

ELEMENTOS DE CÁLCULO NUMÉRICO / CÁLCULO NUMÉRICO

Primer Cuatrimestre 2021

Segundo ejercicio computacional: Caminata de aleatoria

05/04/21 al 12/04/21

Recuerde subir el archivo en formato `ejercicioX_NOMBREPELLIDO.py`

Recuerde al hacer consultas postear su código

Consideremos la caminata aleatoria (*random walk*) que consiste en un proceso que representa un movimiento al azar entre distintas posiciones sobre el espacio. En el caso unidimensional, se avanza o retrocede aleatoriamente, partiendo de un punto inicial (por ejemplo, $x = 0$).

En este ejercicio realizaremos una versión muy sencilla de la caminata, para explorar las funciones `for`, `if` y `while`.

Emplearemos la función `rand` del submodulo `random` del paquete `numpy` que genera un número al azar entre 0 y 1. Con la siguiente modificación, lo movemos al `-0.5, 0.5`:

```
import numpy.random as rd
x = rd.rand()-0.5
```

Escribámoslo dentro de una función, para hacer más cómodo todo lo que sigue:

```
def numero_random():
    x = rd.rand()-0.5
    return(x)
```

- A) Generar una lista `X=[]`, llenarla con `N=100` números aleatorios entre `-0.5` y `0.5`, y graficarla. Calcular el valor promedio de la lista usando `np.mean`. ¿Es acorde al resultado esperado?

Ejemplo de código: usamos una lista vacía y agregamos elementos usando `append`:

```
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
X = []
N = 100
for q in range(N):
    X.append(numero_random())
print(np.mean(X))
plt.plot(X)
plt.savefig('grafico_todos_independientes.png')
```

- B) Usando que `X[-1]` devuelve el último valor de `X`, generar nuevamente la lista `X` pero llenándola de forma tal que cada nuevo valor sea el anterior más un número aleatorio. Es decir, cada nuevo valor de `X` es el último + un número aleatorio. Inicializar `X=[0]`.

```

X= [0]
N = 99
for q in range(N):
    X_nuevo = X[-1] + # COMPLETAR
    X.append(#COMPLETAR)

```

Comparar con el gráfico obtenido en el punto anterior. ¿Hay alguna diferencia? ¿El promedio es distinto?

- C) Construir una función que reciba un número de pasos N y devuelva una lista X como la del punto B), y usarla para graficar 50 trayectorias:

```

def genera_X(N):
    #COMPLETAR
    return(X)

for q in range(50):
    X = #COMPLETAR
    plt.plot(X)
plt.savefig('muchastrayectorias.png')

```

- D) Supongamos que ahora la caminata al azar se corta o bien cuando el caminante llega a N pasos o bien cuando llega a una pared ubicada en W o $-W$. En ese caso queremos cortar la caminata cuando se cumpla alguna de esas dos condiciones. Usando un `while` podemos hacerlo de forma sencilla, todo en uno. Aquí un ejemplo con $N=100$ y $W=3$:

```

X = [0]
W = 3
N = 100
iterador = 0
while np.abs(X[-1])<W and iterador < N:
    X_nuevo = # COMPLETAR
    X.append(#COMPLETAR)
    iterador = iterador +1

```

Usando un ciclo `for` generar 500 trayectorias del caminante (cada una con un máximo de 100 pasos y usando $W=3$) y registrar la cantidad de iteraciones necesarias. ¿Cuántas realiza en promedio?

```

def X_con_W(N,W):
    X = [0]
    iterador = 0
    while np.abs(X[-1])<W and iterador < N:
        X_nuevo = # COMPLETAR
        X.append(#COMPLETAR)
        iterador = iterador +1

```

```

    return(X)

W=3
N=100
Ntrayect = 500
iteraciones = []
for q in range(Ntrayect):
    X = X_con_W(N,W)
    iteraciones.append(len(X))
print(np.mean(#COMPLETAR))

```

- E) Ahora vamos a volver a generar 500 trayectorias del caminante y vamos a usar **for** e **if** para registrar el máximo y mínimo valor de cada una de ellas. ¿Cuál es el mínimo promedio? ¿Y el máximo?

```

def minimo_y_maximo(X):
    minimo = np.inf #Qué representa?
    maximo = -np.inf #Por qué estos valores?
    for x in X:
        if minimo>x:
            # COMPLETAR
        if #COMPLETAR:
            maximo = x
    return([minimo,maximo])

```

```

W=3
N=100
Ntrayect = 500
minTotal = []
maxTotal = []
for q in range(Ntrayect):
    X = X_con_W(N,W)
    minTotal.append(#COMPLETAR)
    maxTotal.append(#COMPLETAR)
print(np.mean(#COMPLETAR))
print(np.mean(#COMPLETAR))

```