PRÁCTICO 4

GENERACIÓN DE VARIABLES ALEATORIAS DISCRETAS.

Ejercicio 1. Se baraja un conjunto de $n = 100$ cartas (numeradas consecutivamente del 1 al 100) y se extrae del mazo una carta por vez. Consideramos que ocurre un "éxito" si la i-ésima carta extraída es aquella cuyo número es i ($i = 1,\dots,n$).

a) Calcule la probabilidad de que

  (i) las primeras r cartas sean coincidencias y dé su valor para $r = 10$.

  (ii) haya exactamente r coincidencias y estén en las primeras r cartas. Dé su valor para $r = 10$.

b) Pruebe que $E(X) = Var(X) = 1$ donde X es el número de coincidencias obtenidas en una baraja de n cartas.

c) Escriba un programa de simulación para estimar la esperanza y la varianza del número total de éxitos, y de los eventos del inciso (a) con $r = 10$, y compare los resultados obtenidos con 100, 1000, 10000 y 100000 iteraciones.

Use el archivo "Problemas de coincidencias" para guiarse.

Ejercicio 2. Se desea construir una aproximación de:

$$\sum_{k=1}^{N} \exp\left(\frac{k}{N}\right) \text{ donde } N = 10000 \, .$$

a) Escriba un algoritmo para estimar la cantidad deseada.

b) Obtenga la aproximación sorteando 100 números aleatorios.

c) Escriba un algoritmo para calcular la suma de los primeros 100 términos, y compare el valor exacto con las dos aproximaciones, y el tiempo de cálculo.

Ejercicio 3. Se lanzan simultáneamente un par de dados legales y se anota el resultado de la suma de ambos. El proceso se repite hasta que todos los resultados posibles: $2,3,\dots,12$ hayan aparecido al menos una vez. Estudiar mediante una simulación la variable N, el número de lanzamientos necesarios para cumplir el proceso. Cada lanzamiento implica arrojar *el par* de dados.

a) Describa la estructura lógica del algoritmo que permite simular en computadora el número de lanzamientos necesarios para cumplir el proceso.

b) Mediante una implementación en computadora,

  (i) estime el valor medio y la desviación estándar del número de lanzamientos, repitiendo el algoritmo: 100, 1000, 10000 y 100000 veces.

  (ii) estime la probabilidad de que N sea por lo menos 15 y la probabilidad de que N sea a lo sumo 9, repitiendo el algoritmo: 100, 1000, 10000 y 100000.

Ejercicio 4. Implemente cuatro métodos para generar una variable X que toma los valores del 1 al 10, con probabilidades $p_1 = 0{,}11,\ p_2 = 0{,}14,\ p_3 = 0{,}09,\ p_4 = 0{,}08,\ p_5 = 0{,}12,\ p_6 = 0{,}10,\ p_7 = 0{,}09,\ p_8 = 0{,}07,\ p_9 = 0{,}11,\ p_{10} = 0{,}09$ usando:

a) Método de rechazo con una uniforme discreta, buscando la cota c más baja posible.

b) Método de rechazo con una uniforme discreta, usando $c = 3$.

c) Transformada inversa.

d) Método de la urna: utilizar un arreglo A de tamaño 100 donde cada valor i está en exactamente $p_i * 100$ posiciones. El método debe devolver $A[k]$ con probabilidad 0,01. ¿Por qué funciona?

Compare la eficiencia de los tres algoritmos realizando 10000 simulaciones.

Ejercicio 5. Implemente dos métodos para generar una binomial $Bin(n, p)$:

I) Usando transformada inversa.

II) Simulando n ensayos con probabilidad de éxito p y contando el número de éxitos.

Para ambos métodos:

a) Compare la eficiencia de ambos algoritmos para $n = 10$ y $p = 0{,}3$, evaluando el tiempo necesario para realizar 10000 simulaciones.

b) Estime el valor con mayor ocurrencia y la proporción de veces que se obtuvieron los valores 0 y 10 respectivamente.

c) Compare estos valores con las probabilidades teóricas de la binomial. Si están alejados, revise el código.

Ejercicio 6. Una variable aleatoria X tiene una función de probabilidad puntual $p_i = P(X = i)$ dada por

$$p_0 = 0{,}15, \quad p_1 = 0{,}20, \quad p_2 = 0{,}10, \quad p_3 = 0{,}35, \quad p_4 = 0{,}20$$

I) Describir mediante un pseudocódigo un algoritmo que simule X utilizando el método de la transformada inversa y que minimice el número esperado de búsquedas.

II) Describir mediante un pseudocódigo un algoritmo que simule X utilizando el método de aceptación y rechazo con una variable soporte Y con distribución binomial B(4,0.45).

III) Compare la eficiencia de los dos algoritmos realizando 10000 simulaciones.

Ejercicio 7. Estime $P(Y > 2)$ con $\lambda = 10$, y 1000 repeticiones para la variable Poisson, simulando con método de transformada inversa común e inversa mejorado.

Ejercicio 8.

a) Desarrolle el método de la Transformada Inversa y el de Rechazo para generar una variable aleatoria X cuya distribución de probabilidad está dada por:

$$P(X = i) = \frac{\frac{\lambda^i}{i!} e^{-\lambda}}{\sum_{j=0}^{k} \frac{\lambda^j}{j!} e^{-\lambda}} \quad (i = 0, \ldots, k)$$

b) Estime $P(X > 2)$ con $k = 10$ y $\lambda = 0{,}7$, y 1000 repeticiones. Compare con el valor exacto.

c) Generalice el problema escribiendo un pseudocódigo para el metodo de rechazo para cualquier variable aleatoria truncada usando como soporte a la variable original (con “cualquier variable aleatoria truncada” nos referimos a una variable como la vista en el inciso (a) pero ahora truncada en cualquier parte $i = a, ..., b$).

Ejercicio 9. Implemente dos métodos para simular una variable geométrica $Geom(p)$:

a) Usando transformada inversa y aplicando la fórmula recursiva para $P(X = i)$.

b) Simulando ensayos con probabilidad de éxito p hasta obtener un éxito.

Compare la eficiencia de estos algoritmos para $p = 0{,}8$ y para $p = 0{,}2$.
Para cada caso, realice 10000 simulaciones y calcule el promedio de los valores obtenidos. Comparar estos valores con el valor esperado de la distribución correspondiente. Si están alejados, revisar el código.

Ejercicio 10.

(a) Desarrolle un método para generar una variable aleatoria X cuya distribución de probabilidad está dada por:

$$P(X = j) = \left(\frac{1}{2}\right)^{j+1} + \frac{\left(\frac{1}{2}\right) 2^{j-1}}{3^j}, \quad j = 1, 2, \ldots$$

(b) Estime $E(X)$ con 1000 repeticiones y compare con la esperanza exacta.

Ejercicio 11. Sea X una variable aleatoria cuya distribución de probabilidad es $P(X = j) = p_j$ con $j = 1, 2, \ldots$. Sea:

$$\lambda_n = P(X = n | X > n-1) = \frac{p_n}{1 - \sum_{j=1}^{n-1} p_j}, \quad n = 1, 2, \ldots$$

Las cantidades λ_n, son las tasas discretas de riesgo. Considerando a X como el tiempo (discreto) de vida de algún artículo, λ_n representa la probabilidad de que habiendo funcionado correctamente hasta el tiempo $n-1$, se rompa en el tiempo n.

a) Muestre que $p_1 = \lambda_1$ y que

$$p_n = (1 - \lambda_1)(1 - \lambda_2) \cdots (1 - \lambda_{n-1}) \lambda_n$$

Método de la tasa discreta de riesgo para simular variables aleatorias discretas: Se genera una sucesión de números aleatorios que termina cuando el n-ésimo número generado es menor que λ_n. El algoritmo puede escribirse como sigue:

```
Paso 1: X = 1
Paso 2: Generar U
Paso 3: Si U <λ_X, terminar.
Paso 4: X = X + 1
Paso 5: Ir al Paso 2
```

b) Muestre que los valores de X que genera este proceso tienen la distribución de probabilidad deseada.

c) Suponga que X es una variable aleatoria geométrica con parámetro p:

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

Determine los valores de $\lambda_n, n \geq 1$. Explique cómo funciona el algoritmo anterior en este caso y por qué es evidente su validez.

Ejercicio 12. ¿Qué distribución tiene la variable simulada por el siguiente algoritmo?

```
def QueDevuelve(p1,p2):
   X = int(np.log(1-random())/np.log(1-p1))+1
   Y = int(np.log(1-random())/np.log(1-p2))+1
   return min(X,Y)
```

Escriba otro algoritmo que utilice un único número aleatorio (random()) y que simule una variable con la misma distribución que la simulada por QueDevuelve(0.05, 0.2).