Q1: Analyzing Markov Chains

1. Introduction

A finite state discrete-time Markov chain models a system where the probability of transitioning from one state to another depends only on the current state and not on how the system arrived at that state. In this case, we're studying the stages of writing a paper, which include reading (r), writing (w), e-mailing (e), and surfing (s).

Transition Probability Matrix P

The transition probabilities are given by the matrix P:

```
        | r   w   e   s |
P = | r | 0.5 0.3 0.0 0.2 |
    | w | 0.2 0.5 0.1 0.2 |
    | e | 0.1 0.3 0.3 0.3 |
    | s | 0.0 0.2 0.3 0.5 |
```

Here, P[i][j] represents the probability of transitioning from state i to state j in one time step.

(a) After 20 minutes

To find the state after 20 minutes, we perform matrix multiplication with the transition matrix P for 20 times:

```
P^20 = P * P * P * ... * P (20 times), the result of which is mentioned below.
```

```
P^20:
[[0.17073182 0.33604338 0.18157173 0.31165307]
 [0.17073177 0.33604337 0.18157177 0.31165309]
 [0.17073168 0.33604336 0.18157183 0.31165313]
 [0.17073159 0.33604334 0.1815719  0.31165317]]
```
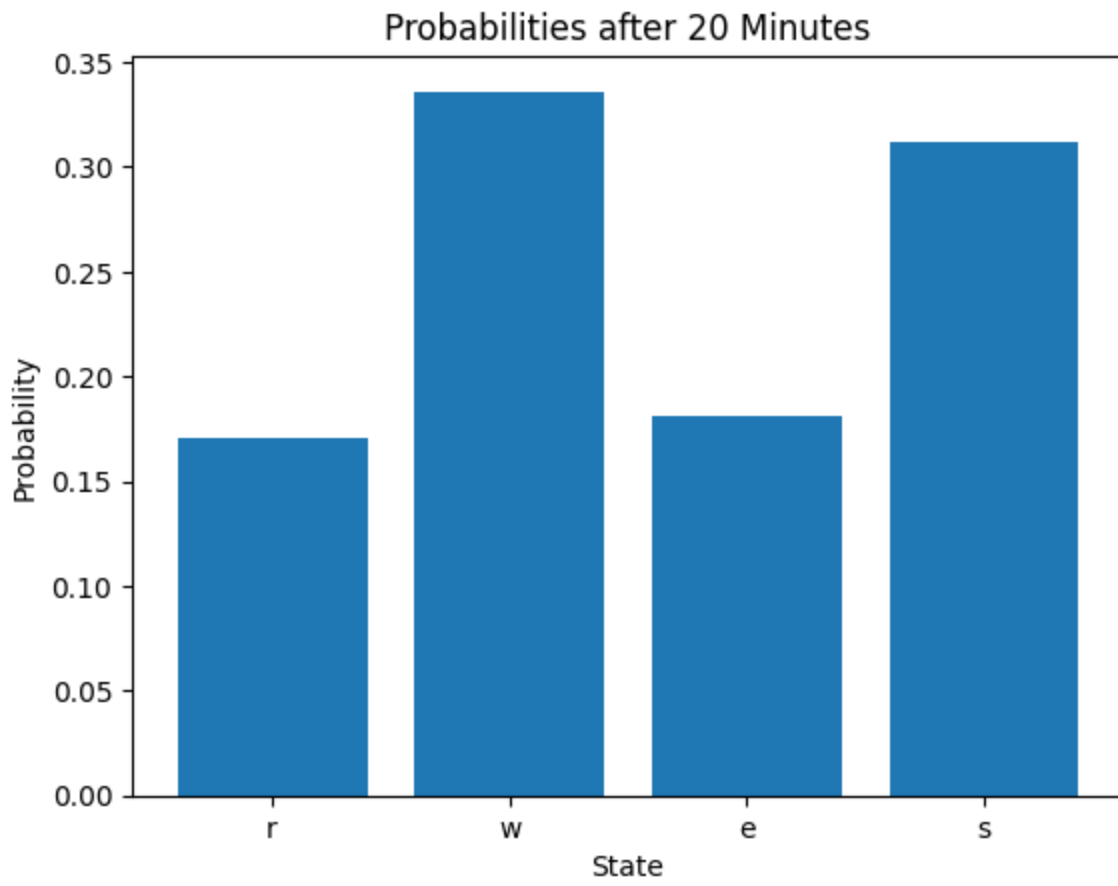
This gives us the probabilities of being in each state after 20 minutes.

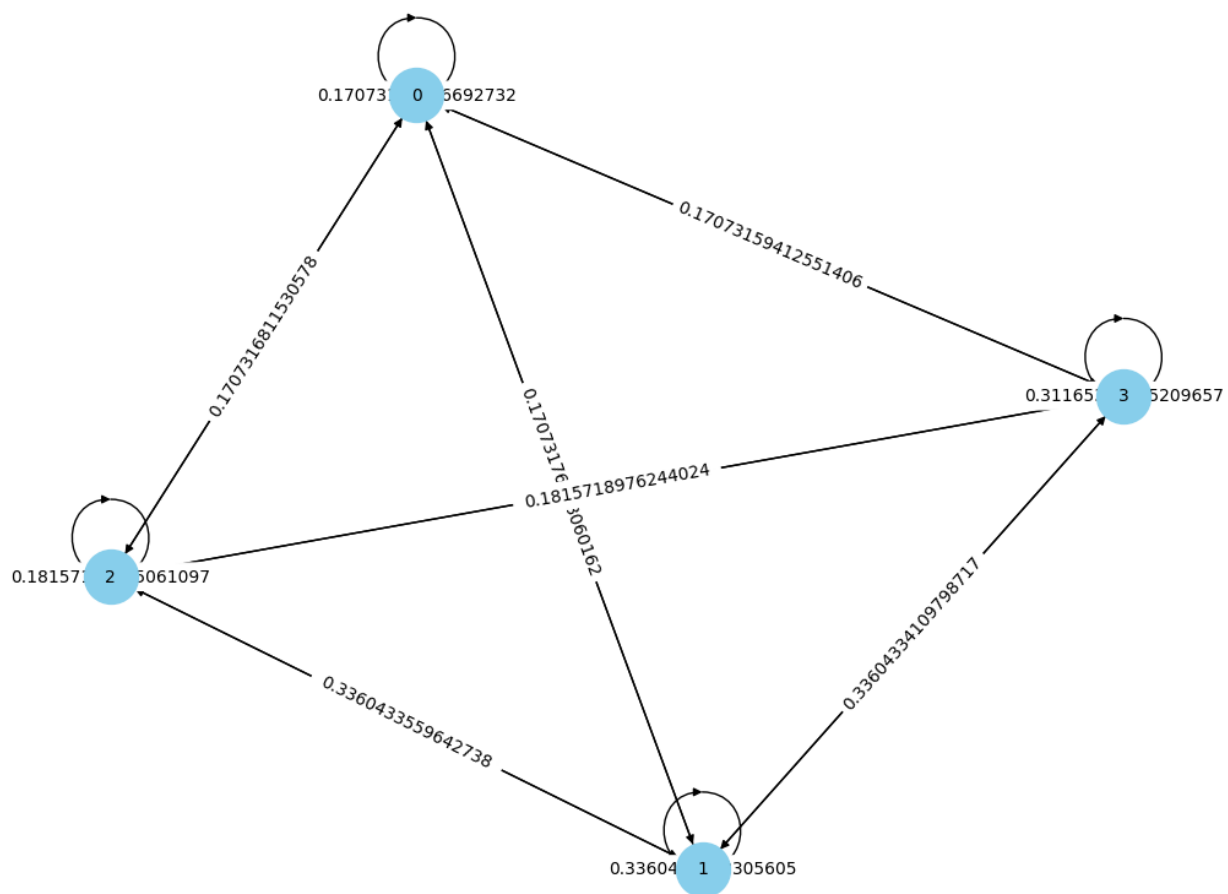$P(X20 = s|X0 = r)$ `= 0.3117 (Rounded-Off to 4 decimal places)`

```
For the precise result,
P(X20 = s | X0 = r) = 0.3116530652582661
```

This is the conditional probability of being in state s after 20 minutes given that we started in state r. This is equivalent to finding the 4th element in the resulting row of P^20 corresponding to starting state r.

Probabilities after 20 Minutes

P_20 Markov Chain

(b) After 25 minutes

Similarly, to find the state after 25 minutes, we perform matrix multiplication with the transition matrix P for 25 times:

```
P^25 = P * P * P * ... * P (25 times)
```

```
P^25:
[[0.17073171  0.33604336  0.18157181  0.31165312]
 [0.17073171  0.33604336  0.18157181  0.31165312]
 [0.17073171  0.33604336  0.18157182  0.31165312]
 [0.1707317   0.33604336  0.18157182  0.31165312]]
```
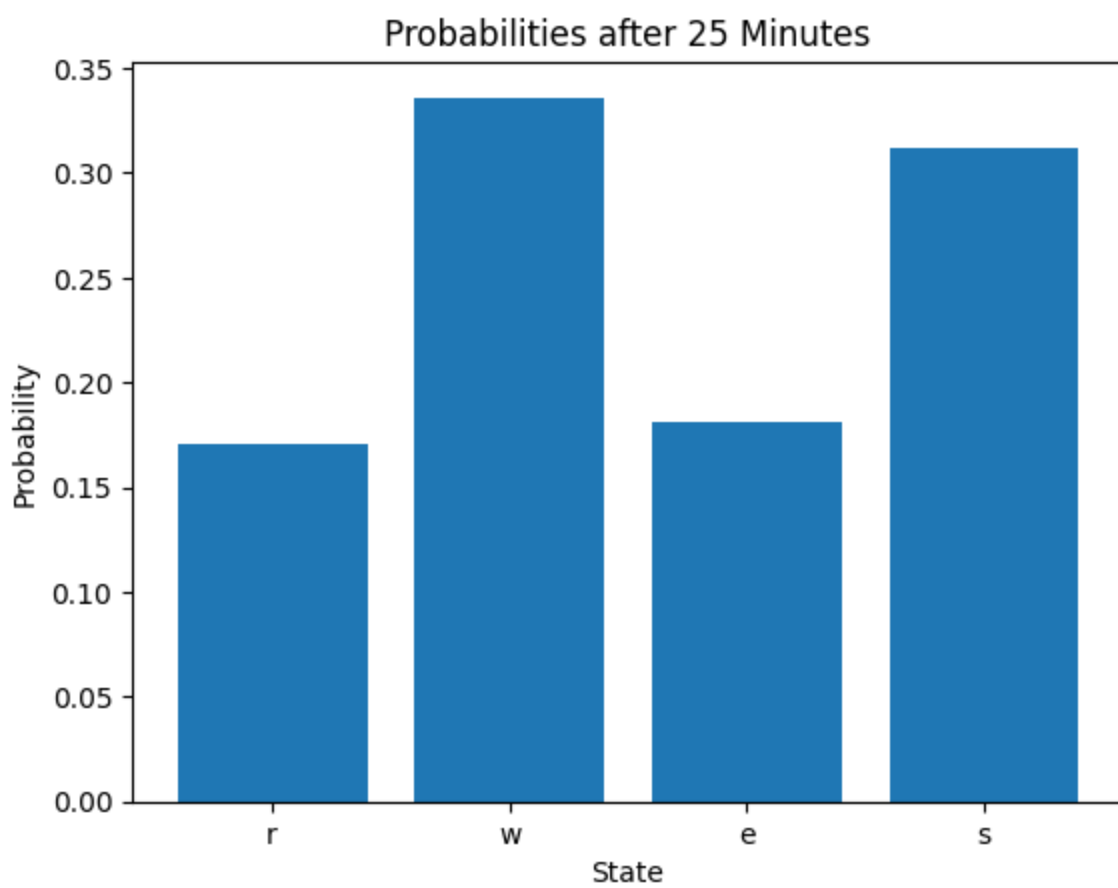
This gives us the probabilities of being in each state after 25 minutes.

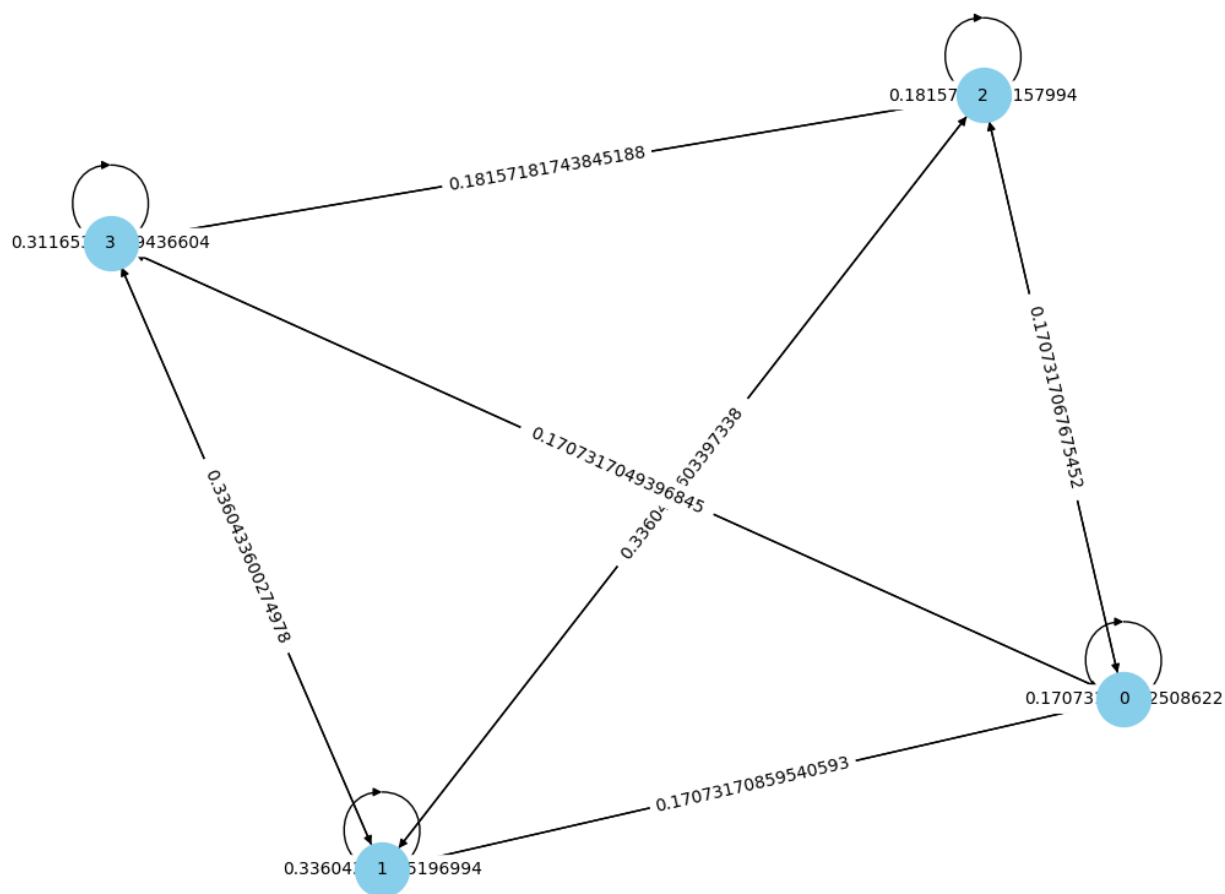$P(X25 = s|X20 = s)$ `= 0.3117 (Rounded-Off to 4 decimal places)`

```
For the precise result,
P(X25 = s | X20 = s)  = 0.31165311759436604
```

This is the conditional probability of being in state s after 25 minutes given that we were in state s after 20 minutes. This is equivalent to finding the 4th element in the resulting row of P^5 (since we're starting from state s).

Probabilities after 25 Minutes

P_25 Markov Chain

0.18157181743845188

0.18157 2 157994

0.3116511 9436604

0.17073170675452

0.17073170493396845 503397338

0.336043 5196994

0.17073 0 2508622

0.3360436 6845

0.33604336002749 78

0.17073170859540593

c) Stationary Distribution

A stationary distribution exists if there is a probability vector π such that:

```
π * P = π
```

where π represents the probability distribution of states. In this case, it means that after any number of steps, the system will eventually converge to this distribution.

To find the stationary distribution, you can solve the equation for π. If it exists, it will represent the long-term probabilities of being in each state.

As per the Python Script's output:

```
Stationary Distribution:
[[0.17073171]
 [0.33604336]
 [0.18157182]
 [0.31165312]]
```

Manual Calculation has also been added below - for reference.

$$P = \begin{array}{c} \\ r \\ w \\ e \\ s \end{array} \begin{array}{cccc} r & w & e & s \\ \begin{bmatrix} 0.5 & 0.3 & 0.0 & 0.2 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.3 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{bmatrix} \end{array}$$

For finding stationary distribution,

let $\pi = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{bmatrix}$

be the required stationary distribution.

Solving the below product would give the required answer:

$$P \cdot \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{bmatrix} = \pi$$

$$\Rightarrow \pi_1 = (0.5)\pi_1 + (0.2)\pi_2 + (0.1)\pi_3$$

$$\pi_2 = (0.3)\pi_1 + (0.5)\pi_2 + (0.3)\pi_3 + (0.2)\pi_4$$

$$\pi_3 = (0.1)\pi_2 + (0.3)\pi_3 + (0.3)\pi_4 + \cancel{1}$$

$$\pi_4 = (0.2)\pi_1 + (0.2)\pi_2 + (0.3)\pi_3 + (0.5)\pi_4$$

$$\Rightarrow (0.5)\pi_1 = (0.2)\pi_2 + (0.1)\pi_3$$

$$(0.5)\pi_2 = (0.3)\pi_1 + (0.3)\pi_3 + (0.2)\pi_4$$

$$(0.7)\pi_3 = (0.1)\pi_2 + (0.3)\pi_4$$

$$(0.5)\pi_4 = (0.2)\pi_1 + (0.2)\pi_2 + (0.3)\pi_3$$

$$\& \quad \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$$

Now, Put these equations into a
4-variable equation solver,
gives the required result.

Simultaneous Linear Equations Solver for Four Variables

Equations		Solved Values
Eq.1:	-0.5W + 0.2X + 0.1Y + 0Z = 0	W = 0.170731707317073
Eq.2:	0.3W + -0.5X + 0.3Y + 0.2Z = 0	X = 0.336043360433604
Eq.3:	0W + 0.1X + -0.7Y + 0.3Z = 0	Y = 0.181571815718157
Eq.4:	1W + 1X + 1Y + 1Z = 1	Z = 0.311653116531165

Here's the interpretation of the result obtained from running the Python Script:

1. State Interpretation:

   - The stationary distribution is a column vector with four elements. Each element corresponds to a state in the Markov chain. The states are in the same order as they appear in the transition matrix (P), i.e., `[r, w, e, s]`.

2. Probability Interpretation:

   - The value `0.17073171` corresponds to the probability of being in state `r` in the long run. Similarly, `0.33604336` corresponds to `w`, `0.18157182` corresponds to `e`, and `0.31165312` corresponds to `s`.

   - These probabilities represent the proportion of time the Markov chain is expected to spend in each state as the number of transitions approaches infinity.

3. Interpreting the Results:

   - The highest probability is associated with state `w` (writing). This suggests that, in the long run, the Markov chain is most likely to be in the "writing" state. This could imply that writing is a pivotal and persistent stage in the process.

   - The lowest probability is associated with state `r` (reading). This means that, in the long run, the chain is least likely to be in the "reading" state. This might indicate that the process doesn't stay in the reading phase for extended periods.

   - States `e` (emailing) and `s` (surfing) have intermediate probabilities, suggesting that these stages have a moderate likelihood of being visited in the long run.

4. Implications:

   - Understanding the stationary distribution provides insights into the long-term behavior of the Markov chain. It helps identify which states are more dominant and which are less dominant in the process.

   - This information can be valuable for making decisions related to resource allocation, optimization, or improving the efficiency of the process represented by the Markov chain.

Stationary Distribution of Markov Chain
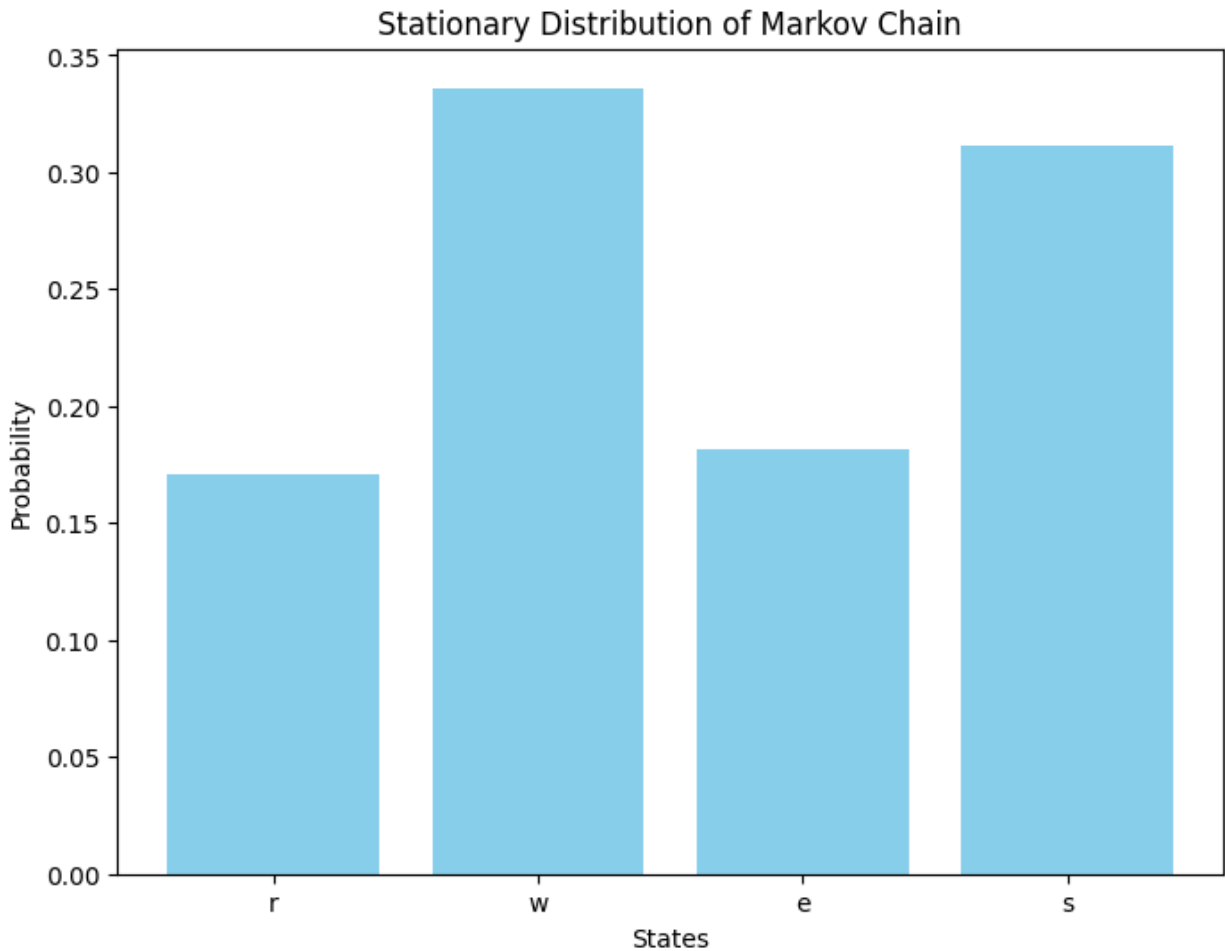
d) Limiting Distribution

In this context, the limiting distribution is the long-term behavior of the Markov chain as time approaches infinity. For a finite state, irreducible and aperiodic Markov chain, the limiting distribution exists and is equal to the stationary distribution.

Since we have already found the stationary distribution in the previous step, we can conclude that the limiting distribution exists and it is the same as the stationary distribution we computed earlier.

The stationary distribution we calculated in the previous step is also the limiting distribution for this Markov chain.

```
Limiting Distribution: [0.17073264 0.33604352 0.18157114 0.3116527 ]
```

```
The Markov chain is irreducible
The Markov chain is aperiodic
```


Limiting Distribution of Markov Chain
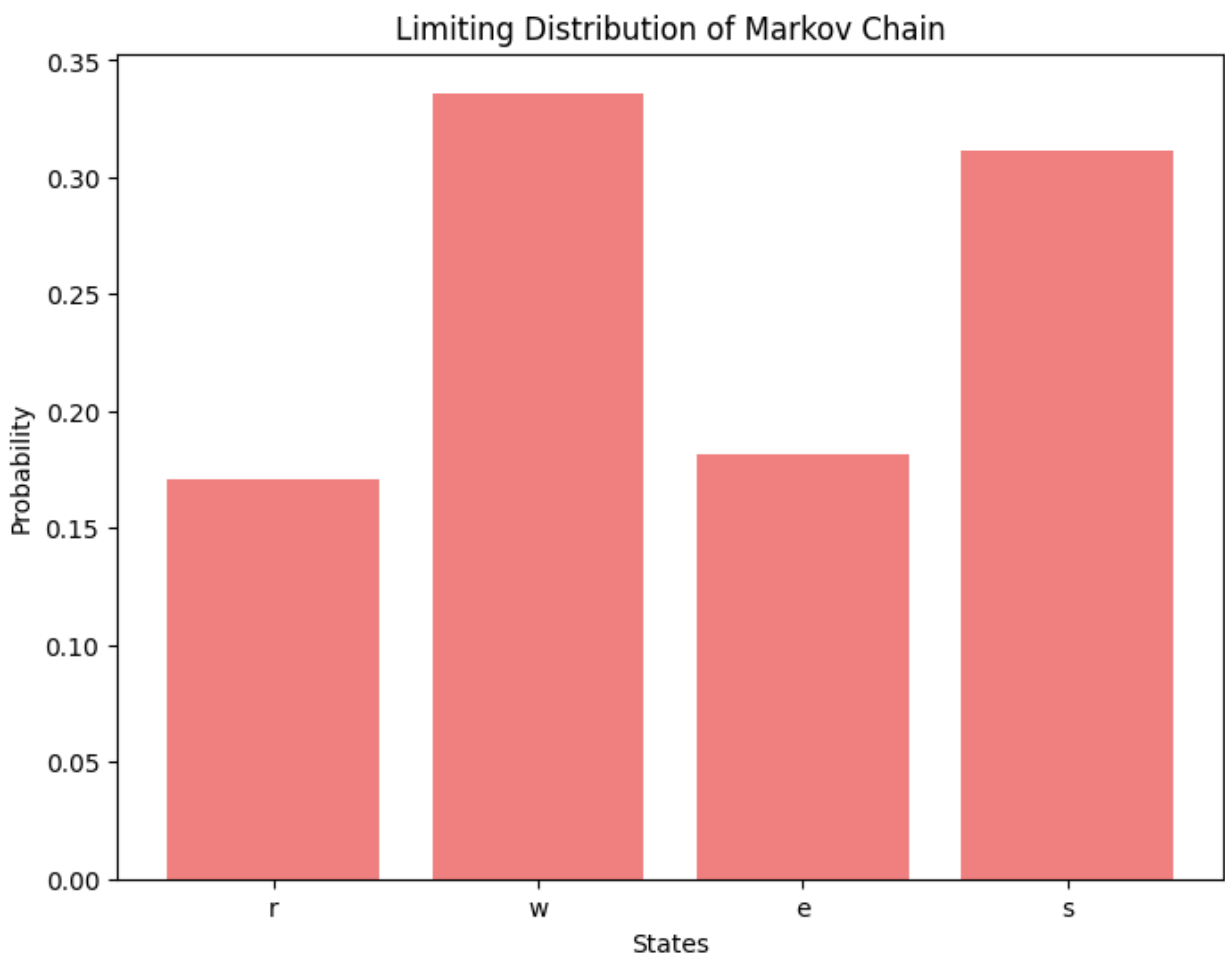
Here's the equivalent theory solution:

Irreducibility Check:

A Markov chain is considered irreducible if, for any pair of states (i) and (j), there exists a positive integer (n) such that the (n)-step transition probability $(P_{ij}^{(n)})$ is greater than 0. In other words, it is possible to transition from any state to any other state in a finite number of steps.

In the given code, the irreducibility check is performed by iterating through all states and transitions ((i) and (j)). For each pair, it calculates the (j)-step transition probability matrix using matrix exponentiation. If all elements in the resulting matrix for state (i) are greater than 0, it signifies that state (i) can be reached from any other state in (j) steps.

If this condition holds true for all states, the Markov chain is classified as irreducible.

Aperiodicity Check:

A Markov chain is considered aperiodic if there are no cycles with a common divisor greater than 1. In other words, the chain does not exhibit regular, repeating patterns.

In the given code, the aperiodicity check involves inspecting each state. If a state has a non-zero transition probability to itself (a self-loop), it implies the existence of a cycle. The length of this cycle is recorded.

The greatest common divisor (gcd) of all cycle lengths is then calculated. If the gcd is 1, it indicates that there are no common factors among the cycle lengths, and the Markov chain is classified as aperiodic. Otherwise, the chain exhibits periodic behavior with a period equal to the gcd.

Q2: Analyzing 1-D Random Walks

Introduction:

The provided code performs simulations of 1-D random walks under different conditions and analyzes the resulting positions. A 1-D random walk models the movement of an entity along a straight line, where at each step, it can move either to the left or right.

Code Walkthrough:

1. Simulation with (p = 0.5):
   - Parameters are set: (p = 0.5) (equal probabilities of left and right steps), and `num_steps = 500` (number of steps).
   - The random walk is simulated, and the positions are stored in the `positions` array.
   - A plot shows the path of the random walker over time.
   - A histogram displays the probability distribution of attaining a particular position, over the course of 500 steps.


1-D Random Walk with p = 0.5 (500 steps)

Probability of Attaining a Particular Position (500 steps)
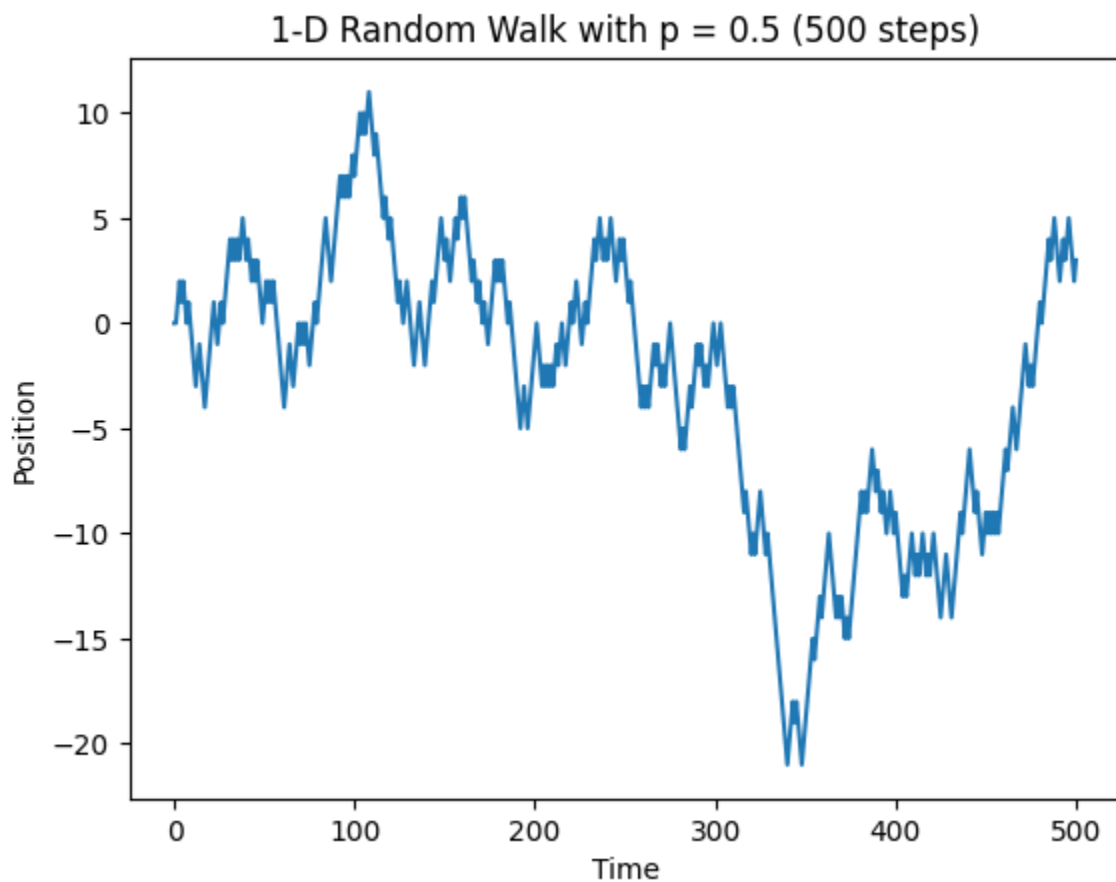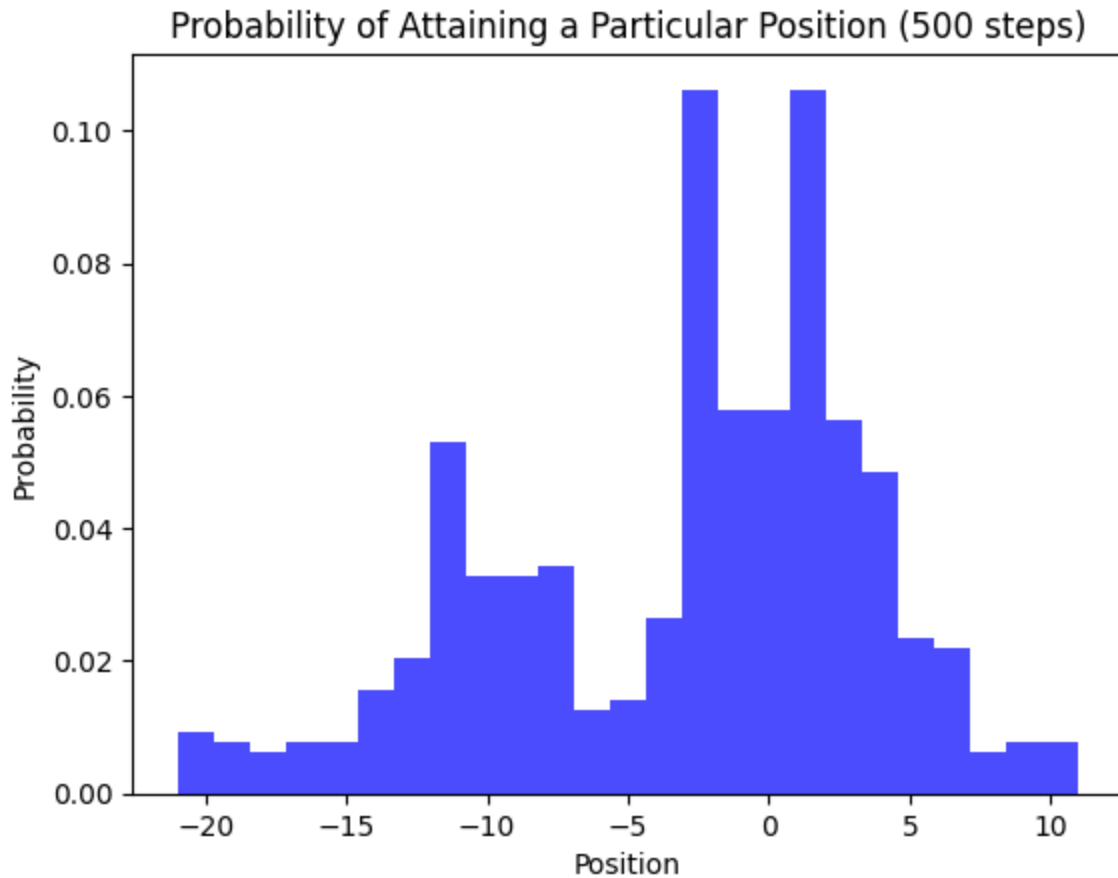
```
Simulated Probability =   0.502
Theoretical Probability = 0.500
```

Inference:

1) The plot shows a path with random fluctuations around the origin.

2) The simulated probability of moving towards the right is about 0.5, indicating that in a balanced random walk, one is equally likely to move in either direction.

2. Simulation with (p = 0.8):
   - Parameters are updated for part (b): (p_b = 0.8) (higher probability of right steps).
   - The random walk is simulated, and the positions are stored in `positions_b`.
   - A plot displays the path of the random walker over time for this case.
   - A histogram illustrates the probability distribution of attaining a particular position, over the course of 500 steps.

1-D Random Walk with p = 0.8 (500 steps)
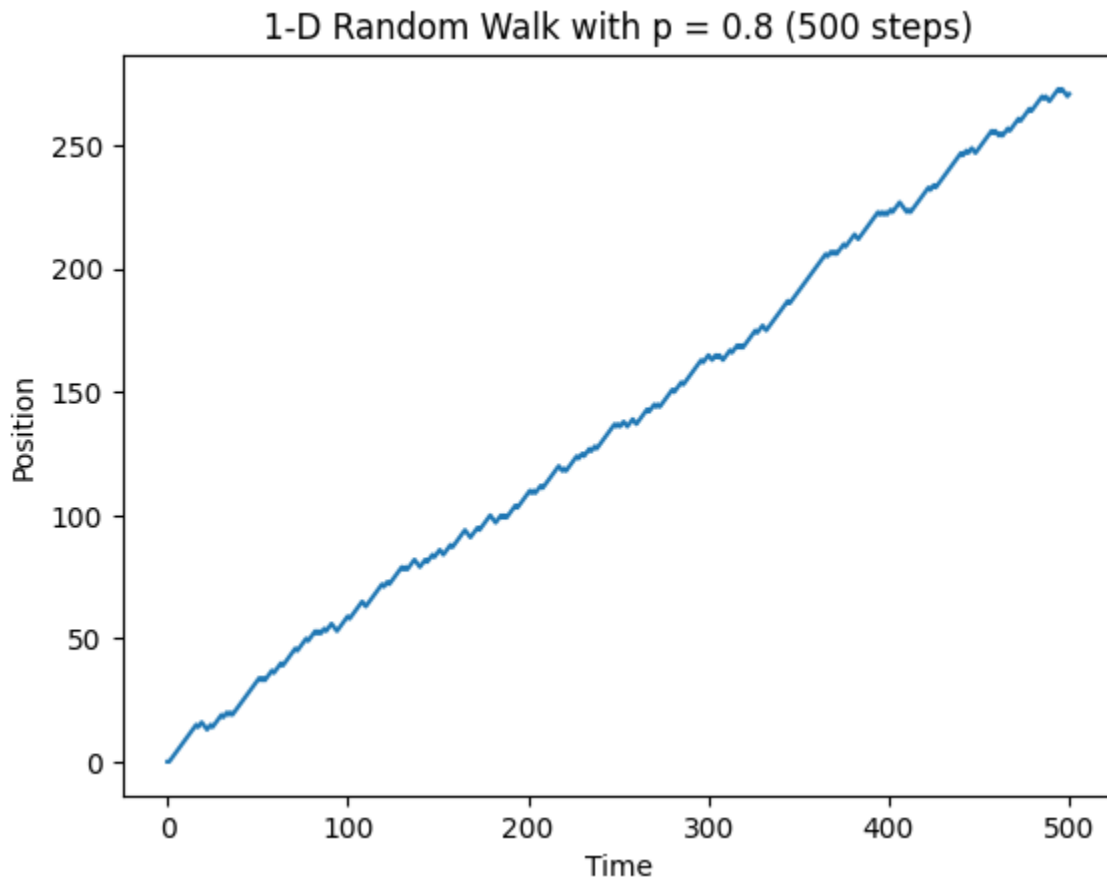
Inference:

1) The plot exhibits a much stronger trend towards positive positions, reflecting the higher probability of moving right, in contrast to the "equally likely" trend of moving in the either direction, of part (a).

2) This is evident from the fact that the positions are now reaching 250, unlike during part (a), where the positions were able to reach till +10.

3) Also, since P = 0.8 in this case, it's very unlikely to reach a negative position below 0, which is clear from the above Random Walk plot.

Probability of Attaining a Particular Position (p = 0.8, 500 steps)
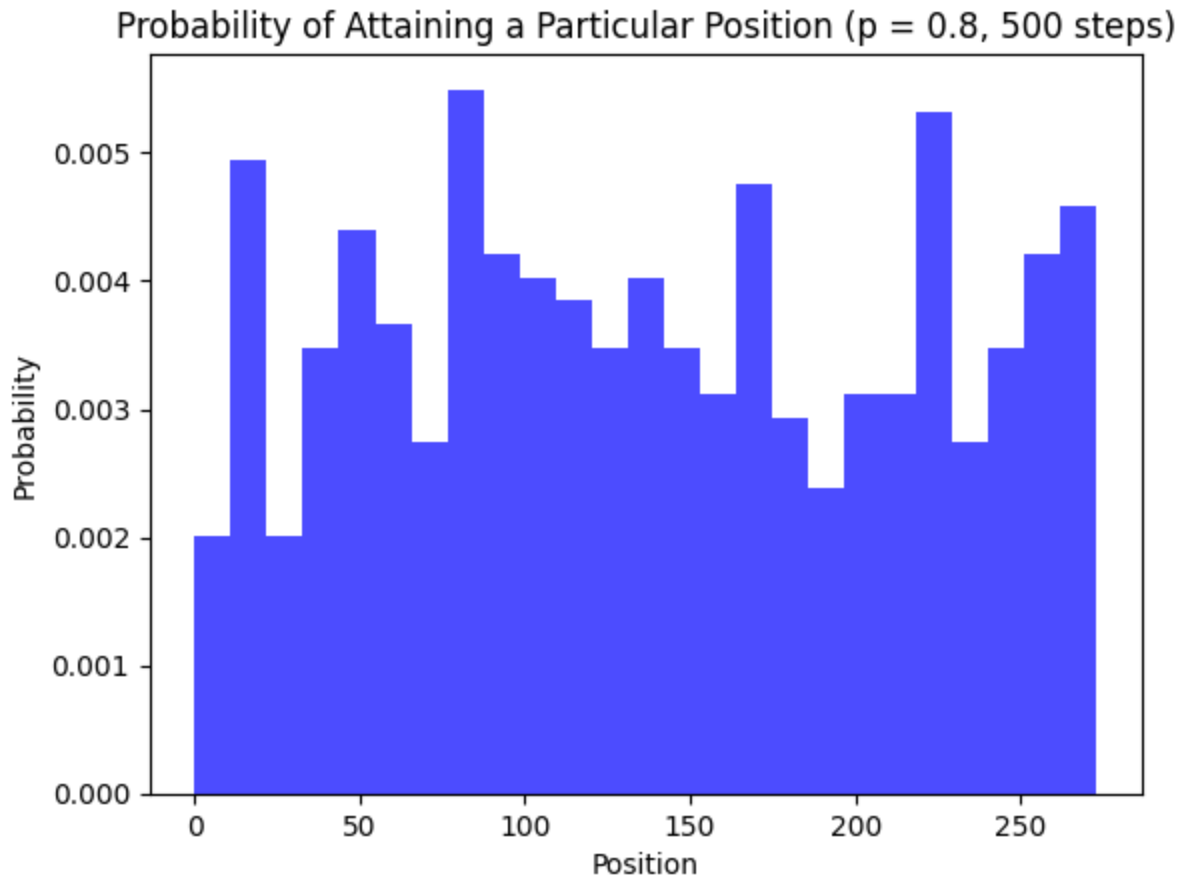
```
Simulated Probability =   0.77
Theoretical Probability = 0.80
```
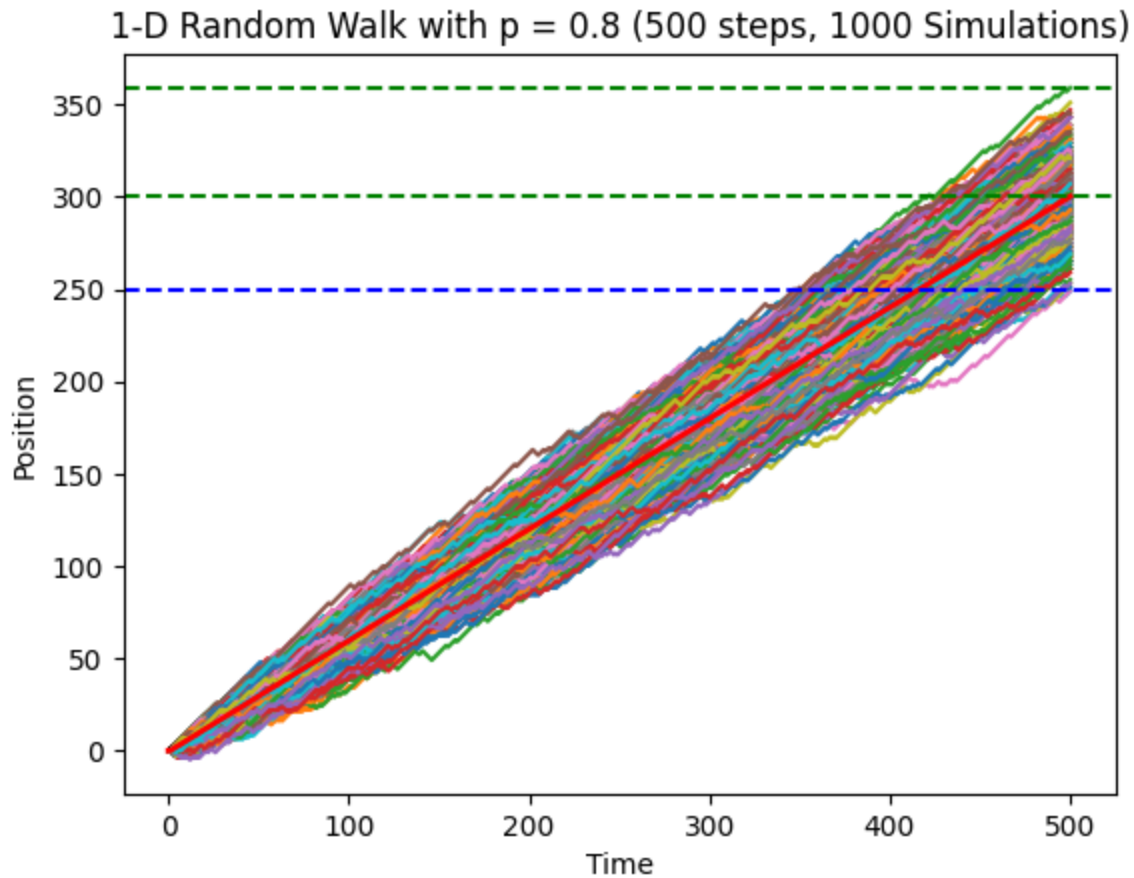
Inference:

1) The Histogram plot exhibits a much stronger trend towards positive positions, reflecting the higher probability of moving right, in contrast to the "equally likely" trend of moving in the either direction, of part (a).

2) This is evident from the fact that the positions are now reaching 250, unlike during part (a), where the positions were able to reach till +10.

3) Also, since P = 0.8 in this case, it's very unlikely to reach a negative position below 0, which is clear from the histogram plot.

3. Increasing # of times we perform the same Simulation (Part c):
  - Parameters are set for part (c): `num_simulations = 1000` (number of simulations).
  - A matrix `positions_c` is initialized to store positions for all simulations.
  - The random walk is simulated 1000 times, and results are stored in `positions_c`.
  - A histogram presents the probability distribution of final positions for this set of simulations.

Note:

1) In the graph below, the straight, red-colored line corresponds to the average value of final position, resulting from the 1000 simulations of the Random Walk Experiment.

2) The average line intersects the y-axis at 300, which is inline with the theoretical result, that the final position, averaged over a lot of experiments, would be very close to 300.

3) This is because 300 = 0.8 * 500 = 400 = # of Steps moved towards the Right.

4) This implies that 500 - 400 = 100 = # of Steps moved towards the Left.

5) This implies that Overall Displacement,
   = (# of Steps moved towards the Right) - (# of Steps moved towards the Left)
   = 400 - 100
   = 300

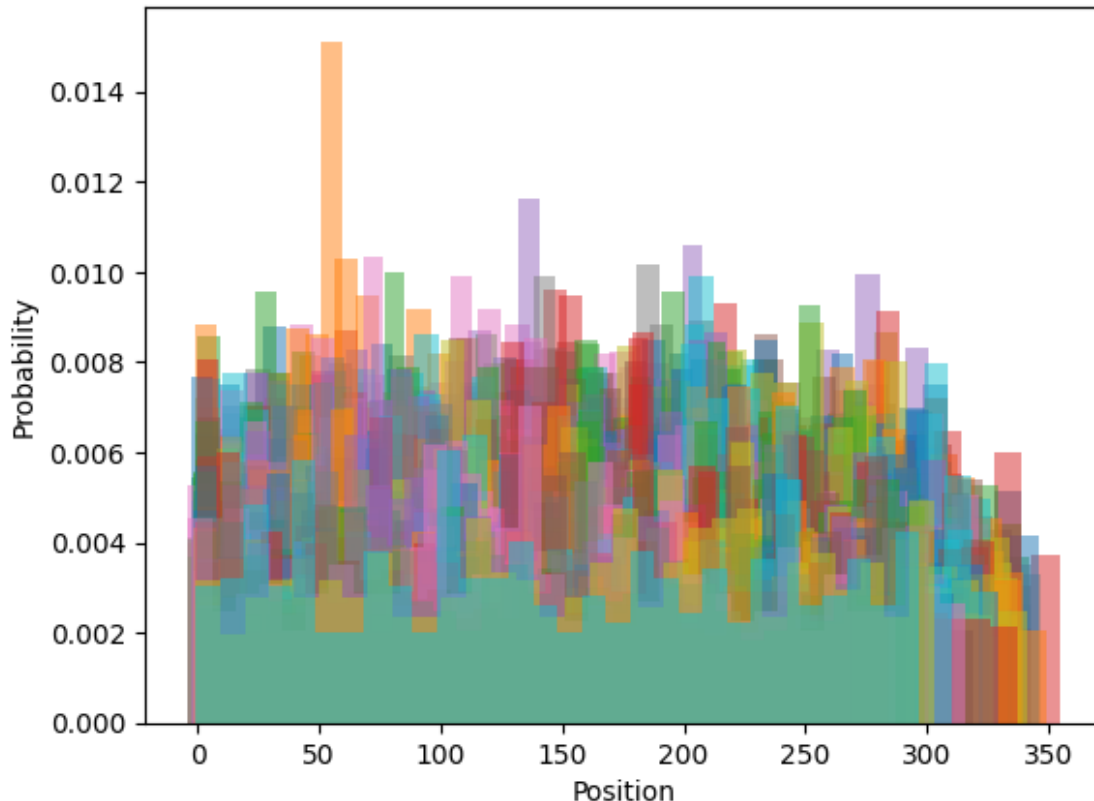1-D Random Walk with p = 0.8 (500 steps, 1000 Simulations)

Inference:

1) As depicted in the figure, the probability of attaining various positions, in the 500 steps of the Random Walk, is well-spread out, with maximum position attained = 350 (albeit with a much smaller probability), compared to the maximum position attained = 250 in part (b).

2) Since we're now performing, essentially the same experiment 1000 times, we can see the differences in each simulation, since the Output of the plots are layered on-top of each other, unlike part (b), where we were simply considering 1 trial, which could have resulted in some potential inaccuracies, especially if were to be an outlier.

3) The average value of Final Positions turns out to be = 300, give-or-take (±)150 displacement from 300.

Note:

1) The figure given below shows the probability of attaining a particular position, in the 500 steps of the Random walk.

2) The plots are layered on-top of each other, for the 1000 simulations of the Random Walk Experiment.



Probability Distribution of Positions (p = 0.8, 500 Steps, 1000 Simulations)

Inference:

4) As depicted in the figure, the probability of attaining various positions, in the 500 steps of the Random Walk, is well-spread out, with maximum position attained = 350 (albeit with a much smaller probability), compared to the maximum position attained = 250 in part (b).

5) Since we're now performing, essentially the same experiment 1000 times, we can see the differences in each simulation, since the Output of the plots are layered on-top of each other, unlike part (b), where we were simply considering 1 trial, which could have resulted in some potential inaccuracies, especially if were to be an outlier.

Probability Distribution of Positions (p = 0.8, 500 Steps, 1000 Simulations)
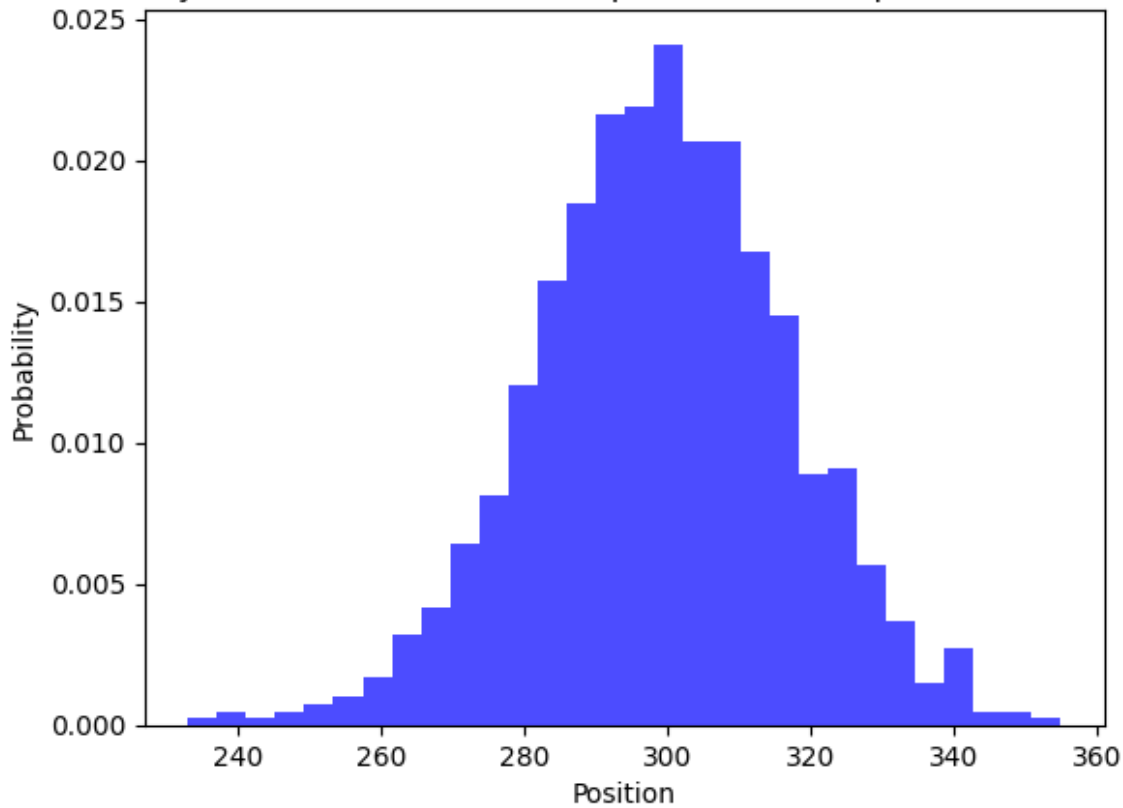
Inference:

1) The histogram of final positions shows a distribution centered around positive values, indicating a higher likelihood of ending up on the right side, particularly around the value 300, which is as expected, based on the calculation shown previously.

2) Compared to the single simulation in part (b), the spread of final positions is broader, showcasing the stochastic nature of the process.

3) It gives the impression that, the Probability Distribution of Final Positions, looks somewhat similar to an "imperfect" Normal-type distribution, centered around the position 300.

Note: Inference (3) must be taken with a grain of salt, since we'd have to perform much larger trials than 1000, to confirm or deny, whether the final positions indeed converge to some form of Normal Distribution, centered around 300.

Conclusion:

The code provides a comprehensive analysis of 1-D random walks, demonstrating how different probabilities of left and right steps influence the behavior of the random walker. It also showcases the use of multiple simulations to gather statistics about the final positions.
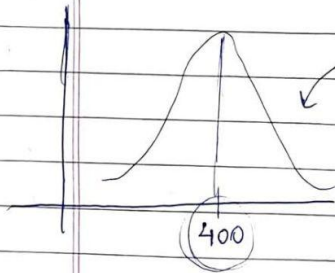
By running this code, one gains insight into the stochastic nature of random walks and how they can be used to model various real-world phenomena.
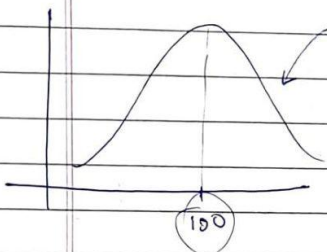
Scratch Work:

$0.8 = \dfrac{\#\ of\ steps\ to\ the\ right}{500}$

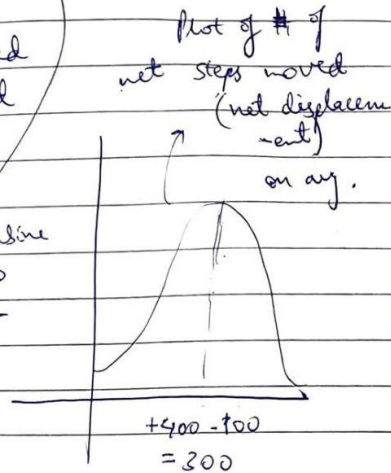$\left(\dfrac{8}{10}\right) 500 = \underline{400}$ (on avg.)

Plot of # of steps moved towards the right (on avg.)

400

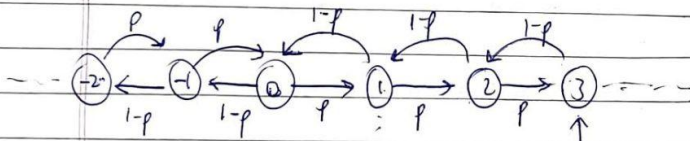Plot of # of net steps moved (not displacement) on avg.

Combine $\Rightarrow$

Plot of # of steps moved towards the left. (on avg)

100

$+400 - 100 = 300$

$p \qquad p \qquad 1-p \qquad 1-p \qquad 1-p$

$(-2) \leftarrow (-1) \leftarrow (0) \rightarrow (1) \rightarrow (2) \rightarrow (3)$

$1-p \quad 1-p \quad p \quad p \quad p$

end up here after 500 steps, 1 simulation

249 $\qquad$ 249

$\dfrac{251}{500} > 0.5$ $\nearrow^{+1}_{n}$

$p = 0.5 \qquad$ simulated $\hat{p} = 0.5$

$0.8 = \dfrac{\#\ of\ steps\ to\ the}{}$ | simulated prop. of moving towards the right

247

$\dfrac{498}{2}$

$(0) \leftarrow (1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \quad (501)$

$1 + 500 = 501$

$\dfrac{2 + 249}{500} = \dfrac{251}{500}$

$(500) \dots (-2) \leftarrow (-1) \leftarrow (0) \rightarrow (1) \rightarrow (2) \quad (499) \rightarrow (500)$

Simp $= (0.5)$
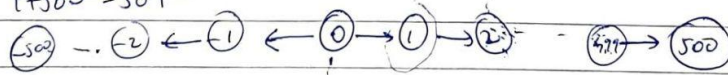
Simp $= \dfrac{\#\ of\ steps\ to\ the\ right}{500}$

$\dfrac{250}{500} \rightarrow \dfrac{251}{500}$

$= \dfrac{251}{500} > 0.5$