Segundo Formulario Simulación II

Solis Procopio Uriel

1. Funciones continuas

1.1. Uniforme

Sea X una variable aleatoria co distribución uniforme continua entonces escribimos $X \sim U(a, b)$

1.1.1. Función de densidad

Si $X \sim U(a, b)$ entonces la función de densidad es

$$f_X(x) = \frac{1}{b-a}$$

Para $x \in [a, b]$

1.1.2. Función de distribución

Si $X \sim U(a, b)$ entonces la función de distribución es

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

1.1.3. Función generadora de momentos

$$M_X(t) = \frac{e^{tb-e^{ta}}}{t(b-a)}$$

1.1.4. Curtosis

$$g_2 = \frac{-6}{5}$$

1.1.5. Media

$$E[X] = \frac{a+b}{2}$$

1.1.6. Varianza

$$\sigma^2 = \frac{(b-a)^2}{12}$$

1.1.7. Simulación

```python
import numpy as np

a = 0
b = 1
n = 10

#Generar 10 numeros aleatorios de la
    distribucion uniforme

uniform numbers = np.random.uniform(a, b, n)

print("Numeros aleatorios generados:" ,
    uniform numbers)
```

1.2. Exponencial

1.2.1. Función de densidad

Se dice que una variable aleatoria continua X tiene una distribución exponencial con parámetros $\lambda > 0$ y escribimos $X \sim Exp(\lambda)$ co fucnión de densidad

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

Para $x > 0$

1.2.2. Función de distribución

La función de distribución esta dada por

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

Para $x > 0$

1.2.3. Función generadora de momentos

$$M_x(t) = \left(1 - \frac{t}{\lambda}\right)^{-1}$$

1.2.4. Curtosis

$$g_2 = 9$$

1.2.5. Media

$$E[X] = \frac{1}{\lambda}$$

1.2.6. Varianza

$$\sigma^2 = \frac{1}{\lambda^2}$$

1.2.7. Simulación

```
import numpy as np

#Parametro lambda
lam = 0.5

#Generar 1000 valores aleatorios de una
    distribucion exponencial con parametro
    lambda

valores = np.random.exponential(1/lam, size =
    1000)

#Imprimir los primeros 10 valores generados
print(valores[:10])
```

1.3. Weibull

1.3.1. Función de densidad

Si X es una variable aleatoria continua, se dice que X tiene una distribución Weibull con parámetros α, $\lambda > 0$ y escribimos $X \sim Weibull(\alpha, \lambda)$ y su función de densidad y sy función esta dada por

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

1.3.2. Función de distribución

La función de distribución esta dada por

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

Para $x > 0$

1.3.3. Función generadora de momentos

$$M_X(t) = \sum_{n=0}^{\infty} \frac{t^n}{n!\lambda^n}\Gamma\left(1 + \frac{n}{\alpha}\right),\ \alpha > 0$$

1.3.4. Media

$$E[X] = \frac{1}{\lambda}\Gamma\left(1 + \frac{n}{\alpha}\right)$$

1.3.5. Varianza

$$\sigma^2 = \frac{1}{\lambda^2}\left[\Gamma\left(1 + \frac{2}{\alpha}\right) - \Gamma^2\left(1 + \frac{1}{\alpha}\right)\right]$$

1.3.6. Simulación

```
import numpy as np
from scipy.stats import uniform

def weibull(alpha, lam, size = 1):
    u = uniform.rvs(size=size)
    x = lam * (-np.log(1-u)**(1/alpha))
    return x

#Simular una muestra de una distribucion de
    Weibull con parametros alpha = 2 y lambda
    = 1
x = weibull(alpha=2, size=1000)
```

1.4. Log-Normal

1.4.1. Función de densidad

Una variable aleatoria positiva X tiene una distribución lognormal con parámetros μ y σ y escribimos $X \sim Lognormal(\mu, \sigma^2)$, si el logaritmo natural de X sigue una distribución normal con media μ y varianza σ^2 esto es $X \sim N(\mu, \sigma^2)$ para obtener

$$f_X(x) = \frac{1}{\sigma x\sqrt{2\pi}e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}}$$

1.4.2. Función de distribución

La función de distribución esta dada por

$$F_X(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right)$$

Donde Φ es la función de distribución acumulada de una normal estándar $N(0, 1)$

1.4.3. Media

$$E[X] = e^{\mu + \frac{\sigma^2}{2}}$$

1.4.4. Varianza

$$\sigma^2 = \left(e^{\sigma^2} - 1\right)e^{2\mu + \sigma^2}$$

1.4.5. Simulación

```
import numpy as np

mu = 0.5
sigma = 0.2
size = 1000

samples = np.random.lognormal(mu, sigma, size)
```

1.5. Normal

1.5.1. Función de densidad

Denotaremos a una distribución normal como sigue $X \sim N(\mu, \sigma^2)$ entonces su función de densidad esta dada por

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

1.5.2. Función de distribución

La función de distribución esta dada por

$$\Phi_{\mu,\sigma^2}(x) = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right], \ x \ \in \ \mathbb{R}$$

Donde **erf=función eror de Gauss** que la podemos expresar como

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

1.5.3. Función generadora de momentos

$$M_X(t) = e^{\mu t + \frac{\sigma^2 t^2}{2}}$$

1.5.4. Curtosis

$$g_2 = 0$$

1.5.5. Media

$$E[X] = \mu$$

1.5.6. Varianza

$$\sigma^2$$

1.5.7. Simulación

```
import numpy as np

mu = 0
sigma = 1
n = 1000

samples = np.random.normal(mu, sigma, n)
```

2. Funciones discretas

2.1. Bernoulli

Si X es una variable aleatoria discreta que mide el número de éxitos y se realiza un único experimento con dos posibles resultados denotamos éxito y fracaso, se dice que la variable aleatoria X se distribuye como una Bernoulli de parámetros p con $0 < p < 1$ y escribimos $X \sim Bernoulli(p)$

2.1.1. Función de Probabilidad

Su función de probabilidad es

$$P[X = x] = p^x(1-p)^{1-x}, \quad x = 0, 1$$

2.1.2. Función de distribución

La función de distribución acumulada de una variable aleatoria Bernoulli esta dado por

$$F(x) = \begin{cases} 0 & si \quad x < 2 \\ 1-p & si \quad 0 \leq x < 1 \\ 1 & si \quad x \geq 1 \end{cases}$$

2.1.3. Función generadora de momentos

$$M_X(t) = 1 + p + pe^t$$

2.1.4. Curtosis

$$g_2 = \frac{3p^2 - 3p + 1}{p(1+p)}$$

2.1.5. Media

$$E[X] = p$$

2.1.6. Varianza

$$\sigma^2 = p(1-p)$$

2.1.7. Simulación

```
import random

def bernoulli(p):
    if random.random() < p:
        return 1
    else:
        return 0
```

2.2. Uniforme discreta

Si X es una variable aleatoria discreta cuyo soporte es el conjunto $\{x_1, x_2, \ldots, x_n\}$ y tiene una distribución uniforme discreta entonces lo escribiremos como $X \sim Uniforme(x_1, x_2, \ldots, x_n)$

2.2.1. Función de probabilidad

La función de probabilidad es

$$P[X = x] = \frac{1}{n}$$

Para $x = x_1, x_2, \ldots, x_n$

2.2.2. Media

$$E[X] = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2.2.3. Varianza

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - E[X])^2$$

2.2.4. Simulación

```
import random

# Genera un numero aleatorio en el rango de 1
    a 6
num_aleatorio = random.randint(1, 6)

print(num_aleatorio)  # Imprime el numero
    aleatorio generado
```

2.3. Binomial

Si una variable aleatoria discreta X tiene una distribución binomial con parámetros $n \in \mathbb{N}$ y p con $0 < p < 1$ entonces escribiremos $X \sim Bin(n, p)$

2.3.1. Función de probabilidad

Su función de probabilidad esta dada por

$$P[X = x] = \binom{n}{x} p^x (1-p)^{n-x}$$

Para $x = 0, 1, 2, \ldots, n$, donde

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

2.3.2. Función de distribución

$$F_X(x) = P[X \leq x] = \sum_{k=0}^{x} \binom{n}{k} p^k (1-p)^{n-k}$$

2.3.3. Función generadora de momentos

$$M_X(t) = (1 - p + pe^t)^n$$

2.3.4. Curtosis

$$g_2 = \frac{1 - 6p(1-p)}{np(1-p)} + 3$$

2.3.5. Media

$$E[X] = np$$

2.3.6. Varianza

$$\sigma^2 = np(1-p)$$

2.3.7. Simulación

```
import numpy as np

n = 20
p = 0.3
size = 10

binomial_values = np.random.binomial(n, p,
    size)
print(binomial_values)
```

3. Geométrica

Si una variable aleatoria discreta X sigue una distribución geométrica con parámetros $0 < p < 1$ entonces escribiremos $X \sim Geometrica(p)$

3.0.1. Función de probabilidad

La función de probabilidad podemos verla como

$$P[X = x] = p(1-p)^{x-1}$$

Para $x = 1, 2, 3, \ldots$

3.0.2. Función de distribución

La función de distribución esta dada por

$$P[X \leq x] = 1 - (1-p)^x$$

Para $x = 0, 1, 2, 3, \ldots$

3.0.3. Función generadora de momentos

$$M_X(t) = \frac{p}{1 - (1-p)e^t}$$

3.0.4. Curtosis

$$g_2 = 6 + \frac{p^2}{1-p}$$

3.0.5. Media

$$E[X] = \frac{1}{p}$$

3.0.6. Varianza

$$\sigma^2 = \frac{1-p}{p^2}$$

3.0.7. Simulación

```
import numpy as np

# Definir el parametro de probabilidad
p = 0.5

# Simular una muestra de tamano 100 de la
    distribucion geometrica con parametro p
muestra_geom = np.random.geometric(p, size
    =100)

# Imprimir los valores generados
print(muestra_geom)
```

3.1. Poisson

Sea $\lambda > 0$ y X una variable aleatoria discreta, si la variable aleatoria X tiene una distribución de Poisson con parámetros λ entonces escribiremos $X \sim Poisson(\lambda)$

3.1.1. Función de probabilidad

La función de probabilidad esta dada por

$$P[X = x] = \frac{e^{-\lambda}\lambda^k}{k!}$$

Donde $k = 0, 1, 2, 3, \ldots$

3.1.2. Función de distribución

La función de distribución queda de la siguiente manera

$$\frac{\Gamma(\lfloor k+1 \rfloor, \lambda)}{\lfloor k \rfloor !}$$

Para $k \geq 0$ donde $\Gamma(x, y)$ es una función gamma incompleta, es decir

$$\Gamma(a, x) = \int_x^{\infty} t^{a-1} e^{-t} dt$$

3.1.3. Función generadora de momentos

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

3.1.4. Curtosis

$$g_2 = 3 + \lambda^{-1}$$

3.1.5. Media

$$E[X] = \lambda$$

3.1.6. Varianza

$$\sigma^2 = \lambda$$

3.1.7. Simulación

```
import numpy as np

 lam = 5 # valor de lambda
 simulacion = np.random.poisson(lam, 1000) #
     simulacion de 1000 valores aleatorios
 print(simulacion)
```

4. Metodología de la simulación

1. Identificación de variables: aquellas cuyo comportamiento define el comportamiento o la evolución global del sistema real

2. Determine la distribución de probabilidad: Elija el tipo de distribución de probabilidad que mejor defina el comportamiento aleatorio de cada una de las variables del sistema identificadas en el paso anterior

3. Modele las variables aleatorias: Construya un modelo fiel de la aleatoriedad de las variables del sistema. Simular el comportamiento del sistema implica generar muestras aleatorias de cada variable que sigan fielmente la distribución de probabilidad correspondiente

4. Defina el modelo del sistema y los objetivos de la simulación: Fije los objetivos y diseñe un modelo del sistema real que abstraiga sus propiedades más relevantes. Todo error en el modelo (o su implementación computacional) puede dar lugar a conclusiones equivocadas.

   Antes de experimentar con el modelo, asegurarse de que refleja fielmente el sistema real validando el mismo a partir de datos reales, para ello compare los resultados obtenidos de la simulación con los producidos por el sistema.

5. Diseño del experimento: Diseñe un experimento así como las ejecuciones concretas del mismo, los escenarios que desea estudiar para alcanzar los objetivos fijados. El experimento consiste en generar valores de las variables cuyo comportamiento define el comportamiento del sistema y analizar este ante dichos valores.

6. Repita el experimento n veces: Así dispondrá de observaciones sobre el comportamiento del sistema, lo que permitirá entender su funcionamiento así como evaluar el desempeño del mismo frente a los diversos escenarios establecidos.

7. Obtener la gráfica de estabilización

8. Calculo de probabilidad

9. Hallar el intervalo de confianza con

$$\overline{x} - z_{\alpha/2}\frac{s}{\sqrt{n}} < \mu < \overline{x} + z_{\alpha/2}\frac{s}{\sqrt{n}}$$

5. Ley débil de los grandes números

Sea $X_1, X_2, \leq$ una sucesión de variables aleatorias independientes e idénticamente distribuidas con media μ. Entonces para cada $\varepsilon > 0$

$$p\left\{\left|\frac{X_1 + \ldots + X_n}{n} - \mu\right| < \varepsilon\right\} \longrightarrow 0, \quad cuando \;\; n \longrightarrow \infty$$

6. Ley fuerte de los grandes números

Bajo las condiciones del teorema anterior, se tiene con probabilidad 1, que

$$\lim_{n\to\infty} \frac{X_1 + \ldots + X_n}{n} = \mu$$

7. Teorema central del limite

$$\lim_{n\to\infty} P\left\{\frac{X_1 + \ldots + X_n - n\mu}{\sigma\sqrt{n}} < x\right\} = \varphi(x)$$

Donde $\varphi = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} dx$, para $-\infty < x < \infty$, es la distribución normal o Gaussiana

8. Cálculo de integrales por el método de Monte Carlo

Consideremos aquí la estimación de integrales unidímensionales

$$I = \int_a^b g(x)dx$$

Suponiendo que

$$0 \leq g(x) \leq c, a \leq x \leq b$$

9. Técnicas de reducción de varianza

9.1. Variables antiteticas

Las variables antitéticas se emplean en cálculos de Monte Carlo con el fin de disminuir la variabilidad del estimador. Este método implica la generación de dos variables aleatorias con la misma distribución, pero con signos opuestos. A continuación, se utiliza la diferencia entre estas dos variables para calcular el valor deseado. Esta técnica reduce la variabilidad al considerar la correlación negativa existente entre las variables antitéticas.

9.2. Variables de control

Las variables de control se emplean con el propósito de disminuir la variabilidad de un estimador al introducir una variable extra que guarda correlación con la variable principal de interés. Esta estrategia se utiliza cuando se dispone de una variable adicional que explica parte de la variabilidad en la variable principal. Al controlar dicha variable adicional, se logra reducir la variabilidad del estimador.

9.3. Jackknife

El jackknife es un método de estimación no paramétrica utilizado para analizar la variabilidad de un estimador. Consiste en obtener múltiples estimaciones al excluir un solo dato en cada iteración, y luego calcular la varianza de estas estimaciones. De esta manera, se obtiene una medida de la variabilidad del estimador original.

9.4. Bootstrap

El bootstrap es una técnica de resampling utilizada para estimar la variabilidad de un estimador sin realizar suposiciones acerca de la distribución de los datos. Consiste en generar muestras aleatorias con reemplazo a partir de la muestra original y calcular el estimador en cada una de estas muestras. Posteriormente, se utiliza la variabilidad de los estimadores calculados en las muestras bootstrap para estimar la variabilidad del estimador original.

10. Simulación de líneas de espera

10.1. Parámetros de simulación

1. Tiempo de simulación (en unidades de tiempo)
2. Número de servidores
3. Tasa de llegada promedio de los clientes (en clientes por unidad de tiempo)
4. Tasa de servicio promedio de cada servidor (en clientes por unidad de tiempo)
5. Capacidad máxima del sistema (opcional, en número máximo de clientes permitidos)

10.2. Métricas de rendimiento a analizar

1. Tiempo promedio de espera de los clientes en la cola
2. Tiempo promedio de espera total de los clientes en el sistema (cola + servicio)
3. Longitud promedio de la cola
4. Utilización promedio de los servidores
5. Probabilidad de que no haya clientes en el sistema
6. Probabilidad de que un cliente tenga que esperar en la cola
7. Probabilidad de que un cliente no sea atendido debido a la capacidad máxima

10.3. Opciones de ejecución

1. Número de simulaciones a realizar
2. Intervalo de confianza deseado para las estimaciones (opcional)

10.4. Resultados de simulación

1. Estimaciones de las métricas de rendimiento calculadas para cada simulación realizada
2. Promedio y desviación estándar de las métricas de rendimiento a través de las simulaciones
3. Intervalo de confianza (si se especificó) para las estimaciones

11. Inventarios

11.1. Formulas

- Cantidad de pedido económico (EOQ, por sus siglas en inglés):

$$\text{EOQ} = \sqrt{\frac{2 \cdot D \cdot S}{H}}$$

- Punto de reorden (ROP, por sus siglas en inglés):

$$\text{ROP} = D \cdot \text{LT}$$

- Inventario promedio:

$$\text{Inventario promedio} = \frac{\text{EOQ}}{2}$$

- Costo total de inventario:

$$\text{Costo total de inventario} = \frac{D \cdot S}{\text{EOQ}} + \frac{\text{EOQ}}{2} \cdot H$$

11.2. Formulas avanzadas

- Modelo de revisión periódica (Q, r):

$$Q = \sqrt{\frac{2 \cdot D \cdot S}{H}}$$

$$r = \text{ROP} - (D \cdot \text{LT})$$

- Nivel de servicio (Service Level):

$$\text{Nivel de servicio} = \frac{\text{Demanda durante el tiempo de entrega} + \text{Stock de seguridad}}{\text{Demanda total}}$$

- Stock de seguridad:

$$\text{Stock de seguridad} = Z \cdot \sqrt{\text{Demanda durante el tiempo de entrega} \cdot \text{Varianza de la demanda durante el tiempo de entrega}}$$

- Punto de pedido (ROP) con demanda variable:

$$\text{ROP} = (D \cdot \text{LT}) + (Z \cdot \sqrt{D \cdot \text{LT} \cdot \text{Varianza de la demanda durante el ti}}$$

- Stock máximo (SM):

$$\text{SM} = \text{ROP} + (Z \cdot \sqrt{D \cdot \text{LT} \cdot \text{Varianza de la demanda durante el t}}$$

12. Simulación de Líneas de Espera

12.1. Parámetros de la simulación

Tiempo de simulación (en unidades de tiempo):
Número de servidores:
Tasa de llegada promedio de los clientes (en clientes por unidad de tiempo):
Tasa de servicio promedio de cada servidor (en clientes por unidad de tiempo):
Capacidad máxima del sistema (opcional, en número máximo de clientes permitidos):

12.2. Métricas de rendimiento a analizar

Tiempo promedio de espera de los clientes en la cola:
Tiempo promedio de espera total de los clientes en el sistema (cola + servicio):
Longitud promedio de la cola:
Utilización promedio de los servidores:
Probabilidad de que no haya clientes en el sistema:
Probabilidad de que un cliente tenga que esperar en la cola:
Probabilidad de que un cliente no sea atendido debido a la capacidad máxima:

12.3. Opciones de ejecución

Número de simulaciones a realizar:
Intervalo de confianza deseado para las estimaciones (opcional):

12.4. Resultados de la simulación

Estimaciones de las métricas de rendimiento calculadas para cada simulación realizada:
Promedio y desviación estándar de las métricas de rendimiento a través de las simulaciones:
Intervalo de confianza (si se especificó) para las estimaciones:

12.5. Análisis y conclusiones

Interpretación de los resultados obtenidos:
Recomendaciones para mejorar el rendimiento del sistema de espera, si corresponde: