

Práctica 1 Movimiento Browniano

Raul Lagunes Rivera

2125677

13 de febrero de 2022

1. Introducción

La práctica número 1 estudia el tema Movimiento Browniano, donde se debe analizar el tiempo de regreso al origen de dicha partícula en 5 dimensiones , variando el número de pasos de la caminata, con 30 repeticiones del experimento para cada combinación[1].

2. Objetivo

El objetivo de la simulación es examinar de manera sistemática los efectos de la dimensión en la distancia Euclideana máxima del origen del movimiento browniano para cinco dimensiones, variando el número de pasos de la caminata (100,1000,10000 pasos), con 30 repeticiones del experimento para cada combinacion y grafica los resultados en una sola figura diagrama caja-bigote

3. Código

En el siguiente código se utilizaron secuencias for para realizar todo el objetivo propuesto en una sola ejecución.El código base se sacó del repertorio de la Dra. Elisa Schaeffer.

Codigo en phyton

<https://github.com/satuelisa/Simulation/blob/master/BrownianMotion>

****Código creado en Python****

```
from random import random, randint
from math import fabs, sqrt
import matplotlib.pyplot as plt
import numpy as np
DIMS=1,2,3,4,5  cuantas dimensiones
caminatas= 100, 1000, 10000
replicas = 30  cuantas veces
CB_100=[]
CB_1000=[]
CB_10000=[]
for dim in DIMS:
    print("##### Dimencion:",dim,"#####")
    cien=[]
    mil=[]
    diezmill=[]
```

```

for replica in range (replicas):
    pos=[0]* dim
    mayor= 0
    re=0
    mayores=[]
    for s in caminatas:
        for paso in range(s):
            eje= randint(0, dim - 1)
            if random()< 0.5:
                pos[eje] +=1
            else:
                pos[eje]-= 1
            mayor = max ( mayor, sqrt(sum([p**2 for p in pos])))
        mayores.append(mayor)
    cien.append(mayores[0])
    mil.append(mayores[1])
    diezmill.append(mayores[2])
    CB_100.append(cien)
    CB_1000.append(mil)
    CB_10000.append(diezmill)

print("cuantos datos de caja bigote",len(CB_100))
pb=plt.boxplot([CB_100[0],CB_1000[0],CB_10000[0],CB_100[1],
CB_1000[1],CB_10000[1],CB_100[2],CB_1000[2],CB_10000[2],
CB_100[3],CB_1000[3],CB_10000[3],CB_100[4],CB_1000[4],CB_10000[4]])
plt.xticks([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],
            ['100','1000','10000','100','1000','10000',
            '100','1000','10000','100','1000','10000','100','1000','10000'])

plt.xlabel('camintas')
plt.ylabel('Distancia máxima')
plt.tick_params(axis='x', rotation=45)
plt.title('Distancia Euclidiana')

colors = ['orange','orange','orange','red','red', 'red',
          'green','green','green', 'pink','pink','pink','blue','blue','blue']

for patch, color in zip(pb['boxes'], colors):
    patch.set_color(color)

plt.plot([], c='orange', label='dimencion 1')
plt.plot([], c='red', label='dimencion 2')
plt.plot([], c='green', label='dimencion 3')
plt.plot([], c='pink', label='dimencion 4 ')
plt.plot([], c='blue', label='dimencion 5')

plt.legend()

plt.savefig('figuraPy.png') # mandar a un archivos

print(pos)

```

4. Resultados

En la figura se muestra como se comporta cada dimension con diferente tamaño de camina , de color amarillo es la dimencion uno y sus tres respectivas caminas, y asi sucesivamente con las sinco dimensiones.

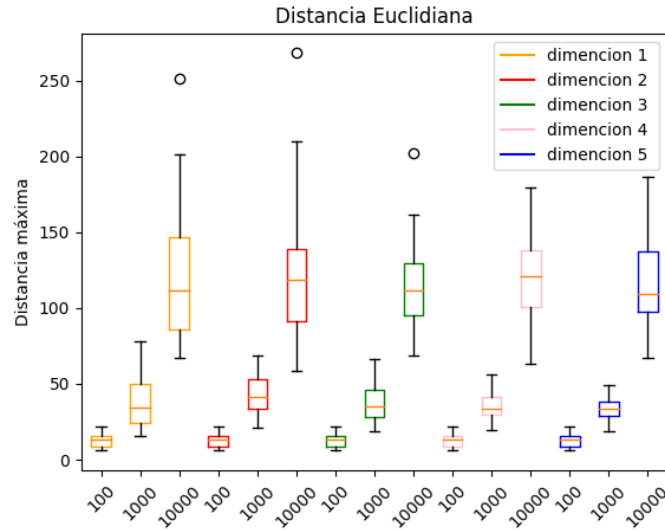


Figura 1: tomado del repositorio de Raul Lagunes Rivera del codigo de python <https://github.com/Raullr28/Resultados/blob/main/P1/figuraPy.png>

5. Conclusiones

se realizo la tarea numero 1 correctamente ,obteniendo la caja-bigote esperada con las caminas propuestas y las dimensiones deseadas

Referencias

- [1] schaafeffer.e. movimiento browniano. 2022. URL <https://elisa.dyndns-web.com/teaching/comp/par/p1.html>.