Actividad 1: Simulación Estocástica

Curso: TEMAS SELECTOS 1 (O25-LAT4032-1)
Profesor: Rubén Blancas Rivera
Alumnos: jujuju jajaja jojojo
Universidad: Universidad de las Américas Puebla
Fecha: 2025-08-15

Export this notebook to PDF with LaTeX using the provided `amsart_template.tplx` for a Times-like, AMS-style layout.

Command:

```
jupyter nbconvert --to pdf --template amsart_template.tplx actividad1_template.ipynb
```

Selection of exercises: *Indicate here whether you solved **evens** or **odds** only (teams max 3).*

1 Ejercicio

Si $x_0 = 5$ y $x_n = 2x_{n-1} \bmod 150$. Encontrar $x_1, \ldots, x_{10}$.

$$x_n = ax_{n-1} \bmod m$$

$10 = 2 * 5 \bmod 15020 = 10 * 5 \bmod 15040 = 20 * 5 \bmod 15080 = 40 * 5 \bmod 15010 = 80 * 5 \bmod 150 \vdots$

[ ]:
```
pseudoaleatorios = []

x0 = 5
a = 2
m = 150

for i in range(10):
    xn = (a * x0) % m
    x0 = xn
    pseudoaleatorios.append(xn)

print(pseudoaleatorios)
```

```
[10, 20, 40, 80, 10, 20, 40, 80, 10, 20]
```

2 Ejercicio

$$\int_0^1 \exp(e^x)\,dx$$

Sea

$$\theta = \int_0^1 \exp(e^x)\,dx.$$

Reescritura como valor esperado con $U \sim \mathrm{Unif}(0,1)$:

$$\theta = \mathbb{E}\left[\exp(e^U)\right].$$

Estimador Monte Carlo con $u_1, \ldots, u_K \overset{\text{iid}}{\sim} \mathrm{Unif}(0,1)$:

$$\hat{\theta}_K = \frac{1}{K}\sum_{i=1}^{K} \exp(e^{u_i}).$$

[ ]:
```
import numpy as np

def h(u):
    return np.exp(np.exp(u))

k = 1000

u = np.random.random(k)

h(u).mean()
```

[ ]: np.float64(6.229521839385485)

3 Ejercicio

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

Sea

$$\theta = \int_{-2}^{2} e^{x+x^2}\,dx.$$

Cambio de variable a ([0,1]):

$$u = \frac{x-(-2)}{2-(-2)} = \frac{x+2}{4}, \qquad x = -2+4u, \qquad dx = 4\,du.$$

Entonces

$$\theta = \int_{0}^{1} 4\,\exp\big[(-2+4u)+(-2+4u)^2\big]\,du.$$

Forma de valor esperado con $U \sim \mathrm{Unif}(0,1)$:

$$\theta = \mathbb{E}[g(U)]\,, \qquad g(u) = 4\,\exp\big[(-2+4u)+(-2+4u)^2\big].$$

Estimador Monte Carlo:

$$\hat{\theta}_K = \frac{1}{K}\sum_{i=1}^{K} g(u_i), \qquad u_i \overset{iid}{\sim} \mathrm{Unif}(0,1).$$

[ ]:
```
def h(u):
    return (b-a)*np.exp(a+(b-a)*u + (a+(b-a)*u)**2)

k = 10000
a = -2
b = 2
u = np.random.random(k)
h(u).mean()
```

[ ]:
```
np.float64(94.60588956711096)
```

4 Ejercicio

$$\int_0^{\infty} \frac{x}{(1+x^2)^2}\,dx$$

Sea

$$\theta = \int_0^{\infty} \frac{x}{(1+x^2)^2}\,dx.$$

Cambio (pág. 21):

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

Entonces

$$\theta = \int_0^1 h(y)\,dy, \qquad h(y) = \frac{g\left(\frac{1}{y}-1\right)}{y^2}, \quad g(x) = \frac{x}{(1+x^2)^2}.$$

Cálculo explícito de h:

$$x = \frac{1-y}{y} \;\Rightarrow\; h(y) = \frac{(1-y)\,y}{\left(1-2y+2y^2\right)^2}, \qquad y \in (0,1).$$

Forma de esperanza con $U \sim \text{Unif}(0,1)$:

$$\theta = \mathbb{E}[\,h(U)\,].$$

Estimador Monte Carlo:

$$\hat{\theta}_K = \frac{1}{K}\sum_{i=1}^{K} h(u_i), \quad u_i \overset{\text{iid}}{\sim} \text{Unif}(0,1).$$

Chequeo analítico:

$$\theta = \int_0^{\infty} \frac{x}{(1+x^2)^2}\,dx = -\tfrac{1}{2}\int_0^{\infty} d\left(\frac{1}{1+x^2}\right) = \tfrac{1}{2}.$$

[ ]:
```
def h(u):
    return ((((1/u)-1)/(1+((1/u)-1)**2)**2))/(u**2)

k = 10000000

u = np.random.random(k)
h(u).mean()
```

[ ]:
```
np.float64(0.5000960390593421)
```

Sea

$$\theta = \int_0^{\infty} \frac{x}{(1+x^2)^2}\,dx.$$

1) Integral impropia:

$$\theta = \lim_{b\to\infty} \int_0^{b} \frac{x}{(1+x^2)^2}\,dx.$$

2) Sustitución $u = 1 + x^2 \Rightarrow du = 2x\,dx$: cuando $x = 0 \Rightarrow u = 1$, cuando $x = b \Rightarrow u = 1 + b^2$. Entonces

$$\int_0^b \frac{x}{(1+x^2)^2}\,dx = \frac{1}{2}\int_1^{1+b^2} u^{-2}\,du.$$

3) Primitiva:

$$\int u^{-2}\,du = -u^{-1} + C.$$

4) Evaluación:

$$\frac{1}{2}\Big[-u^{-1}\Big]_1^{1+b^2} = \frac{1}{2}\left(-\frac{1}{1+b^2} + 1\right).$$

5) Límite:

$$\theta = \lim_{b\to\infty} \frac{1}{2}\left(1 - \frac{1}{1+b^2}\right) = \frac{1}{2}.$$

(Equivalente por antiderivación directa: $\displaystyle\int \frac{x}{(1+x^2)^2}dx = -\frac{1}{2(1+x^2)} + C$, y luego $\theta = \lim_{b\to\infty}\big[-\frac{1}{2(1+x^2)}\big]_0^b = \frac{1}{2}$.)

5 Ejercicio

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

Sea

$$\theta = \int_0^1 \int_0^1 e^{(x+y)^2}\,dy\,dx.$$

Notación de "integrales múltiples":

$$\theta = \int_0^1 \int_0^1 g(x_1, x_2)\,dx_1\,dx_2, \qquad g(x_1, x_2) = e^{(x_1+x_2)^2}.$$

Por las diapositivas:

$$\theta = \mathbb{E}[g(U_1, U_2)], \quad U_1, U_2 \overset{iid}{\sim} \text{Unif}(0,1). \text{ (Ley de la esperanza en el hipercubo)}$$

:contentReferenceoaicite:0

Estimador Monte Carlo con (k) muestras:

$$\hat{\theta}_k = \frac{1}{k}\sum_{i=1}^{k} g(u_{i1}, u_{i2}) = \frac{1}{k}\sum_{i=1}^{k} \exp((u_{i1} + u_{i2})^2), \quad (u_{i1}, u_{i2}) \overset{iid}{\sim} \text{Unif}(0,1)^2.$$

Algoritmo (formato de las diapositivas): inicialice (S←0). Para (i=1,...,k): genere (u_{i1},u_{i2} U(0,1)), actualice (S←S+g(u_{i1},u_{i2})). Devuelva ($\hat{\theta}\_k = S/k$). $: contentReferenceoaicite : 1$

Error de simulación:

$$\text{se}(\hat{\theta}_k) \approx \frac{s_g}{\sqrt{k}}, \quad s_g^2 = \frac{1}{k-1}\sum_{i=1}^{k}\Big(g(u_{i1}, u_{i2}) - \hat{\theta}_k\Big)^2.$$

Sea

$$\theta = \int_0^1 \int_0^1 e^{(x+y)^2}\,dy\,dx = \mathbb{E}[g(U_1, U_2)], \quad g(x,y) = e^{(x+y)^2}, \quad U_1, U_2 \overset{iid}{\sim} \text{Unif}(0,1).$$

Estimador Monte Carlo:

$$\hat{\theta}_k = \frac{1}{k}\sum_{i=1}^{k} g(u_{i1}, u_{i2}).$$

[ ]:
```
def h(u1, u2):
    return np.exp((u1 + u2)**2)

k = 10000

u1 = np.random.random(k)
u2 = np.random.random(k)
h(u1, u2).mean()
```

[ ]: np.float64(1.2140555137345483)

6 Ejercicio

Usar simulación para aproximar $\mathrm{Cov}(U, e^U)$, donde $U \sim \mathcal{U}(0,1)$. Comparar con la respuesta exacta.

Asumo $U \sim \mathrm{Unif}(0,1)$.

6.1 Definición y equivalencia

Por definición,

$$\operatorname{Cov}(X,Y) = \mathbb{E}\big[(X - \mathbb{E}X)(Y - \mathbb{E}Y)\big].$$

Expansión lineal:

$$\operatorname{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y].$$

Aplicando a $X = U$ y $Y = e^U$:

$$\operatorname{Cov}(U,e^U) = \mathbb{E}\big[Ue^U\big] - \mathbb{E}[U]\,\mathbb{E}[e^U].$$

6.2 Cálculo analítico paso a paso

$$\mathbb{E}[U] = \int_0^1 u\,du = \frac{1}{2}.$$

$$\mathbb{E}[e^U] = \int_0^1 e^u\,du = e - 1.$$

$$\mathbb{E}[Ue^U] = \int_0^1 ue^u\,du = \big[ue^u\big]_0^1 - \int_0^1 e^u\,du = e - (e-1) = 1.$$

Por tanto,

$$\boxed{\mathrm{Cov}(U, e^U) = 1 - \tfrac{1}{2}(e-1) = \tfrac{3-e}{2} \;\approx\; 0.140859086}$$

6.3 Estimación Monte Carlo

Muestree $u_1, \dots, u_K \overset{iid}{\sim} \mathrm{Unif}(0,1)$. Defina

$$\hat{\mu}_U = \frac{1}{K}\sum_{i=1}^{K} u_i, \qquad \hat{\mu}_e = \frac{1}{K}\sum_{i=1}^{K} e^{u_i}, \qquad \widehat{m} = \frac{1}{K}\sum_{i=1}^{K} u_i e^{u_i}.$$

Estimador por identidad de momentos:

$$\boxed{\widehat{\mathrm{Cov}}^{(MC)} = \widehat{m} - \hat{\mu}_U\,\hat{\mu}_e}$$

(Consistente cuando $K \to \infty$.)

Opcional, versión insesgada muestral:

$$\boxed{\widehat{\mathrm{Cov}}^{(\mathrm{unb})} = \frac{1}{K-1}\sum_{i=1}^{K}(u_i - \bar{u})(e^{u_i} - \overline{e^u}), \quad \bar{u} = \hat{\mu}_U,\ \overline{e^u} = \hat{\mu}_e}$$

Error Monte Carlo aproximado para $\widehat{\mathrm{Cov}}^{(MC)}$:

$$\mathrm{se}\left(\widehat{\mathrm{Cov}}^{(MC)}\right) \approx \frac{s}{\sqrt{K}},$$

donde s^2 es la varianza empírica de $u_i e^{u_i} - \hat{\mu}_U e^{u_i} - \hat{\mu}_e u_i + \hat{\mu}_U \hat{\mu}_e$.

[47]:
```
def valor_esperado_1(u):
    return u * np.exp(u)

def valor_esperado_2(u):
    return u

def valor_esperado_3(u):
    return np.exp(u)

k = 1000000

u = np.random.random(k)

valor_esperado_1(u).mean() - valor_esperado_2(u).mean() * valor_esperado_3(u).
 ↪mean()
```

[47]: np.float64(0.1409069042870994)

[48]:
```
1 - 1/2*(np.e -1)
```

[48]: 0.14085908577047745

7 Ejercicio

Para variables aleatorias uniformes $U_1, U_2, ...$ definir

$$N = \min\left\{n : \sum_{i=1}^{n} U_i > 1\right\}.$$

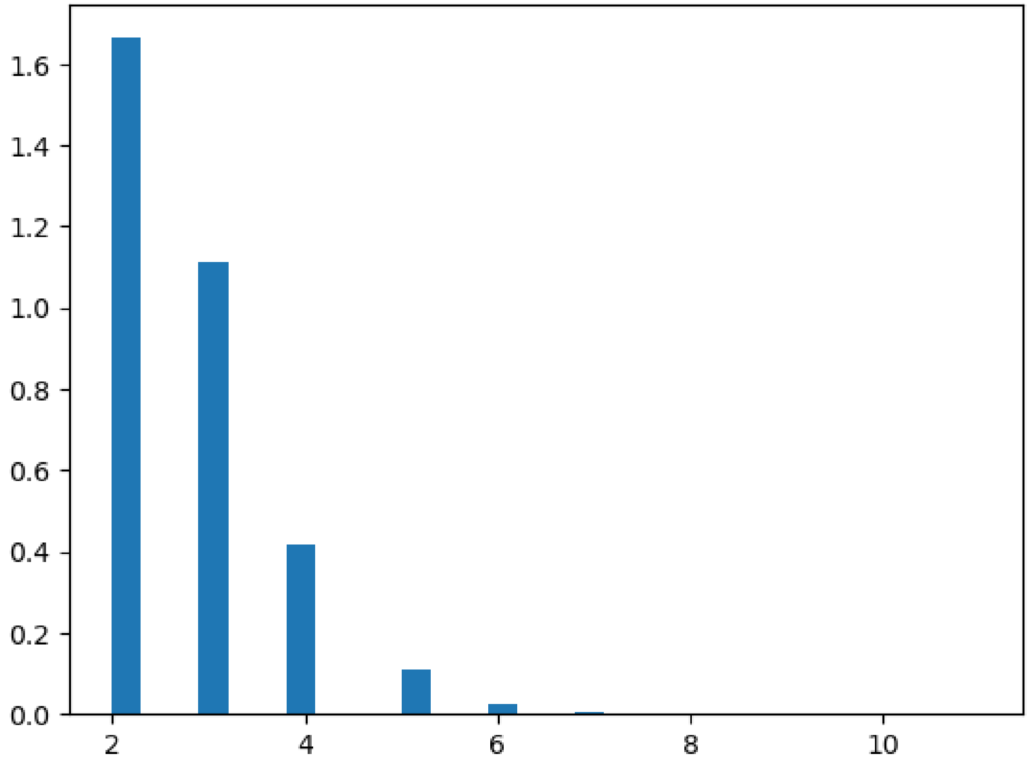
Estimar $\mathbb{E}[N]$ por simulación con: a) 100 valores, b) 1000 valores, c) 10000 valores, d) Discutir el valor esperado.

```
PSEUDOCÓDIGO - MINIMO_N(k)
total_contadores ← 0
PARA i ← 1 HASTA k HACER:
    suma ← 0
    contador ← 0
    MIENTRAS suma < 1 HACER:
        contador ← contador + 1
        suma ← UNIFORME(0,1)
    total_contadores ← total_contadores + contador
RETORNAR total_contadores / k
```

[ ]:
```python
def minimo_N(k):
    lista_contadores = []
    for _ in range(k):
        suma = 0
        contador = 0
        while suma < 1:
            contador += 1
            suma += np.random.random()
        lista_contadores.append(contador)
    return lista_contadores
```

[89]:
```python
lista_para_dist = minimo_N(1000000)
```

[90]:
```python
import matplotlib.pyplot as plt
plt.hist(lista_para_dist, bins=30, density=True)
plt.show()
```



[80]:
```
minimo_N(1_000)
```

[80]: np.float64(2.725)

[81]:
```
minimo_N(10_000)
```

[81]: np.float64(2.7227)