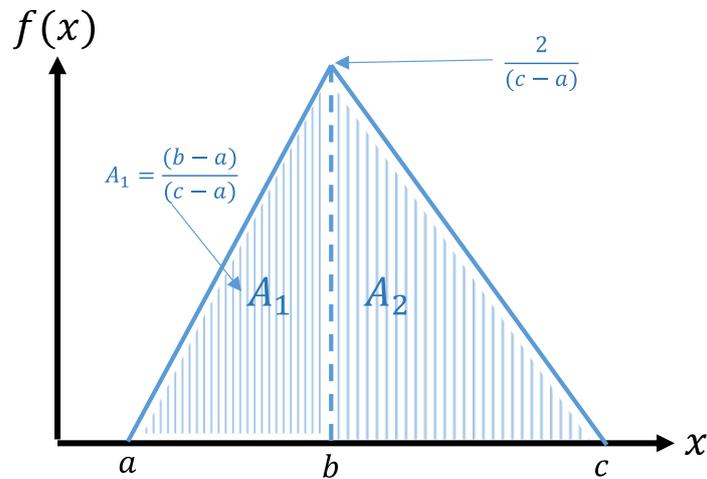
Dado que la función acumulada en $x = 1$ es $F(1) = \frac{1}{2}$ la función acumulada para $1 < x \leq 2$ es:

$$F(x) = \frac{1}{2} + \int_1^x \frac{1}{2} dx = \frac{1}{2} + \frac{1}{2}x \Big|_1^x = \frac{1}{2} + \frac{x}{2} - \frac{1}{2} = \frac{x}{2} \quad \text{si } 1 < x \leq 2$$

$$x = \begin{cases} \sqrt{2r} & \text{si } r \leq 1/2 \\ 2r & \text{si } r > 1/2 \end{cases}$$

r	$\sqrt{2r}$	$2r$	x
0.80783	-	1.61	1.61
0.41605	0.91	-	0.91
0.49807	0.99	-	0.99
0.35482	0.84	-	0.84
0.64382	-	1.29	1.29

Ejemplo: Desarrollar la expresión que permita simular valores de una variable aleatoria triangular cuyo valor mínimo es $x = a$, su moda es $x = b$ y el valor máximo es $x = c$



$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} : \text{si } a \leq x \leq b \\ \frac{-2(x-c)}{(c-b)(c-a)} : \text{si } b < x \leq c \end{cases}$$

Para $a \leq x \leq b$

$$F(x) = \int_a^x \frac{2(x-a)}{(b-a)(c-a)} dx = \frac{2}{(b-a)(c-a)} \int_a^x (x-a) dx = \frac{2}{(b-a)(c-a)} \left[\frac{(x-a)^2}{2} \right]_a^x$$

$$= \frac{(x-a)^2}{(b-a)(c-a)} - \frac{(a-a)^2}{(b-a)(c-a)} = \frac{(x-a)^2}{(b-a)(c-a)}$$

$$r = \frac{(x-a)^2}{(b-a)(c-a)} \rightarrow r(b-a)(c-a) = (x-a)^2 \rightarrow x = a + \sqrt{r(b-a)(c-a)}$$

Considerando que la función acumulada en $x = b$ es $F(b) = \frac{(b-a)}{(c-a)}$; para valores de $b \leq x \leq c$ la función acumulada es:

$$F(x) = \frac{(b-a)}{(c-a)} + \int_b^x \frac{-2(x-c)}{(c-b)(c-a)} dx = \frac{-2}{(c-b)(c-a)} \int_b^x (x-c) dx$$

$$F(x) = \frac{(b-a)}{(c-a)} + \frac{-2}{(c-b)(c-a)} \left[\frac{(x-c)^2}{2} \right]_b^x = \frac{(b-a)}{(c-a)} + \frac{-(x-c)^2}{(c-b)(c-a)} + \frac{(b-c)^2}{(c-b)(c-a)}$$

$$= \frac{(b-a)}{(c-a)} + \frac{(b-c)^2}{(c-b)(c-a)} - \frac{(x-c)^2}{(c-b)(c-a)}$$

$$F(x) = r = 1 - \frac{(x-c)^2}{(c-b)(c-a)}$$

$$r = 1 - \frac{(x-c)^2}{(c-b)(c-a)} \rightarrow x = c - \sqrt{(1-r)(c-b)(c-a)}$$

$$x = \begin{cases} a + \sqrt{r(b-a)(c-a)} & : \text{si } r \leq \frac{(b-a)}{(c-a)} \\ c - \sqrt{(1-r)(c-b)(c-a)} & : \text{si } r > \frac{(b-a)}{(c-a)} \end{cases}$$

Actividad 3

Genere manualmente 10 valores de una distribución aleatoria triangular de parámetros:

Valor mínimo: $a = 1$

Moda: $b = 3$

Valor máximo: $c = 4$

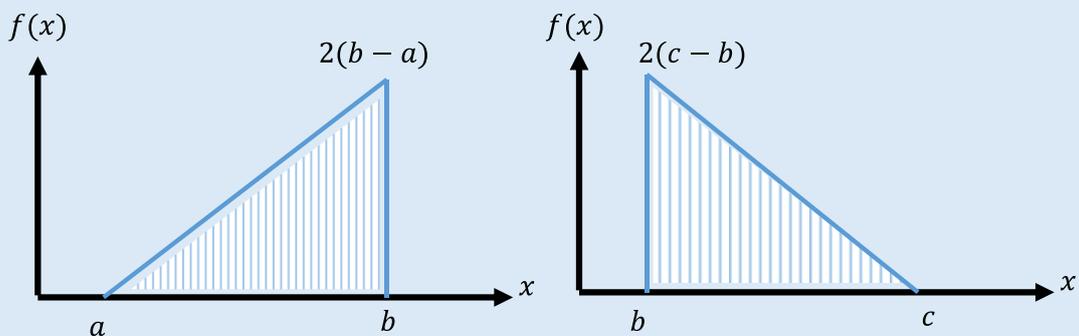
Actividad 4

Verifique que, para la distribución triangular rectangular creciente en (a,b) es:

$$x = a + (b-a)\sqrt{r};$$

y que para la distribución triangular rectangular decreciente en (b,c) es:

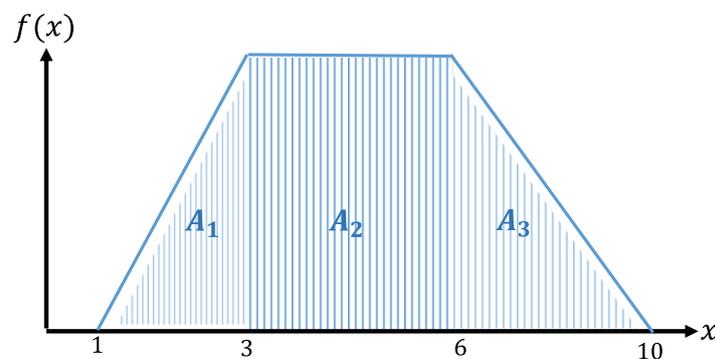
$$x = c - (c-b)\sqrt{r}$$



Método de composición

El método de composición es utilizado para generar variables aleatorias no uniformes. La distribución de probabilidad se divide en varias distribuciones de probabilidad de modo de simplificar el cálculo. El primer paso consiste en dividir la distribución original de probabilidad en sub-distribuciones. Luego se calcula el área de cada sub-distribución respecto del área de la distribución original; es decir la frecuencia de aparición de la sub-distribución en la distribución original. Se generan dos números aleatorios uniformes. Con el primero se selecciona la sub-distribución de acuerdo con la frecuencia de aparición de la sub-distribución. El segundo número aleatorio se utiliza para aplicar el método de la transformada inversa en la sub-distribución seleccionada.

Ejemplo: Aplicando el método de composición, simular 5 valores de la distribución de probabilidad siguiente:



Llamamos h al valor máximo de $f(x)$. Se tiene que cumplir que:

$$(3 - 1) \cdot h/2 + (6 - 3)h + (10 - 6) \cdot h/2 = 1$$

$$h + 3h + 2h = 1$$

$$h = \frac{1}{6}$$

Dividimos la distribución en tres sub-distribuciones; dos triangulares y una rectangular. Las áreas de cada una de ellas son:

$$A_1 = (3 - 1)(1/6)/2 = 1/6$$

$$A_2 = (6 - 3)(1/6) = 1/2$$

$$A_3 = (10 - 6)(1/6)/2 = 1/3$$

Se verifica que $A_1 + A_2 + A_3 = 1$

Con el primer número aleatorio r_1 seleccionamos la sub-distribución de acuerdo con el siguiente criterio:

si $0 \leq r_1 \leq 1/6 \rightarrow$ sub – distribución 1

si $1/6 < r_1 \leq 2/3 \rightarrow$ sub – distribución 2

si $2/3 < r_1 \leq 1 \rightarrow$ sub – distribución 3

El segundo número aleatorio se utiliza en la distribución seleccionada para generar el número aleatorio buscado aplicando el método de la transformada inversa.

- Si la seleccionada por r_1 es la sub-distribución 1 entonces¹²: $x = 1 + (3 - 1) \sqrt{r_2}$
- Si la seleccionada por r_1 es la sub-distribución 2 entonces: $x = 3 + r_2 (6 - 3)$
- Si la seleccionada por r_1 es la sub-distribución 3 entonces: $x = 10 - (10 - 6) \sqrt{r_2}$

r_1	r_2	$x = 1 + 2 \sqrt{r_2}$	$x = 3 + 3 r_2$	$x = 10 - 4 \sqrt{r_2}$	x
0.807	0.416	--	--	7.420	7.420
0.498	0.355	--	4.065	--	4.065
0.644	0.339	--	4.017	--	4.017
0.344	0.832	--	5.496	--	5.496
0.105	0.374	2.223	--	--	2.223

Actividad 5

Utilice el método de composición para obtener 10 valores de una distribución aleatoria triangular de parámetros:

Valor mínimo: $a = 1$

Moda: $b = 3$

Valor máximo: $c = 4$

¹² Ver Actividad 4

Generación de números aleatorios normales

El método de la transformada inversa no es aplicable a la distribución normal ya que no es posible expresar la distribución acumulada en forma explícita. Se han desarrollado varios métodos para generar números aleatorios normales: integración aproximada, tablas de números normales, tablas de la distribución normal acumulada, entre otros. Mencionaremos a continuación dos métodos muy difundidos.

Aplicación del Teorema del límite central:

El teorema del límite central establece que: ***Dada una sucesión de variables aleatorias X_1, X_2, \dots, X_n independientes entre sí, todas con la misma distribución de media μ y desvío estándar σ y la suma $Y_n = X_1 + X_2 + \dots + X_n$; entonces la variable $Z_n = \frac{Y_n - n\mu}{\sigma\sqrt{n}}$ tiende asintóticamente a una Normal estandarizada cuando n tiende a infinito.***¹³

Los números aleatorios rectangulares (0,1) tienen por media 1/2 y varianza 1/12.

Si sumamos n variables aleatorias rectangulares (0,1) tenemos: $Y = \sum_1^n r_i$

La media es: $\bar{Y} = \sum_1^n \frac{1}{2} = \frac{n}{2}$

La varianza es: $\sigma_Y^2 = \sum_1^n \frac{1}{12} = \frac{n}{12}$

Habitualmente se emplean $n = 12$ números aleatorios rectangulares (0,1) lo que nos da una aproximación razonable y simplifica el cálculo. De este modo $\bar{Y} = 6$ y $\sigma_Y^2 = 1$.

Haciendo: $r_n = \sum_1^6 r_i - 6$ obtenemos los números aleatorios normales buscados $N(0,1)$. Este procedimiento requiere que por cada número aleatorio normal, se deben generar 12 números aleatorios uniformes (0,1). Mediante la transformación $r_n\sigma + \mu$ pueden generarse números de cualquier distribución normal $N(\mu, \sigma)$

Procedimiento de Box y Muller¹⁴

Una forma más eficiente de generar números normales aleatorios es la propuesta por Box y Muller. Siendo r_1 y r_2 dos números aleatorios uniformes (0,1) se puede demostrar que las variables x_1 y x_2 son variables aleatorias normales independientes de la misma distribución con media 0 y varianza 1.

$$x_1 = \sqrt{-2 \ln r_1} \cos 2\pi r_2$$

$$x_2 = \sqrt{-2 \ln r_1} \sin 2\pi r_2$$

¹³ Roberto Mariano García. *Inferencia estadística y diseño de experimentos*. 1ª ed. Buenos Aires, Eudeba 2004; (página 40).

¹⁴ G.E.P. Box and Mervin Muller. *A note on the generation of random normal deviates*. The Annals of Mathematical Statistics (1958). 29 (2): 610–611.

EJEMPLO #1: ESTIMACIÓN DEL PESO DE CILINDROS

Utilizaremos la técnica de Simulación de Monte Carlo para estimar el peso de los cilindros producidos por una máquina. Por la experiencia de operación de la máquina se sabe que:

El radio de los cilindros que produce la máquina sigue una distribución triangular de parámetros:

- Valor mínimo: $a = 1.95 \text{ cm}$
- Moda: $b = 2.00 \text{ cm}$
- Valor máximo: $c = 2.01 \text{ cm}$

La longitud de los cilindros que produce la máquina sigue una distribución uniforme de parámetros:

- Valor mínimo: $a = 9.95 \text{ cm}$
- Valor máximo: $b = 10.05 \text{ cm}$

El proveedor actual del material ha sido discontinuado y se estudia contratar con un nuevo proveedor del material para la elaboración de los cilindros. En la fábrica del proveedor se han realizado muestreos sobre el material que indican que el peso específico que entregará tiene una distribución de probabilidad que puede asemejarse a una distribución triangular con los siguientes parámetros:

- Valor mínimo: $a = 7.99 \text{ g/cm}^3$
- Moda: $b = 8.00 \text{ g/cm}^3$
- Valor máximo: $c = 8.01 \text{ g/cm}^3$

Antes de contratar al nuevo proveedor, se desea estimar:

- El peso que presentarán los cilindros producidos con el material de este nuevo proveedor.
- El porcentaje de cilindros que se descartarán, sabiendo que no se admiten cilindros de más de 1000 gramos o de menos de 980 gramos.

Resolución:

1. Fijar la cantidad n de iteraciones a realizar.
2. Generar n valores aleatorios para el radio del cilindro de acuerdo con la distribución triangular adoptada para el radio.
3. Generar n valores aleatorios para la longitud del cilindro de acuerdo con la distribución uniforme adoptada para la longitud.
4. Generar n valores aleatorios para el peso específico de acuerdo con la distribución triangular que estima el nuevo proveedor del material.
5. En función de 2, 3, y 4 calcular n valores para el peso de los cilindros.
6. Calcular la media de la muestra.
7. Calcular el desvío estándar de la muestra.
8. Construir un intervalo de confianza para la media.

9. Contar las veces que el peso superó los 1000 gramos o fue inferior a 980 gramos.

Los resultados obtenidos para N=1000 iteraciones fueron:

- Media muestral $\bar{x} = 991.3$ gramos
- Desvío muestral $S = 13.72$ gramos

Intervalo de Confianza:

Para un nivel de confianza del 95% ($\alpha = 5\%$):

Valor de $t_{999, 1-\frac{\alpha}{2}} = t_{999, 0.975} = 1.96$

- Límite superior:

$$\bar{x} + t_{999, 0.975} \frac{S}{\sqrt{n}} = 991.3 + 1.96 \frac{13.72}{\sqrt{1000}} = 992.16$$

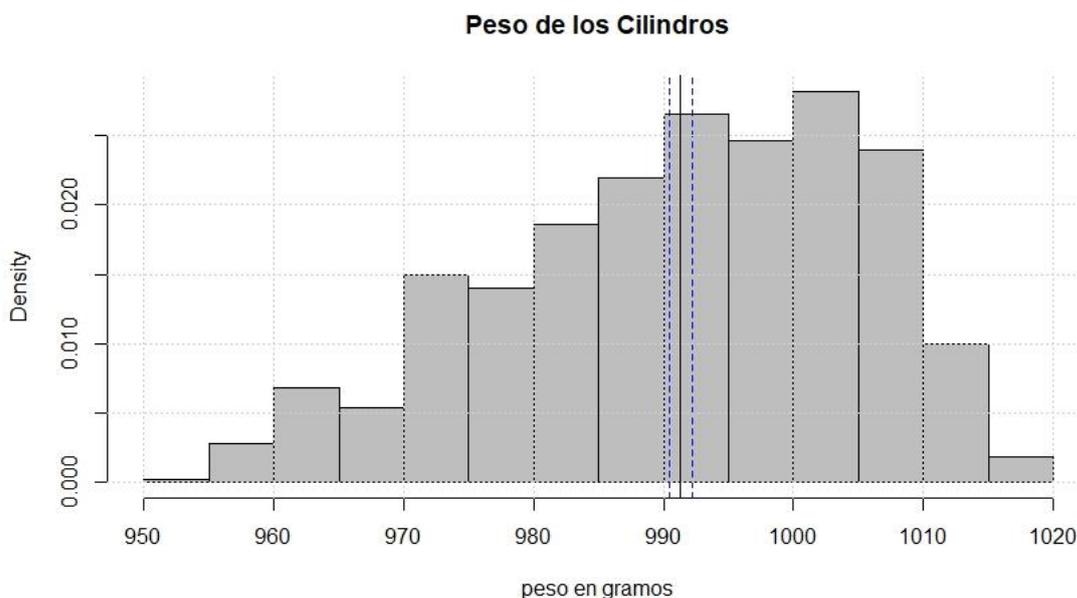
- Límite inferior:

$$\bar{x} - t_{999, 0.975} \frac{S}{\sqrt{n}} = 991.3 - 1.96 \frac{13.72}{\sqrt{1000}} = 990.46$$

Con un nivel de confianza del 95% puede afirmarse que el valor medio del peso de los cilindros será: $990.46 \leq \mu \leq 992.16$

Podemos rastrear en los 1000 resultados obtenidos la cantidad de veces que el peso simulado superó los 1000 gramos o no alcanzó los 980 gramos. Encontramos así que el 54.1% de los cilindros serán descartados por no cumplir esta especificación.

La línea vertical en el histograma indica la media y las líneas punteadas los límites inferior y superior del intervalo de confianza para la media con un nivel de significación del 95%.



INTERVALO DE CONFIANZA Y NÚMERO DE ITERACIONES

Las estimaciones de parámetros desconocidos nunca pueden ser exactas. En este sentido en la simulación de Monte Carlo ocurre lo mismo que con la experimentación sobre un sistema físico. Lo que podemos calcular son rangos dentro de los cuales y con una cierta probabilidad de certeza se encuentre el parámetro real. Si θ es un parámetro que deseamos estimar, lo que podremos calcular son los límites A y B tales que: $P(A < \theta < B) = (1 - \alpha)$ ¹⁵

La probabilidad $(1 - \alpha)$ se llama *nivel de confianza* (es una probabilidad elevada). La probabilidad α es pequeña y se llama *nivel de riesgo*. En el Ejemplo #1 empleamos una probabilidad del 95% como nivel de confianza y una probabilidad del 5% como nivel de riesgo.

La confiabilidad de los resultados de un modelo de simulación de Monte Carlo depende de la cantidad de iteraciones o longitud de la corrida del modelo. En el Ejemplo #1 hemos fijado arbitrariamente el tamaño de la corrida en $n=1000$. Con esta cantidad de iteraciones obtuvimos una estimación del valor medio del peso de los cilindros que nos resultó satisfactoria, porque el ancho del intervalo de confianza nos pareció adecuado.

Si el modelo se encuentra en estado estable y los resultados de las variables son independientes, podemos (como se hizo en el Ejemplo #1) aplicar los procedimientos clásicos de intervalos de confianza. El intervalo de confianza será tanto más estrecho cuanto mayor sea la cantidad de iteraciones.

Intervalo de confianza para la media

Sobre una muestra de n observaciones X_1, X_2, \dots, X_n (tomadas de una población de media μ y desvío σ desconocidos) podemos calcular los indicadores:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

$$S^2 = \frac{1}{(n-1)} \sum_{i=1}^n (X_i - \bar{X})^2$$

El estadístico \bar{X} tiene una media igual a la media de la población ($E(\bar{X}) = \mu$), y un desvío $D(\bar{X}) = \sigma/\sqrt{n}$. Aunque la variable poblacional no sea de distribución normal, igualmente podemos esperar que \bar{X} tenga una distribución normal, o que aproximadamente lo sea si el tamaño de la muestra es suficientemente grande ($(\sigma/\sqrt{n}) < 0.2$). Entonces es posible estandarizar la variable \bar{X} y llevarla a una distribución normal estándar:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} = \frac{(\bar{X} - \mu)\sqrt{n}}{\sigma}$$

$$P \left\{ Z_{\alpha/2} < \frac{(\bar{X} - \mu)\sqrt{n}}{\sigma} < Z_{1-\alpha/2} \right\} = (1 - \alpha)$$

¹⁵ Para un estudio detallado de intervalos de confianza ver: Roberto Mariano García. *Inferencia estadística y diseño de experimentos*. 1ª ed. Buenos Aires, Eudeba 2004; (páginas 50-62).

De aquí podemos establecer:

$$P \left\{ \bar{X} - Z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + Z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right\} = (1 - \alpha)$$

Cuando desconocemos σ (que es lo que habitualmente ocurre), reemplazamos Z por

$$t = \frac{(\bar{X} - \mu)\sqrt{n}}{S}$$

Este estadístico tiene una distribución t de *Student* con $v = (n - 1)$ grados de libertad. El intervalo de confianza se determina como:

$$P \left\{ \bar{X} - t_{v, 1-\alpha/2} \frac{S}{\sqrt{n}} < \mu < \bar{X} + t_{v, 1-\alpha/2} \frac{S}{\sqrt{n}} \right\} = (1 - \alpha)$$

El tamaño de la muestra y la precisión alcanzada quedan entonces estrechamente vinculados y no es posible determinar a priori un tamaño de muestra para alcanzar la precisión deseada. Lo que habitualmente se hace es correr una muestra piloto, obtener los estadísticos \bar{X} y S y con ellos un intervalo de confianza. Si el intervalo de confianza es inadecuado, puede aproximarse la cantidad n de iteraciones necesarias y extender la muestra. Luego el proceso se repite hasta alcanzar el intervalo de confianza que nos brinde la precisión deseada.

Para valores de n grandes la distribución t de *Student* converge a la normal y el valor $t_{v, 1-\alpha/2}$ coincide con $Z_{1-\alpha/2}$. Por ejemplo para $n = 1000$ y $\alpha = 0.05$ es $t_{v, 1-\alpha/2} = 1.96$ y $Z_{1-\alpha/2} = 1.96$.

En el Ejemplo #1 elegimos arbitrariamente $n=1000$ y obtuvimos un ancho de intervalo de confianza de 1,70 gramos. La tabla siguiente muestra el ancho del intervalo de confianza para diferentes valores n :

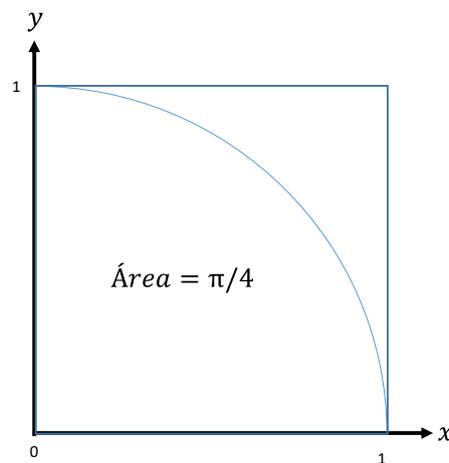
Cantidad de iteraciones	Ancho del Intervalo de Confianza
100	5.11 gramos
500	2.39 gramos
1000	1.70 gramos
5000	0.75 gramos
10000	0.53 gramos
100000	0.17 gramos

EJEMPLO #2: ESTIMACIÓN DE π – INTERVALOS DE CONFIANZA PARA VARIABLES BINOMIALES

Utilizaremos este ejemplo, para exponer el cálculo de intervalos de confianza para variables de tipo Bernoulli; es decir variables cuyos resultados sean de la forma: éxito / fracaso.

Podemos estimar el valor del área de un cuarto de círculo de radio unitario, mediante un proceso que seleccione aleatoriamente pares de valores (x, y) con x e y en el intervalo $(0,1)$. Para cada par de valores (x, y) determinamos si cae dentro o fuera del área del círculo. La proporción de puntos dentro del área del círculo nos permite estimar el valor de π de la siguiente forma:

$$\frac{\text{Área de 1/4 de círculo}}{\text{Área del cuadrado}} = \frac{\pi \cdot r^2 / 4}{r \cdot r} = \frac{\pi}{4}$$



Llamando p a la proporción de pares de valores (x, y) que caen dentro del área del círculo:

$$p = \frac{\pi}{4}$$

El procedimiento entonces consiste en generar un valor aleatorio de x uniformemente distribuido en el intervalo $(0,1)$; generar un valor de y que también distribuya uniformemente en el intervalo $(0,1)$. Calcular el radio $R = x^2 + y^2$ y verificar si es menor o igual que 1. El proceso se repite una cantidad grande de veces (n). El número de veces que $R \leq 1$ dividido por la cantidad total de iteraciones es una aproximación a p .

Resultados:

- $n = 1000$ iteraciones
- $\bar{p} = 0.806$
- $(1 - \bar{p}) = 0.194$

Para tamaños de muestras grandes (y este es generalmente el caso en los modelos de simulación), podemos estimar el intervalo de confianza para una distribución binomial de parámetro p como:

$$P \left\{ \bar{p} - Z_{1-\alpha/2} \sqrt{\frac{\bar{p}(1-\bar{p})}{n-1}} < p < \bar{p} + Z_{1-\alpha/2} \sqrt{\frac{\bar{p}(1-\bar{p})}{n-1}} \right\} = (1-\alpha)$$

Para un nivel de confianza del 95% obtenemos de tablas el valor $Z_{1-\alpha/2} = 1.96$

El extremo superior del intervalo de confianza para p es:

$$p^+ = \bar{p} + Z_{1-\alpha/2} \sqrt{\frac{\bar{p}(1-\bar{p})}{n-1}} = 0.806 + 1.96 \sqrt{\frac{0.806 \cdot 0.194}{999}} = 0.831$$

El extremo inferior del intervalo de confianza para p es:

$$p^- = \bar{p} - Z_{1-\alpha/2} \sqrt{\frac{\bar{p}(1-\bar{p})}{n-1}} = 0.806 - 1.96 \sqrt{\frac{0.806 \cdot 0.194}{999}} = 0.781$$

El valor superior del intervalo de confianza para π será: $4 p^+ = 4 \cdot 0.831 = 3.322$

El valor inferior del intervalo de confianza para π será: $4 p^- = 4 \cdot 0.781 = 3.126$

El intervalo de confianza calculado contiene el valor verdadero de π . El valor central del intervalo $4 \cdot \bar{p} = 4 \cdot 0.806 = 3.224$ tiene un error del 2.6% respecto del valor verdadero de π .

Actividad 6

Utilizando el software de su preferencia elabore un modelo de simulación de Monte Carlo que le permita estimar el área bajo la curva $f(x) = \text{sen}(x)$ en el intervalo $(0, \pi)$.

Expresé el resultado mediante un intervalo de confianza con un nivel de riesgo $\alpha = 0.05$

EJEMPLO #3: COMPARACIÓN DE ALTERNATIVAS DE FINANCIACIÓN

Se desean comparar dos alternativas para la financiación de un préstamo a 15 años de plazo. En ambas alternativas el capital vence al final del año 15. Las alternativas difieren en la tasa de interés:

- Alternativa tasa fija: 6% anual.
- Alternativa tasa variable: la tasa de interés se fija anualmente en función de otras tasas de referencia del mercado. Al inicio del primer año, la tasa variable es 8% anual. Se conocen las probabilidades de cambio de la tasa de un año al siguiente que se resumen en la siguiente tabla:

Variación en la tasa	+2%	+1.5%	+1.0%	+0.5%	+0%	-0.5%	-1.0%	-1.5%	-2.0%
Probabilidad	0.05	0.05	0.15	0.15	0.20	0.15	0.15	0.05	0.05
Probabilidad acumulada	0.05	0.10	0.25	0.40	0.60	0.75	0.90	0.95	1.00

Para comparar las alternativas se empleará como indicador el valor presente calculado con una tasa de descuento del 10% anual. Esta tasa se asume constante para los 15 años de duración del préstamo. La hipótesis de tasa de descuento constante podría flexibilizarse para adaptar el modelo a otro entorno. El pago de los intereses ocurre al final de cada año.

Resolución:

El proceso de resolución puede resumirse de la siguiente manera:

1. Generar $k = 15$ números aleatorios uniformes.
2. Aplicar el método de la transformada inversa para calcular (con los números aleatorios generados en el paso 1) la variación correspondiente en la tasa de interés para cada uno de los 15 años.
3. Con las variaciones calculadas en el punto 2, proyectar la tasa de interés a lo largo de los 15 años del préstamo. Considerar que en el momento inicial la tasa de interés es 8% anual.
4. Calcular el valor presente de los intereses (se obtiene así un resultado del valor presente para la alternativa de tasa variable).
5. Repetir los pasos 1 a 4 una cantidad n de veces.
6. Calcular el valor presente de los intereses a tasa fija.
7. Calcular el porcentaje de las veces que el valor presente a tasa variable es menor que el valor presente a tasa fija.

La tabla siguiente muestra los pasos 1 a 3 para una iteración.

AÑO	r	Δ tasa	tasa
1	0.0900	+1.5	9.5%
2	0.1182	+1.0	10.5%
3	0.0007	+2.0	12.5%
4	0.6912	-0.5	12.0%
5	0.3811	+0.5	12.5%
6	0.4676	+0.0	12.5%
7	0.1097	+1.0	13.5%
8	0.8824	-1.0	12.5%
9	0.0627	+1.5	14.0%
10	0.7223	-0.5	13.5%
11	0.9931	-2.0	11.5%
12	0.9840	-2.0	9.5%
13	0.9141	-1.5	8.0%
14	0.0007	+2.0	10.0%
15	0.0344	+2.0	12.0%

La tasa de descuento para calcular el valor presente es 10%; con esta tasa calculamos el valor presente para la alternativa de tasa variable (paso 4).

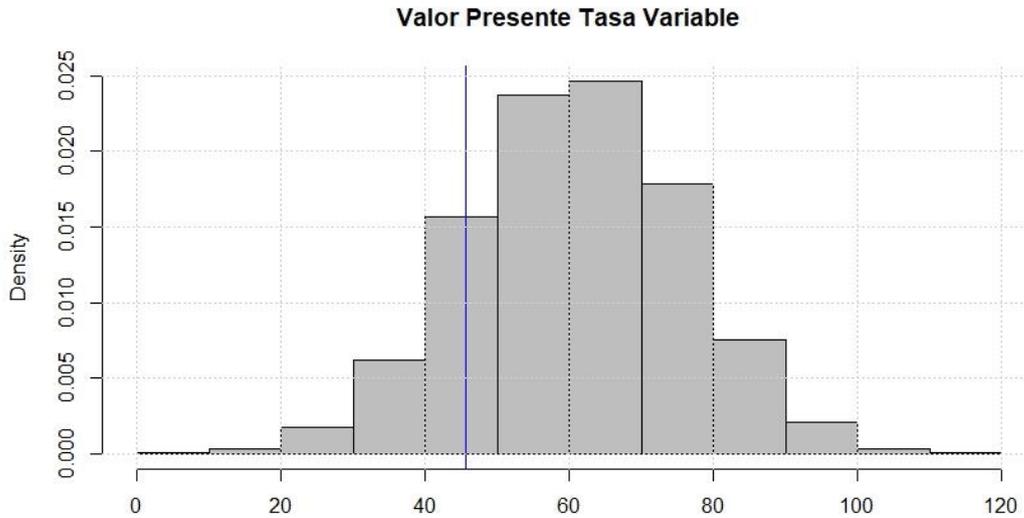
$$VP \text{ tasa variable} = \sum_{k=1}^{15} \frac{\text{tasa proyectada}}{(1+i)^k} = \frac{9.5}{(1.1)} + \frac{10.5}{(1.1)^2} + \frac{12.5}{(1.1)^3} + \dots + \frac{12.0}{(1.1)^{15}} = 88.5$$

Obtenemos así el indicador para una iteración. Esto debe repetirse una cantidad importante de veces (n) de modo de poder estudiar la distribución de probabilidad de la variable valor presente para la alternativa de tasa variable.

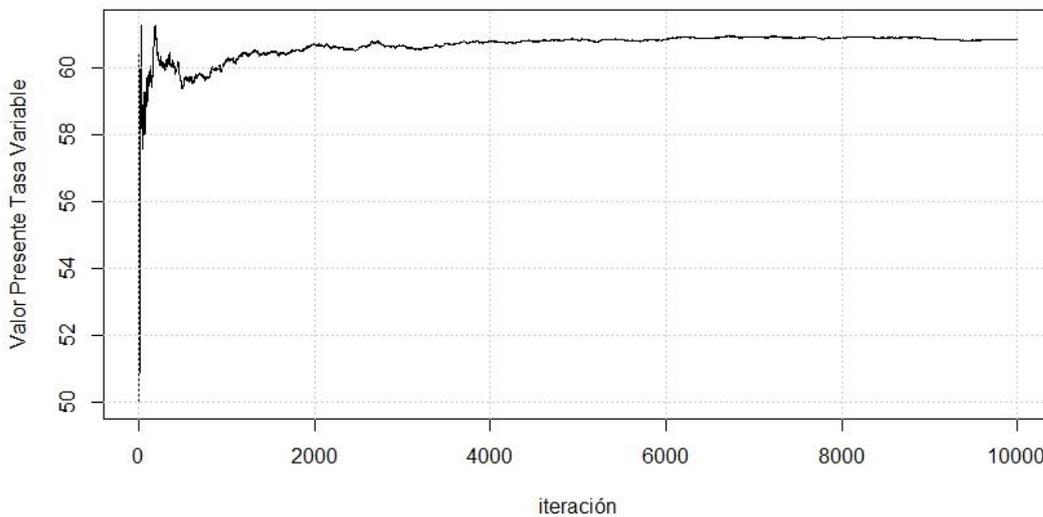
El valor presente para la alternativa de tasa fija debe calcularse una sola vez y es:

$$VP \text{ tasa fija} = \frac{1 - (1+i)^{-k}}{i} \text{ tasa fija} = \frac{1 - (1 + 0.1)^{-15}}{0.1} 6 = 45.6$$

Realizadas $n = 10000$ iteraciones, el valor medio de la variable valor presente para la alternativa de tasa variable es 60.8 que es mayor que el valor presente para la alternativa de tasa fija. En este caso es más útil, como indicador de riesgo, determinar la probabilidad de que la alternativa de tasa variable sea de menor costo que la alternativa de tasa fija. Este indicador se puede obtener computando las veces que el valor presente para la alternativa de tasa variable tiene un costo menor a 45.6. En este caso la probabilidad resulta 16%. Es decir que existe una probabilidad del 16% de que la alternativa de tasa variable tenga un costo inferior al de la alternativa de tasa fija. El gráfico siguiente muestra el resultado para $n = 10000$ iteraciones. La línea vertical azul indica el valor presente para la alternativa de tasa fija.



Normalmente, en las primeras iteraciones el proceso de simulación de Monte Carlo produce resultados muy variables. Acumulada una cantidad suficiente de iteraciones se estabiliza. El gráfico siguiente muestra en el eje horizontal la cantidad de iteraciones y en el eje vertical el valor presente (para la alternativa de tasa variable) promedio acumulado hasta la iteración indicada. El comportamiento variable inicial y la estabilización posterior son típicos en estos modelos.



Al período inicial se lo suele denominar período transitorio o de calentamiento. No existe una forma de anticipar cuántas iteraciones serán necesarias para alcanzar el estado estable. El gráfico muestra que cuanto más larga es la corrida, más probabilidades existen de alcanzar el estado estable; y que el resultado será más preciso cuantas más iteraciones formen la corrida del modelo.

Actividad 7

Para el Ejemplo #3 construya un intervalo de confianza para la probabilidad de que la alternativa de tasa variable resulte de menor costo que la alternativa de tasa fija:

$$P\{VP \text{ tasa variable} < VP \text{ tasa fija}\} = p$$

Dados:

$$\bar{p} = 0.1603$$

Nivel de riesgo $\alpha = 0.05$

$n = 10000$ iteraciones

Actividad 8

En el Ejemplo #3, de $n=10000$ iteraciones, 5170 dieron por resultado un valor presente (para la alternativa de tasa variable) entre 50 y 70; siendo:

$$P\{50 < VP \text{ tasa variable} < 70\} = q$$

Construya un intervalo de confianza para q , dados:

$$\bar{q} = 0.5170$$

Nivel de riesgo $\alpha = 0.05$

$n = 10000$ iteraciones

TIPOS DE MODELOS DE SIMULACIÓN Y MÉTODOS DE RESOLUCIÓN

En general los modelos de simulación pueden clasificarse en modelos *estáticos* y modelos *dinámicos*. En ambos casos el comportamiento es descrito por el estado del sistema y podemos pensar el estado del sistema como un vector \vec{X} que contiene la información de las variables relevantes del sistema.

Los modelos estáticos son aquellos en los que el estado del sistema no depende del tiempo. En los sistemas dinámicos el estado del sistema depende del tiempo de modo que el comportamiento del sistema es descrito por un proceso estocástico aleatorio (discreto o continuo) $\{\vec{X}_t\}$

Modelos de simulación estáticos

Los ejemplos #1 y #2 son modelos de este tipo. El proceso de simulación, en esta clase de modelos, consiste en la generación repetida del estado del sistema. En el ejemplo #1 el estado del sistema puede estar definido por el vector:

$$\vec{X} = (\text{radio}, \text{largo}, \text{peso específico})$$

La variable de interés en el caso planteado es el peso del cilindro es decir la función (*función de desempeño*):

$$H(\vec{X}) = \pi \cdot \text{radio}^2 \cdot \text{largo} \cdot \text{peso específico}$$

En términos generales los modelos estáticos se resuelven con el algoritmo siguiente:

1. Generar n réplicas del modelo $\vec{X}_1, \vec{X}_2, \dots, \dots, \vec{X}_n$ y calcular la función (o funciones) de desempeño relevante: $H(\vec{X}_i)$, $i = 1, 2, \dots, n$
2. Calcular los estimadores puntuales y sus intervalos de confianza. Por ejemplo:

$$\bar{l} = E[H(\vec{X})] = \frac{1}{n} \sum_{i=1}^n H(\vec{X}_i)$$

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (H(\vec{X}_i) - \bar{l})^2$$

$$P \left\{ \bar{l} - Z_{1-\alpha/2} \frac{S}{\sqrt{n}} \leq l \leq \bar{l} + Z_{1-\alpha/2} \frac{S}{\sqrt{n}} \right\} \approx (1 - \alpha)$$

Son ejemplos de modelos estáticos: los modelos de fallas de componentes, de cálculo de áreas, de propagación de errores, de camino crítico (PERT) con variables aleatorias, entre otros.

Modelos de simulación dinámicos

Los modelos de simulación *dinámicos* refieren a sistemas que evolucionan en el tiempo. El objetivo (igual que en los modelos estáticos) es estimar el desempeño del sistema. El estado del sistema está ahora descrito por un proceso estocástico $\{X_t\}$ que puede ser continuo o discreto; nosotros aquí nos centramos en los modelos discretos, y por simplicidad nos referiremos a un escalar estocástico (en lugar de un vector) X_t como variable de estado del sistema.

Una distinción importante dentro de los modelos dinámicos la hacemos entre los modelos dinámicos de *horizonte finito* y los modelos dinámicos de *estado estable*.

En los modelos dinámicos de *horizonte finito* el desempeño del sistema está definido en relación a un intervalo de tiempo $[0, T]$ (T puede ser aleatorio); mientras que en los modelos dinámicos de *estado estable* el desempeño es medido en el largo plazo, es decir cuando el horizonte temporal tiende a infinito.

Una diferencia importante entre ambos tipos de modelos es que en los modelos de simulación de *estado estable*, las medidas de desempeño del sistema alcanzadas cuando $t \rightarrow \infty$ no dependen del estado inicial del sistema. Por el contrario, en los modelos de simulación de *horizonte finito*, las medidas de desempeño del sistema dependen del estado inicial.

El ejemplo #3 es un modelo de simulación de horizonte finito. Se proyecta la variable aleatoria tasa de interés en un período de $T = 15$ años. El estado inicial del sistema (la tasa de interés en un nivel de 8% anual) condiciona la evolución posterior de la tasa de interés y por tanto el valor presente calculado.

En los modelos dinámicos de simulación discreta existen dos formas de generar el avance del tiempo de la simulación (el "reloj" del modelo). Una alternativa es incrementar el tiempo en *lapsos iguales*. Por ejemplo, un modelo de simulación de stock en el que se cuenta con la distribución de probabilidad de la demanda semanal y la simulación se realiza, por tanto, semana a semana. La segunda forma de avanzar el tiempo es mediante *eventos*. Por ejemplo un sistema de colas que avanza con la ocurrencia del evento definido como *llegada de un cliente*. En este caso el tiempo avanza a intervalos aleatorios dado por el régimen de arribos de los clientes.

Modelos de simulación dinámicos de horizonte finito:

El algoritmo de resolución para estimar el desempeño en un horizonte de tiempo finito, $l(T/m)$ (donde m define el estado inicial) es muy similar al de los modelos estáticos:

1. Generar n réplicas del proceso $\{X_t, t \leq T\}$ comenzando cada réplica en el estado $X_0 = m$.
2. Calcular los estimadores puntuales y sus intervalos de confianza de $l(T/m)$ del mismo modo que se hizo para los modelos estáticos.

Modelos de simulación dinámicos de estado estable:

Se refiere a sistemas que muestran alguna forma de comportamiento estacionario o de largo plazo. Podemos pensar que el sistema se ha iniciado en un pasado infinito y que las condiciones iniciales son ya irrelevantes.

La estimación de parámetros en los modelos de estado estable, debe realizarse con algunas precauciones porque los datos de salida están en general correlacionados. A diferencia de los modelos *estáticos* y los modelos dinámicos de *horizonte finito* en los que las observaciones derivadas de cada iteración del modelo son independientes entre sí, aquí las observaciones del modelo a lo largo del tiempo están correlacionadas. Por ejemplo, si estamos simulando un sistema de atención de 5 canales, y el sistema está en el estado vacío, un cliente que arriba no deberá esperar en cola, y tampoco esperarán los 4 clientes que le sigan. Entonces es esperable que a un cliente que no espera, le sigan otros que tampoco lo hagan.

Una forma de realizar la simulación de los sistemas dinámicos de estado estable es realizar M iteraciones, descartar las primeras K y considerar solamente $n = M - K$ para el cálculo de los indicadores. De este modo se descartan los resultados obtenidos en el período transitorio y se cancelan los efectos derivados del estado inicial. Sin embargo, no siempre es claro cuando termina el estado transitorio y comienza el estado estable del modelo.

Otra forma de obtener los parámetros del estado estacionario es el **método de las medias por lotes** (*Batch Means Method*). Se realiza una única corrida de longitud M y se descartan las primeras K iteraciones que se atribuyen al estado transitorio. Las restantes $M - K$ iteraciones se dividen en n lotes con $T = \frac{M-K}{n}$ iteraciones en cada uno de ellos. Si $X_{t,i}$ es la iteración t del lote i , la muestra del lote i es:

$$Y_i = \frac{1}{T} \sum_{t=1}^T X_{t,i} \quad i = 1, 2, \dots, n$$

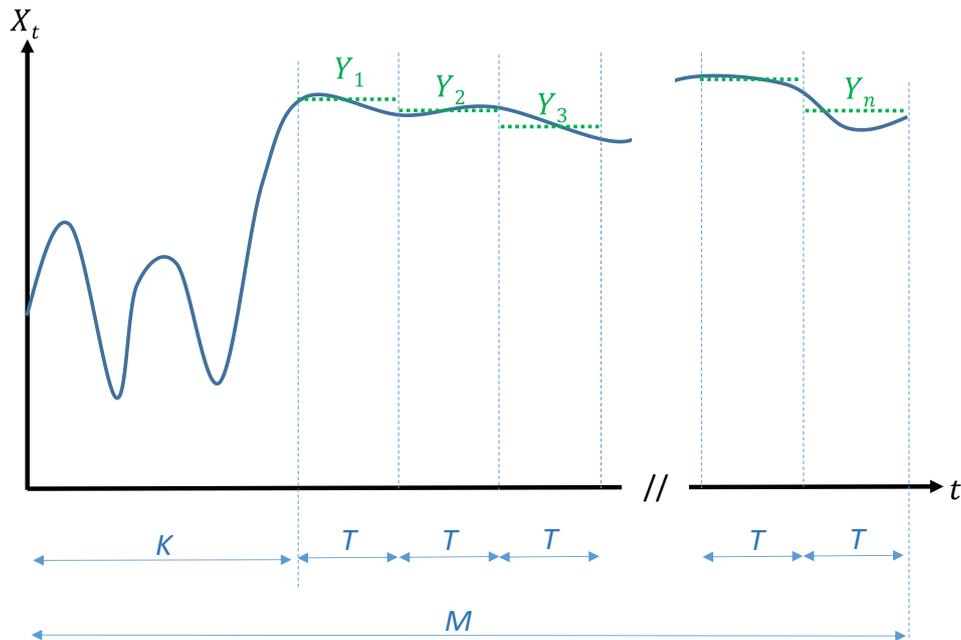
el estimador puntual de l es:

$$\bar{l} = \frac{1}{M - K} \sum_{t=K+1}^M X_t = \frac{1}{n} \sum_{i=1}^n Y_i$$

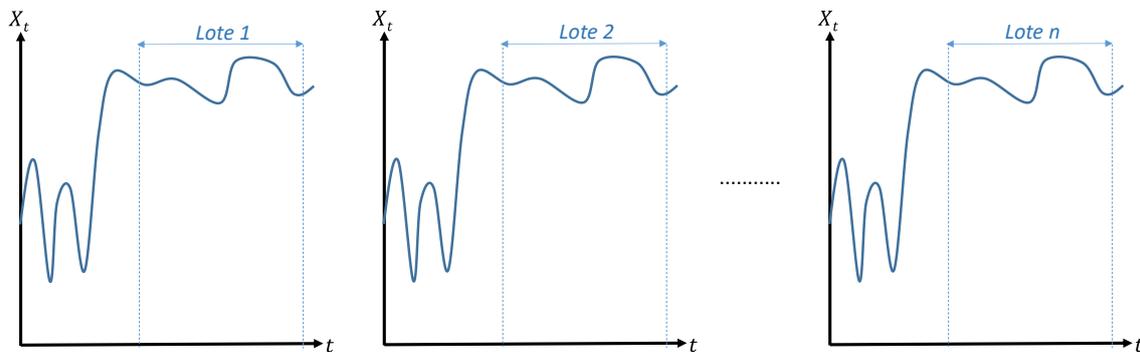
Para garantizar la independencia entre los lotes, el largo T debe ser de tamaño suficiente y la cantidad de lotes debe ser de entre 20 y 30 de modo de poder aplicar el teorema del límite central:

$$S^2 = \frac{1}{n - 1} \sum_{i=1}^n (Y_i - \bar{l})^2$$

$$P \left\{ \bar{l} - Z_{1-\alpha/2} \frac{S}{\sqrt{n}} \leq l \leq \bar{l} + Z_{1-\alpha/2} \frac{S}{\sqrt{n}} \right\} \approx (1 - \alpha)$$



Otra forma de estimar las variables de desempeño del sistema es el **método de replicación-eliminación** (*replication-deletion method*) que consiste en realizar n corridas (réplicas) independientes del modelo de simulación, en cada una de ellas se descartan las iteraciones del período transitorio y se calculan los indicadores puntuales y sus intervalos de confianza como en el método de las medias por lotes. El método de replicación-eliminación tiene un sesgo menor en las estimaciones que el método de las medias por lotes.



Algunos procesos aleatorios tienen la particularidad de reiniciarse en forma aleatoria. Es decir que a intervalos estocásticos de tiempo vuelven a un mismo estado y reinician desde él. Son llamados procesos estocásticos *regenerativos*. Por ejemplo, un sistema de inventarios que a intervalos de tiempo aleatorios agota las existencias, se regenera a partir de ese instante; o podemos pensar que un sistema de atención se reinicia cuando se vacía de clientes. El **método de simulación regenerativa** permite tratar en forma más precisa este tipo de procesos estocásticos ya que reduce la auto-correlación.

EJEMPLO #4: SIMULACIÓN REGENERATIVA

Supongamos un proceso de Markov¹⁶ de dos estados: 1 y 2. Asumimos que el proceso se inicia en el estado 1 y que conocemos la matriz de las probabilidades de transición:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

Admitamos que hemos obtenido mediante un proceso de simulación de Monte Carlo, la siguiente trayectoria del proceso:

$$(x_1, x_2, x_3, x_4 \dots \dots x_{10}) = (1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1).$$

El sistema se regenera cada vez que vuelve al estado inicial 1. La muestra, entonces, tiene 4 ciclos: **1, 2, 2, 2 / 1, 2 / 1 / 1, 2, 2** de longitudes $\tau_1 = 4$; $\tau_2 = 2$; $\tau_3 = 1$; $\tau_4 = 3$

Supongamos que cada transición tiene un costo dado por la siguiente matriz:

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$$

Hemos elegido el costo como parámetro a estimar pero podríamos haber elegido cualquier otro indicador del tipo $R_i = \sum H(X_t)$ (la sumatoria abarca todos los elementos del ciclo i). En nuestro ejemplo resultan:

$$R_1 = 1 + 3 + 3 + 2 = 9$$

$$R_2 = 1$$

$$R_3 = 0$$

$$R_4 = 1 + 3 + 2 = 6$$

En la tabla resumimos las longitudes y los costos de cada uno de los 4 ciclos:

Ciclo	τ_i	R_i
1	4	9
2	2	3
3	1	0
4	3	6
	$\bar{\tau} = 2.5$	$\bar{R} = 4.5$

Notemos que τ y R son variables aleatorias independientes.

El estimador del costo es: $\bar{l} = \frac{\bar{R}}{\bar{\tau}} = \frac{4.5}{2.5} = 1.8$

¹⁶ Tomado de: Reiven Y. Rubinstein, Dirk P. Kroese. *Simulation and the Monte Carlo Method*. 2nd edition. Wiley series in probability and statistics. 2008.

Se puede demostrar que el intervalo de confianza es:

$$\left(\bar{l} \pm \frac{Z_{1-\alpha/2} S}{\bar{\tau} \sqrt{n}} \right)$$

con:

$$S^2 = S_{11} - 2\bar{l} S_{12} + \bar{l}^2 S_{22}$$

$$S_{11} = \frac{1}{n-1} \sum_{i=1}^n (R_i - \bar{R})^2 \quad S_{12} = \frac{1}{n-1} \sum_{i=1}^n (R_i - \bar{R})(\tau_i - \bar{\tau}) \quad S_{22} = \frac{1}{n-1} \sum_{i=1}^n (\tau_i - \bar{\tau})^2$$

En el ejemplo resultan:

$$\left. \begin{array}{l} S_{11} = 15 \\ S_{12} = 5/3 \\ S_{22} = 5 \end{array} \right\} S^2 = 2.4$$

$$\left(\bar{l} \pm \frac{Z_{1-\alpha/2} S}{\bar{\tau} \sqrt{n}} \right) = \left(1.8 \pm \frac{1.96 \sqrt{2.4}}{2.5 \sqrt{4}} \right) \rightarrow \left\{ \begin{array}{l} \bar{l}^+ = 2.4 \\ \bar{l}^- = 1.2 \end{array} \right.$$

EJEMPLO #5: MODELO DE STOCK ALEATORIO

Se debe definir una política de stocks de un insumo crítico para un período de 30 días. La demanda es aleatoria; se la ha estimado como una distribución triangular de valor mínimo $a = 800$ unidades, moda $b = 1000$ unidades y valor máximo $c = 1500$ unidades. Si al inicio del día el stock es inferior a la demanda máxima posible ($c = 1500$ unidades) se realiza un pedido de Q unidades que ingresa en forma inmediata al inventario. El stock inicial al principio del primer día es de 4000 unidades. El costo de mantener una unidad en stock es $c_1 = 0.04 \text{ USD}/(\text{unidad. día})$ y el costo de realizar un pedido es $k = 320 \text{ USD}$. Se desea estimar el tamaño del lote de pedido (Q) que hace que el costo para el período de 30 días sea mínimo.

Si bien los modelos de simulación de Monte Carlo son descriptivos (no de optimización), podemos mediante un proceso de búsqueda encontrar el valor óptimo de un parámetro. En este caso el parámetro es el tamaño del lote de reposición Q . El valor óptimo lo estimaremos corriendo el modelo de simulación para distintos valores de Q .

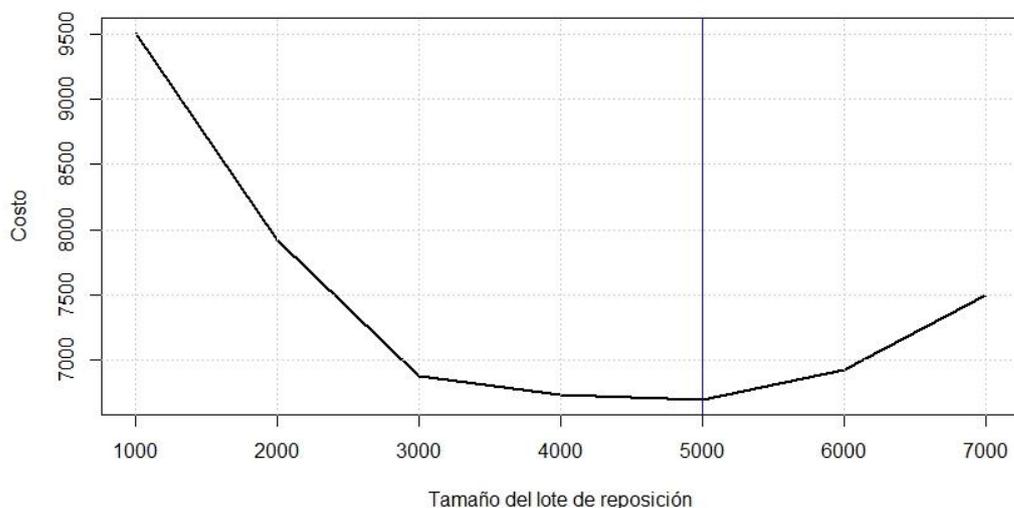
Trataremos el caso como un modelo dinámico de horizonte finito ($T = 30 \text{ días}$) con la siguiente secuencia de cálculo:

1. Fijar el tamaño del lote de reposición. Inicialmente $Q = 4000$.
2. Proyectar la demanda para los días 1 a 30 de acuerdo con la distribución de probabilidad estimada.
3. Proyectar el nivel de stock al inicio y al final de cada día de acuerdo con la política de stock definida.
4. Calcular el costo de almacenamiento de cada día como:
 $\text{Costo Almacenamiento} = \text{Stock inicial} \cdot 0.04 \text{ USD}/(\text{unidad. día})$
5. El costo de reposición de cada día será:
 320 USD si hay reposición ese día; 0 si no hay reposición.
6. Calcular el costo total diario y el costo total mensual.
7. Repetir los pasos 2 a 6 una cantidad n de veces (se adopta $n = 1000$).
8. Analizar los resultados obtenidos en las n iteraciones para el costo total mensual
9. Repetir los pasos 1 a 9 para diferentes valores de Q .

La tabla muestra una iteración (pasos 2 a 6 para $Q = 4000$). El costo de almacenamiento total de esta iteración es de 6785.1 USD.

Día	Stock inicial	Reposición	Demanda	Stock final	Costo reposición	Costo almacenaje	Costo total
1	4000.0	0.0	1103.4	2896.6	0.0	160.0	160.0
2	2896.6	0.0	909.8	1986.9	0.0	115.9	115.9
3	1986.9	0.0	966.4	1020.5	0.0	79.5	79.5
4	1020.5	4000.0	1010.2	4010.3	320.0	200.8	520.8
5	4010.3	0.0	986.7	3023.6	0.0	160.4	160.4
6	3023.6	0.0	1445.2	1578.4	0.0	120.9	120.9
7	1578.4	0.0	955.7	622.7	0.0	63.1	63.1
8	622.7	4000.0	901.5	3721.2	320.0	184.9	504.9
9	3721.2	0.0	1052.4	2668.9	0.0	148.8	148.8
10	2668.9	0.0	1018.1	1650.8	0.0	106.8	106.8
11	1650.8	0.0	1036.1	614.6	0.0	66.0	66.0
12	614.6	4000.0	1316.7	3297.9	320.0	184.6	504.6
13	3297.9	0.0	974.0	2324.0	0.0	131.9	131.9
14	2324.0	0.0	1246.4	1077.6	0.0	93.0	93.0
15	1077.6	4000.0	1031.1	4046.4	320.0	203.1	523.1
16	4046.4	0.0	963.7	3082.7	0.0	161.9	161.9
17	3082.7	0.0	1070.5	2012.2	0.0	123.3	123.3
18	2012.2	0.0	983.0	1029.2	0.0	80.5	80.5
19	1029.2	4000.0	1142.9	3886.3	320.0	201.2	521.2
20	3886.3	0.0	944.7	2941.6	0.0	155.5	155.5
21	2941.6	0.0	975.1	1966.5	0.0	117.7	117.7
22	1966.5	0.0	1143.3	823.3	0.0	78.7	78.7
23	823.3	4000.0	1101.9	3721.4	320.0	192.9	512.9
24	3721.4	0.0	957.4	2764.0	0.0	148.9	148.9
25	2764.0	0.0	1389.0	1375.0	0.0	110.6	110.6
26	1375.0	4000.0	932.0	4442.9	320.0	215.0	535.0
27	4442.9	0.0	944.8	3498.2	0.0	177.7	177.7
28	3498.2	0.0	1166.2	2332.0	0.0	139.9	139.9
29	2332.0	0.0	1119.4	1212.6	0.0	93.3	93.3
30	1212.6	4000.0	965.5	4247.1	320.0	208.5	528.5
TOTAL	4000.0	32000.0	31752.9	4247.1	2560.0	4225.1	6785.1

El paso 9 se ejecuta para cubrir valores del lote de reposición entre $1000 \leq Q \leq 7000$ que permite estimar el costo mínimo de operación para un lote de reposición $Q = 5000$ unidades. Los resultados del paso 9 se sintetizan en el gráfico siguiente:



Los resultados obtenidos para $Q = 5000$ unidades son:

- Costo total = 6700 USD
- Desvío estándar = 214 USD

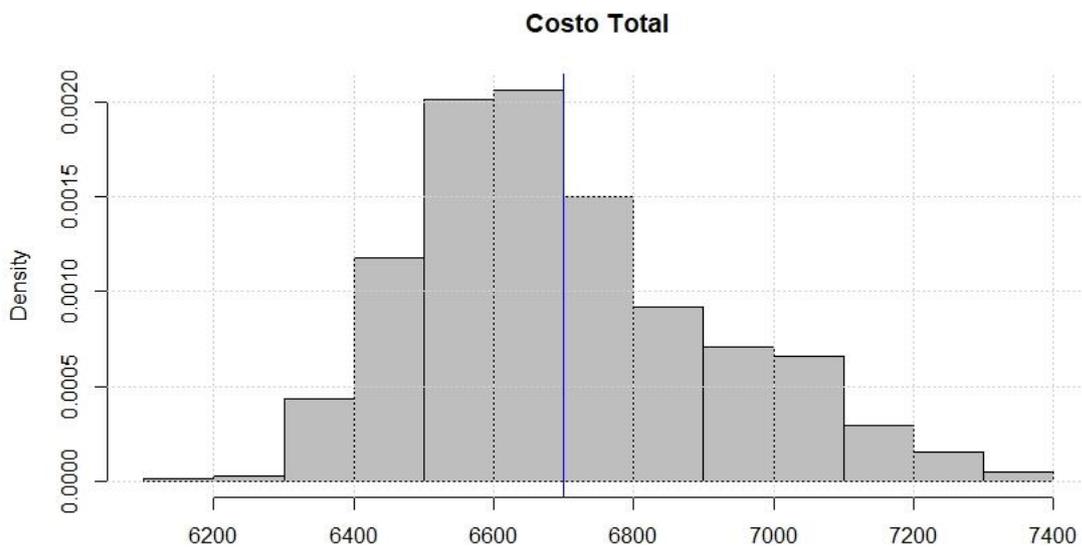
- Intervalo de confianza 95%:

$$P\left\{\bar{l} - Z_{1-\alpha/2} \frac{S}{\sqrt{n}} \leq l \leq \bar{l} + Z_{1-\alpha/2} \frac{S}{\sqrt{n}}\right\} \approx (1 - \alpha)$$

$$P\left\{6700 - 1.96 \frac{214}{\sqrt{1000}} \leq l \leq 6700 + 1.96 \frac{214}{\sqrt{1000}}\right\} \approx 0.95$$

$$P\{6687 \leq l \leq 6713\} \approx 0.95$$

- Histograma del costo total para $Q = 5000$ unidades ($n = 1000$ iteraciones)



Actividad 9

En el Ejemplo #5 el stock inicial al principio del primer día es de 4000 unidades.

Determinar el lote de reposición óptimo Q con la misma política de stock y con un stock inicial al principio del primer día de 2000 unidades.

Verifique que en los modelos dinámicos de horizonte finito, los resultados dependen de las condiciones iniciales del proceso.

GENERACIÓN DE NÚMEROS ALEATORIOS EN EXCEL®

La planilla de cálculo Excel® tiene incorporado un generador de números aleatorios uniformes. La función =ALEATORIO() genera números aleatorios 0-1. Esta función es de naturaleza volátil, es decir que cada vez que se modifica la hoja de cálculo o cada vez que se realiza un recálculo de la hoja presionando la tecla **F9**, todas las celdas del modelo que contengan la función =ALEATORIO() se recalculan.

La función =ALEATORIO.ENTRE(*a*;*b*) genera números rectangulares entre *a* y *b*.

La función =ALEATORIO() puede utilizarse combinada con otras funciones para generar números aleatorios de otras distribuciones. Por ejemplo:

- Distribución exponencial de parámetro *b*
=LN(ALEATORIO()**b*)
- Distribución normal de media μ y desvío σ
=DISTR.NORM.INV(ALEATORIO(); μ ; σ)
- Distribución lognormal de media μ y desvío σ
= DISTR.LOG.INV(ALEATORIO(); μ ; σ)
- Generación de *P* números enteros
=ENTERO(ALEATORIO()**P*)

En las versiones actuales de Excel® no es posible establecer una “semilla” para la función =ALEATORIO(). Se supone que esta semilla es tomada del reloj del sistema.

Las funciones:

- =PROMEDIO(*rango de celdas*)
- =DESVEST(*rango de celdas*)
- =INTERVALO.CONFIANZA.NORM(α , *S*, *N*)
- =INTERVALO.CONFIANZA.T(α , *S*, *N*)

Permiten simplificar el análisis de los resultados.

Una forma alternativa para generar números aleatorios en Excel® es emplear las herramientas que se proveen en la solapa **DATOS/Análisis de datos/Generación de números aleatorios**. Allí se disponen de generadores para variables aleatorias uniformes, normales, bernoulli, binomiales, poisson y discretas. Los números aleatorios generados de esta forma son permanentes. Este modo de generación de números aleatorios permite el empleo de un valor “semilla” que se introduce en el campo: *Iniciar con*.

En la solapa **DATOS/Análisis de datos/Estadística descriptiva** existen herramientas que pueden emplearse para el análisis de los datos de salida de un modelo de simulación de Monte Carlo.

GENERACIÓN DE NÚMEROS ALEATORIOS EN “R”

El lenguaje “R” es básicamente un lenguaje orientado a la estadística. Por esa razón se adapta muy bien para el desarrollo de modelos de simulación de Monte Carlo.

La generación de números aleatorios puede hacerse muy fácilmente. Los siguientes ejemplos ilustran las formas más habituales.

<code>x <- runif(n)</code>	Crea el vector x con <i>n</i> números aleatorios uniformes 0-1
<code>X <- runif(n, a, b)</code>	Crea el vector x con <i>n</i> números aleatorios uniformes <i>a-b</i>
<code>x <- rnorm(n)</code>	Crea el vector x con <i>n</i> números aleatorios normales N(0,1)
<code>x <- rnorm(n, μ, σ)</code>	Crea el vector x con <i>n</i> números aleatorios normales N(<i>μ, σ</i>)
<code>x <- rtriangle(n,a,c,b)</code>	Crea el vector x con <i>n</i> números aleatorios triangulares de mínimo: <i>a</i> ; moda: <i>b</i> ; máximo: <i>c</i> . Requiere la librería “triangle”

Se dispone de una gran cantidad de distribuciones de probabilidad de las que es posible generar muestras de números aleatorios anteponiendo “r” (de random) al nombre de la distribución. Por ejemplo: Beta, Binomial, Cauchy, Chi cuadrado, Exponencial, F, Gamma, Geométrica, Lognormal, Student, Weibull entre otras.

La función **sample(...)** puede emplearse para generar números aleatorios provenientes de distribuciones discretas de probabilidad. El código siguiente permite generar 1000 números aleatorios de la distribución:

x	f(x)
1	0.2
2	0.2
3	0.4
4	0.2

```
x <- seq(1:4)
px <- c(0.2, 0.2, 0.4, 0.2)
r <- sample(x, size=1000, replace=TRUE, prob=px)
```

La función **set.seed(...)** permite establecer una semilla para el generador de números aleatorios. Esto es útil cuando se quiere repetir un ensayo, por ejemplo durante el desarrollo de un modelo.

Análisis de los resultados:

Las funciones **mean(..)** y **sd(...)** calculan la media y el desvío de una muestra. La función **t.test(...)** calcula el intervalo de confianza de una muestra. La función **hist(...)** realiza en forma inmediata el histograma de los datos de una muestra.

En los anexos se exponen los códigos en “R” utilizados para generar los ejemplos.

BIBLIOGRAFÍA

Box, G.E.P; Muller, Mervin. *A note on the generation of random normal deviates.* The Annals of Mathematical Statistics (1958). 29 (2): 610–611.

Coss Bu, Raúl. *Simulación un enfoque práctico.* Limusa. 1986.

García, Roberto Mariano. *Inferencia estadística y diseño de experimentos.* 1ª ed. Buenos Aires, Eudeba 2004.

Hull, T.E. and A.R. Dobell. *Random number generators.* SIAM Review 4.3 (1962) 230-254.

Kuo, Shan. S. *Computer applications of numerical methods.* Addison-Wesley Publishing Company. 1972.

Lehmer, D.H. *Mathematical methods in large-scale computing units.* Annals Comp. Laboratory Harvard Univ. 26 (1951), pp. 141-146

Metropolis, Nicholas; Ulam, Stanislaw. *The Monte Carlo method.* Journal of the American Statistical Association. Vol 44, Nº 247. (Sep. 1949), pp 335-341.

Metropolis, Nicholas. *The beginning of the Monte Carlo method.* Los Alamos Science. Special Issue 1987, pp 125-130.

Naylor, Thomas H; Balintfy, Joseph L; Burdick, Donald S; and Chu, Kong. *Computer simulation techniques.* John Wiley & Sons, New York, 1966.

Robert, Christian P; Casella, George. *Introducing Monte Carlo methods with R.* Springer. 2010.

Rubinstein, Reiven Y; Kroese, Dirk P. *Simulation and the Monte Carlo method.* 2nd edition. Wiley series in probability and statistics. 2008.

Salvador, Fernando. *TP Simulación (mimio).* Centro de Estudiantes de Ingeniería “La Línea Recta”. Buenos Aires 1992.

Taha, Hamdy A. *Investigación de operaciones.* Pearson Education. Mexico 2004.

ANEXO I: CÓDIGO “R” PARA EL EJEMPLO #1

```
#####  
# Ejemplo #1: Cálculo del peso de cilindros  
#  
#-----  
library(triangle)  
dev.off()  
rm(list=ls())  
#-----  
N=1000 # número de iteraciones  
set.seed(26887) # fijo la semilla para reproducir el resultado  
radio<-rtriangle(N,1.95,2.01, 2) # radio del cilindro [cm]  
longitud<-runif(N,9.95,10.05) # longitud del cilindro [cm]  
pespecifico<-rtriangle(N,7.99,8.01,8) # peso específico del material [gramos/cm3]  
  
# Peso del cilindro  
peso<-pi * radio^2 * longitud * pespecifico  
  
# Intervalo de Confianza 95% para la media  
SD=sd(peso) # desvío estandar de la muestra  
PP=mean(peso) # media muestral  
T=qt(p=0.975, df=N-1) # valor de t para nivel de confianza 95% a dos colas  
  
PESO_MAX=PP+T*SD/sqrt(N) # extremo superior del intervalo de confianza  
PESO_MIN=PP-T*SD/sqrt(N) # extremo inferior del intervalo de confianza  
  
# Alternativa  
t.test(peso)  
  
summary(peso)  
SD  
PESO_MAX  
PESO_MIN  
PESO_MAX-PESO_MIN  
  
# Histograma  
hist(peso, col="grey", main="Peso de los Cilindros", xlab="peso en gramos", freq=FALSE)  
abline(v=PP)  
abline(v=PESO_MAX, lty=2, col="blue")  
abline(v=PESO_MIN, lty=2, col="blue")  
  
# Descarte  
Descarte_sup<-sum(peso>1000) # cilindros descartados por peso alto  
Descarte_inf<-sum(peso<980) # cilindros descartados por peso bajo  
Descarte<-(Descarte_sup+Descarte_inf)/N # porcentaje de cilindros descartados  
Descarte
```

ANEXO II: CÓDIGO "R" PARA EL EJEMPLO #2

```
#####  
# Ejemplo #2: Estimación de PI  
#  
#-----  
dev.off()  
rm(list=ls())  
#-----  
N=1000                # número de iteraciones  
set.seed(26887)       # fijo la semilla para reproducir el resultado  
  
x<-runif(N)           # selecciono al azar un punto en 0<x<1  
y<-runif(N)           # selecciono al azar un punto en 0<y<1  
radio<-x^2+y^2        # calculo la distancia de (x,y) a (0,0)  
  
p<-sum(radio<=1)/N    # calculo la p de "exito" (dentro del circulo)  
p  
1-p  
PI=4*p                # el estimador de PI es 4 x p  
PI  
  
# Calculo el intervalo para un nivel de confianza del 95%  
PI_MAX<-4*(p+sqrt(p*(1-p)/(N-1))*qnorm(0.975))  
PI_MIN <-4*(p-sqrt(p*(1-p)/(N-1))*qnorm(0.975))  
PI_MAX  
PI_MIN  
  
# Calculo el error de la estimación  
error<-(PI-pi)/pi  
error
```

ANEXO III: CÓDIGO “R” PARA EL EJEMPLO #3

```
#####  
# EJEMPLO #3: Comparación de alternativas de financiación  
# Tasa variable vs Tasa fija  
#-----  
dev.off()  
rm(list=ls())  
#-----  
# DATOS:  
k=15                # Número de años  
tasa_inicial<-8     # Tasa variable actual (% anual)  
tasa_fija<-6        # Tasa fija (% anual)  
desc<-10            # Tasa de descuento (% anual)  
  
# Distribución de probabilidad de los cambios en la tasa anual  
delta<-c(2.0,1.5,1.0,0.5,0.0,-0.5,-1.0,-1.5,-2.0)  
proba<-c(0.05,0.05,0.15,0.15,0.2,0.15,0.15,0.05,0.05)  
proba_acum<-cumsum(proba)  
  
# SETUP:  
set.seed(26887)     # fijo la semilla para reproducir el resultado  
N<-10000            # Número de iteraciones  
vp_var<-rep(0,N)  
  
#-----  
# Resuelvo para tasa fija  
desc<-desc/100  
vp_fija<-((1-(1+desc)^-k)/desc)*tasa_fija  
vp_fija  
  
#-----  
# Resuelvo para tasa variable. El valor presente de cada iteración se almacena  
# en el vector vp_var[j]  
  
for (j in 1:N){  
  # Genero la tasa de interés proyectada para los k períodos  
  dtasa<-sample(delta, size=15, replace=TRUE, prob=proba)  
  dtasa_acum<-cumsum(dtasa)  
  tasa_var<-(tasa_inicial+dtasa_acum)  
  
  # Calculo el valor presente de la tasa proyectada  
  for(i in 1:k){  
    vp_var[j]<-vp_var[j]+tasa_var[i]/(1+desc)^i  
  }  
}  
#-----
```

```
# Resultados
```

```
vp_fija  
mean(vp_var)
```

```
hist(vp_var, freq=FALSE, main="Valor Presente Tasa Variable", col="grey", xlab=" ")  
abline(v=vp_fija, col="blue")  
grid(col = "lightgray", lty = "dotted")
```

```
# Probabilidad que el valor presente de la tasa variable resulte inferior a la tasa fija  
sum(vp_var<vp_fija)/N
```

```
# Probabilidad que el valor presente de la tasa variable resulte entre 50 y 70.  
(sum(vp_var<50)+sum(vp_var>70))  
(sum(vp_var<50)+sum(vp_var>70))/N
```

```
#-----
```

```
# Gráfico resultado acumulado
```

```
vp_var_acum<-cumsum(vp_var)/(1:N)  
plot(vp_var_acum, type="l", xlab="iteración", ylab="Valor Presente Tasa Variable")  
grid(col = "lightgray", lty = "dotted")
```

```
#-----
```

ANEXO IV: CÓDIGO “R” PARA EL EJEMPLO #5

```
#####  
#Ejemplo #4: Modelo de stock aleatorio  
#  
#-----  
library(triangle)  
dev.off()  
rm(list=ls())  
#-----  
# DATOS:  
N=1000           # número de iteraciones  
t=30             # número de días del mes  
So<-4000        # stock Inicial  
Q<-5000         # lote de reposición  
Sm<-1500        # stock mínimo  
c1<-0.04        # costo almacenamiento USD/(u.día)  
k<-320          # costo de reorden (USD)  
  
# Demanda aleatoria triangular  
a<-800          # demanda mínima  
b<-1000         # demanda moda  
c<-1500         # demanda máxima  
#-----  
set.seed(26887) # fijo la semilla para reproducir el resultado  
Costo_total<-rep(0,N) # inicializo costo total  
  
for (i in 1:N){  
  
  D<-rtriangle(t,a,c,b) # Demanda diaria  
  
  Si<-rep(0,t)          # inicializo stock inicial  
  Sf<-rep(0,t)          # inicializo stock final  
  Cr<-rep(0,t)          # inicializo costo de reposición  
  Ca<-rep(0,t)          # inicializo costo de almacenamiento  
  Si[1]=So              # stock inicial primer período  
  
  for (j in 1:t){      # loop para cada uno de los días del mes  
  
    if (Si[j]<Sm){      # reposición  
      Si[j]<-Si[j]+Q  
      Cr[j]<-k  
    }  
  
    Sf[j]<-Si[j]-D[j]    # stock final del día  
    Ca[j]<-(Si[j]*c1)    # costo de almacenamiento del día  
  
    Si[j+1]<-Sf[j]      # stock inicial del día siguiente  
  }  
}
```

```

}
Costo_total[i]<-sum(Ca)+sum(Cr) # costo total de la iteración
}
#-----
# RESULTADOS:

# Intervalo de Confianza 95% para la media
SD=sd(Costo_total) # desvío estandar de la muestra
PP=mean(Costo_total) # media muestral
T=qt(p=0.975, df=N-1) # valor de t para nivel de confianza 95% a dos colas

Costo_total_max=PP+T*SD/sqrt(N) # extremo superior del intervalo de confianza
Costo_total_min=PP-T*SD/sqrt(N) # extremo inferior del intervalo de confianza

# Alternativa
t.test(Costo_total)

PP
SD
Costo_total_max
Costo_total_min

# Histograma
hist(Costo_total, freq=FALSE, main="Costo Total", xlab=" ", col="grey")
grid(col = "lightgray", lty = "dotted")
abline(v=PP, col="blue")

```

ANEXO V: TABLA DE NÚMEROS ALEATORIOS UNIFORMES

49211	86571	93003	81000	45765	46201	54426	56252
37373	12000	69567	30295	59769	24475	4731	64704
89324	62240	11272	73511	78543	69397	34018	2367
93457	4245	47459	47604	19434	69822	37213	73969
86568	72366	88726	53894	89833	21450	3564	16894
55657	39051	22383	56829	2006	34543	73042	46281
23520	58290	77823	69573	3431	17966	51301	44077
83943	40534	5274	57373	91042	9417	75329	39683
8027	43798	71124	14963	90485	83646	6618	5959
59992	52539	69068	56223	88073	97970	44749	3316
36491	28894	19488	92589	72860	38864	26116	23456
71146	67384	7242	31819	66629	58689	28096	76330
9130	4118	7730	56091	42684	52499	61401	81894
87786	83916	18687	11313	89402	50088	27856	55667
20170	67485	68546	21721	30411	26925	43817	7788
98871	25099	72754	911	31894	58718	17515	85986
31392	89283	44598	31061	35782	24656	26346	54405
17400	74086	4655	2428	3868	64902	2194	56452
23277	41764	21855	81003	8635	84825	91495	28731
85596	30488	6102	62533	10587	78225	46315	80653
22508	10311	7436	20012	38661	24884	26653	90725
34326	15629	23515	92567	76955	17313	21521	75197
13297	6323	13597	98154	23424	44455	74974	29664
52885	6622	47449	15162	13742	25237	73091	143
74041	9738	71025	83228	21839	63304	57112	41375
96346	53289	74258	91583	78953	82178	77853	55137
39911	39720	63528	25145	59907	47522	74380	4855
2146	64029	80826	15771	57818	12107	73335	42499
69141	53444	34657	63651	25749	83483	72873	59795
88447	13573	49964	25897	64053	10731	80713	80787
57048	61820	7165	4679	4112	19199	39562	41607
98138	21272	33016	9270	52267	37541	11802	56132
87044	73751	44417	25209	99448	7157	24294	86542
14791	7395	70097	27023	28693	1325	88430	29222
69260	72209	10488	27926	34947	58453	68965	72087
6313	17202	13950	4658	86566	97181	76866	38292
70124	27491	16275	37871	74465	45394	93415	27132
81181	15999	44149	55270	2459	54323	73721	784
88300	93687	77631	65880	89215	31026	11216	31740
86157	91373	26148	44644	53127	33584	4863	84077

ANEXO VI: TABLA DE NÚMEROS ALEATORIOS NORMALES

0.4411	0.9603	-0.1895	-0.4802	0.7756	-1.5784	1.0653	1.5058
1.1800	1.0362	-0.9115	-1.0205	0.0226	-0.4105	0.5187	0.5638
-2.0718	0.5134	-1.1034	-1.5140	-0.6656	0.9451	0.2927	0.2499
-1.8133	-0.6915	1.6372	1.2380	-0.1525	-0.0735	-0.5760	-0.9135
-0.4951	-0.4933	0.0601	0.6300	0.5946	-0.9277	-0.2958	0.1170
0.7915	-0.2358	-0.2313	1.1393	1.7039	0.2489	-0.8344	0.4863
-0.9825	1.3196	-0.5589	-0.6231	0.9038	0.6161	0.2160	-0.2433
-0.7549	0.7742	1.1016	-0.4091	-0.1141	-0.7828	0.3493	-0.1994
0.0546	0.3348	0.9273	-0.4034	0.2104	-0.7789	-1.2790	-0.2176
-0.7174	0.4271	-0.1018	-0.5045	-0.2964	1.2207	0.4504	-1.1214
0.3220	0.3609	0.5995	1.7043	-0.8662	0.9917	1.1050	-0.9332
0.7761	0.7554	-0.7191	1.0036	1.9346	1.7813	-0.1551	0.7985
-1.1780	-0.8763	0.1745	1.7062	0.5669	-0.3783	0.3581	-0.7722
0.8217	0.7402	0.3462	0.2619	0.3459	0.2945	-0.9141	-1.5544
1.3534	1.2194	1.0964	0.4450	-1.3784	0.0004	0.2117	-0.6567
0.2730	0.1483	-0.8388	-0.8718	0.5185	-0.7901	-0.4125	1.0434
-0.6611	-0.2792	-0.3507	-0.9085	1.3943	2.1842	-0.0465	-0.6085
-0.0854	-0.2428	-0.7356	0.0878	-0.1015	-1.0580	0.5972	0.1066
0.1095	-0.2264	0.1692	1.2659	-0.1819	0.1426	-0.5615	-0.4907
0.2014	-1.0450	0.2360	-0.3737	-0.0955	-2.7985	0.3091	-2.8087
0.1650	0.5296	-0.9098	-0.0151	0.3416	-0.9197	-0.1324	0.1126
0.2632	0.1643	-1.6078	0.4914	-0.8544	-0.1865	0.4974	-1.0040
-0.3866	-0.2262	-1.5649	0.8020	0.2238	-0.5996	-0.0768	-0.1538
-0.7001	1.5009	-0.0204	-0.2302	2.0014	0.4251	-2.0798	1.7384
-0.2439	0.3301	1.3367	0.5336	1.1477	0.7733	-0.1107	1.3322
-0.1306	-0.5452	-0.2979	-0.1189	1.7443	0.9547	0.3148	0.2497
0.7276	0.1138	-0.2061	0.4644	0.3070	-0.7626	-0.0419	-0.3943
0.2382	1.2917	0.9092	1.5282	-1.6941	1.2461	-0.2502	-0.5047
1.2649	0.9160	-0.9462	-0.2439	3.3693	-0.5792	-0.5113	-0.3053
-0.3099	-0.0933	0.0835	0.6435	-0.4897	-0.3757	-0.7590	0.6101
1.0615	-1.5081	0.5535	-1.6254	0.9740	0.2560	-0.3416	-0.5866
-0.5128	0.0068	-0.5150	-0.8740	0.2927	0.8942	-0.8111	0.5480
0.9601	2.5940	0.3442	-1.0761	0.6852	1.3756	0.0363	0.7539
-1.0850	-0.3450	0.6150	0.2951	0.9562	0.7013	-0.9185	-1.2158
1.7158	-1.5259	0.8508	0.8416	-0.9201	-1.0087	-0.1996	-0.7270
-0.1022	0.6403	-0.9511	1.6584	0.5889	-0.8314	0.8513	-1.0169
-0.2507	-0.7378	0.1240	1.2616	-0.1797	-1.3957	0.1348	1.0227
-1.4459	-0.2257	1.5279	-0.7645	-0.4784	0.1569	0.5764	0.2579
0.3723	0.6540	-0.5383	-1.2252	-0.4589	0.1785	-0.0153	-1.5025
0.1134	0.2209	-1.0954	-0.3754	1.3323	1.0101	0.0019	0.5033