14
2Ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

SIMULACION DE DISTRIBUCIONES PROBABILISTICAS

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I A

P R E S E N T A :

MA. ISABEL CARREÑO MARTINEZ

DIRECTOR DE TESIS: DR. MIGUEL ANGEL GARCIA ALVAREZ

1996

VNIVERSIDAD NACIONAL
AVFN°MA DE
MEXICO

M. en C. Virginia Abrín Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
P r e s e n t e

Comunicamos a usted que hemos revisado el trabajo de Tesis: "SIMULACION DE DISTRIBUCIONES PROBABILISTICAS"

realizado por Ma. Isabel Carreño Martinez

con número de cuenta 9060800-1 **, pasante de la carrera de** Actuaria

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis Propietario	Dr. Miguel Angel García Alvarez
Propietario	M. en C. Juan González Hernández
Propietario	Mat. Hugo Villaseñor Hernández
Suplente	M. en C. Guadalupe Carrasco Licea
Suplente	Fis. Guillermo Cesar Anibal Zepeda Perez

Consejo Departamental de Matemáticas
Act. Claudia Carrillo Quiroz

SIMULACIÓN DE DISTRIBUCIONES PROBABILÍSTICAS

DEDICATORIA

A mi Padre Gabino Carreño Venegas, por su apoyo económico , moral y desinteresado ya que gracias a el pude continuar con mis estudios.

A mi Madre Josefina Martinez Saucedo, por su apoyo moral y por que siempre me ayudo a continuar y no darme por vencida.

A mis hermanos Arturo, Laura, Gabino, Rosario, por que siempre conté con su apoyo y comprensión.

A mi Asesor Dr. Miguel Ángel García Alvarez, por su paciencia y dedicación a este trabajo y a mi.

A los profesores que me ayudaron a la revisión de mi trabajo por el tiempo dedicado y por su amable cooperación.

INDICE

INTRODUCCIÓN

CONTENIDO

1. INTRODUCCIÓN

Dado el gran avance en el uso de la simulación y el desarrollo de las computadoras, se han extendido rápidamente técnicas y algoritmos para proyectos que necesitan de mucho tiempo para conocer el resultado, sobre todo entre las personas dedicadas a la investigación.

La simulación fué primeramente usada por los científicos Von Neuman y Ulam, quienes trabajaron en un proyecto llamado Monte Carlo, en la Segunda Guerra Mundial.

Este trabajo incluía una simulación directa de problemas probabilísticos relacionados con la difusión de neutrones.

Este fué el principio, podríamos decir, de un gran desarrollo de la simulación en muchos campos, que abarcan políticos, económicos y sociales, entre otros.

Muchas de las técnicas de simulación requieren la generación de datos que se comporten como valores de variables aleatorias asociadas con algún fenomeno estocástico; Es por esto que se requieren de métodos que permitan realizar esto de manera más eficiente.

El objetivo de este trabajo es el estudio de diferentes métodos que permitan generar una colección de números que se distribuyen como si fueran valores de una variable aleatoria con una distribución dada.

En el primer capítulo se resumen algunos conceptos básicos de probabilidad y estadística, los cuales serán utilizados en los capítulos posteriores.

En el capítulo dos se introducen los conceptos básicos de simulación, se introduce el concepto de número pseudoaleatorio y se estudian los métodos que permiten generarlos. Se analizará en esta parte la importancia de la elección de un buen generador de números pseudoaleatorios pues, como se verá, son la base para simular una distribución probabilística arbitraria.

En el capítulo tercero se estudian dos métodos generales que permiten generar colecciones de números con una distribución arbitraria dada.

En el capítulo cuarto se estudian algunos métodos particulares que se aplican para generar colecciones de números con una distribución particular dada.

En el capítulo quinto se describen los algoritmos utilizados en este trabajo para simular las distribuciones más importantes de la teoría de la probabilidad y la estadística.

Los algoritmos estudiados fueron programados en el lenguaje Pascal. Cabe mencionar que pueden desarrollarse en cualquier otro lenguaje de programación como Fortran, Basic,C, etc. o con algún lenguaje específicamente usado en simulación como el GPSS, DYNAMO, SIMSCRIPT, entre otros.

Para verificar la consistencia de las colecciones de números generados se aplicaron métodos estadísticos de pruebas de hipótesis y se graficaron histogramas para compararlos con las gráficas de las densidades teóricas.

Además, para tener una idea de la eficiencia de cada método se midió el tiempo de cómputo que utilizó la simulación de cada distribución.

0.1 CONCEPTOS BÁSICOS DE LA TEORÍA DE LA PROBABILIDAD Y ESTADÍSTICA

0.1.1 Espacios de probabilidad

Un experimento aleatorio es cualquier experimento que admita diferentes posibles resultados.

Al conjunto de todos los posibles resultados de un experimento aleatorio se le llama *espacio muestral*.Este conjunto puede ser finito, infinito numerable o infinito no numerable.

Se dice que un espacio muestral es *discreto*, si éste se puede poner en correspondencia uno a uno con algún conjunto de enteros positivos.

Un espacio muestral es *continuo* si sus elementos constituyen un intervalo de números reales.

Un *evento* es un conjunto de resultados contenidos en el espacio muestral. En un problema concreto un evento se define en base a una o varias propiedades que lo caracterizan y entonces el subconjunto del espacio muestral que se le asocia está formado por los resultados que satisfacen tal o tales propiedades.

Si un evento está formado por todos los posibles resultados en $E_1 o E_2$ o en ambos entonces a ese evento se le llama la *unión* de E_1 y E_2 y se le denota por $E_1 \cup E_2$.

Si un evento está formado por resultados comunes es decir, tanto de E_1 como E_2, entonces a ese evento se le llama la *intersección* de E_1 y E_2 y se le denota por $E_1 \cap E_2$.

Si un evento está formado por todos los posibles resultados que no están en E entonces a ese evento se le llama el complemento de E y se le denota por E^c.

Dada una colección de eventos, se dice que éstos son mutuamente excluyentes si cualquier par de ellos tiene una intersección vacía.

La probabilidad de un evento es un número real que mide la posibilidad de que tal evento ocurra cuando el correspondiente experimento aleatorio se lleve a cabo. Se tiene así una función real definida sobre la familia de eventos, a la cual se le pide que satisfaga las siguientes propiedades:

$P(E) \geq 0$

$P(\Omega) = 1$

Si se tienen eventos $E_1, E_2, E_3, \ldots$, tales que $E_i \cap E_j = \emptyset$ para toda $i \neq j$, entonces

$P(E_1 \cup E_2 \cup \ldots) = P(E_1) + P(E_2) + \ldots$

Un espacio de probabilidad esta dado por la tripleta $(\Omega, \Lambda, P[.])$, donde Ω es el espacio muestral, Λ es el conjunto de eventos, donde

$i)\Omega \in \Lambda$

$ii)$ Si $A \in \Lambda$ entonces $\overline{A} \in \Lambda$

$iii)$ Si A_1y $A_2 \in \Lambda$ entonces $A_1 \cup A_2 \in \Lambda$

y $P[.]$ es la función de probabilidad que tiene como dominio el conjunto de eventos y contradominio el intervalo $[0, 1]$.

0.1.2 Reglas de la suma, del producto y de la probabilidad total.

La última propiedad de la función de probabilidad nos permite calcular la probabilidad de una unión de eventos mutuamente excluyentes. Si A y B son eventos cualesquiera, se tiene la siguiente regla general, la cual es llamada la regla de la suma:

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

cuya forma general es como sigue:

$P(A_1 \cup A_2, ..., \cup A_n) = P(A_1) + P(A_2 +, \ldots, +P(A_i)$

$-P(A_1 A_2) - P(A_1 A_3) -, \ldots, -P(A_{n-1} A_n)$

$+P(A_1 A_2 A_3) + P(A_1 A_2 A_4) + \ldots + P(A_{n-2} A_{n-1} A_n)$

....

$+(-1)^{n-1} P(A_1 A_2 \ldots A_n)$

En una forma más simplificada tenemos que

$P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i) - \sum_{i<j}^{n} P(A_i A_j) + \sum_{i<j<k}^{n} P(A_i A_j A_k)$

$- \sum_{i<j<k<l}^{n} P(A_i Aj A_k A_l) + \ldots + (-1)^{n-1} P(A_1 A_2 \ldots A_n)$

Diremos que se tiene una partición del espacio muestral cuando se tiene una colección finita o infinita numerable de eventos mutuamente excluyentes $A_1, A_2, \ldots$ cuya unión es todo el espacio muestral. Cuando se tiene una colección de este tipo se puede obtener la probabilidad de un evento A de la siguiente manera

$P(A) = \sum_i P(A \cap A_i)$

Sean A y B dos eventos con $P(B) > 0$, se define la probabilidad condicional de A dado que ocurre el evento B como el cociente de la probabilidad conjunta de A y B entre la probabilidad de B y se denota a esta probabilidad como $P(A/B)$, es decir,

$P(A/B) = \frac{P(A \cap B)}{P(B)}$.

La fórmula que define la probabilidad condicional puede escribirse como producto, lo que da como resultado la *regla del producto* de probabilidades, es decir

$P(A \cap B) = P(B)P(A/B)$.

La regla del producto, en su forma general, es la siguiente

$P(A_1 A_2 \ldots A_n) = P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1 A_2) \ldots P(A_n \mid A_1 A_1 \ldots A_{n-1})$

Combinando lo anterior se obtiene lo que se conoce como la regla de la probabilidad total, la cual establece que si $A_1, A_2, \ldots$ es una partición del espacio muestral y A es cualquier evento, entonces

$P(A) = \sum_i P(A/A_i)P(A_i)$

0.1.3 Independencia

Se dice que dos eventos A y B son independiente si la probabilidad de ocurrencia de uno no está influenciada por la del otro,esto es si $P(A/B) = P(A)$ y si $P(B/A) = P(B)$.

Si un evento A es independiente de otro B, se cumple

$P(A) = P(A/B) = \frac{P(A \cap B)}{P(B)}$.

Así que

$P(A \cap B) = P(A)P(B)$

Esta última relación no requiere la restricción de que los eventos tengan probabilidad positiva y es por esto que se toma como la definición de independencia.

Para el caso de más de dos eventos, diremos que los eventos $A_1, A_2 \ldots A_n$ son independientes si para cualquier subcolección de ellos, $A_{k1}, A_{k2}, \ldots A_{km}$, se tiene $P(A_1 \cap, \ldots \cap A_k) = P(A_1) \ldots P(A_k) = \prod_{i=1}^{k} P(A_i)$

0.1.4 Variables Aleatorias y Distribuciones de Probabilidad.

A continuación mencionaremos la definiciones de variable aleatoria, de la función de distribución, y de las propiedades de estas, tanto en caso discreto como continuo.

0.1.5 Variable Aleatoria

Para un espacio de probabilidad dado $(\Omega, \mathcal{A}, P[.])$ una variable aleatoria, denotada por X ó $X(.)$, es una función con dominio Ω y contradominio la recta real.

La función $X(.)$ debe ser tal que el conjunto A_r, definido por $A_r = \{w : X(w) \leq r\}$ pertenece a el espacio de eventos $\mathcal{A}$ para todo número real r.

En otras palabras diremos que una variable aleatoria es un resultado numérico de algún experimento.

0.1.6 Función de Distribución

La función $F(x) = P(X \leq x)$, definida para cualquier $x \in \Re$, es llamada función de distribución de la variable aleatoria X. Ésta es también llamada función de distribución acumulativa.

La probabilidad de que x tome valores en un intervalo $(a, b]$ se obtiene por medio de la función de distribución acumulativa como una diferencia, es decir $P(a < X \leq b) = F(b) - F(a)$.

Propiedades de la función de distribución.

Primero, debido a que $F(x)$ está definida como una probabilidad, sus valores están entre cero y uno. Además, se tiene

$F(-\infty) = \lim_{x \to -\infty} F(x) = \lim_{x \to -\infty} P(X \leq x) = 0.$

$F(\infty) = \lim_{x \to \infty} F(x) = \lim_{x \to \infty} P(X \leq x) = 1.$

Segundo, $F(x)$ es una función no decreciente y continua por la derecha en todos los puntos x.

Existen dos tipos importantes de variables aleatorias, las discretas y las absolutamente continuas.

Variable aleatoria discreta.

Una variable aleatoria es discreta si el número de valores que puede tomar es numerable (finito o infinito).

Variable aleatoria continua.

Una variable aleatoria es llamada continua si su correspondiente función de distribución es continua.

Variable aleatoria absolutamente continua

Una variable aleatoria X es llamada absolutamente continua si existe una función $f(x)$ no negativa, llamada la función de densidad de X, tal que

$\int_{-\infty}^{\infty} f(y)dy = 1$

$P(X \leq x) = \int_{-\infty}^{x} f(y)dy$ para todo $x \in \Re$.

En particular, una variable absolutamente continua es continua.

0.1.7 Distribución de Probabilidad de variables aleatorias Discretas.

Sea X una v.a discreta, a la función $p(x) \equiv P(X = x)$, para $x \in \Re$, se le llama la función de probabilidad de X y satisface las siguientes propiedades:

1. $p(x) \geq 0$ para todos los valores $x \in \Re$;
2. $\sum_{\{x:p(x)>0\}} p(x) = 1$

La función de distribución acumulativa de la variable aleatoria discreta X es la función

$F : \Re \to [0,1]$, definida por

$F(x) = P(X \leq x) = \sum_{x_i \leq x} p(x_i)$

En el caso discreto, una variable aleatoria X está caracterizada por la función de probabilidad puntual $p(x)$, la cual determina la probabilidad puntual de que $X = x$, y por la función de distribución acumulativa $F(x)$, la que representa la suma de las probabilidades puntuales hasta el valor x de X inclusive.

0.1.8 Distribución de probabilidad de variables aleatorias continuas.

La distribución de probabilidad de una variable aleatoria absolutamente continua X está caracterizada por una función $f(x)$ que es su función de densidad de probabilidad. En este caso se tiene

$P(a \leq X \leq b) = \int_a^b f(x)dx$ para cualesquiera a y $b \in \Re$.

Como puede verse, en este caso la probabilidad de que X tome valores en el intervalo $[a, b]$ es el área acotada por la función de densidad y las rectas $x = a$ y $x = b$.

La función de densidad $f(x)$ del caso absolutamente continuo no es la misma función de probabilidad que para el caso discreto ya que en el caso continuo la probabilidad de que X tome el valor específico x es cero, así que la función de densidad de probabilidad no representa la probabilidad de que $X = x$, más bien ésta determina la probabilidad de un intervalo $a \leq X \leq b$.

La función de distribución acumulativa de una variable aleatoria absolutamente continua X es la probabilidad de que X tome un valor menor o igual a un x específico, es decir

$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$,

De aquí puede verse que la función de distribución acumulativa $F(x)$ es el área acotada por la función de densidad entre $-\infty$ y x. Además se tiene

$P(X = x) = \int_x^x f(t)dt = 0$,

$P(X \leq x) = P(X < x) = F(x)$.

0.1.9 Valor Esperado de una variable aleatoria

Si X es una variable aleatoria discreta, diremos que ésta tiene esperanza finita si la serie $\sum_x xp(x)$ converge absolutamente, donde $p(x)$ es la función de probabilidad de X. En este caso denotamos la esperanza de X, como $E(X)$, y se representa de la siguiente manera

$E(X) = \sum_x xp(x)$

En el caso absolutamente continuo, diremos que una variable aleatoria X tiene esperanza finita si la integral $\int_{-\infty}^{\infty} \mid x \mid f(x)\,dx$ es finita, donde $f(x)$ es la función de densidad de X. En este caso denotamos la esperanza de X, $E(X)$, de la siguiente manera

$E(X) = \int_{-\infty}^{\infty} xf(x)\,dx$

La esperanza de una variable aleatoria X no es una cantidad aleatoria, sino un número fijo que depende únicamente de la distribución de probabilidad de X.

El valor esperado (o esperanza) de una variable aleatoria X representa el promedio de X después de un número grande de experimentos

0.1.10 Momentos de una variable aleatoria

Los momento de una variable aleatoria X son los valores esperados de ciertas funciones de X. Los momentos de X pueden definirse alrededor de cualquier punto de referencia, pero generalmente se definen alrededor del cero o del valor esperado de X.

Para una variable aleatoria X y un entero positivo k, la esperanza $E(X^k)$ es llamada el k-ésimo momento de X.

El momento central de orden k de una variable aleatoria X se define como $\mu_k = E\left[(X - E(X))^k\right]$.

El momento central cero de cualquier variable aleatoria X es uno, ya que $\mu_0 = E\left[(X - \mu)^0\right] = E(1) = 1$

De manera similar, el primer momento central de cualquier variable aleatoria es cero, dado que

$\mu_1 = E(X - \mu) = E(X) - \mu = 0,$

el segundo momento central, $\mu_2 = E(X - \mu)^2$, recibe el nombre de *varianza* de la variable aleatoria y se le denota por

$Var(X)$ o σ_X^2.

0.1.11 Varianza de una variable aleatoria

La varianza queda expresada de la siguiente manera

Sea $\mu = E(X)$,

entonces,

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

Se tiene así que la varianza de cualquier variable aleatoria es el segundo momento alrededor del origen menos el cuadrado de la media. La varianza de una variable aleatoria es una medida de la dispersión de la distribución de probabilidad de ésta. Para el caso continuo, si la mayor parte del área por debajo de la curva de distribución se encuentra cercana a la media, la varianza es pequeña; mientras que si la mayor parte del área se encuentra muy dispersa alrededor de la media, la varianza será grande.

la varianza de una variable aleatoria es invariante bajo traslaciones, en efecto, tenemos

$Var(X + b) = Var(X)$ para cualquier constante b.

A la raíz cuadrada positiva de la varianza se le llama *desviación estándar* de X y se le denota por σ_x .La desviación estandar, al igual que la variaza, es una medida de dispersión de los valores de la variable aleatoria.

La *covarianza* de una pareja de variables aleatorias X y Y nos dice la medida en que dos variables "varían juntas" y está definida como

$$Cov(X,Y) = E[(X - E[X])(Y - E[Y])).$$

El coeficiente de correlación indica la dirección de relación y el grado de dependencia lineal entre las variables X y Y y está definido como

$$Corr(X,Y) = \frac{cov(X,Y)}{\sqrt{[var(X)Var(y)]}}$$

A manera de resumen, diremos que el valor esperado de una variable aleatoria X es frecuentemente usada como medida de localización de la distribución de X, mientras que la varianza proporciona una medida de espaciamiento de la distribución.

0.1.12 Distribuciones Discretas de Probabilidad

Aquí se examinarán varias distribuciones de probabilidad discretas y se mencionarán las características más generales para el estudio de éstas, como por ejemplo, su función de probabilidad, función de distribución acumulativa. su media,su varianza y la relación existente entre cada distribución, ya que como se vera más adelante esto toma un papel muy importante para la generación de números aleatorios.

0.1.13 Distribución Uniforme discreta.

La distribución uniforme juega un papel muy importante en la generación de números pseudoaleatorios

Se dice que una variable aleatoria está distribuida uniformemente en el conjunto $\{x_1, x_2, \ldots, x_N\}$ si su función de densidad discreta está dada por

$$f(x;N) = \begin{cases} \frac{1}{N} & \text{para } x = x_1, x_2, \ldots, x_N \\ 0 & \text{e.o.c.} \end{cases}$$

donde el parámetro N tiene como rango los enteros positivos.

Una variable aleatoria así definida es llamada varialde aleatoria discreta uniforme

Si una variable aleatoria X está distribuída uniformemente en el conjunto $\{1,2,\ldots,N\}$, entonces

$$E(X) = \frac{N+1}{2} \qquad Var(X) = \frac{(N+1)(N-1)}{12}$$

0.1.14 Distribución Bernoulli

Aquí se consideran repeticiones sucesivas de un experimento u observación en la cual solo hay dos posibles resultados; cada repetición es llamada un ensayo.

Algunos ejemplos de esta situación son los siguientes:

·El observar un huevo de gallina que ha sido empollado y determinar si el polluelo es masculino o femenino.

·La prueba de un antibiótico en un ratón registrando si la reacción del ratón es positiva o negativa.

Cuando se tienen ensayos repetidos e independientes se les llama *ensayos de Bernoulli.*

En cada ensayo sólo hay dos resultados posibles y sus propiedades son las mismas en todos los ensayos. El espacio muestral en los n ensayos de Bernoulli contiene 2^n puntos, o sucesiones de n símbolos

Ensayos de Bernoulli

1) Cada ensayo da solo dos resultados llamados técnicamente éxito (e) o fracaso (f)

2) Para cada ensayo, la probabilidad de un éxito $P(e)$ es la misma y se denota por $p = P(e)$, la probabilidad de un fracaso es entonces $q = 1 - p$

3) Los ensayos son independientes, es decir, la probabilidad de un éxito en un ensayo no cambia dada la información acerca del resultado de otro ensayo

Se dice que una variable aleatoria X tiene distribución Bernoulli si su función de densidad discreta está dada por:

$$f(x;p) = \begin{cases} p^x(1-p)^{1-x} & \text{para} \quad x = 0 \text{ ó } 1 \\ 0 & \text{e.o.c.} \end{cases}$$

donde el parámetro p satisface $0 \leq p \leq 1$ y $1 - p$ es denotado por q

Si una variable X tiene una distribución Bernoulli entonces su esperanza y varianza son

$E(X) = p \quad Var(X) = pq$, respectivamente

0.1.15 La Distribución Binomial

La distribución binomial es una de las distribuciones de probabilidad más útiles, esto debido a sus multiples áreas de aplicación, las cuales incluyen inspección de calidad, ventas, mercadotecnia, etc.

Cuando un número n dado de repeticiones de ensayos de Bernoulli es conducido con probabilidad de éxito p y consideramos la variable aleatoria X, la cual representa el conteo de los éxitos en los n ensayos, entonces la distribución de X es llamada una distribución Binomial con parámetros n y p.

Sea X es una variable aleatoria que representa el número de éxitos en n ensayos y sea p la probabilidad de éxito, entonces la función de densidad de probabilidad de X está dada por

$$p(x;n,p) = \begin{cases} \binom{n}{x}p^x(1-p)^{n-x} & x = 0,1,2,\ldots,n \\ 0 & \text{en otro valor de } x \end{cases}$$

La probabilidad de que una variable aleatoria X sea menor o igual a un valor específico de x, se determina por la función de distribución acumulativa.

$$P(X \leq x) = F(x;n,p) = \begin{cases} 0 & \text{si } x < 0 \\ \sum_{i=1}^{[|x|]} \binom{n}{i}p^i(1-p)^{n-i} & \text{si } 0 \leq x \leq n \\ 1 & \text{si } x > n \end{cases}$$

Se puede observar que si $n = 1$, la función de probabilidad Binomial se reduce a:

$$p(x;p) = \begin{cases} p^x(1-p)^{1-x} & x = 0,1 \\ & \text{e.o.c.}^2 \end{cases}.$$

que es una distribución Bernoulli.

Si X es una variable aleatoria binomial X entonces

$E(X) = np$ y $Var(X) = np(1-p)$.

Se ve entonces que la media de una variable aleatoria binomial es el producto entre el número de ensayos y la probabilidad de éxito en cada uno de éstos y la varianza es el producto de la media por la probabilidad de tener un fracaso, de manera que la varianza de una variable aleatoria binomial siempre es menor que el valor de su media.

Por último diremos que la distribución binomial se aplica cuando se muestrea con sustitución (la probabilidad de obtener un artículo defectuoso, por ejemplo, es constante.)

0.1.16 Distribución Geométrica

La distribución geométrica es otra distribución discreta en el contexto de ensayos de Bernoulli.

Cuando se lleva a cabo un número dado de n de estos ensayos, el número de éxitos es una variable aleatoria con distribución binomial $b(n,p)$.

Si en lugar de un número dado de ensayos se quisiera llevar a cabo ensayos de Bernoulli hasta que el primer éxito ocurra, entonces el número de éxitos está dado por 1 y el número de ensayos es una variable aleatoria.

Sea X el número de fracasos que se obtienen antes del primer éxito, entonces para $x = 0, 1, 2, \ldots$, el evento $[X = x]$ ocurre si y solo si tenemos una cadena de x fracasos seguida por 1 éxito, sabiendo que los ensayos son terminados tan pronto como un éxito es obtenido.

Usando la condición de independencia de ensayos de obtiene

$$P[X = x] = \left[\underbrace{FF\ldots F}_{x} S\right] = q^x p$$

Es decir, si una variable aleatoria X tiene distribución geométrica, función de probabilidad está dada por

$f_x(x) = f_x(x,p) = p(1-p)^x$ para $x = 0, 1, 2\ldots$

donde el parámetro p satisface $0 \leq p \leq 1$ y $1 - p = q$

La probabilidad de que una variable aleatoria X sea menor o igual a un valor específico de x se determina por la función de distribución acumulativa, dada por

$$F(x) = \begin{cases} 1-(1-p)^{\|\|x\|\|+1} & \text{si} \quad x \geq 0 \\ 0 & \text{e.o.c.} \end{cases}$$

Si la variable aleatoria X tiene una distribución geométrica, entonces

$E[X] = \frac{q}{p}$ y $Var[X] = \frac{q}{p^2}$

0.1.17 Distribución Hipergeométrica

Se dice que una variable aleatoria X tiene distribución hipergeométrica si su función de probabilidad está dada por

$$p(x) = \begin{cases} \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}} & x = 0, 1, \dots n, \quad x \leq k, \quad n - x \leq N - k \\ 0 & \text{e.o.c.} \end{cases}$$

donde N es un entero positivo, k es un entero no negativo tal que $k \leq M$, y n es un entero positivo tal que $n \leq N$.

Si X tiene una distribución hipergeométrica entonces su esperanza y su varianza están dadas por:

$E(X) = \frac{nk}{N}$ y $Var(X) = np(1-p)\left(\frac{N-n}{N-1}\right)$, respectivamente.

0.1.18 Distribución Poisson.

Esta es otra distribución discreta de probabilidad muy útil en la que la variable aleatoria representa el número de eventos que ocurren hasta un cierto tiempo cuando estos eventos se presentan aleatoriamente. Además es una aproximación muy buena a la función de probabilidad binomial cuando p es pequeña y n grande.

Se dice que una variable aleatoria X tiene distribución Poisson si su función de densidad de probabilidad está dada por:

$$P(x;\lambda) = \begin{cases} \frac{e^{-\lambda}\lambda^x}{x!} & x = 0, 1 \dots; \quad \lambda > 0 \\ 0 & \text{e.o.c.} \end{cases}$$

La probabilidad de que una variable aleatoria X con distribución de Poisson sea menor o igual a un valor de x se determina por la función de distribución acumulativa

$$P(X \leq x) = F(x;\lambda) = \begin{cases} 0 & \text{si } x < 0 \\ \sum_{i=0}^{[[x]]} \frac{e^{-\lambda}\lambda^i}{i!} & \text{si } x \geq 0 \end{cases}$$

Si una variable aleatoria X se distribuye Poisson su valor esperado y su varianza están dadas por $E(X) = \lambda$ y $Var(X) = \lambda$, respectivamente.

La distribución Poisson se emplea entre otras cosas para modelar el número de eventos aleatorios independientes que ocurren, por ejemplo el número de llamadas por hora en un conmutador, el número de defectos por unidad de algún material.

Naturalmente no todos los conteos pueden ser modelados realisticamente con una distribución Poisson, pero, por ejemplo, el número de accidentes automovilisticos o el número de seguros reclamados en alguna unidad de tiempo frecuentemente se asume como una variable aleatoria con una distribución Poisson.Cada una de estas instancias o sucesos pueden pensarse como un proceso que genera un número de cambios, accidentes, reclamos, etc., en un intervalo dado(tiempo o espacio).

Cuando el valor de n es grande y el valor de p es cercano a cero, la distribución binomial con parámetros n y p puede ser aproximada por una distribución Poisson con media np.

En efecto, supongamos que para cada entero positivo n se tiene una variable aleatoria X con distribución binomial de parámetros n y p de tal manera que el producto $\lambda = np$ se mantiene constante, entonces

$f(x/n,p) = \frac{n(n-1)\ldots(n-x+1)}{x!}p^x(1-p)^{n-x} = \frac{\lambda^x}{x!}\frac{n}{n}\cdot\frac{n-1}{n}\ldots\frac{n-x+1}{n}(1-\frac{\lambda}{n})^n(1-\frac{\lambda}{n})^{-x}$

$\lim_{x\to\infty}\frac{n}{n}\cdot\frac{n-1}{n}\ldots\frac{n-x+1}{n}(1-\frac{\lambda}{n})^{-x} = 1$

$\lim_{n\to\infty}(1-\frac{\lambda}{n})^n = e^{-\lambda}$

Por lo tanto,

$f(x/n,p) \to \frac{e^{-\lambda}\lambda^x}{x!}$

La expresión del lado derecho de esta última relación, es la función de probabilidad $f(x/\lambda)$ de la distribución Poisson con media λ .

Por lo tanto cuando n es grande y p cercano a cero, el valor de la función de probabilidad $f(x/n,p)$ de la distribución binomial puede ser aproximada por el valor de la función de probabilidad $f(x/\lambda)$ de la distribución Poisson para la cual $\lambda = np$.

0.1.19 Distribución Binomial Negativa

Consideremos una sucesión de ensayos de Bernoulli independientes, con probabilidad de éxito p en cada ensayo. Sea X el número de ensayos necesarios para alcanzar, de manera exacta, k éxitos. La función de probabilidad de X está dada por

$$p(x;k,p) = \begin{cases} \binom{k+x-1}{k-1}p^k(1-p)^x & x=0,1,2,\ldots \quad 0\le p\le 1, \quad k=1,2,\ldots \\ 0 & \text{e.o.c.} \end{cases}$$

A una distribución de este tipo se le llama "binomial negativa".

En vista de lo anterior cabe hacer notar que si $k = 1$, se obtiene la distribución geométrica.

Si una variable aleatoria X tiene distribución Binomial Negativa, entonces $E(X) = \frac{k(1-p)}{p}$ y $Var(X) = \frac{k(1-p)}{p^2}$.

Como siguiente punto definiremos las distribuciones continuas de mayor uso en este trabajo.

Al igual que en el caso discreto, se mencionara su función de probabilidad, su función de densidad acumulativa, su esperanza y varianza, respectivamente y la relación que exite entre ellas, esto con el fin de posteriormente ser generadas.

0.1.20 Distribuciones continuas de probabilidad

A continuación mencionaremos las distribuciones continuas, como en el caso discreto mencionaremos, su función de densidad, su función de distribución acumulativa, su esperanza y varianza, y la relación que existe entre cada una de ellas.Comenzaremos por la distribución normal

0.1.21 La distribución Normal y el teorema del límite central

La distribución normal es quizá la más importante y la de mayor uso de todas las distribuciones continuas de probabilidad.

La apariencia gráfica de la distribución noromal es una curva simétrica con forma de campana, que se extiende sin límite tanto en la dirección positiva como en la negativa.

Se dice que una variable aleatoria X se distribuye normalmente si su función de densidad está dada por:

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], -\infty < x < \infty, \quad -\infty < \mu < \infty, \quad \sigma > 0$$

donde los parámetros μ y σ^2 representan la media y la varianza, respectivamente, de X.

La probabilidad de que una variable aleatoria normalmente distribuída sea menor o igual a un valor específico, x está dada por la función de *distribución acumulativa*

$$P(X \leq x) = F(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} exp[-(t-\mu)^2/2\sigma^2]\,dt,$$

la integral no puede evaluarse en forma cerrada, pero puede ser tabulada y debido a que existen un número infinito de valores de μ y de σ se emplea una transformación.

Sea Z una variable aleatoria definida como $Z = (X - \mu)/\sigma$, en donde μ y σ son la media y la desviación estandar de X, respectivamente. Z es entonces una variable aleatoria estandarizada con media cero, y desviación estándar uno.

Tenemos entonces a su función acumulativa definida de la siguiente manera

$$\begin{aligned} P(X \leq x) &= P[Z \leq (x-\mu)/\sigma] \\ &= \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{(x-\mu)/\sigma} exp(-z^2/2)(\sigma dz) \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{(x-\mu)/\sigma} exp(-z^2/2)dz \end{aligned}$$

Es decir, se tiene

$P(X \leq x) = P(Z \leq z)$, entonces

$F_X(\frac{x-\mu}{\sigma}) = F_Z(z)$, donde

$z = (x-\mu)/\sigma$ y $F_z(z;0,1)$

Donde z es la función de distribución acumulativa de la función de densidad normal estandarizada.

La distribución normal es llamada así debido a su ocurrencia natural, la cual está ligada al teorema del límite central.

El teorema nos dice que bajo ciertas condiciones, si se suma un número grande de variables aleatorias de cierto experimento que quizá no sean normales , entonces el resultado de esta suma nos da una buena aproximación a la distribución normal. La forma precisa del teorema se puede formular de la siguiente manera:

Sean $X_1, X_2, X_3, \ldots, X_n, \ldots$ variables aleatorias independientes e idénticamente distribuídas, con esperaza $E[X_i] = \mu$ y varianza $Var(X_i) = \sigma^2$, entonces para cualquier número real x,

$$\lim_{n\to\infty} \Pr\left\{\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(\frac{X_i-\mu}{\sigma}\right) \leq x\right\} = \Phi(x)$$

donde $\Phi(x)$ es la función de distribución acumulativa normal con media cero y varianza 1

Como caso particular tenemos que la distribución binomial (n, p) que fué definida como la distribución que gobierna el número de éxitos X en n ensayos independientes de un experimento que tiene una probabilidad de éxito p en cada ensayo, puede aproximarse, para n grande y p no cercana a 0 o 1, mediante una distribución normal.

Sabiendo que la variable aleatoria binomial X tiene media $\mu = np$ y una desviación estándar $\sigma = \sqrt{np(1-p)}$, cuando n es grande, pero σ^2 es moderada, tal que cuando np o $n(1-p)$ es moderada, podemos tratar a la binomial X como si ésta tuviera una distribución Normal $N(np, \sqrt{np(1-p)})$.

De esta forma, si Z tiene distribución normal estándar, se tiene

$$P[a \leq X \leq b] \approx P\left[\frac{a-np}{\sqrt{np(1-p)}} \leq Z \leq \frac{b-np}{\sqrt{np(1-p)}}\right]$$

Cuando se aproxima una distribución discreta con una distribución continua, es conveniente realizar un ajuste que se conoce como corrección continua. Por ejemplo, en el caso de la binomial, se logra una mejor aproximación mediante la fórmula siguiente:

$$P[a \leq X \leq b] \approx \left[\frac{a-0.5-np}{\sqrt{np(1-p)}} \leq Z \leq \frac{b+0.5-np}{\sqrt{np(1-p)}}\right]$$

0.1.22 Distribución Uniforme

Se dice que una variable aleatoria X está distribuida uniformemente sobre el intervalo (a, b) si su función de densidad es:

$$f(x; a, b) = \begin{cases} 1/(b-a) & a < x < b \\ 0 & \text{c.o.c.} \end{cases}$$

La función de densidad de probabilidad de una distribución uniforme es contante en el intervalo (a, b).

La función de distribución acumulativa se determina de manera fácil y está dada por

$$P(X \leq x) = F(x;a,b) = \begin{cases} 0 & x < a \\ (b-a)^{-1}\int_a^x dt & a \leq x \leq b \\ 1 & x > b \end{cases}$$

$$= \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x \leq b \\ 1 & x > b \end{cases}$$

Si X se distribuye uniformemente, entonces

$E(X) = (b-a)^{-1}\int_a^b x dx = (b-a)^{-1}(\frac{x^2}{2})\,|_a^b$

$= (b-a)^{-1}b^2 - a^2/2 = (a+b)/2$

y $Var(X) = (b-a)^2/12$,

Al caso en que $a = 0$ y $b = 1$ se le conoce como la distribución uniforme sobre el intervalo unitario$(0,1)$ con su función de densidad de probabilidad dada por $f(x;0,1) = 1, \quad 0 \leq x \leq 1$, esta distribución es muy importante ya que, como se mencionó anteriormente, juega un papel clave en la simulación por computadora de los valores de una variable aleatoria con una función de distribución específica.

0.1.23 Distribución Gama

Para un entero positivo α sea el valor $\Gamma(\alpha)$ definido por la siguiente integral

$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx$

esta función es llamada la función gama

Obsérvese que si integramos por partes haciendo

$dv = e^{-x}dx$,

$v = -e^{-x}$,

$u = x^{\alpha-1}$,

$du = (\alpha-1)x^{\alpha-2}dx$; entonces se tiene

$\Gamma(\alpha) = \int_0^\infty x\alpha^{\alpha-1}e^{-x}dx$

$= -x^{\alpha-1}e^{-x}\,|_0^\infty - (\alpha-1)\int_0^\infty x^{\alpha-2}e^{-x}dx$

$= 0 + (\alpha-1)\Gamma(\alpha-1) = (\alpha-1)\Gamma(\alpha-1)$

la cual es una importante relación recursiva de la función Gama. En base a esta propiedad, si $n \geq 2$ es un entero positivo, entonces

$\Gamma(n) = (n-1)\Gamma(n-1) = (n-1)(n-2)\Gamma(n-2)......\Gamma(1)$.

pero $\Gamma(1) = \int_0^\infty e^{-x}dx = 1$ y por lo tanto $\Gamma(n) = (n-1)!$.

Se dice que la variable aleatoria X tiene una distribución gama si su función de densidad de probabilidad está dada por:

$$f(x;\alpha,\beta) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} exp^{-\beta x} & x,\alpha,\beta > 0 \\ 0 & \text{e.o.c.} \end{cases} \qquad (a)$$

Si $\alpha = 1$ la ecuación (a) se transforma en $f(x) = \beta e^{-\beta x}$ para $x > 0$, la cual es llamada una densidad exponencial.

Si α no es un entero positivo, no existe una forma cerrada para el cálculo de la integral en la función de densidad, pero si α es un entero positivo, la distribución Gama puede escribirse como

$F(x) = 1 - e^{-x/\beta} \sum_{k=0}^{\alpha-1} \frac{(x/\beta)^k}{k!}$ si $x > 0$

Si la variable aleatoria X tiene una distribución Gama, entonces la esperanza y la varianza están dadas por $E(X) = \alpha/\beta$ y $Var(X) = \alpha(\alpha+1)/\beta^2 - (\frac{\alpha}{\beta})^2 = \alpha/\beta^2$.

0.1.24 Distribución Exponencial

Está distribución, como ya se mencionó, es un caso especial de la distribución gama.

Si una variable aleatoria X tiene una distribución exponencial su función de densidad está dada por:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & 0 \leq x \leq \infty \\ 0 & \text{e.o.c.} \end{cases}$$

donde λ es el parámetro y es igual al número promedio de eventos por unidad que ocurren cuando tales eventos se presentan de manera aleatoria.

La función de distribución acumulativa está dada por

$P(X \leq x) = F(x;\lambda) = 1 - \exp^{-\lambda x}$ para $x > 0$.

Si la variable aleatoria X tiene distribución exponencial, su esperanza y varianza están dadas por

$E(X) = 1/\lambda$ y $Var(X) = 1/\lambda^2$.

Por otra parte, bajo cierta hipótesis de independencia, la longitud del tiempo que transcurre en acontecimientos sucesivos, que ocurren aleatoriamente, se distribuye exponencialmente. Por esta razón, la distribución exponencial se emplea para estudiar problemas como el tiempo que transcurre antes de que ocurra una falla en una máquina, el número de llamadas recibidas en un lapso de tiempo específico, etc.

Debido a estas características, la distribución Exponencial y la Poisson se encuentran relacionadas.

Se mencionó que bajo ciertas condiciones, el conteo del número de acontecimientos en un intervalo de tiempo dado tiene una distribución Poisson con parámetro proporcional a la longitud del intervalo.

Supongamos ahora que uno de estos acontecimientos ha ocurrido y queremos determinar la distribución del tiempo Y que transcurre hasta el próximo acontecimiento

$P[Y > t] = P$[no ocurre ningún acontecimiento en el intervalo de longitud t] = e^{-vt}, donde v es el número promedio de ocurrencias por unidad de tiempo, esto es

$$F_Y(t) = P[Y \leq t]$$
$$= 1 - P[Y > t] = 1 - e^{-vt} \qquad \text{para } t > 0;$$

Así que Y tiene una distribución exponencial. Por otro lado, puede demostrarse que bajo cierta presunción de independencia, si tenemos acontecimientos que ocurren en el tiempo de tal manera que la distribución de la longitud del tiempo entre acontecimientos sucesivos es exponencial, entonces el número de acontecimientos en un intervalo de tiempo dado se distribuye como una Poisson.

0.1.25 Distribución Beta

Esta distribución se utiliza para representar variables físicas cuyos valores se encuentran restringidos a un intervalo de longitud finita y para encontrar ciertas cantidades que son conocidas como límites de tolerancia sin necesidad de la hipótesis de una distribución normal.

Se dice que una variable aleatoria X posee una distribución beta si su función de densidad está dada por:

$$f(x;\alpha,\beta)=\begin{cases}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} & 0<x<1,\alpha,\beta>0\\ 0 & \text{e.o.c.}\end{cases}$$

Las cantidades α,β son ambas parámetros de perfil.

Está distribución proviene de su asociación con la función beta que se encuentra definida como $B(\alpha,\beta)=\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx$ y que se encuentra relacionada con la función gama por la expresión $B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

La función de distribución acumulativa se encuentra dada por:

$$P(X\leq x)=F(x;\alpha,\beta)=\begin{cases}0 & x\leq 0\\ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt & 0<x<1\\ 0 & x\geq 1\end{cases}$$

donde la integral $B_x(\alpha,\beta)=\int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt$ es la función Beta incompleta. De esta forma, la función de distribución beta, puede expresarse como un cociente de funciones beta incompletas es

$$F(x;\alpha,\beta)=B_x(\alpha,\beta)/B(\alpha,\beta)=I_x(\alpha,\beta)\qquad 0<x<1$$

Algunas áreas en la que se emplea la distribución beta incluyen la distribución de artículos defectuosos sobre un intervalo de tiempo específico.

Si X tiene una distribución Beta, entonces su esperanza y varianza son

$E(X)=\frac{\alpha}{\alpha+\beta}$ y $Var(X)=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$,

respectivamente.

0.1.26 Distribución Chi-cuadrada

Esta distribución es un caso especial muy importante de la distribución gama,se obtiene si $\alpha=1/2$ y $r=n/2$ donde n es un entero positivo.

Se obtiene así una familia de distribuciones de un parámetro con función de densidad dada por:

$$f(z)=\begin{cases}\frac{1}{2^{n/2}\Gamma(n/2)}x^{(n/2)-1}e^{-x/2} & x>0\\ 0 & \text{e.o.c.}\end{cases}\qquad (b)$$

Si una variable aleatoria X tiene su función de densidad dada por (b), se dice que tiene una distribución χ-cuadrada con n grados de libertad y se denota por χ^2_n.

Si una variable aleatoria tiene función de distribución $\chi-$ cuadrada entonces su esperanza y varianza son $E(X) = n$ y $Var(X) = 2n$, respectivamente.

Cuando la distribución χ-cuadrada tiene 2 grados de libertad se obtiene una distribución exponencial con parámetro $1/2$ o, equivalentemente, una distribución exponencial con media igual a 2.

0.1.27 Distribución Cauchy

Supongamos que una variable aleatoria X tiene una distribución continua para la cual su función de densidad está dada por

$$f(x) = \beta/\pi\left[\beta^2 + (x-\alpha)^2\right] \qquad \text{para} \quad -\infty < x < \infty \qquad (1)$$

en donde $\alpha > 0$ y $\beta > 0$. Dicha distribución es llamada la distribución Cauchy

La función de densidad especificada por (1) es una función simétrica alrededor de $x = 0$, por eso, si la media de la distribución Cauchy existe, este valor debería ser cero, sin embargo $\int_{-\infty}^{\infty} | x | f(x)dx \geq \beta/\pi \int_{\alpha}^{\infty} \frac{x}{\beta^2+(x-\alpha)^2}dx \geq \beta/\pi \int_{\alpha}^{\infty} \frac{x-\alpha}{\beta^2+(x-\alpha)^2}dx$
$= \beta/\pi \int_{0}^{\infty} \frac{y}{\beta^2+y^2}dy = \infty$

Por lo tanto la media de la distribución Cauchy no existe, porque la integral no converge.

0.1.28 Distribución t-student

Consideremos dos variables aleatorias independientes Y y Z tales que Y tiene una distribución Normal estándar y Z tiene una distribución $\chi-$cuadrada con n grados de libertad.

Supongamos que una variable aleatoria X está definida por

$$X = Y/(Z/n)^{1/2}$$

La función de densidad de X está entonces dada por

$$p(t) = \frac{\Gamma[(n+1)/2]}{\sqrt{\pi n}\Gamma(n/2)}\left[1 + \frac{t^2}{n}\right]^{-(n+1)/2}$$

Se dice que una variable aleatoria X de este tipo tiene una distribución t-student con n grados de libertad.

Obsérvese que si $n = 1$ se obtiene una distribución de Cauchy.

Si la variable aleatoria X se distribuye como una t-student, entonces su esperanza y varianza están dadas por $E[X] = \mu_t = 0$ para $n \leq 2$ y $E[(X-\mu)^2] = \sigma_t^2 = \frac{n}{n-2}$ para $n \geq 2$

0.1.29 Distribución F

La distribución F tiene mucha importancia en problemas de prueba de hipótesis en la cual dos o más distribuciones normales son comparadas en base a las muestras aleatorias.

Consideremos 2 variables aleatorias Y y Z tales que Y tiene una distribución χ-cuadrada con m grados de libertad y Z tiene una distribución χ-cuadrada con n grados de libertad, donde m y n son enteros positivos. Se define una nueva variable aleatoria X como sigue

$$X = \frac{Y/m}{Z/n} = \frac{nY}{mZ}$$

entonces a la distribución de X se le llama la distribución F con m y n grados de libertad

Si la variable aleatoria X tiene una distribución F con m y n grados de libertad, entonces su función de densidad $f(x)$ está dada por

$$f(x) = \frac{\Gamma[1/2(m+n)]m^{m/2}n^{n/2}}{\Gamma(\frac{1}{2}m)\Gamma(\frac{1}{2}n)} \cdot \frac{x^{(m/2)-1}}{(mx+n)^{(m+n)/2}}$$

Si X es una variable aleatoria con distribución F con m y n grados de libertad entonces su esperanza y su varianza están dadas por

$$E[X] = \frac{n}{n-2} \quad \text{para } n > 2 \quad \text{y} \quad Var[X] = \frac{2n^2(m+n-2)}{m(n-2)^2(n-4)} \quad \text{para } n > 4$$

A continuación se mencionan algunos conceptos básicos sobre estadística, los cuales serán utilizados en capítulos posteriores.

0.1.30 Conceptos Básicos de Estadística

Como parte de los conceptos básicos que nos serviran en el transcurso de nuestro trabajo mencionaremos los que se relacionan con la parte de estadística; asi como las pruebas estadísticas que nos serviran para comprobar la aleatoriedad de nuestros números generados

0.1.31 Estadística

La estadística es el estudio de los fenómenos aleatorios, así como una metodología que ha sido desarrollada para interpretar y extraer concluciones de los datos.

La metodología desarrollada para pasar de observaciones de la muestra a afirmaciones acerca de la población es llamada *Inferencia Estadística*

0.1.32 Población

Es la colección de toda la posible información que caracteriza a un fenómeno, o también se puede decir que es cualquier colección, ya sea de un número finito de mediciones o una colección virtualmente infinita de datos acerca de algo de interés,es decir la parte más representativa para la investigación.

0.1.33 Muestra

Es un subconjunto representativo el cual es seleccionado de una población.

En la estadística la inferencia es inductiva porque se proyecta de lo específico(muestra) hacia lo general(población).

La información seleccionada puede ser agrupada en clases, esto con el fin de poder identificar de alguna forma los patrones relevantes en un conjunto de datos.

Al número de observaciones en una clase se le da el nombre de *frecuencia de clase* y al cociente de una frecuencia de clase con respecto al número combinado de observaciones en todas las clases se conoce como *frecuencia relativa* de esa clase.

Después de obtener nuestra muestra se buscará la forma de decidir la mejor opción para poder dar un resultado al experimento, por esto a continuación se define la forma que nos ayudará a llegar al mejor resultado.

Definiremos primeramente lo que se entiende por rechazarr o aceptar cierta hipótesis H_0

Una prueba de hipótesis estadística con respecto a alguna característica desconocida de la población de interés es cualquier regla para decidir si se rechaza la hipótesis con base a una muestra aleatoria de la población.

A la probabilidad de rechazar H_0, dado que H_0 es cierta se le define como la probabilidad (o tamaño) del Error tipo I y se le denota por α, $0 \leq \alpha \leq 1$.

A la probabilidad de no rechazar H_0, dada que H_0 es falsa se le define como la probabilidad (o tamaño) del error tipo II y se le denota por β, $0 \leq \beta \leq 1$.

Por lo tanto, las probabilidades de los errores de tipo I y tipo II están dadas por las proposiciones:

P(rechazar H_0 | H_0 es cierta)= α, y P(no rechazar H_0 | H_0 es falsa)= β .

0.1.34 Bondad de Ajuste

Una prueba de bondad de ajuste se emplea para decidir si un conjunto de datos se apega a una distribución de probabilidad dada. Es decir, las pruebas de bondad de ajuste checan la consistencia entre un conjunto de datos y un modelo propuesto.

En general el procedimiento de bondad de ajuste es usado para decidir si la población tiene cierta distribución conocida.

Varias de las pruebas presentadas en este trabajo son pruebas de ajuste. Por ejemplo la prueba Chi-Cuadrada y de Kormogorov-Smirlov.

Una manera de saber si cierta población se distribuye de forma normal, binomial, exponencial,uniforme,etc.,es por medio de hipótesis. Es decir enunciando la hipótesis de nulidad (H_0) .La hipótesis de nulidad es una hipótesis de diferencias nulas. Es formulada por lo común con la intención expresa de ser rechazada. Si se rechaza, puede aceptarse la hipótesis alterna (H_1). La hipótesis alterna es la afirmación operacional de la hipótesis de investigación del experimentador.

Con esta comparación entre hipótesis, se puede determinar si la muestra sometida a estudio se distribuye de manera tal que pueda ser aceptada.

0.1.35 PRUEBAS ESTADÍSTICAS.

Acontinuación mencionaremos algunas pruebas estadísticas que son usadas para probar aleatoriedad de los números simulados por nuestro generador.

Estas pruebas no ayudarán a ver la consistencia de aleatoriedad de los números generados

0.1.36 Prueba de Promedios

Para fundamentar esta prueba mencionaremos el siguiente teorema, el cual es una consecuencia inmediata del teorema del límite central.

Teorema

Sea n un entero positivo y $U_1, \ldots, U_n$, variables aleatorias independientes e idénticamente distribuídas, todas con distribución uniforme en el intervalo $(0, 1)$, entonces, cuando n es grande, la variable aleatoria $Z_n = \frac{(\bar{x}-1/2)\sqrt{n}}{\sqrt{1/2}}$ tiene aproximadamente una distribución normal estándar, en donde $\bar{x}$ es la media muestral.

En base a este teorema es posible plantear una prueba de hipótesis, con la cual se trata de probar que los números pseudoaleatorios generados provienen de una distribución uniforme con media $1/2$ planteada de la siguiente forma:

$H_0 : u = 1/2$

$H_1 : u \neq 1/2$

Esta prueba necesita una muestra de tamaño N (N números pseudoaleatorios), y su promedio aritmético es evaluado de acuerdo a la siguiente expresión: $\bar{x} = \frac{U_1+U_2+...U_n}{N}$,en seguida se determina el valor del estadístico $Z_0 = \frac{(\bar{x}-1/2)\sqrt{N}}{\sqrt{1/12}}$. Dado un nivel de significancia α para la prueba, se determina la región de rechazo $\left\{|z| > z_{\alpha/2}\right\}$ en base a la distribución normal estándar. De esta manera, si $|Z_0| < z_{\alpha/2}$ entonces no se puede rechazar la hipótesis y por lo tanto diremos que los números pseudoaleatorios generados provienen de una distribución uniforme con media $1/2$.

0.1.37 Prueba de Frecuencias o Chi-cuadrada

La prueba chi-cuadrada con bondad de ajuste fué propuesta por Pearson en 1900, y es quizá la más conocida de las pruebas estadísticas. Es, además, una de las pruebas de más importancia sobre aleatoriedad de los números pseudoaleatorios.

La prueba χ^2 es adecuada para analizar datos referente a lo observado y lo esperado. Esta técnica es del tipo de bondad de ajuste, ya que se usa para probar una diferencia significativa entre un número observado de objetos de cada categoría y un número esperado, basado en la hipótesis de nulidad.

Sea $X_1, \ldots, X_N$ una muestra de una población con función de distribución acumulativa desconocida $F_x(x)$.

Consideremos una hipótesis nula del tipo

$$H_0 : F_x(x) = F_0(x) \qquad \text{para toda} \qquad x,$$

contra la alternativa

$$H_1 : F_x(x) \neq F_0(x) \qquad \text{para alguna} \qquad x$$

donde $F_0(x)$ es una función de distribución acumulativa completamente especificada. Se asume además que las N observaciones han sido agrupadas dentro de k categorías mutuamente exclusivas. Denotemos por O_i y E_i, el número observado de resultados y el número esperado para la i–ésima categoría , $i = 1, 2, \ldots, k$, respectivamente, cuando H_0 es verdadera.

La hipótesis de nulidad puede probarse mediante el estadístico

$Y = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$,

el cual tiende a ser pequeño cuando H_0 es verdadera y grande cuando H_0 es falsa.

Para muestras grandes Y tiene una distribución que es aproximadamente una chi-cuadrada con $k-1$ grados de libertad.

Bajo la hipótesis de H_0 se espera que

$P(Y > X^2_{1-\alpha}) = \alpha$,

donde α es el nivel de significancia; el cuantil $X^2_{1-\alpha}$ corresponde a la probabilidad $1-\alpha$, que está dada en la tabla de la distribución chi-cuadrada.

Para probar aleatoriedad, se divide el intervalo $(0, 1)$ en n subintervalos para luego ser comparada la frecuencia esperada con la frecuencia observada.Si estas frecuencias son bastante parecidas entonces la muestra proviene de una distribución uniforme.

El estadístico de esta prueba está dado por:

$$X_0^2 = \sum_{i=1}^{n} \frac{(FO_i - FE_i)^2}{FE_i}, \text{ donde:}$$

FO_i =Frecuencia observada del i–ésimo subintervalo

FE_i =Frecuencia esperada del i – ésimo subintervalo (N/n)

N =Tamaño de la muestra

n = Número de subintervalos

Como X_0^2 tiende a tener una distribución chi-cuadrada con $n-1$ grados de libertad, entonces, dado un nivel de significancia α para la prueba, se puede obtener una región de rechazo $\{x > x_\alpha\}$ en base a una distribución chi-cuadrada con $n-1$ grados de libertad. De esta manera, si $X_0^2 < x_\alpha$, no se puede rechazar la hipótesis de que la muestra proviene de una distribución uniforme.

0.1.38 Prueba de Kolmogorov-Smirnov

Otra prueba muy conocida en estadística es la propuesta por Kolmogorov y desarrollada por Smirnov, la cual consiste en lo siguiente.

Sea $X_1, \ldots, X_N$ una muestra aleatoria de una población con función de distribución acumulativa desconocida $F_X(x)$. La función de distribución acumulativa de la muestra, denotada por $F_N(x)$, está definida como

$$\begin{aligned} F_N(x) &= \tfrac{1}{N} \text{ (número de } X_i \text{ menores o iguales a } x) \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^{N} I_{(-\infty,x]}(X_i) \end{aligned}$$

donde $I_{(-\infty,x]}$ es la función indicadora del intervalo $(-\infty, x]$, es decir,

$$I_{(-\infty,x]}(X) = \begin{cases} 1 & si\ X \le x \\ 0 & e.o.c. \end{cases}$$

Para una x dada, $F_N(x)$ es una variable aleatoria, es decir una función de la muestra.

Si $V_i = I_{(-\infty,x]}(X_i)$; entonces V_i tiene una distribución bernoulli con parámetro $P(V_i = 1) = P(X_i \le x) = F_X(x)$. Por lo tanto, $\sum_{i=1}^{N} V_i$ tiene una distribución binomial con parámetros N y $F_X(x)$, y como $F_N(x) = (1/N)\sum_{i=1}^{N} V_i$, se sigue que

$$P\left[F_N(x) = \tfrac{k}{N}\right] = \binom{N}{K} [F_X(x)]^k [1 - F_X(x)]^{N-k}$$

$E[F_N(x)] = \sum_{k=0}^{N} \frac{k}{N}\binom{N}{k}[F_x(x)]^k[1 - F_X(x)]^{N-k} = F_X(x)$

$Var\ F_N(x) = \frac{1}{N}F_X(x)[1 - F_X(x)]$

Las ecuaciones anteriores muestran que, para una x dada, $F_N(x)$ es un estimador insesgado de $F_X(x)$. Por otra parte, debido a que la distribución de $F_N(x)$ es binomial, del teorema del límite central se sigue que $F_N(x)$ tiene una distribución asintóticamente normal, con media $F_X(x)$ y varianza $(1/N)F_X(x)[1 - F_X(x)]$.

Por otra parte, el teorema de *Glivenko-Cantelli establece que*

$P\left[\lim_{N\to\infty}\left(\sup_{-\infty<x<\infty} \mid F_N(x) - F_X(x) \mid\right) = 0\right] = 1$

Es decir, con probabilidad 1, la función $F_N(x)$ converge uniformemente a la función de distribución $F_X(x)$. Esto es para N grande, la desviación $\mid F_N(x) - F_X(x) \mid$ entre la verdadera función $F_X(x)$ y la imagen estadística $F_N(x)$ es pequeña para todos los valores de x.

A la cantidad aleatoria

$D_N = \sup_{-\infty<x<\infty} \mid F_N(x) - F_X(x) \mid$,

se le llama la *estadística para una muestra de Kolmogorov-Smirnov*

La prueba lleva consigo la especificación de la función de distribución acumulativa que ocurriría bajo la distribución teórica y su comparación con la distribución de frecuencia acumulativa observada. Se determina entonces el punto en que las dos distribuciones, la teórica y la observada, muestran la mayor divergencia y ese es el valor de D_N.

Dado un nivel de significancia α para la prueba, la región de rechazo está dada por $\{d \geq d_\alpha\}$, en donde d_α es el más grande número real tal que, asumiendo que H_0 es verdadera, $P[D_N \geq d_\alpha] \leq \alpha$.

La prueba de Kolmogorov-Smirnov trata las observaciones separadamente y así, a diferencia de la prueba chi-cuadrada, no se pierde información al combinar categorías.

Con esta prueba finalizamos lo referente a las pruebas estadísticas, que son usadas para probar aletoriedad y uniformidad.

Especificamente usaremos la prueba chi-cuadrada, para probar uniformidad en nuestro generador de números pseudoaleatorios y de los métodos de generación.

0.2 GENERACIÓN DE NÚMEROS PSEUDOALEATORIOS

En este capítulo hablaremos acerca de lo que se entiende por simulación, en base a qué se utiliza la simulación y cúales son sus ventajas y desventajas, también mencionaremos lo que se entiende por modelos, generadores de números pseudoaletorios; y elegiremos un generador de números pseudoaletorios que nos servirá para aplicar las distintas técnicas de simulación a las distribuciones de probabilidad.

0.2.1 Simulación

La *simulación* es el proceso de desarrollar un modelo simple de un sistema complejo ya que dicho modelo es utilizado para analizar y predecir el comportamiento del sistema original. Se utiliza simulación ya que en general los sistemas de la vida real son complejos y a veces difíciles o imposibles de ser resueltos en su totalidad.

En la simulación debe buscarse que los elementos que forman parte del sistema sean relevantes, con esto nos referimos a que dichos elementos deben ser parte del objetivo principal del sistema, por otra parte se debe también tener cierta precaución para ignorar aquellos elementos que no formen parte primordial del objetivo (aunque la tarea no sea nada fácil debido a la complejidad del problema).

Para saber que se debe simular es necesario tener una definición exacta del sistema , para esto se tiene que hacer un análisis preliminar con el fin de determinar la interacción del sistema con otros sistemas y tomar en cuenta las restricciones y las variables que interactúan dentro del mismo y en base a esto definir los resultados que se desean obtener.

Las técnicas de simulación fueron primeramente desarrolladas en la física y la ingeniería y más tarde se desarrollaron y perfeccionaron por la industria militarizada y aeroespacial.

Los distintos campos de aplicación de la simulación son por ejemplo; los procesos de fabricación de semiconductores, los sistemas de distribución de energía eléctrica, sistemas nucleares, etc..Para este tipo de sistemas se supone un modelo matemático que consiste de ecuaciones que relacionan variables internas del sistema con otras variables que interactuan fuera del sistema.

Estos modelos deben de describir de una manera muy clara la idea original, con el fin de ser programados en una computadora, ya que como veremos más adelante es de gran uso la implementación de éstos, debido a la eficacia y rapidez con que son hechos los cálculos

0.2.2 Elementos de la simulación

Debido a lo anterior la simulación es una poderosa herramienta para el análisis y diseño de sistemas complejos.

Los elementos básicos del proceso de simulación son

- *El sistema real.* Se refiere a la fuente de los datos observados.Es decir que se debe hacer un análisis preliminar del sistema con otros sistemas y sus interrelaciones para definir y estudiar los resultados que se esperan obtener.
- *Formar el modelo,* Este elemento describe un conjunto límite de circunstancias bajo las cuales el sistema real es observado o probado, esto es, se debe formular un modelo y definir todas las variables que forman parte de él.
- *La Colección de los Datos.*Este elemento es la explicación hipotetica y completa de el sistema, por consiguiente, es muy importante que se definan con claridad y exactitud los datos que el modelo va a requerir para producir los resultados deseados.
- *La computadora.* Ésta es una herramienta que es de gran ayuda para estudiar el comportamiento de las partes del modelo con gran rapidez .Es decir ya que se tiene el modelo definido, se puede decidir entre la utilización de un lenguaje de computación como Pascal, Fortran, C, etc. para ser implantados en la computadora o un lenguaje especialmente hecho para la simulación como GPSS,SIMULA, etc.

- *El algoritmo.* Es el elemento que nos ayuda a describir parte por parte la ecuación del modelo para ser calculada mediante la computadora. La ejecución del algoritmo simula el comportamiento del sistema.
- *La Validación.* A través de esta etapa es posible detallar deficiencias en la formulación del modelo o en los datos que formaron parte del modelo.

0.2.3 Usos y propósitos de la simulación

La simulación puede ser usada en las siguientes áreas

1. En situaciones en las cuales las técnicas estadísticas y matemáticas usan métodos analíticos que fallan porque el problema es complejo y el usar otras técnicas como la simulación podrián ayudar a derivar el comportamiento del sistema.

2. En la situación en la cual es imposible obtener una experiencia práctica, debido a que el fenómeno a considerar no ha ocurrido aun, por ejemplo el diseño de un nuevo aeroplano,el diseño de supercomputadoras,equipo médico,etc.

3. En un proyecto de inversión; En esté caso es recomendable usar simulación debido a que existen en la práctica una cantidad muy grande de proyectos de inversión, donde la incertidumbre con respecto a los flujos de efectivo que el proyecto genera, a las distintas tasas de interés a las distintas tasas de inflación, etc. hacen difícil o a veces aun imposible analizarlo en forma analítica, es decir por medio de cálculos numéricos exactos.

La simulación tiene los siguientes propósitos

1. Predecir las consecuencias de cambios en política, condiciones o métodos; por ejemplo el cambio de tipo de motor de una linea de automóviles, la alteración de la información cuando la fuente cambia de exclusivamente periódico a exclusivamente radio.

2. El aprendizaje de nuevos sistemas en el orden de rediseñar o de redefinir éstos, por ejemplo la alta definición en TV, estaciones espaciales, etc.

3. Verificar o demostrar una nueva proposición de comportamiento con el sistema, por ejemplo una nueva estructura en los datos para un programa ya existente.

4. Proyectar el comportamiento futuro de un sistema, por ejemplo el efecto de tasas de interés en las casas de cambio.

Dificultades para usar simulación

Estas dificultades se pueden clasificar como limitaciones inherentes.

Limitaciones inherentes de simulación.

Una limitación inherente se da debido a que los datos son obtenidos de un conjunto límite, obtenido por inducción. Para reducir estas limitaciones se usan pruebas estadísticas, análisis de sensibilidad, etc.

Una segunda limitación inherente es que los métodos de simulación por si solos no pueden resolver sistemas complejos. Por esto la gran ventaja de usar simulación es debido a los algoritmos implantados en computadora.

Una tercer limitante es por ejemplo cuando se da una sola respuesta a un problema, cuando éste tiene más de una o cuando se da una respuesta cuando no la hay.

Además también existe la opinión de que los datos apropiados no existen, aunque esto no sea del todo entendido debido a que los datos son generados por el propio investigador, y se hacen los cálculos necesarios para que se acoplen a las necesidades propias.

También el conocimiento acerca de varias técnicas estadísticas y matemáticas en la evaluación de la simulación es inadecuada

Resumiendo esto de la manera más general posible, diremos que para modelar un sistema, se debe tener un conocimiento muy amplio del modelo a simular debido a que no existe una receta que nos diga cuales son los pasos que nos ayuden a resolver nuestro problema de la manera más adecuada.

Por todo lo anterior la simulación es una herramienta muy poderosa y de mucha ayuda ya que con la construcción de modelos se puede ahorrar mucho tiempo en los cálculos y en el costo.

Después de la creación de un modelo de simulación, que represente nuestro problema lo más real posible se debe de validar, como ya se a mencionado en las etapas de la simulación, las formas más comunes de validar un modelo son:

La opinión de expertos sobre los resultados de simulación.

La comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real.

La aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el experimento de simulación.

Ya que hemos hablado tanto de la importancia en la creación y elaboración de un modelo diremos a continuación que se entiende por modelo, los distintos tipos de modelos y el tipo de modelo que se usara para la elaboración de nuestro trabajo en simulación.

0.2.4 MODELOS

Un *modelo* es la representación simplificada de un sistema.

Los modelos, como ya hemos visto, son de gran utilidad, ya que pueden representar en una forma simplificada cualquier problema de la vida real.

El modelo se hace ya que se tengan definidas las variables y los resultados que se esperan.

De acuerdo a la forma en que ha sido creado el modelo se debe de someter a pruebas estadísticas muy rigurosas, esto con el fin de garantizar cierto nivel de seguridad en la inferencia que pueda hacerse.

El modelo debe ser construído del tal forma que represente el comportamiento del sistema,en base a variables matemáticas o lógicas.

Se deben coleccionar datos que nos ayuden a la formulación del modelo y hacer una extracción de los datos relevantes de nuestro sistema, e implementar el modelo mediante el empleo de algoritmos numéricos, que pueden repetirse exactamente en lenguaje de programación y con esto obtener con más rapidez los resultados deseados.

Ya que se obtuvieron los resultados, se deben de interpretar para poder tomar una decisión que nos lleve a la respuesta exacta de nuestro problema.

Los modelos se pueden clasificar de la siguiente forma:

De acuerdo a las variables de interés, las cuales pueden ser continuas, discretas, combinadas y determinísticas o estocásticas.

Está clasificacion se basa en el comportamiento de la variable dependiente del modelo; es decir por ejemplo, si una variable puede tomar un valor dentro de un rango específico, el modelo es *continuo*; por ejemplo cualquier objeto que esté suspendido es contiuno porque la posición del objeto(variable dependiente) varía continuamente sobre el tiempo.

Si la variable dependiente puede tomar solo un número finito o infinito numerable de valores específicos, entoces se dice que el modelo es *discreto*.

Si la variable dependiente se encuentra en un modelo continuo y en uno discreto se dice que el modelo es *combinado*.

Si la variable dependiente es completamente determinada por la condición inicial, estados del modelo, y salida(s) para el modelo se dice entonces que el modelo es *determinístico*.

En el caso de que un modelo sea *estocástico*, las variables dependientes no pueden ser evaluadas con base al conocimiento de las condiciones iniciales, los estados y salidas del modelo.

Cabe destacar que cuando se trabaja con un sistema real continuo, el modelo a desarrollar no necesariamente es continuo y similarmente cuando el sistema es discreto, el modelo no necesariamente debe de ser discreto y así sucesivamente.

Un modelo *cieutífico* es aquel que proporciona de alguna forma el análisis para determinar como uno o más cambios en algunos aspectos del sistema modelado puede afectar otros aspectos del sistema o al sistema como un todo, esto es que si se propone hacer algun cambio debe no solo de pensarse en alguna parte del problema, sino que también se debe ver en que forma o como afecta al problema original ese cambio.

Debido a esto se debe de construir una función objetivo y en base a ésta trabajar todos los cambios y modificaciones del modelo, esta función objetivo debe ser una función matemática que represente las caracteristicas del problema con variables de decisión.

Un modelo *simbólico* requiere operaciones lógicas o matemáticas que puedan ser usadas para formular una solución al problema dado.

Dada esta clasificación se deben de contruir modelos matemáticos porque ofrecen o describen en forma más general el problema a modelar.

Como siguiente paso, después de haber construído el tipo de modelo que refleje de alguna manera la idea más cercana a la realidad, es resolverlo, y tenemos entonces que los métodos de solución más practicos o conocidos son analíticos y numéricos.

Los métodos de solución *analítica* se obtienen de su representación matemática en forma de fórmula.

Los métodos de solución *numérica* son una aproximación, obtenida como resultado de la sustitución de las variables y parámetros del modelo, por valores numéricos. Muchos métodos numéricos son recursivos, es decir que se necesita de un resultado previo para obtener el siguiente.

Uno de los métodos más conocidos y que trabaja de forma recursiva es el método de Monte-Carlo, esté método fué usado en la segunda guerra mundial (bomba atómica) en una simulación directa concerniente a la difusión de neutrones y está basado en la generación de números aleatorios.

Los problemas considerados por el método de Monte-Carlo son determinísticos y probabilísticos . El aprovechamiento más simple del metódo de Monte-Carlo es la observación de números aleatorios y en base a la simulación se eligen de alguna manera estos números para inferir la solución deseada.

Esté metódo es sin lugar a duda una de los más importantes en el uso de la simulación. existen muchos estudios relacionados con éste.

Diremos además que en las últimas décadas se han encontrado usos extensivos en base a los números aleatorios; en investigación de operaciones,en experimentos fisiconucleares, en biología, etc.

En el siguiente tema definiremos más ampliamente que se entiende por números pseudoaleatorios, cuál es su uso, cuál es la forma de generación de éstos y por qué su creación..

0.2.5 NÚMEROS PSEUDOALEATORIOS

Bajo un método matemático un número es definido como *pseudoaleatorio* debido a que éste no es realmente aleatorio, ya que es generado mediante relaciones de recurrencia que son determinísticas.

Sin embargo, esta objeción puede superarse, al menos parcialmente, al tomar el punto de vista un tanto pragmático de que una sucesión puede considerarse aleatoria si satisface un cierto conjunto de pruebas estadísticas de aleatoriedad.

El objetivo de crear este tipo de números, es porque pueden ser usados como variables aleatorias y dar la prueba de laboratorio que necesitamos, para poder definir los resultados esperados de cierto modelo matemático.

La generación de números pseudoaleatorios se da por medio de una distribución uniforme estándar denotada por $U(0,1)$ y bajo esta distribución, otras distribuciones de probabilidad pueden obtenerse bajo tranformaciones, que más adelante explicaremos.

Una de las muchas técnicas que ha sido usada en años recientes para generar números pseudoaleatorios es la siguiente.

La técnica que fué propuesta por John Von Neumann llamada método de Mínimos Cuadrados que principalmente consiste en:

- Elegir un número de n dígitos (n par) a este número se le llama semilla y a partir de éste se generan números pseudoaleatorios de n dígitos en forma recursiva.
- Se eleva al cuadrado y al número resultante se le agrega ceros a la izquierda, si es que hace falta para completar $2n$ dígitos.
- El siguiente número será formado por los n dígitos centrales del número obtenido en el paso anterior. Y así sucesivamente hasta obtener la cantidad deseada.

Es decir por ejemplo sea 3979 la semilla, obtener una sucesión de 10 números pseudoaleatorios:

1.- $(3979)^2 = 15832441$

2.- $(8324)^2 = 69288976$

3.- $(2889)^2 = 08346321$

....

10.- $(0001)^2 = 00000001$

Con esta semilla se obtiene un período de 10 a partir del cual el resultado es cero, a esto llamamos una sucesión determinística y que además aparenta ser aleatoria.

Debido a que los números generados por este método tienen un período muy corto, el método de mínimos cuadrados ha caído en desuso.

En base a esto, y debido a la forma en que se rechaza este método para generar números pseudoaleatorios, se plantea que para la generación de estos números se deben garantizar:

1.- Que estén uniformente distribuídos

2.- Que sean estadísticamente independientes

3.- Que sean reproducibles

4.- Que tengan período largo(sin repetición dentro de una longitud determinada de la sucesión.).

Ya que hemos hablado de las características que deben tener los números pseudoaleatorios, como siguente tema definiremos que es un generador y cómo se debe elegir el generador para que cumpla con los requisitos mencionados.

0.2.6 TIPOS DE GENERADORES CONGRUENCIALES.

Los siguientes conceptos nos serán de gran utilidad en el desarrollo de este capítulo.

Entenderemos como congruencia la siguiente definición:

para cada entero $p > 1$ y para toda a y b en Z, se define a congruente con b módulo p y se escribe $a \equiv b(mod\ p)$, si

$a - b$ es divisible por p, esto es existe $q \in Z$ tal que $a = qp + b$,

en particular $a \equiv 0(mod\ p)$ si para alguna $q \in Z$, $a = qp$.

Empezaremos por definir algunas técnicas básicas para generar números pseudoaleatorios bajo una distribución uniformes $U(0,1)$.

Las técnicas de las que hablamos son; los *métodos congruenciales*. En este caso, enteros aleatorios, sobre un rango conocido son primero generados , para después ser utilizados en las técnicas de simulación.

0.2.7 Generadores Congruenciales Lineales.

Los métodos más conocidos fueron sugeridos por Lehmer y estos son; congruencial mixto y congruencial multiplicativo.

0.2.8 Congruencial Mixto

Con los generadores congruenciales lineales mixtos se obtiene una sucesión de números pseudoaleatorios, la secuencia congruencial puede tomar formas diferentes, pero el método más comunmente usado está definido de la siguiente forma.

El siguiente número pseudoaleatorio es determinado a partir del último número generado, es decir el número pseudoaleatorio X_{n+1} es derivado del número pseudoaleatorio X_n donde la relación de recurrencia es la siguiente:

$$X_{i+1} = (aX_i + c) \; mod \quad M \qquad (1)$$

donde los números X_i, a, c y M son enteros no negativos y $0 \leq X_i < M$

a es el multiplicativo ($a > 0$)

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

M es el módulo ($M > X_0, M > a$ y $M > c$),

Está relación de recurrencia nos dice que X_{n+1} es el residuo de dividir aX_n+c entre el módulo, esto significa que los valores posibles de X_{n+1} son $\{0, 1, 2, \ldots, M-1\}$,

donde M representa el número posible de valores diferentes que pueden ser generados ya que los enteros producidos por la fórmula (1)

están en el intervalo $[0, M]$.

Mediante la transformación $U_i = X_i/M$ se obtienen números en el intervalo $[0, 1)$, los cuales se distribuyen uniformemente aproximadamente.

0.2.9 Congruencial Multiplicativo

Al igual que el generador congruencial mixto, el generador congruencial multiplicativo determina al siguiente número pseudoaleatorio a partir del último número generado, en base a la relación de recurrencia;

$$X_{n+1} = aX_n \; mod \quad M$$

X_0 es la semilla ($X_0 > 0$)

a es la constante de multiplicación

La recomendación principal de estos dos generadores es la elección de la semilla, a,c (en el caso mixto) y M, ya que de estas constantes depende de que el generador tenga o no un período completo, entendiendo por esto que;

El *período* es el tamaño de la longitud de una secuencia de números pseudoaleatorios que se pueden generar antes de que se repita la sucesión.

Sea P el período de una sucesión, se dice que es completo si $P = M$. Para generadores multiplicativos el período máximo es $M - 1$ pues si el cero alguna vez se presenta éste se repetirá indefinidamente.

Los siguientes teoremas nos sugieren las condiciones para conseguir el período máximo, para el generador de (1)

Teorema 1

Un generador congruencial mixto tiene período M si y solo si

i) c es un primo relativo a M

ii) $a \equiv 1 \quad mod \quad p$ para cada factor primo p de M

iii) $a \equiv 1 \quad mod \quad 4$ si 4 es un factor de M

Cabe hacer notar que si M es primo, existe período completo solo si $a = 1$

Teorema 2

Un generador multiplicativo con modulo $M = 2^{\beta} \geq 16$ tiene período máximo $M/4$ si y solamente si $a \equiv 3$ mod 8 o $a \equiv 5$ mod 8. En el caso $a \equiv 5$ *mod* 8,

si $b = X_0$ *mod* 4, entonces

$(U_i - b/M)$ es la secuencia de salida del generador de período completo

$X_i = \{aX_{i-1} + b(a-1)/4\} mod\ M/4$

Aunque para aplicar estos teoremas en la elección de nuestro generador, es necesario en primer lugar tener una computadora muy poderosa(en cuanto a capacidad), ya que es necesario utilizar un módulo bastante grande.

0.2.10 ELECCIÓN DE UN GENERADOR

Está elección debe ser de acuerdo a nuestras necesidades, por ejemplo si limitamos nuestra atención para obtener generadores de un período completo y además buscamos generadores multiplicativos con módulo primo y período máximo, los generadores indicados serían los generadores congruenciales mixto y multiplicativo, ya que estos toman valores regularmente espaciados en el intervalo $[0, 1)$, y cada uno ocurre una vez por ciclo.

Si se proporciona a M un valor suficientemente grande,

la colección de números (U_i) podrá tener una distribución uniforme.

Después de haber mencionado diferentes tipo de generadores, nosotros elegimos en primera instancia como nuestro generador de números pseudoaleatorios, al generador congruencial mixto, porque consideramos que era un buen generador, pero al ser usado en nuestro algorimo y ser implantado en lenguaje de computación (Pascal), resultó no ser el más optimo, primeramente porque no teníamos una computadora lo suficientemente poderosa y en segunda instancia porque el lenguaje de Pascal tiene ciertas restricciones es cuanto al rango de números y el generador congruencial mixto, en algunos cálculos teníamos que poner muchas restricciones para que pudiera ser generado cierto porcentaje de números.

Es por esto que preferimos elegir el generador que nos proporciona Pascal (RANDOM), los números generados también tienen una distribución uniforme en el intervalo $(0, 1)$.

RANDOM nos permite un porcentaje mayor de número generados, y éste es por así decirlo el principal objetivo, para poder dar respuesta a nuestro problema que nos planteamos al inicio de este trabajo.

Después de ser generados los números pseudoaleatorios, el siguiente paso es aplicarles la prueba estadística, que nos ayudara para probar aleatoriedad en nuestros números generados por RANDOM , la prueba que usaremos será la Chi-Cuadrada.

0.3 TÉCNICAS GENERALES PARA SIMULAR DISTRIBUCIONES DE PROBABILIDAD

En primer lugar mencionaremos los métodos generales, es decir los métodos de aceptación-rechazo y el método de la transformación inversa, y como parte de otro capítulo mencionaremos los métodos particulares para la simulación de algunas variables aleatorias, en ambos casos continuas o discretas.

Llamamos particulares a los métodos que solo nos sirven para simular una distribución de probabilidad.

0.3.1 El Método de la Transformación Inversa

Este método se refiere a la simulación de una variable aleatoria continua X con función de distribución acumulativa $F(x)$ i.e. $F(x) = P(X \leq x)$ y está basado en el siguiente resultado.

Proposición.

Sea U una variable aleatoria que se distribuye uniformemente en el intervalo $(0,1)$ y sea F una función de distribución de probabilidad continua y estrictamente creciente, entonces la variable aleatoria $X = F^{-1}(U)$ tiene como función de distribución a F.

Dem.

Si $X = F^{-1}(U)$, entonces $\Pr(X \leq x) = \Pr(F^{-1}(U) \leq x)$

Pero, debido a que $F(x)$ es una función continua monótona y estrictamente creciente de x, podemos escribir

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

Y como U es una variable aleatoria uniforme $U(0,1)$, tenemos que

$\Pr(U \leq F(x)) = F(x)$

Por lo tanto,

$\Pr(X \leq x) = F(x)$.

De esta forma, la variable aleatoria $X = F^{-1}(U)$ tiene la distribución requerida.

Para ejemplificar este método, sea X una variable aleatoria con distribución exponencial de parámetro λ, entonces su función de densidad es

$f(x) = \lambda e^{-\lambda x}$ para $x \geq 0$,

y su función de distribución acumulativa está dada por

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

Por lo tanto, la función inversa de la distribución acumulativa es

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

Así que por la proposición demostrada anteriormente, si U es una variable aleatoria con distribución uniforme en el intervalo $(0,1)$, entonces

$F^{-1}(U) = -\frac{1}{\lambda}\log(1-U)$ tiene una distribución exponencial de parámetro λ. Pero si U es una variable aleatoria uniforme $U(0,1)$ entonces $(1-U)$ también tiene esta distribución por lo tanto la variable aleatoria $X = -\frac{1}{\lambda}\log U$ tiene también una distribución exponencial con parámetro λ.

El método de la transformación inversa es uno de los más utilizados, al menos en nuestro trabajo, debido primero a que la transformación inversa F^{-1} de las distribuciones simuladas no ha sido demasiado dificil de encontrar ya que son muy conocidas, y segundo, porque siempre necesitamos para su simulación exactamente una variable aleatoria $U(0,1)$,que nos da el valor de la X deseada, ya que como veremos en el método siguiente se necesitan varias variables aleatorias distribuidas uniformemente es $U(0,1)$ para poder obtener el valor de X.

En forma general diremos que el principal problema que presenta este método es el evaluar $F^{-1}(u)$, es decir si no nos es posible escribir una fórmula para F^{-1} en forma cerrada, no es posible utilizar este método, como por ejemplo para las distribuciones normal y gama no es posible usar este método ya que como sabemos no existe la fórmula F^{-1} en forma cerrada, aunque para estos casos se dan métodos alternativos para poder simular dichas distribuciones, y que pueden ser mejores y más rápidos en el tiempo de cómputo

0.3.2 Método de Aceptación-Rechazo

Debido a los numerosos avances en los métodos para generar números aleatorios, se menciona un método alternativo al método de la transformación inversa.

Este método es conocido como el método de rechazo(algunas veces llamado de aceptación-rechazo) atribuido a Von Neumann.

Podemos decir que este método es menos directo en su aprovechamiento y puede ser útil cuando los métodos directos fallan o son ineficientes.

A continuación mencionaremos este método, que puede ser como una alternativa o para ser comparado con los resultados, en el tiempo de cómputo principalmente con el método de la transformación inversa.

Supongamos que deseamos simular una variable aleatoria X

Esta variable tiene una función de distribución $F(x)$ y una función de densidad $f(x)$.

El método de aceptación-rechazo requiere de la especificación de una función $h(x)$, tal que $h(x) \geq f(x)$..

Sea X generada de $f_x(x), x \in I$ .Para efectuar este método se representa a $f(x)$ como

$f_x(x) = Ch(x)g(x)$

donde $C \geq 1, h(x)$,es también una función de densidad de probabilidad, y $0 < g(x) \leq 1$.

Se generan dos variables aleatorias U que se distribuye uniformente en $U(0,1)$ y Y como $h(y)$, respectivamente, y la prueba consiste en ver si la desigualdad $U \leq g(Y)$ se cumple de la siguiente manera

1. Si la desigualdad se cumple, entonces aceptamos Y , como una variable generada de $f_x(x)$.

2. Si la desigualdad no se cumple, rechazamos el par U y Y e intentamos de nuevo.

La teoría dada en este método se basa en lo siguiente.

Teorema. Sea X una variable aleatoria distribuida con función de densidad de probabilidad $f_X(x), x \in I$, la cual es representada como

$f_X(x) = Ch(x)g(x),$

donde $C \geq 1, 0 < g(x) \leq 1$, y $h(x)$, es también una función de densidad de probabilidad.

Sea U y Y independientemente distribuidas como $U(0,1)$ y $h(x)$, respectivamente. Entonces

$f_Y(x \mid U \leq g(Y)) = f_X(x).$

Dem.

Como U y Y son independientes, se tiene

$P[U \leq g(Y) \mid Y = x] = P[U \leq g(x)] = g(x)$

Así que

$P[u \leq g(Y)] = \int_{-\infty}^{\infty} P[u \leq g(Y) \mid Y = x]h(x)dx = \int_{-\infty}^{\infty} g(x)h(x)dx = \int_{-\infty}^{\infty} \frac{f_X(x)}{c} = 1/c$

Ahora bien, por la fórmula de Bayes,

$f_Y(x \mid u \leq g(Y)) = \frac{P[u \leq g(Y)|Y=x]h(x)}{P[u \leq g(Y)]} = cg(x)h(x) = f_X(x)$

Concluimos la exposición de este método deciendo que la efeciencia del método de aceptación-rechazo es determinada por la desigualdad $U \leq g(Y)$

Para ejemplificar este método mencionaremos como generar una función de distribución Normal

Para simular una variable aleatoria normal Z (con media 0 y varianza 1), cabe hacer notar que el valor absoluto de Z tiene una función de densidad de probabilidad.

$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2} \qquad 0 < x < \infty$

como se menciona anteriormente se debe de encontrar una función $h(x)$, que en este caso es una función de densidad exponencial con media 1, tal que

$h(x) \geq f(x)$

aplicando el procedimiento tenemos que

$$\frac{f(x)}{h(x)} = \sqrt{2/\pi} \exp\left\{\frac{-(x^2-2x)}{2}\right\} \qquad = \sqrt{2/\pi} \exp\left\{\frac{-(x^2-2x+1)}{2} + \frac{1}{2}\right\}$$
$$= \sqrt{2e/\pi} \exp\left\{\frac{-(x-1)^2}{2}\right\} \leq \sqrt{2e/\pi}$$

si tomamos a $c = \sqrt{2e/\pi}$ tenemos que

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

con esto podemos simular el valor absoluto de una variable aleatoria normal

0.4 TÉCNICAS PARTICULARES PARA GENERAR DISTRIBUCIONES DE PROBABILIDAD

Una técnica que fué desarrollada para simular una distribución normal a partir de una distribución uniforme es la de coordenadas polares.

0.4.1 Simulación de una distribución normal

Esta simulación se hizo de acuerdo al método de coordenadas polares

Este método está basado en la siguiente proposición

Proposición

Sean U y V variables aleatorias independientes, ambas con distribución uniforme en el intervalo $(0,1)$. Definamos

$X = \sqrt{-2\ln U}\cos(2\pi V)$ y

$Y = \sqrt{-2\ln U}sen(2\pi V)$,

entonces X y Y son variables aleatorias independientes, ambas con distribución normal estándar.

Dem:

Sea $Z = -2\ln U$ y sean $F_{X,Y}$ y $f_{X,Y}$ la función de distribución y de densidad conjunta, respectivamente, de X y Y; se tiene entonces,

$F_{X,Y}(x_0, y_0) = P\left[\sqrt{Z}\cos 2\pi U \leq x_0, \sqrt{Z}sen 2\pi U \leq y_0\right]$

$= \iint_{\{(z,u)|\sqrt{z}\cos 2\pi u \leq x_0, \sqrt{z}sen 2\pi u \leq y_0\}} f_Z(z)f_U(u)dzdu$

Haciendo el cambio de variable

$x = \sqrt{z}\cos 2\pi u$,

$y = \sqrt{z}sen 2\pi u$, se obtiene,

$F_{X,Y}(x_0, y_0) = \frac{1}{\pi}\iint_{\{(x,y)|x\leq x_0, y\leq y_0\}} f_Z(x^2+y^2)f_U(\frac{1}{2\pi}\arctan\frac{y}{x})dxdy$

$= \frac{1}{\pi}\iint_{\{(x,y)|x\leq x_0, y\leq y_0\}} f_Z(x^2+y^2)dxdy$

Así que,

$f_{X,Y}(x,y) = \frac{1}{\pi}f_Z(x^2+y^2) = \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}$

Por lo tanto, X y Y son independientes y cada una de ellas tiene distribución normal estándar.

0.4.2 Simulación de una distribución gama

Para simular una distribución Gama con parámetro α igual a un entero positivo, se puede utilizar el hecho de que la suma de n variables aleatorias independientes, todas con distribución exponencial de parámetro λ, tiene una distribución gama con parámetros $\alpha = n$ y λ.

Por lo tanto, si $U_1, \ldots, U_n$ son variables aleatorias independientes, todas con distribución uniforme en el intervalo $(0,1)$, y $\lambda > 0$, entonces, la variable aleatoria

$X = -\sum_{i=1}^{n} \frac{1}{\lambda} \log U_i$

$= -\frac{1}{\lambda} \log(\prod_{i=1}^{n} U_i)$

tiene una distribución gama con parámetros $\alpha = n$ y λ.

0.4.3 Simulación de una distribución Uniforme discreta

La distribución uniforme discreta en el conjunto $\{k, k+1, \ldots, j\}$ se puede obtener directamente de una distribución uniforme en el intervalo $(0,1)$.

En efecto, si U tiene una distribución uniforme en el intervalo $(0,1)$, entonces, para $i = 0, 1, 2, \ldots, j-k$,

$P\left[U \in \left[\frac{i}{j-k+1}, \frac{i+1}{j-k+1}\right)\right] = \frac{1}{j-k+1}$, de manera que si definimos

$X = k + [[(j-k+1)U]]$,

en donde $[[\,]]$ denota a la función mayor entero, entonces X tiene una distribución uniforme en el conjunto $\{k, k+1, \ldots, j\}$

0.4.4 Simulación de una distribución Geométrica

La distribución Geométrica también formó parte del estudio, esta distribución fue calculada en base a su relación con la distribución exponencial.

Sea $0 < p < 1$ y Y una variable aleatoria con distribución exponencial de parámetro $\lambda = -\ln(1-p)$, entonces, si $x \geq 0$, tenemos

$P(x \leq Y < x+1)$

$= \lambda \int_x^{x+1} e^{-\lambda y} dy$

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

$= p(1-p)^x$

De manera que

$X = [[Y]]$ tiene una distribución geométrica de parámetro p.

Por lo tanto, si U tiene una distribución uniforme en el intervalo $(0,1)$, entonces

$X = [[\ln U/\ln(1-p)]]$

tiene una distribución geométrica de parámetro p.

0.4.5 Simulación de una distribución Binomial

Si $U_1,\ldots,U_n$ son variables aleatorias independientes, todas con distribución uniforme en el intervalo $(0,1)$.

Entonces si decimos que hay éxito cuando $U_i < p$, la variable aleatoria

$X = \sum_{i=1}^{n} I_{(0,p)}(U_i)$

representa el número de éxitos en n ensayos, por lo tanto tiene una distribución binomial con parámetros n y p.

0.4.6 Simulación de una distribución Poisson

Una distribución Poisson puede simularse utilizando su relación con la distribución exponencial.

En efecto, consideremos eventos que ocurren aleatoriamente en el tiempo de tal manera, que el número de ocurrencias hasta el tiempo t tiene una distribución Poisson de parámetro λt, en donde λ es el número promedio de ocurrencias por unidad de tiempo.

Sabemos que en esta situación, los tiempos entre ocurrencias sucesivas son independientes y tienen, cada uno de ellos, una distribución exponencial de parámetro λ.

Denotemos por X_t al número de ocurrencias hasta el tiempo t y por $Y_1, Y_2, \ldots$ a los tiempo entre ocurrencias sucesivas, comenzando desde el tiempo 0, en el cual no ha habido ninguna ocurrencia.

Entonces $X_1 = k$ si y solo si

$Y_1 + Y_2 + \cdots + Y_k \leq t$ y

$Y_1 + Y_2 + \cdots + Y_k + Y_{k+1} > t.$

Por lo tanto,

$X_t = \max\left\{i : \sum_{j=1}^{i} Y_j \leq t\right\}$

En particular, la variable aleatoria

$X = \max\left\{i : \sum_{j=1}^{i} Y_j \leq 1\right\}$ tiene una distribución Poisson de parámetro λ.

De esta manera, la distribución Poisson se puede obtener generando distribuciones exponenciales, las cuales a su vez se obtienen generando distribuciones uniformes.

De manera más especifica, se puede concluir lo siguiente:

Sean $U_1, U_2, \ldots$ variables aleatorias independientes, todas con distribución uniforme en el intervalo $(0,1)$,

entonces la variable aleatoria

$$
\begin{aligned}
X &= \max\left\{i : -\tfrac{1}{\lambda}\textstyle\sum_{j=1}^{i} \ln U_j \leq 1\right\} \\
&= \max\left\{i : -\ln \textstyle\prod_{j=1}^{i} U_j \leq \lambda\right\} \\
&= \max\left\{i : \textstyle\prod_{j=1}^{i} U_j \geq e^{-\lambda}\right\} \\
&= \min\left\{i : \textstyle\prod_{j=1}^{i} U_j < e^{-\lambda}\right\} - 1
\end{aligned}
$$

tiene una distribución Poisson de parámetro λ.

En el siguiente capítulo concluiremos en base a los métodos utilizados cual fué el mejor.

Además mencionaremos los algoritmos que utilizamos para la generación de números pseudoaleatorios, mostraremos las gráficas y también se tendrán los resultados obtenidos al aplicar la prueba estadística, y el tiempo de cómputo en la simulación de cada una de las funciones de distribución, con todas estas pruebas veremos en forma más precisa cuando podamos aceptar o no los métodos utilizados, diciendo que nos aproximaron de la mejor manera a la verdadera función de distribución.

0.5 APLICACIÓN DE LA SIMULACIÓN

En este capítulo hablaremos de la forma en que usamos la simulación, los resultados obtenidos al aplicar los distintos métodos y las gráficas de ciertas funciones de distribución, así como el tiempo de cómputo.

Como ya vimos en capítulos anteriores, para la generación de números pseudoaleatorios utilizamos el generador RANDOM de Pascal, con este método generamos los números uniformes que nos sirven para la simulación de ciertas distribuciones de probabilidad, tanto discretas como continuas.

Conforme a estos métodos de simulación se generaron números que nos dieron muy buenas aproximaciones a las distribuciones reales.

La simulación de los métodos mencionados se realizaron tanto en distribuciones continuas como en discretas, comenzamos por mencionar los resultados obtenidos en las distribuciones continuas que formaron parte de nuestro análisis.

0.5.1 La distribución Normal.

Como primer paso la distribución normal fué simulada por el método de coordenadas polares. Los resultados obtenidos al aplicar el método de coordenadas polares se pueden ver en el apéndice **A**.

Dado los resultados, el tiempo de ejecución del programa y la gráfica, podemos decir que este método nos da una buena aproximación a la verdadera distribución.

La simulación de esta distribución se hizo en base al algoritmo siguiente

Algoritmo N

1 Generar U_1, U_2 independientes, con distribución $U(0,1)$

2 Calcular $R^2 = -2\ln(U_1)$ y $\theta = 2\pi(U_2)$

3 Hacer $X = R\cos\theta$

Esta distribución también fué simulada en base al método de rechazo, los resultados se pueden ver en el apendice **A1**

Conforme a los resultados, el tiempo de cómputo y la gráfica de la distribución, concluimos que también es un buen método

El algoritmo en el cual nos basamos para simular dicha distribución es el siguiente

Algoritmo N1

1 Generar V_1 y V_2 independientes, con distribución exp(1)

2 Si $V_2 < (V_1 - 1)^2/2$ regresar al paso 1

3 Generar U, independiente de V_1 y V_2, con distribución $U(0,1)$

4 Si $U \geq 0.5$, entonces $Z = -V_1$

5 Si no, $Z = V_1$

0.5.2 Distribución exponencial

La siguiente función de distribución simulada fué la exponencial, esta distribución fue simulada por el método de transformación inversa.

Los datos obtenidos aplicando el método de tranformación inversa se pueden ver en el apéndice **A2**, así como el tiempo de ejecución y la gráfica de la función simulada.

Dados estos resultados podemos concluir que esté método nos da una buena aproximación a la verdadera función de distribución.

La distribución exponencial fué simulada en base al siguiente algoritmo

Algoritmo E

1 Generar U con distribución $U(0,1)$

2 Hacer $X = -\lambda \ln U$

0.5.3 Distribución Gama

La siguiente distribución que formó parte del estudio fué la distribución gama, obteniéndola como una suma de variables aleatorias independientes, todas con distribución exponencial con el mismo parámetro.

Los datos obtenidos aplicando el método se pueden ver en el apéndice **A3**, así como el tiempo de ejecución y la gráfica de la función simulada.

Dados estos resultados podemos concluir que este método nos da una buena aproximación a la verdadera función de distribución.

La obtención de la distribución gama se basó en el siguiente algoritmo

Algoritmo G

1 Generar $U_1, U_2, \ldots, U_n$ independientes, con distribución $U(0,1)$

2 Hacer $X = -\ln\left(\prod_{i=1}^{n} U_i\right)$

0.5.4 Distribución Cauchy

Otra de las distribuciones que formó parte de nuestro análisis fué la Cauchy.Ésta fue simulada por el método de la transformación inversa.

Sea $F_x(X) = \frac{1}{2} + \pi^{-1}\tan^{-1}\left(\frac{X-\alpha}{\beta}\right)$ la función acumulativa de una Cauchy

Aplicando el método a la función acumulativa tenemos

$X = F_x^{-1}(U) = \alpha + \beta\tan\left[\pi\left(U - \frac{1}{2}\right)\right] = \alpha - \frac{\beta}{\tan(\pi U)}$

La simulación de esta distribución nos dio una buena aproximación a la verdadera distribución. Los datos obtenidos,el tiempo de cómputo y la gráfica de la función se pueden apreciar en el apéndice **A4**.

El algoritmo que sirvió para simular a la distribución Cauchy es el siguiente

Algoritmo C

1 Generar U con distribución $U(0,1)$

2 Hacer $X = \alpha - \frac{\beta}{\tan(\pi U)}$

0.5.5 Distribución Chi- Cuadrada

La distribución Chi-Cuadrada (χ^2), fué simulada con 20 grados de libertad como un caso particular de la densidad Gama, con parámetros $\alpha = 1/2$ y $\beta = 2$, respectivamente.

El algoritmo en el cual nos basamos fué el siguiente

1 Generar números aleatorios con distribución uniforme $U \sim (0,1)$

2 Calcular a $Y = -2 * \ln(\prod_{i=1}^{k/2} Ui)$, donde k son los grados de libertad

3 Y, tiene la distribución deseada

De acuerdo con los resultados obtenidos, podemos decir que este algoritmo nos da una buena aproximación a la verdadera función de distribución

Los resultados se pueden ver en el apéndice **A5**

0.5.6 Distribución t-Student

Esta distribución fué simulada, de acuerdo a la relación que existe con la distribución Normal y la Distribución χ^2.

El algoritmo en el cula nos basamos fué el siguiente

1 Generar a N con distribución Nomal

2 Generar a Y con distribución χ^2

3 Calcular a $X = N/\sqrt{Y/k}$, donde k son los grados de libertad

4 X , tiene la distribución deseada

Los resultados que se pueden ver en el apéndice **A6,** nos dan una buena aproximación a la verdadera función de distribución.

A continuación mencionaremos los métodos aplicados, así como los resultados obtenidos en las distribuciones discretas que formaron parte de nuestro análisis.

0.5.7 Distribución Uniforme discreta

La distribución uniforme discreta en el conjunto $\{k, k+1, \ldots, j\}$, fué simulada directamente a partir de una distribución uniforme en el intervalo $(0, 1)$ como se mencionó anteriormente. Es decir, si U tiene una distribución uniforme en el intervalo $(0, 1)$, entonces $X = k + [[(j - k + 1)U]]$ tiene una distribución uniforme en el conjunto $\{k, k+1, \ldots, j\}$.

El cálculo de X nos dió una buena aproximación a la verdadera función Uniforme.Los resultados,el tiempo de cómputo, así como la gráfica de la función se pueden apreciar en el apéndice **B.**

El algoritmo en que se basó el cálculo de esta función fué el siguiente

Algoritmo U

1 Generar U con distribución $U(0, 1)$

2 Hacer $X = k + [(j - k + 1)U]$

0.5.8 Distribución Geométrica

La distribución Geométrica también formó parte del estudio, esta distribución fue calculada considerando que si U es una variable aleatoria con distribución uniforme en el intervalo $(0, 1)$, entonces $X = [[\ln U / \ln(1-p)]]$ tiene una distribución geométrica de parámetro p.

El cálculo de X nos dio una buena aproximación a la distribución verdadera, los resultados,el tiempo de cómputo y la gráfica se pueden ver en el apéndice **B1**.

El algoritmo en el cual fué basada la generación de dicha distribución es el siguiente

Algoritmo G

1 Generar U con distribución $U(0, 1)$

2 Hacer $X = [[\ln U / \ln(1-p)]]$

0.5.9 Distribución Binomial

La distribución binomial fue simulada considerando que si $U_1, \ldots, U_n$ son variables aleatorias independientes, todas con distribución uniforme en el intervalo $(0, 1)$, entonces la variable aleatoria $X = \sum_{i=1}^{n} I_{(0,p)}(U_i)$ tiene una distribución binomial con parámetros n y p.

Los resultados al aplicar este método se pueden apreciar en el apéndice **B2**

Dados los resultados podemos decir que este método nos dió una buena aproximación a la verdadera distribución.

El algoritmo en el cual nos basamos es el siguiente

Algoritmo Bi

1 Generar $U_1, U_2, \ldots, U_n$ independientes, con distribución $U(0, 1)$

2 Hacer $X = \sum_{i=1}^{n} I_{(0,p)}(U_i)$

3 X se distribuye con distribución geométrica

0.5.10 Distribución Binomial Negativa

La distribución binomial negativa con parámetros (n, p), fué simulada de acuerdo con su relación con la distribución geométrica con parámetro (p).

Es decir si tenemos a $Y_1, Y_2, \ldots Y_n$, variables aleatorias independientes con distribución geométrica, entonces $Y_1 + Y_2 + \ldots Y_n$ se distribuyen como una distribución binomial negativa.

El algoritmo fué el siguiente

Algoritmo Bn

1 Generar $Y_1, Y_2, \ldots Y_n$ variables aleatorias con distribución geométrica

2 Hace a $X = Y_1 + Y_2 + \ldots Y_n$

3 X tiene la distribución deseada.

Los resultados obtenidos, al simular esta distribución nos dan una buena aproximación a la verdadera fución de distribución y se pueden apreciar en el apendice **B3**

0.5.11 Distribución Poisson

La distribución Poisson (λ), fué simulada en relación con la distribución exponencial $(1/\lambda)$

El algoritmo en el cual se basó para dicha distribución es el siguiente

1 Sea $k = e^{-\lambda}$, $b = 1$ y $r = 0$

2 Generamos $u1 \sim U(0,1)$ y remplazamos a b por $b * u1$, si $b < k$, entonces $X = r$, si no ir a 3

3 Remplazar a r por $r + 1$ y regresar a 2.

De acuerdo con los resultados obetenidos, podemos decir que este método nos da una buena aproximación a la verdadera función de distribución, estos resultados se pueden apreciar en el apendice **B4**.

1. CONCLUSIÓN

Mencionaremos como parte importante y que nos ocurrió en el transcurso de este trabajo lo siguiente; al elegir la manera en la cual íbamos a generar nuestros números pseudoaleatorios, elegimos en primera instancia al Generador Congruencial Mixto, que fué mencionado en este trabajo, pero al ser programado en el lenguaje de Pascal tuvimos dificultades en cuanto al rango de números que nos proporciona, ya que es importante obtener el mayor porcentaje de números, y una opción que encontramos a este problema fue el utilizar el generador que nos da Pascal, RANDOM y este generador mejoró nuestros resultados.

En base a los resultados obtenidos, que fueron el tiempo de cómputo, los histogramas de cada distribución y la aplicación de la prueba estadística (chi-cuadrada), podemos concluir diciendo que los métodos estudiados pueden considerarse buenos pues las distribuciones probabilísticas generadas se aproximan aceptablemente a las distribuciones teórica.

BIBLIOGRAFÍA

Cramer Herald. "Elementos de la téoria de probabilidad y algunas de sus aplicaciones". Aguilar Editores.

Canavos George C. "Probabilidad y Estadística Aplicaciones y Métodos". Mc-GrawHill, books company.

Mood Alexander M., Graybill Franklin A., Boes Duane C., "Introduction to the theory of statistics" Mc-GrawHill, books company.

Berry Donald A., Lindgren Bernard W. "Statistics: Theory and Methods", Books/Cole Publishing Company, Pacific Grove, California.

Hoel Paul G., Port Sidney C., Stone Charles J. "Introduction to Probability Theory". University of California, Los Angeles.

Coss Bu Raúl. "Simulación: Un enfoque practico". Limusa; México 1982.

Matloff Norman S. "Probability Modeling and Computer Simulation"

Rubinstein Reuven Y. "Simulation and the Monte Carlo Method" Mc-GrawHill, books company.

Morgan Byron J. T. "Elements of Simulation". Mathematical Institute. Chapman and Hall.

Ripley Brian D. "Stochastics Simulation"

Mendenhall William. "Introduction to Probability and Statistics

Conover W.J. "Practical Nonparametris Statistics", Wiley, New York 1971

Devroye Luc. "Non-Uniform Random Variate Generation". School of Computer Sciencie, Mc-GrawHill, books company, Montreal, Canadá.

Kahaner David, Nash Stephen, Moler Cleve B. "Numerical Methods and Software", Prentice-Hall, Englewood Cliffs, New Yersey, 1989.

APÉNDICE

A

(La simulacin en este programa de la distribucin normal, se basa en el m
todo de coordenadas polares.)

```
program generador_normal;
    USES
       crt,graph,dos;
   type division = array[1..80] of integer;

 {Declaracin de variables}

   var
      x1, x2, i, a, c, m, k, z,v :longint;
      u1, u2, R2, X, teta, FreRel :real;
      bandera: boolean;
      celda: division;
      hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
      ch:char;
      espacio:string;
      arch:text;
procedure inicia_graficos;          {se inicia graficos}
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;

Function LeadingZero(w:Word):String;   {formato de hora}
 Var
  s:String;
   Begin
    Str(w:0,s);
     if Length(s)=1 then
       s:='0' + s;
       LeadingZero:=s;
   end;
     BEGIN                         {programa principal}
       CLRSCR;
       FOR i:= 1  to  80 DO
       celda[i]:= 0;

 { Muestra en pantalla el tiempo de inicio del programa}

       GetTime(hr1,mn1,seg1,cseg1);
        writeln('la hora inicial es:',LeadingZero(hr1),':',
        LeadingZero(mn1),':',  LeadingZero(seg1),':',
        LeadingZero(cseg1));
        readln;
     Begin
     RANDOMIZE;
     For i:=1 to 25000  DO
      Begin
```

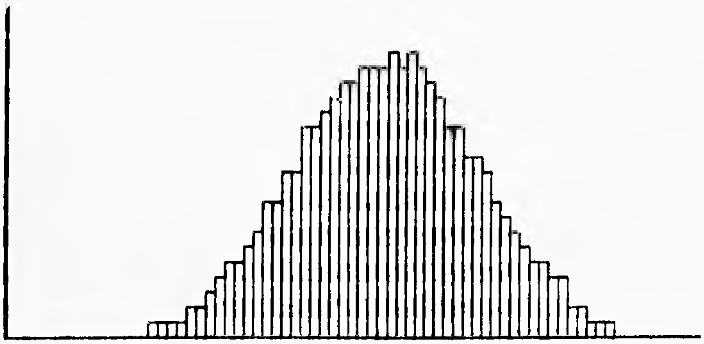
```
        u1:=random;
        u2:=random;
        R2 := -2 * ln(u1);
        teta := 2 * (pi) * (u2);
        X:=sqrt(R2)*cos(teta);          {normal}
        writeln(x:3:2);
          if ((x<4) and (x>-4)) then
            begin
              z:=40+trunc(x/0.10);
            if (x>=0) then
              z:=z+1;
              celda[z] := celda[z] + 1;
            end;
         end;
       end;
     FOR z:= 1  to  80  DO
         begin
          WRITELN('no. de valores en la celda ', z, ':  ', celda[z]);
         end;
          WRITELN;

  { Muestra el tiempo final de laejecucin del programa, calculando
   el tiempo de ejecucin}

         GetTime(hr2,mn2,seg2,cseg2);
            writeln('la hora final es:',LeadingZero(hr2),':',
            LeadingZero(mn2),':', LeadingZero(seg2),':',
            LeadingZero(cseg2));
            writeln('el tiempo de computo es:');
             if (cseg2 < cseg1) then
         begin
           seg2:= seg2 - 1;
           cseg2:= cseg2 + 100;
         end;
         if (seg2 < seg1) then
         begin
           mn2:= mn2 - 1;
           seg2:= seg2 + 60;
         end;
       writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
               leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
       readln;

   { aqui ya se tiene el numero de valores que cayeron en cada celda
    se debe de obtener la frecuencia relativa de cada celda dividiendo
    celda[z]/i
    finalmente graficar el histograma, que se hace a continuacion. }
    inicia_graficos;
    line(1,1,1,getmaxy-1);
    for z:= 1 to 80 do
     begin
      freRel:=celda[z]/i;
      rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-tr
unc(frerel*getmaxy)*10);
     end;
readln
end.
```

```
{Aplicacion de la prueba estadística a los distintos números generados,
de las distintas distribuciones.}


program generador_normal;
   USES
      crt,graph,dos;
   type division = array[1..100] of integer;

 {Declaracin de variables}

   var
      x1, x2, i,  k, z, v, summa1, ob, l :longint;
      u1, u2, R2, X, teta, FreRel, summa, e :real;
      bandera: boolean;
      celda: division;
      hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
      ch:char;
      espacio:string;
      arch:text;
{ Con este procedimiento hacemos el llamado para la elaboracin de la
grafica de la distribucin}

procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

{Esta funcin nos permite dar un formato a la presentacin de salida
de la hora de ejecucin del programa}

Function LeadingZero(w:Word):String;
 Var
  s:String;
  Begin
   Str(w:0,s);
    if Length(s) = 1 then
     s:='0' + s;
     LeadingZero:=s;
  end;
   BEGIN                     {programa principal}
     CLRSCR;
     FOR i:= 1  to  80 DO
     celda[i]:= 0;
     GetTime(hr1,mn1,seg1,cseg1);
      writeln('la hora inicial es:',LeadingZero(hr1),':',
      LeadingZero(mn1),':',  LeadingZero(seg1),':',
      LeadingZero(cseg1));
      readln;

{Comienza la generacin de nmeros con distribucin uniformes
 en base al generador congruencial lineal mixto}
    Begin
```

```
    RANDOMIZE;
    For i:=1 to 25000  DO
     Begin
      u1:=random;
      u2:=random;
      R2 := -2 * ln(u1);
      teta := 2 * (pi) * (u2);
      X:=sqrt(R2)*cos(teta);          {normal}
      writeln(x:3:2);
 {De acuerdo a este rango los nmeros con distribucin normal son guardados
 y posteriormente graficados}

        if ((x<4) and (x>-4)) then
          begin
           z:=40+trunc(x/0.10);
          if (x>=0) then
           z:=z+1;
           celda[z] := celda[z] + 1;
          end;
       end;
     end;
    readln;
 summa:=0;
 summa1:=0;
 FOR z:= 1  to  100  DO
    begin
     summa1:=summa1 + celda[z];
      if (z mod 20 = 0)  then
        begin
          l:= z div 20;
          WRITELN('no. de valores en la celda ', z, ':  ', summa1);
          writeln;
          e:=i/20;
          ob:=summa1;
          summa:=summa + SQR(ob-e)/e;
          writeln;
          summa1:=0;
        end;
    end;
    writeln('el valor esperado E es:', e:3:2);
    writeln;
    writeln('el valor del estadistico es:', summa:3:2);
    writeln;
 GetTime(hr2,mn2,seg2,cseg2);
      writeln('la hora final es:',LeadingZero(hr2),':',
        LeadingZero(mn2),':', LeadingZero(seg2),':',
        LeadingZero(cseg2));
     writeln('el tiempo de computo es:');
    if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
     if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
     writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
            leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
```

```
      readln;

   { aqui ya se tiene el numero de valores que cayeron en cada celda
    se debe de obtener la frecuencia relativa de cada celda dividiendo
    celda[z]/i
    finalmente graficar el histograma, que se hace a continuacion. }
    inicia_graficos;
    line(1,1,1,getmaxy-1);
    for z:= 1 to 80 do
     begin
      freRel:=celda[z]/i;
      rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
     end;
  readln
  end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TÉORICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



NORMAL(COORDENADAS POLARES)

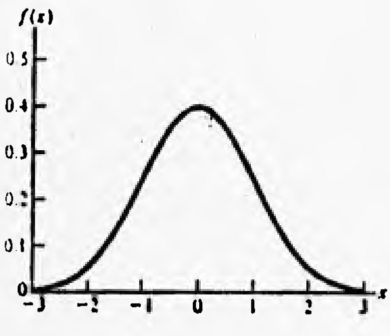


Figure 5.5 N(0, 1) density function.

APÉNDICE

A1

```
{La simulacin en este programa de la distribucin normal, se basa
en el m
       todo de rechazo.}


program normal;
    USES
       crt,graph,dos;
   type division = array[1..80] of integer;

 {Declaracin de variables}

   var
      i, k, z,ld :longint;
      ul, u2,u3, R2,x,V1,V2, teta, FreRel :real;
      bandera: boolean;
      celda: division;
      hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
      ch:char;
      espacio:string;
      arch:text;
procedure inicia_graficos;    {inicia graficos}
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;
Function LeadingZero(w:Word):String;   {formato de hora de inicio y fin}
 Var
  s:String;
   Begin
    Str(w:0,s);
     if Length(s)=1 then
       s:='0' + s;
       LeadingZero:=s;
    end;

     BEGIN            {programa principal}
      CLRSCR;
       FOR i:= 1  to  80 DO
       celda[i]:= 0;
       GetTime(hr1,mn1,seg1,cseg1);     {muetra tiempo inicial}
        writeln('la hora inicial es:',LeadingZero(hr1),':',
        LeadingZero(mn1),':',  LeadingZero(seg1),':',
        LeadingZero(cseg1));
        readln;
        Begin
          ld:=1;
          RANDOMIZE;
           For i:=1 to 25000  Do
            Begin
            Repeat
              ul:=random;
```

```
            u2:=random;
            V1:=(-ld*ln(u1));      {exponencial}
            V2:=(-ld*ln(u2));
         Until V2 >= sqr(V1-1)/2;
            u3:=random;
            If u3 >= 0.5  then
              X:= -V1                  {normal}
            Else
              X:=V1;
             writeln(X:3:2);
       if (x > -4)  and (x < 4) then
        begin
          z:=40 + trunc(x/0.10);
          z:=z+1;
          celda[z] := celda[z] + 1;
       end;
       end;
    end;
  FOR z:= 1  to  80  DO    {los números son guardados en casillas}
      begin
      WRITELN('no. de valores en la celda ', z, ':  ', celda[z]);
   end;
      WRITELN;
      GetTime(hr2,mn2,seg2,cseg2);    {tiempo final de ejecución}
         writeln('la hora final es:',LeadingZero(hr2),':',
         LeadingZero(mn2),':', LeadingZero(seg2),':',
         LeadingZero(cseg2));
         writeln('el tiempo de computo es:');
          if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
            leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma, que se hace a continuacion. }
  inicia_graficos;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 80 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-tr
unc(frerel*getmaxy)*10);
   end;
readln
end.
```

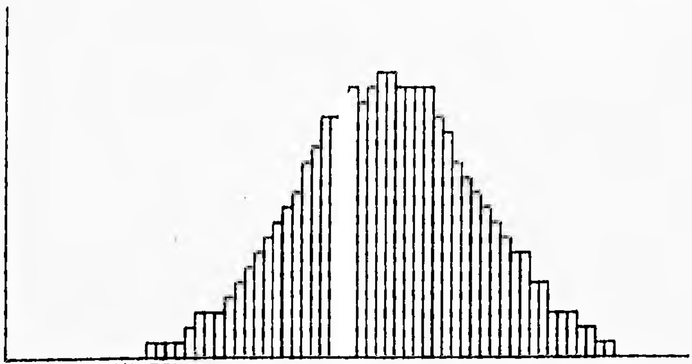
{Aplicacion de la prueba estadistica a los numeros generados conforme a las distintas distribuciones.}

```
program normal;
   USES
     crt,graph,dos;
  type division = array[1..100] of integer;

 {Declaracin de variables}

  var
    i, k, z,summa1, ob, l, ld :longint;
    u1, u2,u3, R2,x,V1,V2, teta, FreRel,summa, e :real;
    bandera: boolean;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
    ch:char;
    espacio:string;
    arch:text;
{ Con este procedimiento hacemos el llamado para la elaboracin de la
grafica de la distribucin}

procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

{Esta funcin nos permite dar un formato a la presentacin de salida
de la hora de ejecucin del programa}

Function LeadingZero(w:Word):String;
 Var
  s:String:
   Begin
    Str(w:0,s);
     if Length(s)=1 then
      s:='0' + s;
      LeadingZero:=s;
    end;

     BEGIN          {programa principal}
      CLRSCR;
       FOR i:= 1 to 100 DO
       celda[i]:= 0;
 { Muestra en pantalla el tiempo de inicio del programa}

       GetTime(hr1,mn1,seg1,cseg1):
        writeln('la hora inicial es:',LeadingZero(hr1),':',
        LeadingZero(mn1),':',  LeadingZero(seg1),':',
        LeadingZero(cseg1));
        readln;
        Begin
```

```
        ld:=1;
        RANDOMIZE;
         For i:=1 to 25000  Do
          Begin
          Repeat
            u1:=random;
            u2:=random;
            V1:=(-ld*ln(u1));    {exponencial}
            V2:=(-ld*ln(u2));
         Until V2 >= sqr(V1-1)/2;
            u3:=random;
            If u3 >= 0.5  then
              X:= -V1
            Else
              X:=V1;
             writeln(X:3:2);
{De acuerdo a este rango los nmeros con distribucin normal son guardados
 y posteriormente graficados}

       if (x > -4)  and (x < 4) then
        begin
          z:=40 + trunc(x/0.10);
           if (x >= 0) then
          z:=z+1;
          celda[z] := celda[z] + 1;
       end;
      end;
    end;
    readln;
 summa:=0;
 summa1:=0;
 FOR z:= 1  to  100  DO
   begin
     summa1:=summa1 + celda[z];
       if (z mod 20 = 0)  then
        begin
          i:= z div 20;
          WRITELN('no. de valores en la celda ', z, ':  ', summa1);
          writeln;
          e:=i/20;
          ob:=summa1;
          summa:=summa + SQR(ob-e)/e;
          writeln;
          summa1:=0;
        end;
   end;
   writeln('el valor esperado E es:',e:3:2);
   writeln;
   writeln('el valor del estadistico es:',summa:3:2);
   writeln;
{ Muestra el tiempo final de laejecucin del programa, calculando
 el tiempo de ejecucin}

   GetTime(hr2,mn2,seg2,cseg2);
      writeln('la hora final es:',LeadingZero(hr2),':',
          LeadingZero(mn2),':', LeadingZero(seg2),':',
          LeadingZero(cseg2));
     writeln('el tiempo de computo es:');
       if (cseg2 < cseg1) then
     begin
```

```
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma, que se hace a continuacion. }
  inicia_graficos;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 100 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
   end;
readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



NORMAL(RECHAZO)

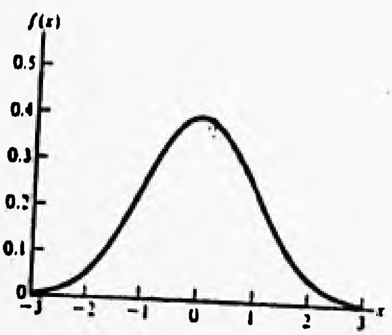


Figure 5.5 N(0, 1) density function.

APÉNDICE

A2

(Usando el m
todo de la transformacin inversa, que se basa en la funcin de distribucin acumulativa F(x)=1-exp-x/landa,resolviendo para x tenemos que X=-1/ld*ln(U),con ld=1.)

```
program generador_exponencial;
    USES
       crt,dos,graph;
   type division = array[1..80] of integer;
   var
      xi, xi_1, i, a, c, m, k, z, v:longint;
      ul, X, FreRel,ld :real;
      bandera: boolean;
      celda: division;
      ch:char;
      arch,arch2:text;
      hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;

procedure inicia_graficos;    (inician graficos)
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;

    Function LeadingZero(w:Word):String;     (formato de hora)
     Var
      s:String;
       Begin
        Str(w:0,s);
         if Length(s)=1 then
          s:='0' + s;
         LeadingZero:=s;
        end;
   BEGIN                         (programa principal)
      CLRSCR;
       FOR i:= 1  to  80 DO
       celda[i]:= 0;
       GetTime(hr1,mn1,seg1,cseg1);
        writeln('la hora inicial es:',LeadingZero(hr1),':',
                 LeadingZero(mn1),':',  LeadingZero(seg1),':',
                 LeadingZero(cseg1));
        readln;

     Begin
      ld:=1;
     RANDOMIZE;
      For i:=1 to 25000  Do
       Begin
        ul:=random;
        X:=(-ld*ln(ul));    (exponencial)
        writeln(x:3:2);
```

```
              IF x < 4 then
                Begin
                  z:=trunc(x/0.10);
                  z:=z+1;
                  celda[z] := celda[z] + 1;
                End;
            END;
          end;
        FOR z:= 1  to  80  DO
        begin
          WRITELN(' no. de valores en la celda ', z, ':  ', celda[z]);
       end;
          WRITELN;
           Gettime(hr2,mn2,seg2,cseg2);
           writeln('la hora final es ',leadingZero(hr2),':',
                  leadingZero(mn2),':',leadingZero(seg2),
                  '.', leadingZero(cseg2));
            writeln('el tiempo de computo fue: ');
             if (cseg2 < cseg1) then
          begin
            seg2:= seg2 - 1;
            cseg2:= cseg2 + 100;
          end;
          if (seg2 < seg1) then
          begin
            mn2:= mn2 - i;
            seg2:= seg2 + 60;
          end;
        writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
                leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
        readln;

   { aqui ya se tiene el numero de valores que cayeron en cada celda
    se debe de obtener la frecuencia relativa de cada celda dividiendo
    celda[z]/i
    finalmente graficar el histograma, que se hace a continuacion. }

    inicia_graficos;
    line(1,1,1,getmaxy-1);
    for z:= 1 to 80 do
     begin
      freRel:=celda[z]/i;
      rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-tr
unc(frerel*getmaxy)*10);

     end;
readln
end.
```

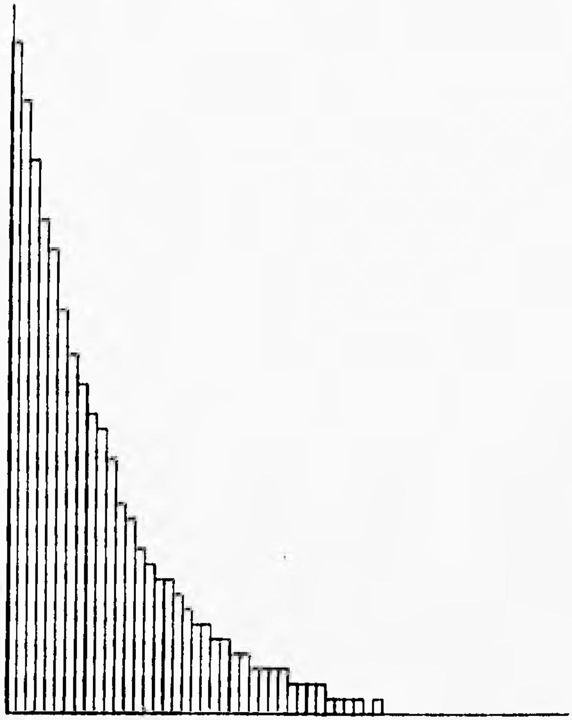
{Aplicacion de la prueba estaistica a los numeros generados de las distintas distribuciones.}

```
program generador_exponencial;
   USES
     crt,dos,graph;
  type division = array[1..100] of integer;
  var
    xi, xi_1, i, summa1, ob, l, k, z, v:longint;
    u1, X, FreRel,Id, summa, e :real;
    bandera: boolean;
    celda: division;
    ch:char;
    arch,arch2:text;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;

 { Con este procedimiento hacemos el llamado para la elaboracin de la
   grafica de la distribucin}

procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

{Esta funcin nos permite dar un formato a la presentacin de salida
de la hora de ejecucin del programa}

   Function LeadingZero(w:Word):String;
    Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
         s:='0' + s;
        LeadingZero:=s;
       end;
 {Inicia el programa principal}

  BEGIN
    clrscr;
     assign(arch, 'c:\sciword\docs\uniforme.tex');
     rewrite(arch);          {inicializacin del archivo para lectura}
     assign(arch2,'c:\sciword\docs\expon.tex');
     rewrite(arch2);
     FOR i:= 1  to  100 DO
     celda[i]:= 0;

 { Muestra en pantalla el tiempo de inicio del programa}

     GetTime(hr1,mn1,seg1,cseg1);
      writeln('la hora inicial es:',LeadingZero(hr1),':',
            LeadingZero(mn1),':',  LeadingZero(seg1),':',
```

```
            LeadingZero(cseg1));
        readln;

{Comienza la generacin de nmeros con distribucin uniformes}
  Begin
   ld:=1;
  RANDOMIZE;
   For i:=1 to 25000  Do
    Begin
     u1:=random;
        v:= i mod 10;
         case v of
          1,6: write(arch,i:3,'.- ', u1:3:2);
          2,7: write(arch,'     ',i:3,'.- ', u1:3:2);
          3,8: write(arch,'      ',i:3,'.- ', u1:3:2);
          4,9: write(arch,'       ',i:3,'.- ', u1:3:2);
          5,0: writeln(arch,'        ',i:3,'.- ', u1:3:2);
         end;
        X:=(-ld*ln(u1));    {exponencial}
        v:= i mod 10;
         case v of
          1,6: write(arch2,i:3,'.- ', X:3:2);
          2,7: write(arch2,'     ',i:3,'.- ', X:3:2);
          3,8: write(arch2,'      ',i:3,'.- ', X:3:2);
          4,9: write(arch2,'       ',i:3,'.- ', X:3:2);
          5,0: writeln(arch2,'        ',i:3,'.- ', X:3:2);
         end;
        writeln;

{De acuerdo a este rango los nmeros con distribucin exponencial
 son guardados y posteriormente graficados}

        IF x < 4 then
         Begin
          z:=trunc(x/0.10);
          z:=z+1;
          celda[z] := celda[z] + 1;
         End;
      END;
    end;
    close(arch);
    readln;

 {Con este ciclo sabemos cuantos nmeros calleron dentro de cada
una de las casillas}
  summa:=0;
  summa1:=0;
  FOR z:= 1  to  100  DO
   begin
    summa1:=summa1 + celda[z];
     if (z mod 20 = 0)  then
      begin
       l:=z div 20;
       WRITELN(' no. de valores en la celda ', z, ':  ', summa1);
       writeln;
       e:=i/20;
       ob:=summa1;
       summa:=summa + SQR(ob-e)/e;
       writeln;
       summa1:=0;
```

```
        end;
    end;
    writeln('el valor esperado E es:',e:3:2);
    writeln;
    writeln('el valor del estadistico es:',summa:3:2);
    writeln;
    close(arch2);
   Gettime(hr2,mn2,seg2,cseg2);
        writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
             '.', leadingZero(cseg2));
         writeln('el tiempo de computo fue: ');
          if (cseg2 < cseg1) then
       begin
         seg2:= seg2 - 1;
         cseg2:= cseg2 + 100;
       end;
       if (seg2 < seg1) then
       begin
         mn2:= mn2 - 1;
         seg2:= seg2 + 60;
       end;
     writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
     readln;

  { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma, que se hace a continuacion. }

   inicia_graficos;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
    begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);

    end;
 readln
 end.
```

COMPARACIÓN DE GRÁFICAS TEÓRICAS Y LA OBTENIDAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



EXPONENCIAL

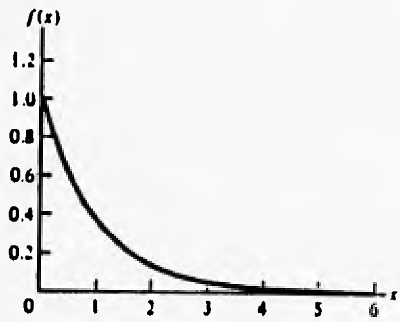


Figure 5.2 expo(1) density function.

APÉNDICE

A3

```
{Para generar una funcion gama con parametros(n,landa(ld),cuando n
 es un entero se usa el factor de que la suma de n variables alea-
 torias exponenciales independientes,cada una con rango ld tiene una
 distribucion gamma.}
program gamma2;

    USES
      crt,dos,graph;
   type division = array[1..80] of integer;
   var
      xi, xi_1, i, a, c, m, k, z,n,v,l:longint;
      u1,X, FreRel,ld,b,producto:real;
      bandera: boolean;
      celda: division;
      ch:char;
      arch,arch2:text;
      hr1,mn1,seg1,cseg1,
      hr2,mn2,seg2,cseg2 :word;

Function LeadingZero(w:word):string;    {formato de hora}
var
   s:string;
Begin
   Str(w:0,s);
   if length(s) = 1 then
   s:='0' + s;
   LeadingZero:=s;
end;

procedure inicia_graficos;     {inician graficos}
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;
   BEGIN                        {programa principal}
      CLRSCR;
       FOR i:= 1  to  80 DO
       celda[i]:= 0;
       Gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es ',leadingZero(hr1),':',
              leadingZero(mn1),':', leadingZero(seg1),
              '.', leadingZero(cseg1));
       readln;
       Begin
        ld:=1;
        producto:=1;
        RANDOMIZE;
         For i:=1 to 25000  Do
         BEGIN
           u1 :=random;
             begin
               X:=(-ld*ln(producto));   {gamma}
               producto:= 1;
                Begin
```

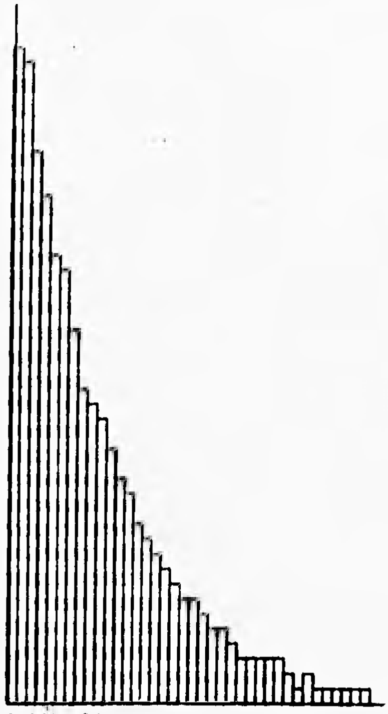
```
          if X<4 then
           z:=trunc(X/0.10);
           z:=z+1;
           celda[z] := celda[z] + 1;
          end;
        end;
      END;
    end;
   FOR z:= 1 to 80 DO
   begin
     WRITELN(' no. de valores en la celda ', z, ':  ', celda[z]);
   end;
   WRITELN;

   Gettime(hr2,mn2,seg2,cseg2);
   writeln('la hora final es ',leadingZero(hr2),':',
          leadingZero(mn2),':',leadingZero(seg2),
          '.', leadingZero(cseg2));
   writeln('el tiempo de computo fue: ');
    if (cseg2 < cseg1) then
     begin
       seg2:= seg2 - 1;
       cseg2:= cseg2 + 100;
     end;
     if (seg2 < seg1) then
     begin
       mn2:= mn2 - 1;
       seg2:= seg2 + 60;
     end;
   writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
         leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
   readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma, que se hace a continuacion. }
  inicia_graficos;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 80 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);

  end;
readln
end.
```

```
{Aplicacion de la prueba estadistica a los numeros generados deacuerdo a las
distintas distribuciones}
program gamma2;

    USES
      crt,dos,graph;
   type division = array[1..100] of integer;
   var
      xi, xi_1, i, summa1, ob, k, z,n,v,l:longint;
      u1,X, FreRel, ld, b, summa, e, producto:real;
      bandera: boolean;
      celda: division;
      ch:char;
      arch,arch2:text;
      hr1,mn1,seg1,cseg1,
      hr2,mn2,seg2,cseg2 :word;

 {Esta funcin nos permite dar un formato a la presentacin de salida
de la hora de ejecucin del programa}

Function LeadingZero(w:word):string;
var
   s:string;
Begin
   Str(w:0,s);
   if length(s)=1 then
   s:='0' + s;
   LeadingZero:=s;
end;

{ Con este procedimiento hacemos el llamado para la elaboracin de la
   grafica de la distribucin}

procedure inicia_graficos;
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;
   BEGIN
      CLRSCR;
       FOR i:= 1  to  100 DO
       celda[i]:= 0;
 {Muestra la hora de inicio del programa}

       Gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es ',leadingZero(hr1),':',
              leadingZero(mn1),':', leadingZero(seg1),
              '.', leadingZero(cseg1));
       readln;
        ld:=1;
        producto:=1;
        RANDOMIZE;
         For i:=1 to 25000  Do
         BEGIN
```

```
          u1 :=random;
          writeln(u1:3:2);
          producto:= producto*u1;
           begin
             X:=(-ld*ln(producto));   {gamma}
             producto:= 1;
              Begin
               if X<4 then
                z:=trunc(X/0.10);
                z:=z+1;
                celda[z] := celda[z] + 1;
              end;
           end;
        end;
 summa:=0;
 summa1:=0;
  FOR z:= 1  to  100  DO
    begin
      summa1:=summa1 + celda[z];
       if (z mod 20 = 0)  then
        begin
         l:= z div 20;
         WRITELN(' no. de valores en la celda ', z, ':  ', summa1);
         writeln;
         e:=i/20;
         ob:=summa1;
         summa:=summa + SQR(ob-e)/e;
         writeln;
         summa1:=0;
       end;
    end;
    writeln('el valor esperado E es:', e:3:2);
    writeln;
    writeln('el valor del estadistico es:', summa:3:2);
    writeln;
Gettime(hr2,mn2,seg2,cseg2);
  writeln('la hora final es ',leadingZero(hr2),':',
          leadingZero(mn2),':',leadingZero(seg2),
          '.', leadingZero(cseg2));
   writeln('el tiempo de computo fue: ');
     if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;

{ aqui ya se tiene el numero de valores que cayeron en cada celda
 se debe de obtener la frecuencia relativa de cada celda dividiendo
 celda[z]/i
 finalmente graficar el histograma, que se hace a continuacion. }
 inicia_graficos;
 line(1,1,1,getmaxy-1);
```

```
  for z:= 1 to 100 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);

   end;
readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



GAMA(0.5,1)

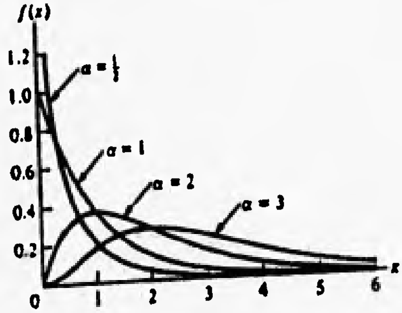


Figure 5.3 gamma(α, 1) density functions.

APÉNDICE

A4

{La distribucion cauchy es simulada por el metodo de la transformacion inversa, es decir obteniendo la inversa de la funcion de distribucion acumulativa.}

```
program generador_cauchy;
   USES
     crt,dos,graph;
  type division = array[1..100] of integer;
  var
    xi, xi_1, i, a, c, m, k, z, alfa,beta,v:longint;
    u1, X, FreRel, tan :real;
    bandera: boolean;
    celda: division;
    ch:char;
    espacio:string;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
    arch,arch2:text;
procedure inicia_graficos;      {inician graficos}
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

function LeadingZero(w : Word) : String;    {formato de hora}
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
   s := '0' + s;
  LeadingZero := s;
end;

  BEGIN                              {programa principal}
  CLRSCR;
   FOR i:= 1 to 100 DO
   celda[i]:= 0;
    GetTime(hr1,mn1,seg1,cseg1);
    WriteLn('La hora inicial es ',LeadingZero(hr1),':',
    LeadingZero(mn1),':',LeadingZero(seg1),
          '.',LeadingZero(cseg1));
    readln;
    Begin
      alfa:=1;
      beta:=1;
      RANDOMIZE;
      For i:=1 to 25000 Do
       Begin
        u1:=random;
        writeln(u1:3:2);
        begin
          tan:=sin(pi*u1)/cos(pi*u1);
           X:=alfa -beta/tan;          {cauchy}
```

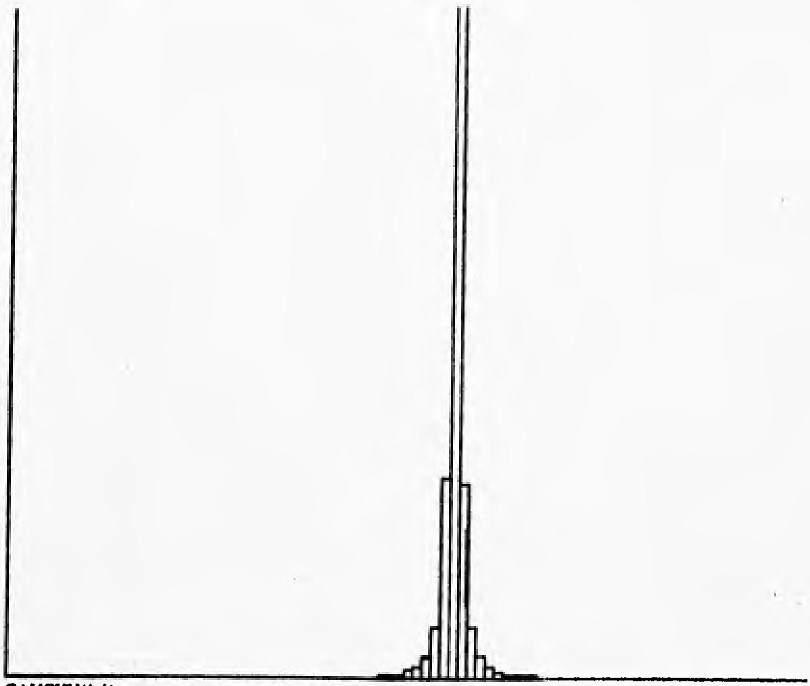
```
          if (x > -100) and (x < 100) then
          begin
             x: =x/2;
             if x > =0 then
                x: =x+1;
             z: =trunc(x)+50;
             celda[z] : = celda[z] + 1;
          end;
        end;
      END;
    end;
    FOR z: = 1 to 100 DO
    begin
      WRITELN(' no. de valores en la celda ', z, ':  ', celda[z]);
   end;
      WRITELN;
      Gettime(hr2,mn2,seg2,cseg2);
      writeln('la hora final es ',leadingZero(hr2),':',
           leadingZero(mn2),':',leadingZero(seg2),
           '.', leadingZero(cseg2));
      writeln('el tiempo de computo fue: ');
      if (cseg2 < cseg1) then
      begin
        seg2: = seg2 - 1;
        cseg2: = cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2: = mn2 - 1;
        seg2: = seg2 + 60;
      end;
        writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
           leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
       readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma, que se hace a continuacion. }

  inicia_graficos;
  line(1,1,1,getmaxy-1);
  for z: = 1 to 100 do
   begin
    freRel: =celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*2);
   end;
readln
end.
```

```
{Aplicacion de la prueba estadistica a los numeros generados por las
distintas distribuciones.}


program generador_cauchy;
   USES
     crt,dos,graph;
  type division = array[1..100] of integer;
  var
     xi, xi_1, i, k, z, summa1, ob, l, alfa,beta,v:longint;
     u1, X, FreRel, tan,summa,e :real;
     bandera: boolean;
     celda: division;
     ch:char;
     espacio:string;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
     arch,arch2:text;
{ Con este procedimiento hacemos el llamado para la elaboracin de la
  grfica de la distribucin}

procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

{Esta funcin nos permite dar un formato a la presentacin de salida
de la hora de ejecucin del programa}

function LeadingZero(w : Word) : String;
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
    s := '0' + s;
  LeadingZero := s;
end;

{Comienza el programa principal}

  BEGIN
  CLRSCR;
   FOR i:= 1 to 100 DO
   celda[i]:= 0;

 {Muestra en pantalla la hora inicial en la ejecucin del programa}

    GetTime(hr1,mn1,seg1,cseg1);
    WriteLn('La hora inicial es ',LeadingZero(hr1),':',
    LeadingZero(mn1),':',LeadingZero(seg1),
           '.',LeadingZero(cseg1));
    readln;
     alfa:=1;
```

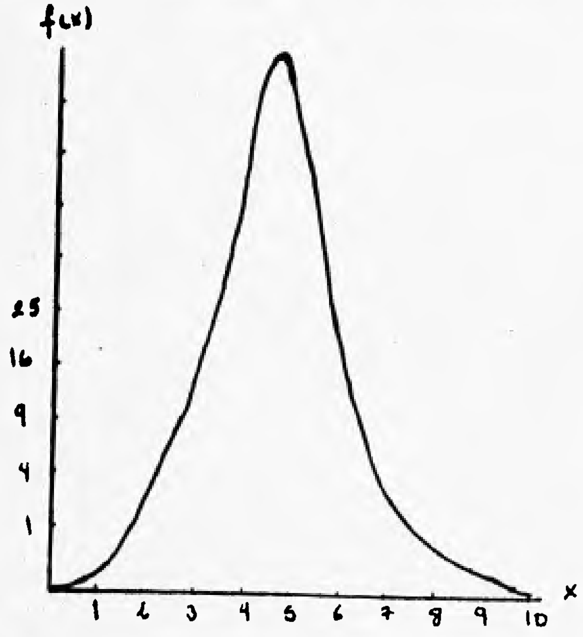
```
beta:=1;
RANDOMIZE;
For i:=1 to 25000  Do
 Begin
  u1:=random;
  writeln(u1:3:2);
  begin
    tan:=sin(pi*u1)/cos(pi*u1);
     X:=alfa -beta/tan;         {cauchy}
     if (x>-100) and (x<100) then
     begin
        x:=x/2;
        if x>=0 then
           x:=x+1;
        z:=trunc(x)+50;
        celda[z] := celda[z] + 1;
     end;
   end;
 end;
 readln;
summa:=0;
summa1:=0;
FOR z:= 1  to  100  DO
 begin
   summa1:=summa1  + celda[z];
     if (z mod 20 = 0)  then
       begin
        l:= z div 20;
        WRITELN(' no. de valores en la celda ', z, ':  ', celda[z]);
        writeln;
        e:=i/20;
        ob:=summa1;
        summa:=summa + SQR(ob-e)/e;
        writeln;
        summa1:=0;
       end;
 end;
 writeln('el valor esperado E es:',e:3:2);
 writeln;
 writeln('el valor del estadistico es:',summa:3:2);
 writeln;
 WRITELN;
Gettime(hr2,mn2,seg2,cseg2);
  writeln('la hora final es ',leadingZero(hr2),':',
         leadingZero(mn2),':',leadingZero(seg2),
         '.', leadingZero(cseg2));
   writeln('el tiempo de computo fue: ');
    if (cseg2 < cseg1) then
    begin
      seg2:= seg2 - 1;
      cseg2:= cseg2 + 100;
    end;
    if (seg2 < seg1) then
    begin
      mn2:= mn2 - 1;
      seg2:= seg2 + 60;
    end;
      writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
         leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
     readln;
```

```
  { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma, que se hace a continuacion. }

   inicia_graficos;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
    begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*2);
    end;
readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



CAUCHY(1,1)



APÉNDICE

A5

{Usando el factor de que x2 con k grados de libertad, es un caso particular de una densidad gama con parmetros, alfa y beta iguales, a k/2 y 2, respectivamente}

```
program generador_ji_cda;
     USES
        crt,dos,graph;
    type division = array[1..100] of integer;
    var
       xi, xi_1, i, a, c, m, k, l, z, v :longint;
       ul,FreRel, producto :real;
       bandera: boolean;
       celda: division;
       ch:char;
       hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
       j:array[1..3550] of real;
       arch, arch2:text;

procedure inicia_graficos;        {inician graficos}
var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'c:\tp\bgi');
    ErrCode := GraphResult;
    if ErrCode <> grOk then halt(1);
end;

function LeadingZero(w : Word) : String;    {formato de salida}
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
    s := '0' + s;
  LeadingZero := s;
end;

BEGIN                             {Comienza el programa pincipal}
   CLRSCR;
   FOR i:= 1  to  100 DO
      celda[i]:= 0;
   FOR i:=1 to 3550 DO
      j[i]:=0;

 { Muestra en pantalla el tiempo de inicio del programa}

   GetTime(hr1,mn1,seg1,cseg1);
   WriteLn('La hora inicial es ',LeadingZero(hr1),':',
   LeadingZero(mn1),':',LeadingZero(seg1),
         '.',LeadingZero(cseg1));
   readln;
   Begin
    producto:=1;
```

```
RANDOMIZE;
 For i:=1 to 35500  DO
  Begin
    ul:=random;
    producto:= producto*ul;
     begin
       if (i mod 10 = 0) then
        begin
         l:= i div 10;
          j[l]:= -2 * ln(producto);               {ji-cuadrada}
          writeln(j[l]:3:2);
          producto:=1;
           if  (j[l] < 50) then
             begin
               j[l]:=j[l]/2;
              if j[l] >=0  then
               z:=trunc(j[l]) + 25;
               celda[z] := celda[z] + 1;
             end;
        end;
      end;
    end;
  end;
  FOR z:= 1  to  100  DO
  begin
    WRITELN('no. de valores en la celda ', z, ':  ', celda[z]);
  end;
    WRITELN;
  Gettime(hr2,mn2,seg2,cseg2);
  writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
             '.', leadingZero(cseg2));
  writeln('el tiempo de computo fue: ');
  if (cseg2 < cseg1) then
    begin
      seg2:= seg2 - 1;
      cseg2:= cseg2 + 100;
    end;
    if (seg2 < seg1) then
    begin
      mn2:= mn2 - 1;
      seg2:= seg2 + 60;
    end;
  writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
           leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
  readln;

{ aqui ya se tiene el numero de valores que cayeron en cada celda
 se debe de obtener la frecuencia relativa de cada celda dividiendo
 celda[z]/i
 finalmente graficar el histograma, que se hace a continuacion. }

 inicia_graficos;
 line(1,1,1,getmaxy-1);
 for z:= 1 to 100 do
  begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-t
runc(frerel*getmaxy)*20);
  end;
readln
end.
```

{Aplicacion de la prueba estadistica a los numeros generados conforme a las distintas distribuciones}

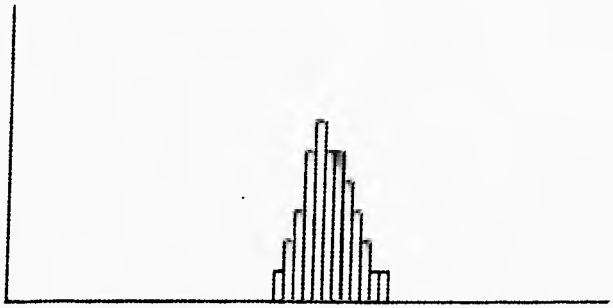
```
program generador_ji_cda;
   USES
      crt,dos,overlay,graph;
  type division = array[1..100] of integer;
  var
     xi, xi_1, i, summa1,ob, k, l, z, v :longint;
     u1,FreRel, producto, summa,e :real;
     bandera: boolean;
     celda: division;
     ch:char;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
     j:array[1..3550] of real;
     arch, arch2:text;

procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;

function LeadingZero(w : Word) : String;
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
    s := '0' + s;
  LeadingZero := s;
end;

BEGIN                         {Comienza el programa pincipal}
  CLRSCR;
  FOR i:= 1 to 100 DO
    celda[i]:= 0;
  FOR i:=1 to 3550 DO
    j[i]:=0;

 { Muestra en pantalla el tiempo de inicio del programa}

  GetTime(hr1,mn1,seg1,cseg1);
  WriteLn('La hora inicial es ',LeadingZero(hr1),':',
  LeadingZero(mn1),':',LeadingZero(seg1),
       '.',LeadingZero(cseg1));
  readln;
  producto:=1;
  RANDOMIZE;
    For i:=1 to 35500 DO
      Begin
        u1:=random;
        producto:= producto*u1;
```

```
        begin
         if (i mod 10 = 0) then
           begin
            l:= i div 10;
            j[l]:= -2 * ln(producto);              {ji-cuadrada}
            writeln(j[l]:3:2);
            producto:=1;
             if  (j[l] < 50) then
               begin
                j[l]:=j[l]/2;
               if j[l] >=0  then
                 z:=trunc(j[l]) + 25;
                 celda[z] := celda[z] + 1;
               end;
           end;
        end;
    end;
    readln;
summa:=0;
summa1:=0;
FOR z:= 1  to  100  DO
   begin
    summa1:=summa1 + celda[z];
     if (z mod 20 = 0)  then
       begin
        l:=z div 20;
        WRITELN('no. de valores en la celda ', z, ':  ', summa1);
        writeln;
        e:=i/20;
        ob:=summa1;
        summa:=summa + SQR(ob-e)/e;
        summa1:=0;
       end;
   end;
   writeln('el valor esperado E es:',e:3:2);
   writeln;
   writeln('el valor del estadistico es:',e:3:2);
   writeln;
Gettime(hr2,mn2,seg2,cseg2);
    writeln('la hora final es ',leadingZero(hr2),':',
           leadingZero(mn2),':',leadingZero(seg2),
            '.', leadingZero(cseg2));
    writeln('el tiempo de computo fue: ');
    if (cseg2 < cseg1) then
      begin
       seg2:= seg2 - 1;
       cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
       mn2:= mn2 - 1;
       seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;

{ aqui ya se tiene el numero de valores que cayeron en cada celda
 se debe de obtener la frecuencia relativa de cada celda dividiendo
 celda[z]/i
```

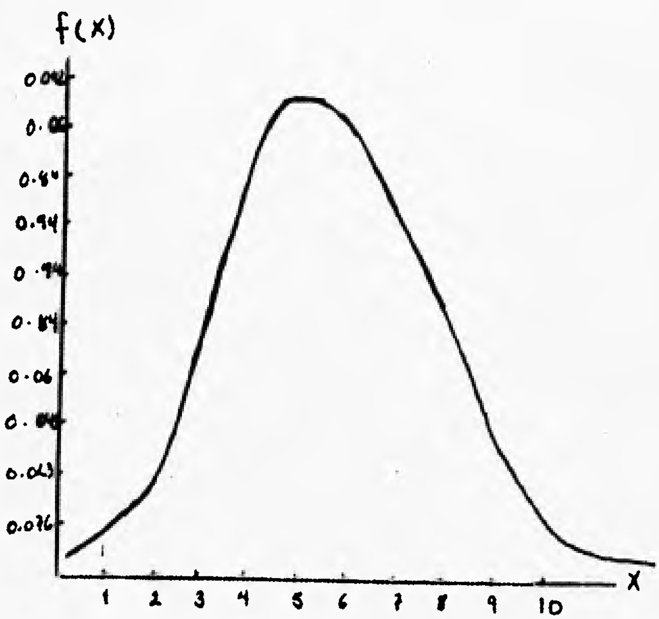
finalmente graficar el histograma, que se hace a continuacion. }

```
   inicia_graficos;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
    begin
      freRel:=celda[z]/i;
      rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*20);
    end;
 readln
 end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



CHI-CUADRADA(10 gl.)



χ^2_{10}

APÉNDICE

A6

{La distribucion t_student, para ser generada necesita de una v.a. X con una distribucion normal y una Y con distribucion Xi_cuadrada.entonces t=X/sqrt(y/k), donde k son los grados de libertad.}

```
program generador_tstudent;

    USES
      crt,dos,graph;
   type division = array[1..80] of integer;
   var
     x1, x2, i, a, c, m, k, z,gl,v,l :longint;
     u1, u2,u3, R2, N, teta, FreRel,T,y,producto,s :real;
     bandera: boolean;
     celda: division;
     X:array[1..5000] of real;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
     ch:char;
     arch,arch2:text;
procedure inicia_graficos;         {se inician graficos}
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;
Function LeadingZero(w:word):string;          {formato de hora}
var
  s:string;
Begin
  Str(w:0,s);
  if length(s)=1 then
  s:='0' + s;
  LeadingZero:=s;
end;
  BEGIN                          {programa principal}
    CLRSCR;
     FOR i:= 1  TO  100 DO
       celda[i]:= 0;
     FOR i:= 1 TO 5000  DO
       x[i]:= 0;
     Gettime(hr1,mn1,seg1,cseg1);
     writeln('la hora inicial es ',leadingZero(hr1),':',
            leadingZero(mn1),':', leadingZero(seg1),
            '.', leadingZero(cseg1));
     readln;
     Begin
      producto:=1;
      RANDOMIZE;
      For i:=1 to 25000  Do
       Begin
        u1 := random;                   {numeros uniformes}
        u2 := random;
        R2 := -2 * ln(u1);
        teta := 2 * (pi) * (u2);
```

```
          N:=sqrt(R2)*cos(teta);          {normal}
          writeln('la normal es :',n:3:2);
          u3:=random;
           producto:= producto*u1;
         begin
           if (i mod 5 = 0) then
             begin
              l:= i div 5;
              X[l]:= -2 * ln(producto);          {ji-cuadrada con 10/2 gl}
              writeln('la ji es :',x[l]:3:2);
              producto := 1;
              s:=sqrt(X[l]/10);
              T:= N/s;
              writeln(T:3:2);
              if (T<4) and (T>-4) then
               begin
                z:=40+trunc(t/0.10);
               if t >= 0 then
                z:=z+1;
                celda[z] := celda[z] + 1;
             end;
         end;
        end;
       end;
       END;
     FOR z:= 1  to  80  DO
     begin
       WRITELN(' no. de valores en la celda ', z, ':  ', celda[z]);
     end;
       WRITELN;
       Gettime(hr2,mn2,seg2,cseg2);
       writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
             '.', leadingZero(cseg2));
       writeln('el tiempo de computo fue: ');
       write(leadingZero(hr2-hr1),':',leadingZero(mn2-mn1),':');
        if (cseg2 < cseg1) then
        begin
          seg2:= seg2 - 1;
          cseg2:= cseg2 + 100;
          writeln(leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg2));
        end;
        readln;
  { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma, que se hace a continuacion. }
   inicia_graficos;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 80 do
   begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*40);
   end;
 readln
 end.
```

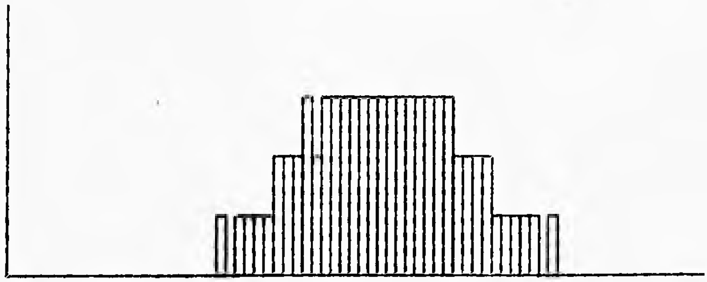
{Aplicacion de la prueba estadistica a los numeros generados, de las distintas distribuciones.}

```
program generador_tstudent;

   USES
     crt,dos,graph;
  type division = array[1..100] of integer;
  var
     x1, x2, i,  k, z, gl, v, l, ob, summa1 :longint;
     u1, u2,u3, R2, N, teta, FreRel,T,y,producto,s,summa,e :real;
     bandera: boolean;
     celda: division;
     X:array[1..5000] of real;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2:word;
     ch:char;
     arch,arch2:text;
procedure inicia_graficos;
var
   grDriver : Integer;
   grMode   : Integer;
   ErrCode  : Integer;
begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
end;
Function LeadingZero(w:word):string;
var
  s:string;
Begin
  Str(w:0,s);
  if length(s)=1 then
  s:='0' + s;
  LeadingZero:=s;
end;
  BEGIN
    CLRSCR;
     FOR i:= 1  TO  100 DO
       celda[i]:= 0;
     FOR i:= 1 TO 5000  DO
       x[i]:= 0;
     Gettime(hr1,mn1,seg1,cseg1);
     writeln('la hora inicial es ',leadingZero(hr1),':',
            leadingZero(mn1),':', leadingZero(seg1),
            '.', leadingZero(cseg1));
     readln;
    producto:=1;
    RANDOMIZE;
    For i:=1 to 25000  Do
     Begin
        u1 := random;                    {nmeros uniformes}
        u2 := random;
        R2 := -2 * ln(u1);
        teta := 2 * (pi) * (u2);
        N:=sqrt(R2)*cos(teta);          {normal}
        writeln(n:3:2);
```

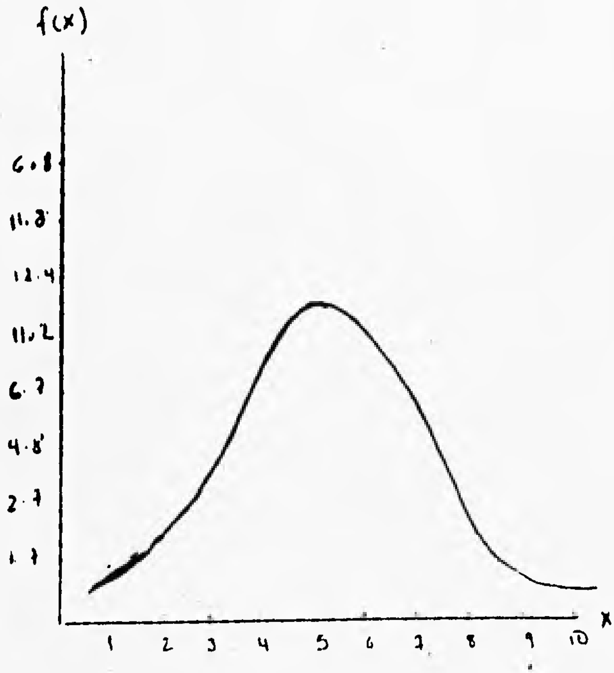
```
        u3:=random;
        producto:= producto*u1;
       begin
        if (i mod 5 = 0) then
         begin
          l:= i div 5;
          X[l]:= -2 * ln(producto);          {ji-cuadrada con 10/2 gl}
          writeln(x[l]:3:2);
          producto := 1;
          s:=sqrt(X[l]/20);
          T:= N/s;
          writeln(T:3:2);
         if (T<4) and (T>-4) then
          begin
           z:=40+trunc(t/0.10);
          if t >= 0 then
           z:=z+1;
           celda[z] := celda[z] + 1;
          end;
         end;
       end;
     END;
    readln;
 summa:=0;
 summa1:=0;
 FOR z:= 1 to 100 DO
   begin
     summa1:=summa1 + celda[z];
      if (z mod 20 = 0) then
       begin
         l:=z div 20;
         WRITELN(' no. de valores en la celda ', z, ': ', summa1);
         writeln;
         e:=i/20;
         ob:=summa1;
         summa:=summa + SQR(ob-e)/e;
         writeln;
         summa1:=0;
       end;
   end;
   writeln('el valor esperado E es:',e:3:2);
   writeln;
   writeln('el valor del estadistico es:',summa:3:2);
   writeln;
 Gettime(hr2,mn2,seg2,cseg2);
     writeln('la hora final es ',leadingZero(hr2),':',
          leadingZero(mn2),':',leadingZero(seg2),
          '.', leadingZero(cseg2));
     writeln('el tiempo de computo fue: ');
     write(leadingZero(hr2-hr1),':',leadingZero(mn2-mn1),':');
      if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
        writeln(leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg2));
      end;
      readln;
{ aqui ya se tiene el numero de valores que cayeron en cada celda
 se debe de obtener la frecuencia relativa de cada celda dividiendo
 celda[z]/i
```

```
   finalmente graficar el histograma, que se hace a continuacion. }
   inicia_graficos;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
   begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*40);
   end;
 readln
 end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



STUDENT(ji de 5gl.)



Student

APÉNDICE

B

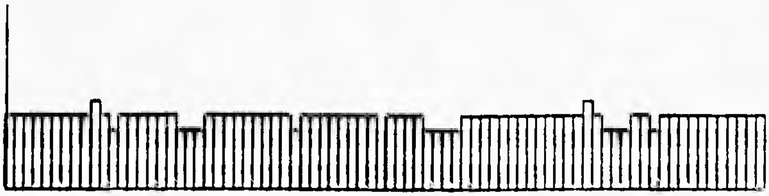
```
{La distribucion uniforme fue generada en base a su definicion}
program uniforme;
  uses
   crt,dos,graph;
  type division = array[1..80] of integer;
  var
    xi, xi_1, i, a,c,m,z,k,j,l:longint;
    u1,p,frerel,e,x:real;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
    ch:char;

   Function LeadingZero(w:Word):String;   {formato de hora}
    Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
           s:='0' + s;
        LeadingZero:=s;
       end;

   Procedure inicia;              {se inician graficos}
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk  then halt(1);
    end;

  BEGIN               {programa principal}
   CLRSCR;
    for i:=1 to 80 do
    celda[i]:=0;
       gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es:',LeadingZero(hr1),':',
              LeadingZero(mn1),':', LeadingZero(seg1),':',
              LeadingZero(cseg1));
       readln;
     Begin
      k:=-4;
      j:=4;
      RANDOMIZE;
       For i:=1 to 25000  DO
        Begin
         u1:=random;
          l:=j-k+1;
          x:=k+(l * u1);
          writeln(x:2:1);
        if (x>-4) and (x<4) then
         begin
            z:=40+trunc( x/0.10);
           if (x>=0)  then
            z:= z +1;
            celda[z] := celda[z] + 1;
         end;
```

```
        end;
       end;
       For z:=1 to 80 do
        Begin
         writeln (' no. de valores en la celda ', z, ':  ', celda[z]);
         writeln;
        end;
         gettime(hr2,mn2,seg2,cseg2);
         writeln('la hora final es:',LeadingZero(hr2),':',
         LeadingZero(mn2),':', LeadingZero(seg2),':',
         LeadingZero(cseg2));
         writeln('el tiempo de computo es:');
          if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;
 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma}
  inicia;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 80 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
   end;
   readln
end.
```

```
{Aplicacion de la prueba estadistica a los numeros generados, de las
distintas distribuciones}
program uniforme;
  uses
   crt,dos,graph;
  type division = array[1..100] of integer;
  var
    xi, xi_1, i, summa1, ob, z, k, j, l:longint;
    u1, p, frerel, e, x, summa:real;
    bandera: boolean;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
    ch:char;

   Function LeadingZero(w:Word):String;
    Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
          s:='0' + s;
        LeadingZero:=s;
       end;

   Procedure inicia;
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk  then halt(1);
    end;

  BEGIN
   CLRSCR;
    for i:=1 to 100 do
    celda[i]:=0;
       gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es:',LeadingZero(hr1),':',
               LeadingZero(mn1),':',  LeadingZero(seg1),':',
               LeadingZero(cseg1));
       readln;
    k:=-4;
    j:=4;
     RANDOMIZE;
       For i:=1 to 25000  DO
         Begin
           u1:=random;
            l:=j-k+1;
            x:=k+(l * u1);
            writeln(x:2:1);
          if (x>-4) and (x<4) then
            begin
              z:=40+trunc( x/0.10);
             if (x>=0)  then
              z:= z +1;
              celda[z] := celda[z] + 1;
```

```
          end;
        end;
        readln;
  summa:=0;
  summa1:=0;
   For z:=1 to 100 do
     Begin
       summa1:=summa1 + celda[z];
        if (z mod 20 = 0)  then
          begin
           i:= z div 20;
           writeln (' no. de valores en la celda ', z, ':  ', summa1);
           writeln;
           e:=i/20;
           ob:=summa1;
           summa:=summa + SQR(ob-e)/e;
           writeln;
           summa1:=0;
          end;
     end;
     writeln('el valor esperado E es:',e:3:2);
     writeln;
     writeln('el valor del estadistico es:',summa:3:2);
     writeln;
  gettime(hr2,mn2,seg2,cseg2);
      writeln('la hora final es:',LeadingZero(hr2),':',
        LeadingZero(mn2),':', LeadingZero(seg2),':',
        LeadingZero(cseg2));
      writeln('el tiempo de computo es:');
       if (cseg2 < cseg1) then
      begin
        seg2:= seg2 - 1;
        cseg2:= cseg2 + 100;
      end;
      if (seg2 < seg1) then
      begin
        mn2:= mn2 - 1;
        seg2:= seg2 + 60;
      end;
    writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
    readln;
 { aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma}
  inicia;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 100 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
   end;
   readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



UNIFORME

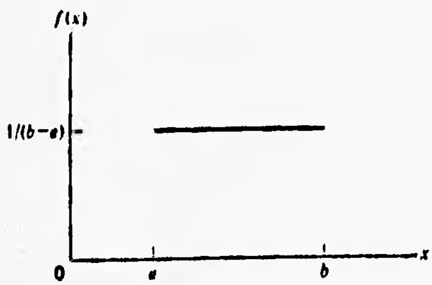


Figure 5.1 U(*a*, *b*) density function.

APÉNDICE

B1

```
{La distribucion geometica fue simulada en relacion a la distribucion
exponencial}
program geometrica;
  uses
   crt,dos,graph;
  type division = array[1..80] of integer;
  var
    xi, xi_1, i, a,c,m,z:longint;
    u1,frerel,y,p,x,suma:real;
    bandera: boolean;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
    ch:char;

   Function LeadingZero(w:Word):String;    {formato de hora}
    Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
           s:='0' + s;
        LeadingZero:=s;
        end;

   Procedure inicia;                {inician graficos}
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk  then halt(1);
    end;
                              {programa principal}
  BEGIN
   CLRSCR;
    for i:=1 to 80 do
      celda[i]:=0;
      gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es:',LeadingZero(hr1),':',
              LeadingZero(mn1),':',  LeadingZero(seg1),':',
              LeadingZero(cseg1));
     readln;
    Begin
     p:=0.5;
      RANDOMIZE;
      For i:=1 to  25000  DO
       Begin
       u1:=random;
       x:=(ln(u1)/ln(1-p));       {geometrica}
       writeln(x:3:2);
        if  (x<4) then
          begin
            z:= trunc(x/0.10);
            z:= z +1;
            celda[z] := celda[z] + 1;
          end;
         end;
```

```
        end;
         readln;
         For z:=1 to 80 do
          Begin
           writeln (' no. de valores en la celda ', z, ':  ', celda[z]);
           writeln;
          end;
           gettime(hr2,mn2,seg2,cseg2);
           writeln('la hora final es ',leadingZero(hr2),':',
              leadingZero(mn2),':',leadingZero(seg2),
               '.', leadingZero(cseg2));
           writeln('el tiempo de computo fue: ');
           if (cseg2 < cseg1) then
          begin
           seg2:= seg2 - 1;
           cseg2:= cseg2 + 100;
          end;
          if (seg2 < seg1) then
           begin
            mn2:= mn2 - 1;
            seg2:= seg2 + 60;
           end;
           writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
             leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));

  { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma}
   inicia;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 80 do
    begin
     freRel: =celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
    end;
    readln
 end.
```

```
{Aplicacion de las pruebas estadistica a los numeros generados conforme a las
distintas distribuciones}
program geometrica;
  uses
   crt,dos,graph;
  type division = array[1..100] of integer;
  var
    xi, xi_1, i,l,ob,z,summa1 :longint;
    u1, frerel, y, p, x, e, summa:real;
    bandera: boolean;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
    ch:char;

  Function LeadingZero(w:Word):String;
   Var
   s:String;
    Begin
     Str(w:0,s);
     if Length(s)=1 then
        s:='0' + s;
     LeadingZero:=s;
     end;

  Procedure inicia;
   var
    grDriver : Integer;
    grMode   : Integer;
    ErrCode  : Integer;
  begin
   grDriver := Detect;
   InitGraph(grDriver,grMode,'c:\tp\bgi');
   ErrCode := GraphResult;
   if ErrCode <> grOk then halt(1);
  end;

 BEGIN
  CLRSCR;
  for i:=1 to 100 do
    celda[i]:=0;
    gettime(hr1,mn1,seg1,cseg1);
     writeln('la hora inicial es:',LeadingZero(hr1),':',
            LeadingZero(mn1),':', LeadingZero(seg1),':',
            LeadingZero(cseg1));
   readln;
  Begin
   p:=0.5;
   RANDOMIZE;
   For i:=1 to 25000 DO
    Begin
    u1:=random;
    x:=(ln(u1)/ln(1-p));      {geometrica}
    writeln(x:3:2);
     if (x < 4) then
      begin
        z:= trunc(x/0.10);
        z:= z +1;
        celda[z] := celda[z] + 1;
      end;
     end;
```

```
      end;
       readln;
       summa:=0;
       summa1:=0;
   For z:=1 to 100 do
     Begin
      summa1:=summa1 + celda[z];
      if (z  mod 20 = 0)  then
        begin
         l:=z div 20;
          writeln (' no. de valores en la celda ', z, ':  ', summa1);
          writeln;
          e:=i/20;
          ob:=summa1;
          summa:=summa + SQR(ob-e)/e;
          writeln;
          summa1:=0;
        end;
    end;
    writeln('el valor esperado E es:',e:3:2);
    writeln;
    writeln('el valor del estadistico es:',summa:3:2);
    writeln;
         gettime(hr2,mn2,seg2,cseg2);
         writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
              '.', leadingZero(cseg2));
         writeln('el tiempo de computo fue: ');
          if (cseg2 < cseg1) then
         begin
          seg2:= seg2 - 1;
          cseg2:= cseg2 + 100;
         end;
         if (seg2 < seg1) then
          begin
           mn2:= mn2 - 1;
           seg2:= seg2 + 60;
          end;
          writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
            leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
          readln;
{ aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma}
  inicia;
  line(1,1,1,getmaxy-1);
  for z:= 1 to 100 do
   begin
    freRel:=celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
   end;
   readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS

GEOMETRICA(0.5)

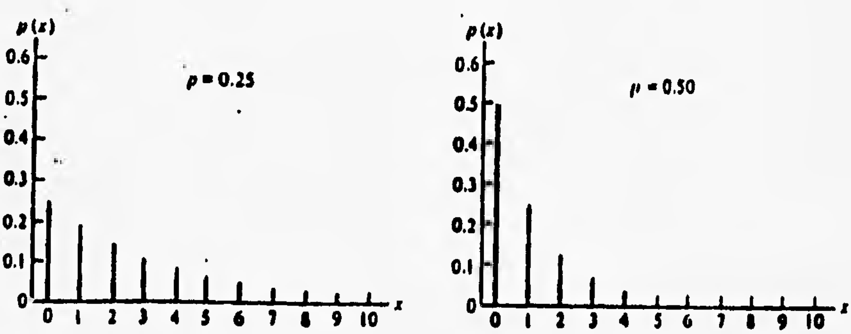


Figure 5.12 geom(p) mass functions.

APÉNDICE

B2

```
{La simulación de la distribución binomial, fu conforme a su relación
con la distribución bernoulli}
program binomial;
   uses
    crt,dos,graph;
   type division = array[1..100] of integer;
   var
      xi, xi_1,x, i, a,c,m,z,suma,n,l:longint;
      ul,p,frerel:real;
      celda: division;
      hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
      ch:char;
      b:array [1..2500] of real;
    Function LeadingZero(w:Word):String;  {formato de hora}
     Var
      s:String;
       Begin
        Str(w:0,s);
         if Length(s)=1 then
            s:='0' + s;
         LeadingZero:=s;
        end;

    Procedure inicia;                   { se inician graficos}
      var
       grDriver : Integer;
       grMode   : Integer;
       ErrCode  : Integer;
     begin
      grDriver := Detect;
      InitGraph(grDriver,grMode,'c:\tp\bgi');
      ErrCode := GraphResult;
      if ErrCode <> grOk  then halt(1);
     end;

   BEGIN                                {programa principal}
    CLRSCR;
     for i:=1 to 100 do
        celda[i]:=0;
     for i:=1 to 2500 do
        b[i]:=0;
        gettime(hr1,mn1,seg1,cseg1);
        writeln('la hora inicial es:',LeadingZero(hr1),':',
                 LeadingZero(mn1),':',  LeadingZero(seg1),':',
                 LeadingZero(cseg1));
        readln;
       Begin
       suma:=0;
        p:=0.5;
        n:=10;
       RANDOMIZE;
        For i:=1 to 25000  Do
         begin
           ul :=random;
            if ul <= p then
               x:=1
            else
```

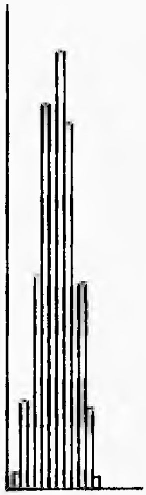
```
            x:=0;                    {bernoulli}
           suma:=suma + x;
          if (i mod 10 = 0) then
          begin
             l:=i div 10;                  {binomial}
             b[l]:=suma;
             writeln(b[l]:3:2);
             suma:=0;
          if  (b[l] < 10) then
          begin
             z:=trunc(b[l]/0.10);
             z:=z + 1;
             celda[z] := celda[z] + 1;
          end;
        end;
       end;
      end;
      readln;
        For z:=1 to 80 do
         Begin
          writeln (' no. de valores en la celda ', z, ':  ', celda[z]);
          writeln;
         end;
          Gettime(hr2,mn2,seg2,cseg2);
          writeln('la hora final es ',leadingZero(hr2),':',
              leadingZero(mn2),':',leadingZero(seg2),
              '.', leadingZero(cseg2));
          writeln('el tiempo de computo fue: ');
       if (cseg2 < cseg1) then
       begin
         seg2:= seg2 - 1;
         cseg2:= cseg2 + 100;
       end;
       if (seg2 < seg1) then
       begin
         mn2:= mn2 - 1;
         seg2:= seg2 + 60;
       end;
         writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
             leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
        readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma}
   inicia;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
    begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-tr
unc(frerel*getmaxy)*10);
    end;
    readln
end.
```

```
{La simulacion de la distribucion binomial, se baso en la relacion
que existe con la distribucion bernoulli}
program binomial;
  uses
   crt,dos,graph;
  type division = array[1..100] of integer;
  var
    xi, xi_1,x, i, a,c,m,z,suma,summa1,k,ob,n,l:longint;
    u1,p,e,summa,frerel:real;
    celda: division;
    hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
    ch:char;
    b:array [1..2500] of real;
   Function LeadingZero(w:Word):String;  {formato de hora}
    Var
     s:String;
     Begin
      Str(w:0,s);
       if Length(s)=1 then
         s:='0' + s;
       LeadingZero:=s;
      end;

   Procedure inicia;             { se inician graficos}
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk  then halt(1);
    end;

  BEGIN                          {programa principal}
   CLRSCR;
    for i:=1 to 100 do
       celda[i]:=0;
    for i:=1 to 2500 do
       b[i]:=0;
       gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora Inicial es:',LeadingZero(hr1),':',
             LeadingZero(mn1),':', LeadingZero(seg1),':',
             LeadingZero(cseg1));
       readln;
       suma:=0;
       p:=0.5;
       n:=10;
      RANDOMIZE;
       For i:=1 to 25000  Do
        begin
          u1 :=random;
           if u1  <= p then
             x:=1
           else
             x:=0;               {bernoulli}
           suma:=suma + x;
          if (i mod 10 = 0) then
          begin
```

```
            l:=i div 10;                    {binomial}
            b[l]:=suma;
            writeln(b[l]:3:2);
            suma:=0;
          end;
          if  (b[l] < 10) then
          begin
            z:=trunc(b[l]/0.70);
            z:=z + 1;
            celda[z] := celda[z] + 1;
          end;
      {  end;}
       end;
       readln;
         summa:=0;
         summa1:=0;
   For z:=1 to 100  do
     begin
       summa1:=summa1  + celda[z];
         if (z mod  20 = 0)  then
           begin
             l:=z div 20;
             writeln (' no. de valores en la celda ', z, ':  ', summa1);
             writeln;
             e:=i/20;
             ob:=summa1;
             summa:=summa +SQR(ob-e)/e;
             writeln;
             summa1:=0;
           end;
    end;
     writeln('el valor esperado E es:',e:3:2);
     writeln ;
     writeln('el valor del estadistico es:',summa:3:2);
     writeln;
     Gettime(hr2,mn2,seg2,cseg2);
     writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
              '.', leadingZero(cseg2));
    writeln('el tiempo de computo fue:  ');
      if (cseg2 < cseg1) then
        begin
          seg2:= seg2 - 1;
          cseg2:= cseg2 + 100;
        end;
         if (seg2 < seg1) then
           begin
             mn2:= mn2 - 1;
             seg2:= seg2 + 60;
           end;
          writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
              leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
         readln;

 { aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma}
   inicia;
   line(1,1,1,getmaxy-1);
```

```
   for z:= 1 to 50 do
    begin
     freRel: =celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/100), getmaxy, z*trunc(getmaxx/100), getmaxy-trunc(frerel*getmaxy)*3);
    end;
    readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



BINOMIAL (10,0.5)

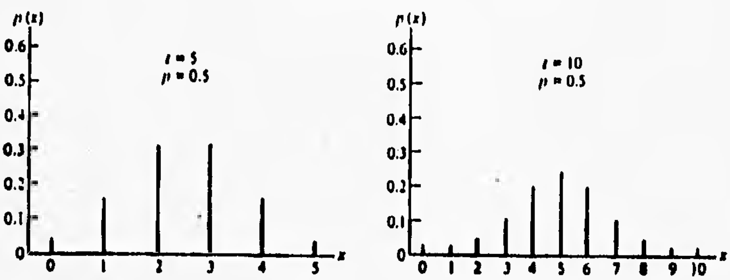


Figure 5.11 bin(t, p) mass functions.

APÉNDICE

B3

```
{La simulacin de esta distribución se baso en su relación con
 la distribución geomtrica }

program binomial negativa;
   uses
    crt,dos,graph;
   type division = array[1..100] of integer;
   var
      xi, xi_1, i, a,c,m,z,n,l:longint;
      ul,frerel,y,p,x,suma:real;
      celda: division;
      hrl,mnl,segl,csegl,hr2,mn2,seg2,cseg2 :word;
      ch:char;
      bn:array [1..2500] of real;

    Function LeadingZero(w:Word):String;    {Formato de la hora}
     Var
      s:String;
       Begin
        Str(w:0,s);
         if Length(s)=1 then
            s:='0' + s;
         LeadingZero:=s;
         end;

    Procedure inicia;                          {llamado a graficos}
      var
       grDriver : Integer;
       grMode   : Integer;
       ErrCode  : Integer;
     begin
      grDriver := Detect;
      InitGraph(grDriver,grMode,'c:\tp\bgi');
      ErrCode := GraphResult;
      if ErrCode <> grOk  then halt(1);
     end;

   BEGIN                                        {programa principal}
    CLRSCR;
     for i:=1 to 100 do
     celda[i]:=0;
     for i:=1 to 2500 do
     bn[i]:=0;
      gettime(hrl,mnl,segl,csegl);
       writeln('la hora inicial es:',LeadingZero(hrl),':',
                  LeadingZero(mnl),':',  LeadingZero(segl),':',
                  LeadingZero(csegl));
       readln;
       Begin
       suma:=0;
        n:=10;
        p:=0.5;
       RANDOMIZE;
        For i:=1 to 25000  Do
         begin
          ul:=random;
          x:=(ln(ul)/ln(1-p));         {geometrica}
          suma:=suma + x;
```

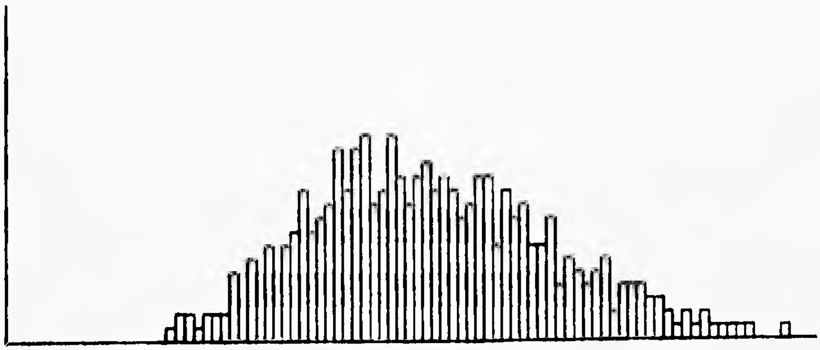
```
       if (i mod 10 = 0)  then
        begin
           l:=i div 10;
           bn[l]:=suma;             (binomial negativa)
           writeln(bn[l]:3:2);
           suma:=0;
         end;
       if  (bn[l] < 30) then
          begin
             z:= trunc(bn[l]/0.30) ;
             z:= z +1;
             celda[z] := celda[z] + 1;
          end;
        end;
       end;
      readln;
       For z:=1 to 80 do            (los números generados son guardados)
        Begin
         writeln (' no. de valores en la celda ', z, ':  ', celda[z]);
         writeln;
        end;
         gettime(hr2,mn2,seg2,cseg2);
         writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
             '.', leadingZero(cseg2));
         writeln('el tiempo de computo fue: ');
          if (cseg2 < cseg1) then
         begin
          seg2:= seg2 - 1;
          cseg2:= cseg2 + 100;
         end;
         if (seg2 < seg1) then
          begin
           mn2:= mn2 - 1;
           seg2:= seg2 + 60;
          end;
          writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
          LeadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
         readln;

 ( aqui ya se tiene el numero de valores que cayeron en cada celda
   se debe de obtener la frecuencia relativa de cada celda dividiendo
   celda[z]/i
   finalmente graficar el histograma)
   inicia;
   line(1,1,1,getmaxy-1);
   for z:= 1 to 100 do
    begin
     freRel:=celda[z]/i;
     rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-tr
unc(frerel*getmaxy)*10);
    end;
    readln
end.
```

```
{Apliacacion de la prueba estadistica a los numeros generados, conforme a las
distintas distribuciones}
program binega;
   uses
    crt,dos,graph;
   type division = array[1..100] of integer;
   var
     xi, xi_1, i, a,c,m,z,n,l,ob,summa1:longint;
     u1,frerel,y,p,x,suma,summa,e:real;
     bandera: boolean;
     celda: division;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
     ch:char;
     bn:array [1..2500] of real;

    Function LeadingZero(w:Word):String;
     Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
           s:='0' + s;
        LeadingZero:=s;
        end;

    Procedure inicia;
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk  then halt(1);
    end;

   BEGIN
    CLRSCR;
     for i:=1 to 100 do
     celda[i]:=0;
     for i:=1 to 2500 do
     bn[i]:=0;
      gettime(hr1,mn1,seg1,cseg1);
       writeln('la hora inicial es:',LeadingZero(hr1),':',
              LeadingZero(mn1),':', LeadingZero(seg1),':',
              LeadingZero(cseg1));
          readln;
   suma:=0;
   n:=10;
   p:=0.5;
   RANDOMIZE;
     For i:=1 to 25000  Do
       begin
         u1:=random;
         x:=(ln(u1)/ln(1-p));       {geometrica}
         suma:=suma + x;
        if (i mod 10 = 0)  then
         begin
           l:=i div 10;
```

```
            bn[l]:=suma;        {binomial negativa}
            writeln(bn[l]:3:2);
            suma:=0;
        if (bn[l] < 30) then
          begin
            z:= trunc(bn[l]/0.30) ;
            z:= z + l;
            celda[z] := celda[z] + l;
          end;
        end;
       end;
      readln;
  summa:=0;
  summal:=0;
   For z:=l to 100 do
     Begin
      summal:=summal + celda[z];
      if (z mod 20 = 0)  then
       begin
         l:=z div 20;
         writeln (' no. de valores en la celda ', z, ':  ', summal);
         writeln;
         e:=i/20;
         ob:=summal;
         summa:=summa + SQR(ob-e)/e;
         writeln;
         summal:=0;
       end;
     end;
     writeln('el valor esperado E es:', e:3:2);
     writeln;
     writeln('el valor del estadistico es:', summa:3:2);
     writeln;
         gettime(hr2,mn2,seg2,cseg2);
         writeln('la hora final es ',leadingZero(hr2),':',
             leadingZero(mn2),':',leadingZero(seg2),
             '.', leadingZero(cseg2));
         writeln('el tiempo de computo fue: ');
          if (cseg2 < csegl) then
         begin
          seg2:= seg2 - l;
          cseg2:= cseg2 + 100;
         end;
         if (seg2 < segl) then
          begin
           mn2:= mn2 - l;
           seg2:= seg2 + 60;
          end;
          writeln(leadingZero(hr2-hrl),':', leadingZero(mn2-mnl),':',
          LeadingZero(seg2-segl),'.',leadingZero(cseg2-csegl));
         readln;

{ aqui ya se tiene el numero de valores que cayeron en cada celda
  se debe de obtener la frecuencia relativa de cada celda dividiendo
  celda[z]/i
  finalmente graficar el histograma}
  inicia;
  line(l,l,l,getmaxy-l);
  for z:= l to 100 do
   begin
```

```
    freRel: =celda[z]/i;
    rectangle((z-1)*trunc(getmaxx/80), getmaxy, z*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*10);
   end;
   readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



BINOMIAL NEGATIVA(10,0.5)

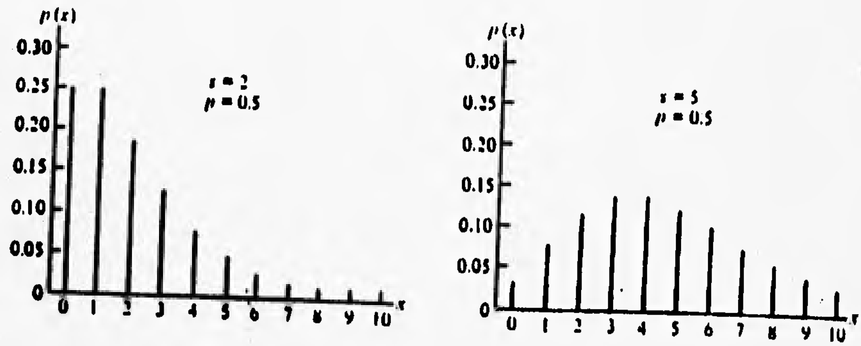


Figure 5.13 negbin(s, p) mass functions.

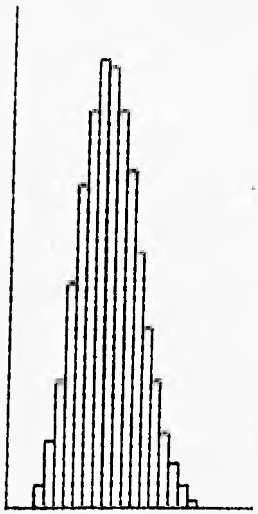
APÉNDICE

B4

```
{La distribucion poisson fue simulada en base al metodo particular
mencionado en el trabajo}
program poisson;
  uses
   crt,dos,graph;
  type division = array[1..100] of integer;
  var
     i, a,z,j,r,x:longint;
     u1,p,frerel,k,ld,b:real;
     bandera: boolean;
     celda: division;
     hr1,mn1,seg1,cseg1,hr2,mn2,seg2,cseg2 :word;
     ch:char;

   Function LeadingZero(w:Word):String;
    Var
     s:String;
      Begin
       Str(w:0,s);
        if Length(s)=1 then
           s:='0' + s;
        LeadingZero:=s;
      end;
   Procedure inicia;
     var
      grDriver : Integer;
      grMode   : Integer;
      ErrCode  : Integer;
    begin
     grDriver := Detect;
     InitGraph(grDriver,grMode,'c:\tp\bgi');
     ErrCode := GraphResult;
     if ErrCode <> grOk then halt(1);
    end;

 BEGIN
  CLRSCR;
  for i:=1 to 80 do
    celda[i]:=0;
  gettime(hr1,mn1,seg1,cseg1);
  writeln('la hora inicial es:',LeadingZero(hr1),':',
         LeadingZero(mn1),':', LeadingZero(seg1),':',
         LeadingZero(cseg1));
  readln;
  RANDOMIZE;
  For i:=1 to 25000 do
    begin
     ld:=10.0;
     b:=1;
     r:=0;
     k:=exp(-ld);
     Repeat
        u1:=random;
        b:=b*u1;
        x:=r;
        r:=r+1;
     until b < k;
     if (x < 80) then
        celda[x+1] := celda[x+1] + 1;
   end;
```

```
For z:=5 to 15 Do
  Begin
    writeln (' no. de valores en la celda ', z, ':  ', celda[z+1]);
    writeln;
  end;
readln;
gettime(hr2,mn2,seg2,cseg2);
writeln('la hora final es:',LeadingZero(hr2),':',
LeadingZero(mn2),':', LeadingZero(seg2),':',
      LeadingZero(cseg2));
writeln('el tiempo de computo fue: ');
if (cseg2 < cseg1) then
   begin
     seg2:= seg2 - 1;
     cseg2:= cseg2 + 100;
   end;
if (seg2 < seg1) then
   begin
     mn2:= mn2 - 1;
     seg2:= seg2 + 60;
   end;
writeln(leadingZero(hr2-hr1),':', leadingZero(mn2-mn1),':',
       leadingZero(seg2-seg1),'.',leadingZero(cseg2-cseg1));
readln;
{ aqui ya se tiene el numero de valores que cayeron en cada celda
se debe de obtener la frecuencia relativa de cada celda dividiendo
celda[z]/i
finalmente graficar el histograma}
inicia;
line(1,1,1,getmaxy-1);
for z:= 0 to 79 do
 begin
  freRel:=celda[z+1]/i;
  rectangle(z*trunc(getmaxx/80), getmaxy, (z+1)*trunc(getmaxx/80), getmaxy-trunc(frerel*getmaxy)*5);
 end;
 readln
end.
```

COMPARACIÓN DE GRÁFICAS OBTENIDAS Y LAS TEÓRICAS DE ACUERDO A LA SIMULACIÓN CON 25000 NÚMEROS GENERADOS



POISSON (10)

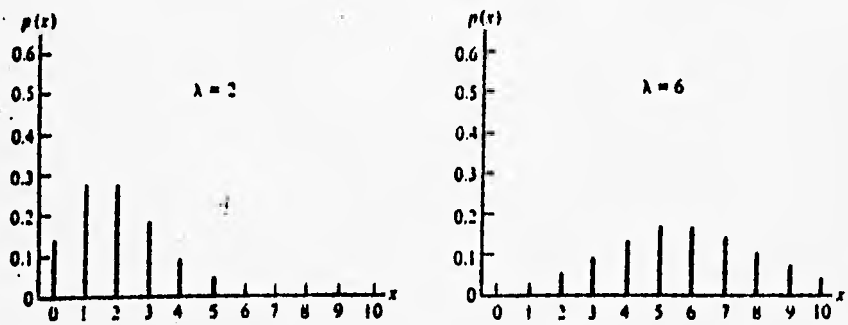


Figure 5.14 Poisson(λ) mass functions.