**Maths Assignment Q-3**

Question: A particle starts at the origin of the real line and moves along the line in jumps of one unit. For each jump the probability is p that the particle will jump one unit to the left and the probability is 1−p that the particle will jump one unit to the right. Let Xn be the position of the particle after n units. Find E(Xn) and V(Xn). (This is known as a random walk.)

We will simulate this in python so that we can understand it.

Before the code, what is random walk?

This concept/Theory is used in the stock market and it says that “Random walk theory is based on the idea that stock prices reflect all available information and adjust quickly to new information, making it impossible to act on it.” This is definition for stock markets(Found in <https://www.investopedia.com/terms/r/randomwalktheory.asp>)

Each step taken by the object in any direction has a probability associated with it. Hence, **the final position is completely independent of the point of origin.**

A simple example of a random walk is a [drunkard’s walk](https://en.wikipedia.org/wiki/The_Drunkard%27s_Walk). A drunk man has no preferential direction. Therefore, he’s equally likely to move in all directions.

This is the premise, it is equally likely to move in as many directions as possible based on the case. But in our case, it’s a one dimensional line. So there are only 2 directions or 2 choices and both are equally likely. All we can say is For each jump the probability is **p** that the particle will jump one unit to the left and the probability is **1−p** that the particle will jump one unit to the right.

Now lets start with the code:

1. **Initialization**: Initialize variables for the current position (**Xn**), expectation (**EXn**), sample space size (**n**), left counter (**l**), right counter (**r**), probability of moving left (**p\_l**), and probability of moving right (**p\_r**).

import random

xn = 0

exn = 0 # EXPECTATION

n = 10\*\*7 # SAMPLE SPACE

l = 0 # LEFT COUNTER

r = 0 # RIGHT COUNTER

p\_l = 0 # PROBABILITY OF MOVING LEFT

p\_r = 0 # PROBABILITY OF MOVING RIGHT

1. **Random Walk Simulation**: Perform a random walk simulation by generating a random number (**z**) between 1 and 2 (inclusive) for each step in the sample space.

for i in range(n):

z = random.randint(1, 2) # GENERATING 1 OR 2 RANDOMLY

1. **Move Left or Right**: Update the position of the particle (**Xn**) based on the random number generated. If the random number is 1, move left and increment the left counter (**l**); otherwise, move right and increment the right counter (**r**).

# RANDOMLY WALKING LEFT OR RIGHT IF EITHER 1 OR 2 IS GENERATED RESPECTIVELY

if z == 1:

l += 1 # IF Z WAS 1, WE INCREMENT THIS VARIABLE

xn -= 1 # HERE WE MAKE THE PARTICLE MOVE TO THE LEFT

else:

r += 1 # IF Z WAS 2, WE INCREMENT THIS VARIABLE

xn += 1 # HERE WE MAKE THE PARTICLE MOVE TO THE RIGHT

1. **Calculate Probabilities**: Calculate the probabilities of moving left (**p\_l**) and moving right (**p\_r**) based on the counters.

p\_l = l / n

p\_r = 1 - p\_l

1. **Estimate Expectation**: Estimate the expectation (**EXn**) using the calculated probabilities.

exn = (-1) \* p\_l + (1) \* p\_r # ESTIMATING EXPECTATION THAT IS '(-1)p + (1)(p-1)' WHERE 'p' IS THE PROBABILITY

1. **Calculate Variance**: Calculate the variance (**var\_x**) of the particle using the calculated probabilities.

var\_x = 4 \* p\_l - 4 \* p\_l\*\*2 # CALCULATING VARIANCE OF THE PARTICLE THAT IS E[x^2] - E[x]^2 OR '4^p - 4\*(p)^2' where E[x] IS EXPECTATION AND 'p' IS THE PROBABILITY

1. **Print Results**: Print the expected position, Expectation and variance.

# EXPECTED VALUE

print(f"\nThe expected position(Xn) of the particle after walking randomly for {n} steps is {xn} and the expectation(E[x]) is {exn}\n")

# VARIANCE

print(f"The variance[Var(x)] for the particle is {var\_x}\n")

Output:

A computer screen with white text

Description automatically generated

Full code:

import random

import matplotlib.pyplot as plt

plt.style.use('dark\_background')

xn = 0

exn = 0 # EXPECTATION

n = 10\*\*7 # SAMPLE SPACE

l = 0 # LEFT COUNTER

r = 0 # RIGHT COUNTER

p\_l = 0 # PROBABILITY OF MOVING LEFT

p\_r = 0 # PROBABILITY OF MOVING RIGHT

expectation\_values = []

variance\_values = []

for i in range(n):

z = random.randint(1, 2) # GENERATING 1 OR 2 RANDOMLY

# RANDOMLY WALKING LEFT OR RIGHT IF EITHER 1 OR 2 IS GENERATED RESPECTIVELY

if z == 1:

l += 1 # IF Z WAS 1, WE INCREMENT THIS VARIABLE

xn -= 1 # HERE WE MAKE THE PARTICLE MOVE TO THE LEFT

else:

r += 1 # IF Z WAS 2, WE INCREMENT THIS VARIABLE

xn += 1 # HERE WE MAKE THE PARTICLE MOVE TO THE RIGHT

p\_l = l / n

p\_r = 1 - p\_l

exn = (-1) \* p\_l + (1) \* p\_r # ESTIMATING EXPECTATION THAT IS '(-1)p + (1)(p-1)' WHERE 'p' IS THE PROBABILITY

var\_x = 4 \* p\_l - 4 \* p\_l\*\*2 # CALCULATING VARIANCE OF THE PARTICLE THAT IS E[x^2] - E[x]^2 OR '4^p - 4\*(p)^2' where E[x] IS EXPECTATION AND 'p' IS THE PROBABILITY

expectation\_values.append(exn)

variance\_values.append(var\_x)

# EXPECTED VALUE

print(f"\nThe expected position(Xn) of the particle after walking randomly for {n} steps is {xn} and the expectation(E[x]) is {exn}\n")

# VARIANCE

print(f"The variance[Var(x)] for the particle is {var\_x}\n")

fig, ax1 = plt.subplots(figsize=(8, 8))

# Instantiate a second axes that shares the same x-axis

ax2 = ax1.twinx()

ax2.set\_ylim(4, 20)

plt.title('Random Walk')

ax1.plot(n,expectation\_values)

ax2.plot(n,variance\_values)

plt.show()