Project 4 : Discrete-Time-Finite-State [DTFS] Markov Chains

Author: Ishan Mohanty

USC ID: 4461-3447-18

NET ID: imohanty

Email: imohanty@usc.edu

/##

Import Libraries and Other Dependencies

In [507]:

```
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg
from numpy.linalg.linalg import matrix_power
```

Problem 1:

DTFS MC Stationary distributions:

Find stationary distributions for a discrete time finite state (DTFS) Markov chain { X_n } having transition matrix:

$$P = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$$

when (i) a=1/10 and b=1/15, (ii) a=0.5 and b=0.5, (iii) a=1 and b=1, (iv) a=0 and b=0

In [508]:

```
"""
Generates Transition Matrix.
@param a,b = values
@return P: Transition Matrix
"""

def transition_matrix(a,b):
    P = np.array([[1-a,a],[b,1-b]])
    return P
```

In [509]:

```
"""
Finds Stationary Distribution
@param P: Transition Matrix
@return pi_stat: Stationary Distribution
"""

def find_stat_dist(P):
    eig_val,eig_vec = linalg.eig(P.T)
    if eig_val[0]==1 and eig_val[1] == 1:
        special_pi_stat = []
        special_pi_stat.append(eig_vec[:,0].T / np.sum(eig_vec[:,0]) )
        special_pi_stat.append(eig_vec[:,1].T / np.sum(eig_vec[:,1]) )
        return special_pi_stat
    pos = np.argwhere(eig_val==1)
    pi_stat = eig_vec[:,pos].T
    pi_stat = pi_stat/np.sum(pi_stat)
    return pi_stat
```

In [510]:

```
P1 = transition_matrix(1/10,1/15)
print("Stationary Distribution for P1:",find_stat_dist(P1))
```

```
Stationary Distribution for P1: [[[0.4 0.6]]]
```

In [511]:

```
P2 = transition_matrix(0.5,0.5)
print("Stationary Distribution for P2:",find_stat_dist(P2))
```

```
Stationary Distribution for P2: [[[0.5 0.5]]]
```

In [512]:

```
P3 = transition_matrix(1,1)
print("Stationary Distribution for P3:",find_stat_dist(P3))
```

```
Stationary Distribution for P3: [[[0.5 0.5]]]
```

In [513]:

```
P4 = transition_matrix(0,0)
pi_stat = find_stat_dist(P4)
print("Stationary Distribution for P4:",pi_stat[0],"and", pi_stat[1])
```

```
Stationary Distribution for P4: [1. 0.] and [0. 1.]
```

Summary:

1. We can find the stationary distributions for a DTFS Markov Chain $\{X_n\}$ through two methods. (i) Algebraic Solution through solving simultaneous equation (ii) Eigen Decomposition Method.
2. (i) Simultaneous equation method

    1. Find the 2 equations based on

$$\pi P = \pi$$

    2. Solve and get the stationary distribution

$$\pi_* = \frac{1}{a+b} * (b, a)$$

(ii) Eigen Decomposition Method

    1. Find the Eigen-Values and Vectors according to the equation.

$$Ax = \lambda x$$

Here, we take

$$Ax = \lambda x$$

$$P^T \pi^T = \pi^T$$

Hence we find the right Eigen-vector of P^T

    2. choose the right Eigen-Vector corresponding to the eigen value

$$\lambda = 1$$

1. The Eigen-vector should be normalized and should be non-negative. This is the stationary distribution π_* and it must obey the following equation:

$$\sum \pi_i = 1$$

Results:

1.
$$P1 = \begin{pmatrix} 0.9 & 0.1 \\ 0.0666 & 0.9333 \end{pmatrix}$$

$$\pi_* = [0.4, 0.6]$$

1.
$$P2 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

$$\pi_* = [0.5, 0.5]$$

1. $$P3 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\pi_* = [0.5, 0.5]$$

1. $$P4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\pi_* = [1, 0]$$

$$\pi_* = [0, 1]$$

```
The stationary distribution is not unique because it has both eigen-values
```

Problem 2:

DTFS MC Simulation:

Simulate 75 sample paths of length 500 for the DTFS Markov chains below using the update equation:

$\pi(t+1)$ = $\pi(t)$P

All MCs start with the initial probability vector $\pi(t=0)$ = (1,0).

$$P1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$P2 = \begin{pmatrix} 0.75 & 0.25 \\ 0.1 & 0.9 \end{pmatrix}$$

$$P3 = \begin{pmatrix} 0.48 & 0.48 & 0.02 \\ 0.22 & 0.7 & 0.08 \\ 0 & 0 & 1 \end{pmatrix}$$

Check for MC convergence by applying a goodness-of-fit test to the last 75 samples of each sample path. Compare the ensemble and time averages for the 3 Markov chains. Based on these averages, which of the chains are ergodic?

In [514]:

```
"""
simulates Markov Chain State Transition.
@param num_states: number of states in Markov Chain
       N: Number of Iterations
       pi_init_state: intial state distribution
       P: Transition Matrix
@return state_transitions: list of states
"""

def simulate_markov_chain(num_states,P,N,pi_init_state):

    state_transitions = []
    state_transitions.append(1)

    for i in range(N-1):

        if num_states == 2:
            pi_state = np.array([0,0]).reshape(1,2)
        else:
            pi_state = np.array([0,0,0]).reshape(1,3)

        pi_state = pi_init_state@matrix_power(P,i+1)

        if (1 in pi_state) == True:
            state_prob = np.argmax(pi_state)
            state_transitions.append(state_prob+1)

        else:
            rv = np.random.uniform(0,1)
            min_state_prob = np.argmin(pi_state)
            max_state_prob = np.argmax(pi_state)

            if num_states == 2:
                if rv <= pi_state[0,min_state_prob]:
                    state_transitions.append(min_state_prob+1)
                else:
                    state_transitions.append(max_state_prob+1)

            else:
                states = [0,1,2]
                indices = [min_state_prob,max_state_prob]
                new_state = np.delete(states,indices)
                intermediate_state = np.asscalar(new_state)
                if rv <= pi_state[0,min_state_prob]:
                    state_transitions.append(min_state_prob+1)
                elif rv > pi_state[0,min_state_prob] and rv <= pi_state[0,intermediate_
state]:
                    state_transitions.append(intermediate_state+1)
                else:
                    state_transitions.append(max_state_prob+1)

    return state_transitions
```

In [515]:

```
"""
Produces more spaced and enlarged plot.
@param s_p: state_transitions
"""
def enlarged_plot(s_p):
    plt.plot(s_p)
    plt.xticks(np.arange(500))
    plt.yticks([1,2])
    plt.grid()
    plt.gca().margins(x=0)
    plt.gcf().canvas.draw()
    tick_l = plt.gca().get_xticklabels()
    m_size = max([t.get_window_extent().width for t in tick_l])
    m = 0.005
    s = m_size/plt.gcf().dpi*800+2*m
    margin = m/plt.gcf().get_size_inches()[0]
    plt.gcf().subplots_adjust(left=margin, right=1.-margin)
    plt.gcf().set_size_inches(s, plt.gcf().get_size_inches()[1])
    plt.tight_layout()
    plt.show()
```

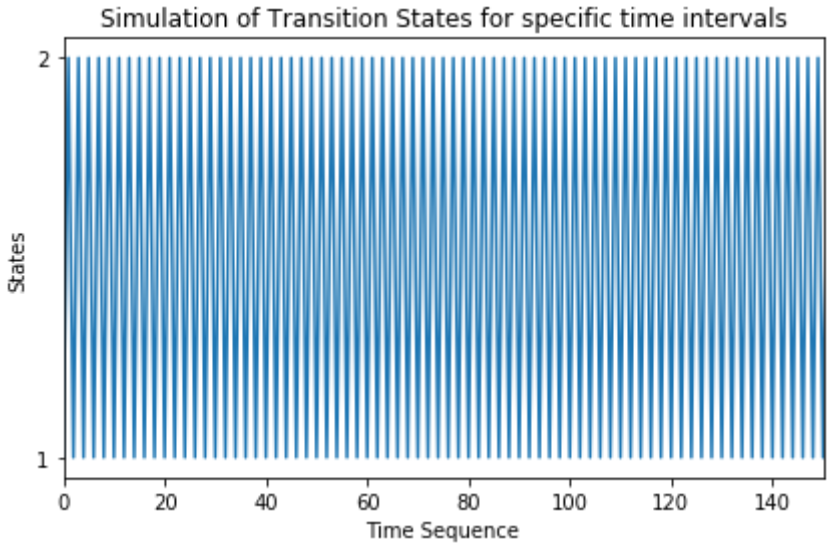
P1 with time length t=500 [Time Average]

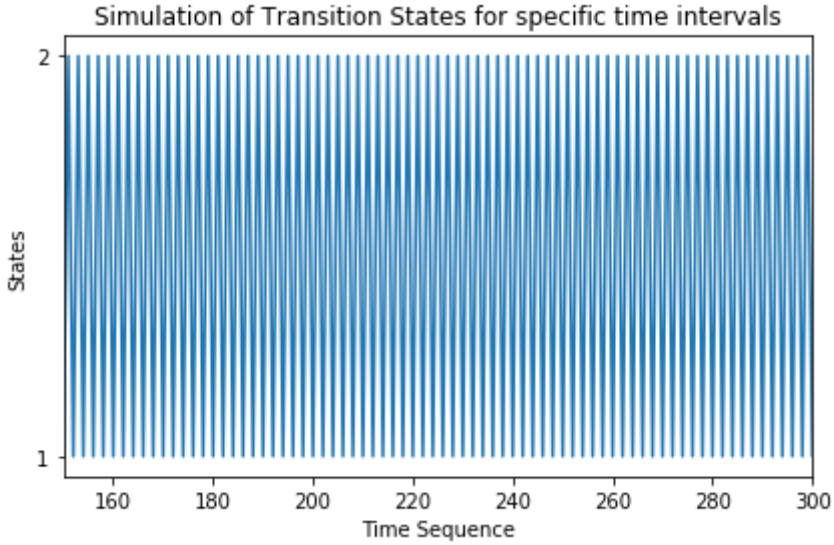
In [516]:

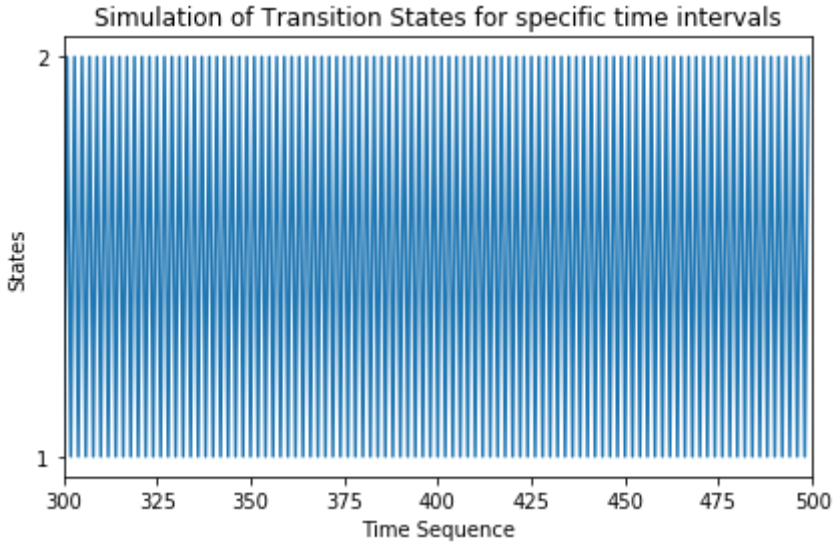
```
P1 = np.array([[0,1],[1,0]])
pi_init_state = np.array([1,0]).reshape(1,2)
s_p1 = simulate_markov_chain(2,P1,500,pi_init_state)
```

In [517]:

```
plt.plot(s_p1)
plt.xlim(0,150)
plt.yticks([1, 2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.tight_layout()
plt.show()

plt.plot(s_p1)
plt.xlim(150,300)
plt.yticks([1, 2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.tight_layout()
plt.show()

plt.plot(s_p1)
plt.xlim(300,500)
plt.yticks([1, 2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.tight_layout()
plt.show()
```

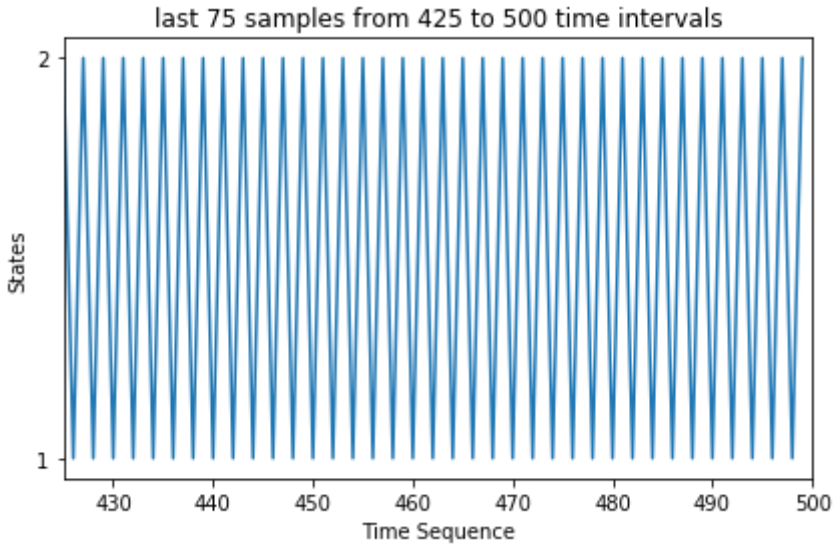

Simulation of Transition States for specific time intervals
States
Time Sequence
2
1
0
20
40
60
80
100
120
140


Simulation of Transition States for specific time intervals
States
Time Sequence
2
1
160
180
200
220
240
260
280
300


Simulation of Transition States for specific time intervals
States
Time Sequence
2
1
300
325
350
375
400
425
450
475
500

In [518]:

```
plt.plot(s_p1)
plt.xlim(425,500)
plt.yticks([1, 2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('last 75 samples from 425 to 500 time intervals')
plt.tight_layout()
plt.show()
```

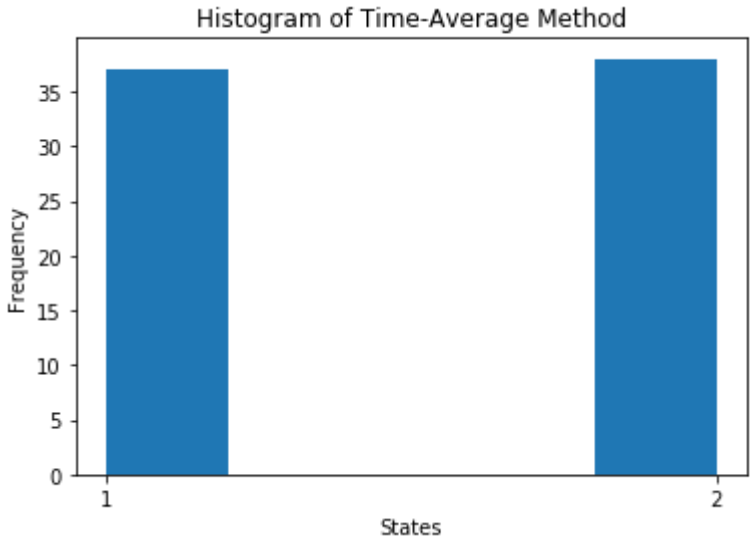


In [519]:

```
enlarged_plot(s_p1)
```

In [520]:

```
plt.hist(s_p1[425:500],bins=5)
plt.xticks([1,2])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Time-Average Method')
plt.show()
```



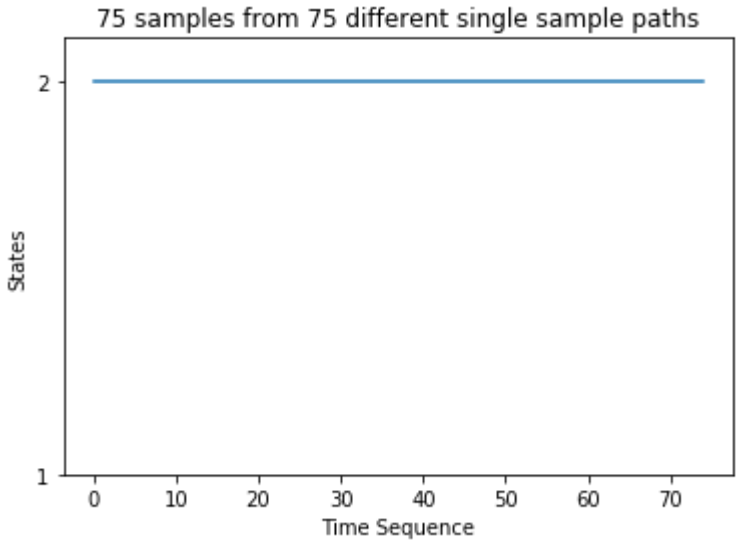
P1 with time length t=500 [Ensemble Average]

In [521]:

```
sample_paths = []
for sp in range(75):
    state_trans = simulate_markov_chain(2,P1,500,pi_init_state)
    sample_paths.append(state_trans[499])
```

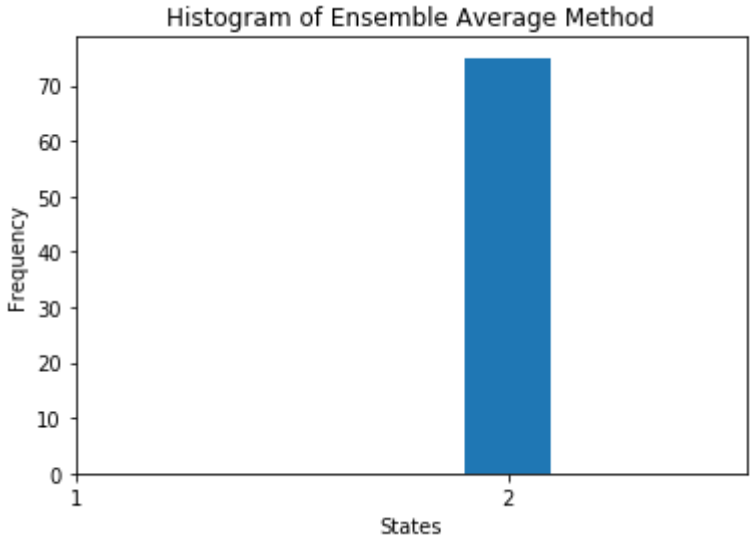
In [522]:

```
plt.plot(sample_paths)
plt.yticks([1,2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('75 samples from 75 different single sample paths')
plt.show()
```



In [523]:

```
plt.hist(sample_paths,bins=5)
plt.xticks([1,2])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Ensemble Average Method')
plt.show()
```



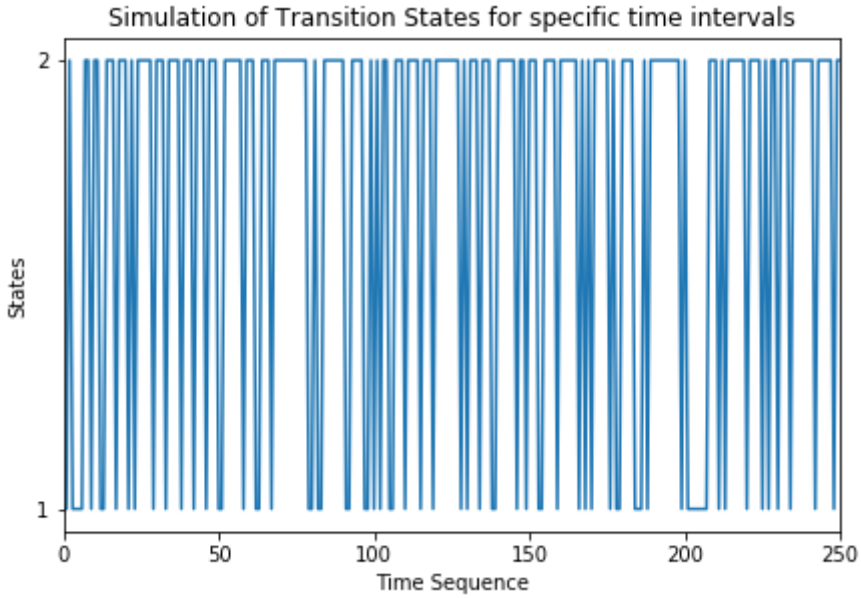
P2 with time length t=500 [Time Average]

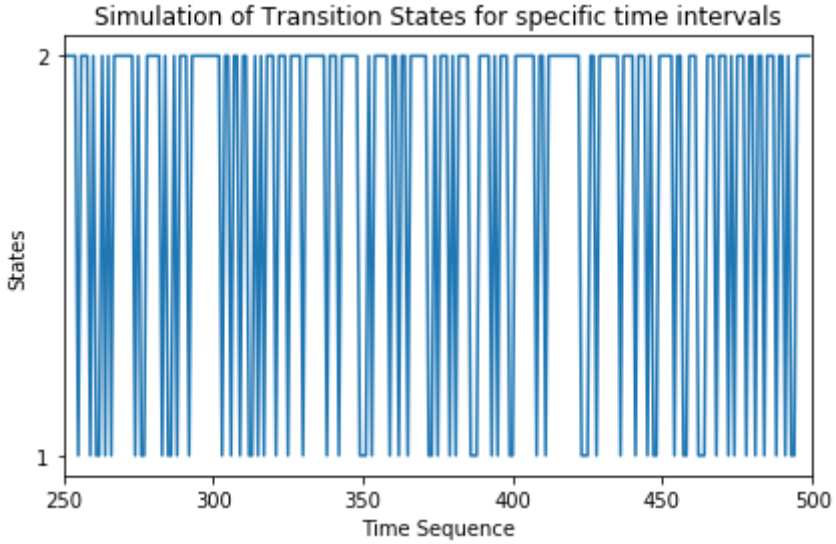
In [524]:

```
P2 = np.array([[0.75,0.25],[0.1,0.9]])
pi_init_state = np.array([1,0]).reshape(1,2)
s_p2 = simulate_markov_chain(2,P2,500,pi_init_state)
```

In [525]:

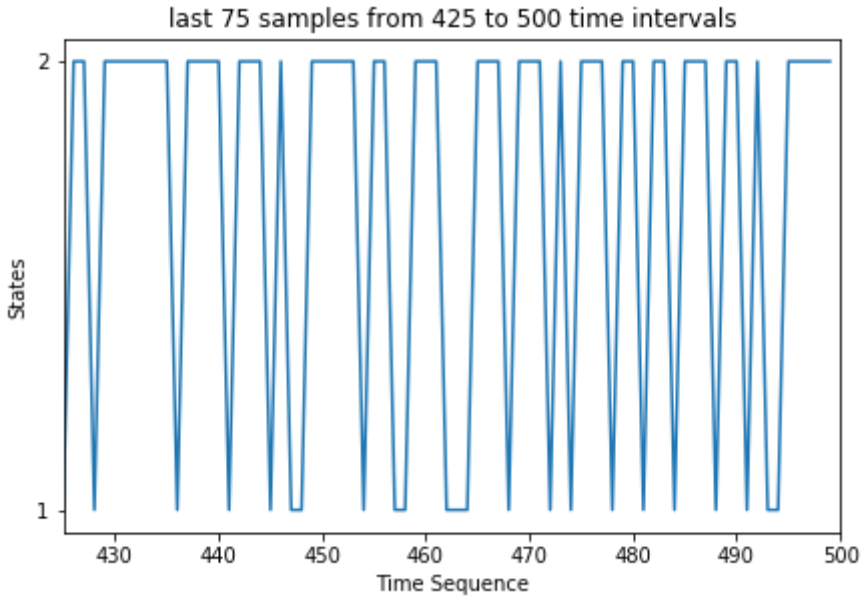
```
plt.plot(s_p2)
plt.xlim(0,250)
plt.yticks([1, 2])
plt.tight_layout()
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.show()

plt.plot(s_p2)
plt.xlim(250,500)
plt.yticks([1, 2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.tight_layout()
plt.show()
```





In [526]:

```
plt.plot(s_p2)
plt.xlim(425,500)
plt.yticks([1, 2])
plt.tight_layout()
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('last 75 samples from 425 to 500 time intervals')
plt.show()
```

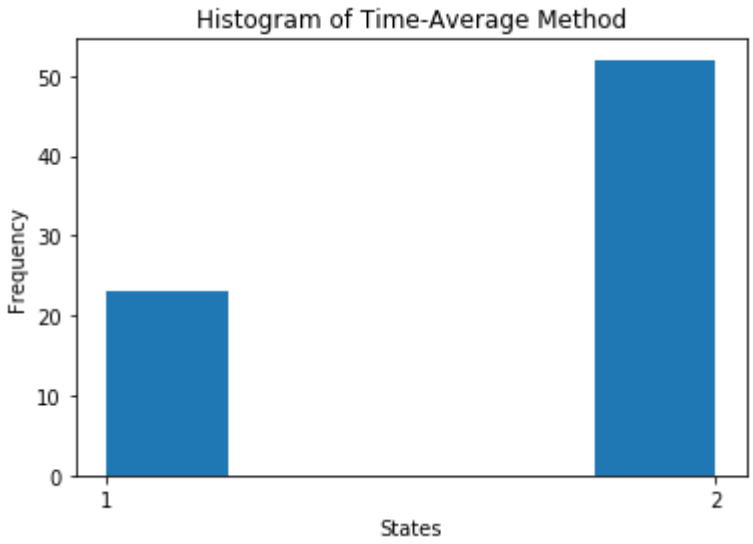


In [527]:

```
enlarged_plot(s_p2)
```

In [528]:

```
plt.hist(s_p2[425:500],bins=5)
plt.xticks([1,2])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Time-Average Method')
plt.show()
```



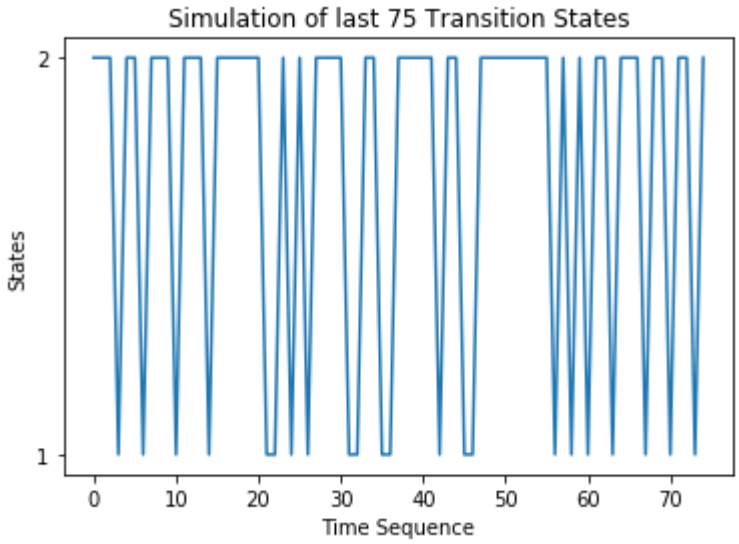
P2 with time length t=500 [Ensemble Average]

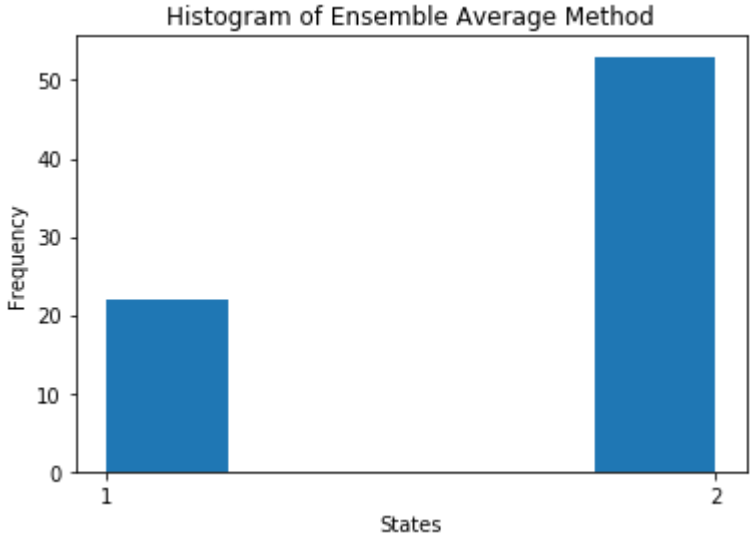
In [529]:

```
ens_states = []
for sp in range(75):
    state_trans = simulate_markov_chain(2,P2,500,pi_init_state)
    ens_states.append(state_trans[499])
```

In [530]:

```
plt.plot(ens_states)
plt.yticks([1,2])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of last 75 Transition States')
plt.show()

print()

plt.hist(ens_states,bins=5)
plt.xticks([1,2])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Ensemble Average Method')
plt.show()
```





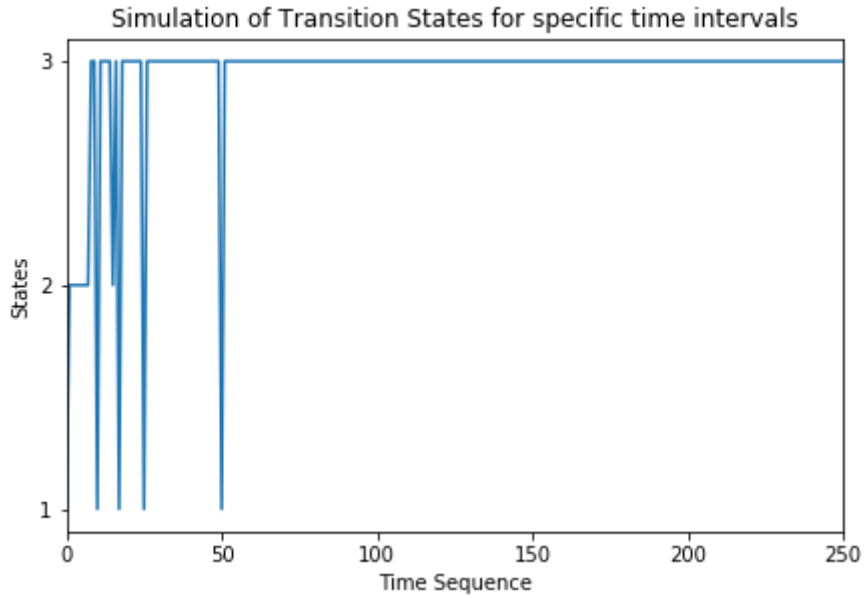
P3 with time length t=500 [Time Average]

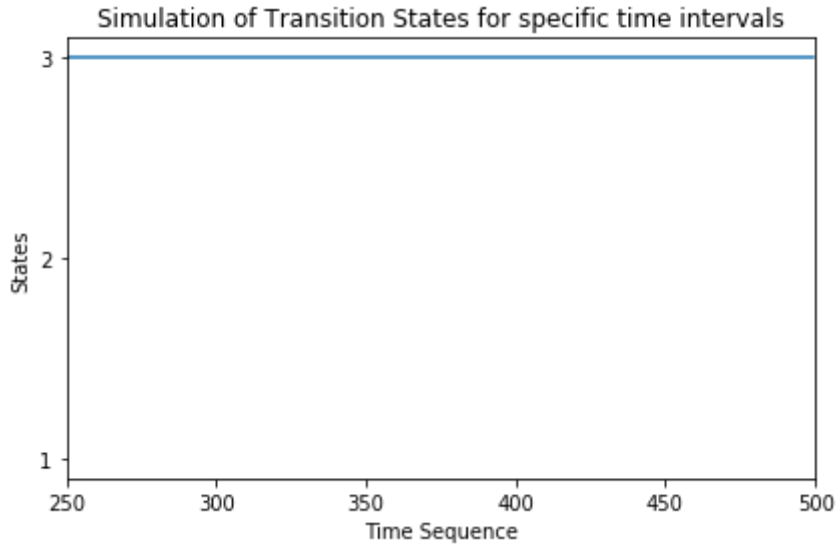
In [531]:

```
P3 = np.array([[0.48,0.48,0.04],[0.22,0.7,0.08],[0,0,1]])
pi_init_state_p3 = np.array([1,0,0]).reshape(1,3)
s_p3 = simulate_markov_chain(3,P3,500,pi_init_state_p3)
```

In [532]:

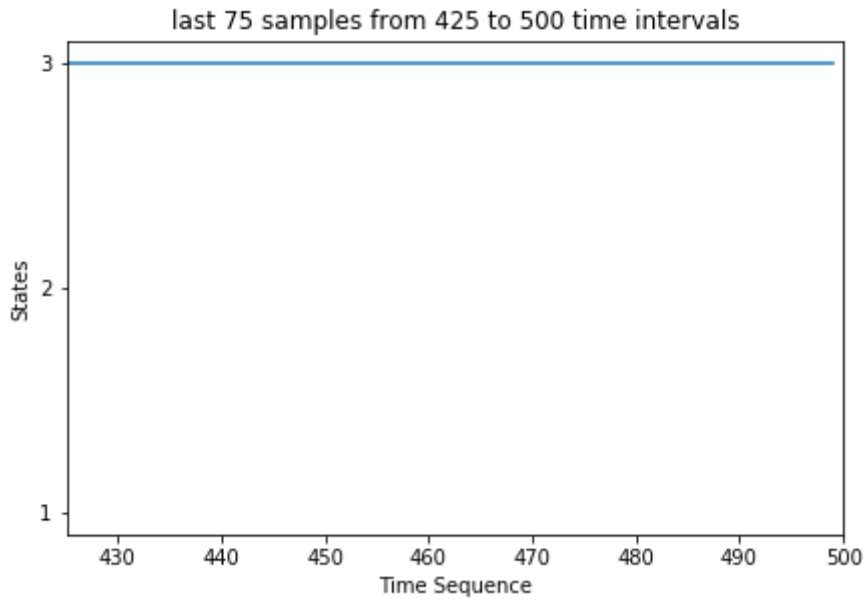
```
plt.plot(s_p3)
plt.xlim(0,250)
plt.yticks([1, 2, 3])
plt.tight_layout()
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.show()

plt.plot(s_p3)
plt.xlim(250,500)
plt.yticks([1, 2, 3])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of Transition States for specific time intervals')
plt.tight_layout()
plt.show()
```


Simulation of Transition States for specific time intervals
States
Time Sequence
1
2
3
0
50
100
150
200
250


Simulation of Transition States for specific time intervals
States
Time Sequence
1
2
3
250
300
350
400
450
500

In [533]:

```
plt.plot(s_p3)
plt.xlim(425,500)
plt.yticks([1, 2, 3])
plt.tight_layout()
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('last 75 samples from 425 to 500 time intervals')
plt.show()
```

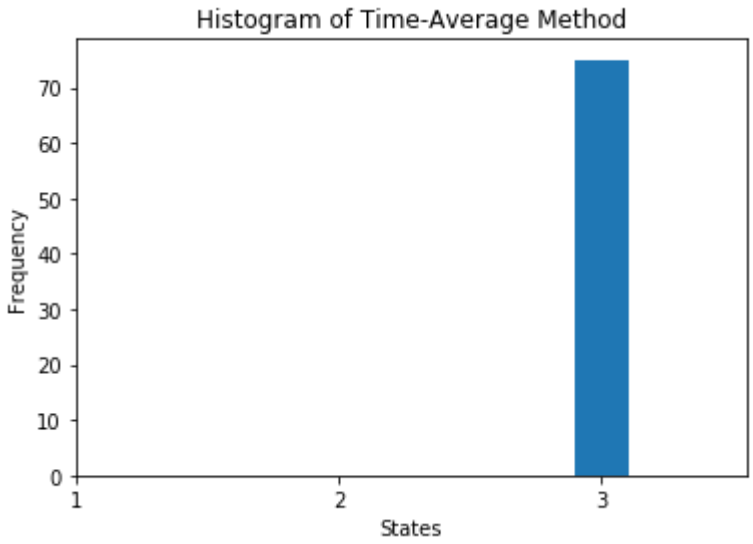


In [534]:

```
enlarged_plot(s_p3)
```

In [535]:

```
plt.hist(s_p3[425:500],bins=5)
plt.xticks([1,2,3])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Time-Average Method')
plt.show()
```



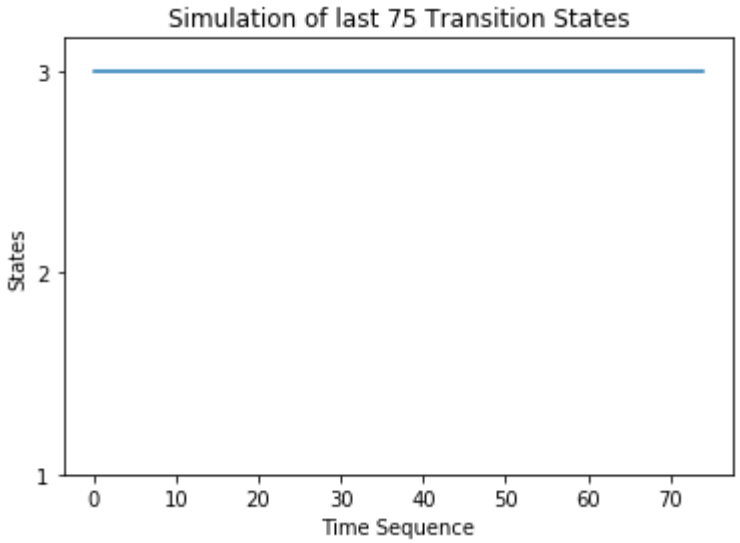
P3 with time length t=500 [Ensemble Average]

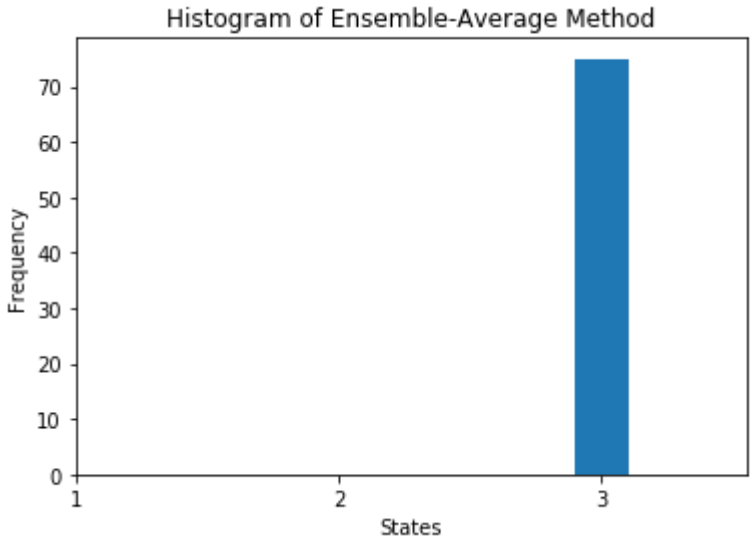
In [536]:

```
ens_states_p3 = []
for sp in range(75):
    state_trans = simulate_markov_chain(3,P3,500,pi_init_state_p3)
    ens_states_p3.append(state_trans[499])
```

In [537]:

```
plt.plot(ens_states_p3)
plt.yticks([1,2,3])
plt.xlabel('Time Sequence')
plt.ylabel('States')
plt.title('Simulation of last 75 Transition States')
plt.show()

print()

plt.hist(ens_states_p3,bins=5)
plt.xticks([1,2,3])
plt.xlabel('States')
plt.ylabel('Frequency')
plt.title('Histogram of Ensemble-Average Method')
plt.show()
```





Summary:

1. For the simulation of the DTFS Markov chains. we first start with state 1 or,

   $\pi(t=0)$ = (1,0)

2. We then use the chapman-kolomogorv equation to get the next state $\pi(t=1)$,

   $\pi(t+1)$ = $\pi(t)$P

   This is a recursive formula as the Markov-Chain is Homogeneous and results in,

   $\pi(t=N)$ = $\pi(t=0)P^N$

3. For Every $\pi(t=N)$, we take a decision which state it transits to based on the results of a uniform random number generation. If the particular probablistic state is less than or equal to the the probability value we select that state else we go to the other state with the higher probability. This method is for the Markov-Chain with 2 states. For the 3 state Markov-chain we threshold on the random number generated keeping the probabilities of the 3 states in mind and decide to move to the next state.

1. We this method we generate the single path time-average simulation for 500 time sequences and plot the histogram of the state occurrences for the last 75 time sequences.

1. we use the above method and run the single path time-average simulation for 75 times and take the 500th or last time sequence to obtain 75 ensemble-average path simulation. We then plot the histogram of the state occurrences for this.

1. we compare the histograms from the time-average and ensemble-average method for each of the three transition matrices P1,P2 and P3 and draw conclusions about the convergence and ergodicity.

Results:

1. For

$$P1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The histogram for time-average method has almost equal distribution of occurrences of states 1 and 2, whereas the ensemble-average method shows us that the all the occurrences belong to state 2. They are hence, not equal.

Convergence: by the histogram goodness-of-fit statistic we can say that since both the histograms (time and ensemble-average) are not nearly equal. we say that the long-run(time-average) and the stationary(ensemble average) distribution do not converge.

ergodicity: since both the histograms are not equal or near equal they are hence not ergodic. Based on the transition matrix, the periodicity of getting from either of the states d(i) where i = 1,2 is equal to 2, we conclude that the Markov chain is periodic and is Irreducible. But for Ergodic Markov Chains we require them to be Aperiodic and Irreducible. Therefore the Markov Chain with transition probability P1 is not Ergodic.

1. For,

$$P2 = \begin{pmatrix} 0.75 & 0.25 \\ 0.1 & 0.9 \end{pmatrix}$$

The histogram for time-average method has twice as many states in 2 rather than in 1 and even the ensemble-average method shows us this behaviour. Hence, they are roughly/approximately equal.

Convergence: by the histogram goodness-of-fit statistic we can say that since both the histograms (time and ensemble-average) are nearly equal. we say that the long-run(time-average) and the stationary(ensemble average) distribution do successfully converge.

ergodicity: since both the histograms are nearly equal they are hence Ergodic. From the transition matrix as well we conclude that the Markov chain is Aperiodic and is also Irreducible. Therefore, the Markov Chain with transition probability P2 is Ergodic.

1. For,

$$P3 = \begin{pmatrix} 0.48 & 0.48 & 0.02 \\ 0.22 & 0.7 & 0.08 \\ 0 & 0 & 1 \end{pmatrix}$$

The histogram for time-average and the ensemble-average method is exactly the same.

Convergence: by the histogram goodness-of-fit statistic we can say that since both the histograms (time and ensemble-average) are exactly equal. Therefore, we could say that they do converge but are not sure looking at the transition matrix P3.

ergodicity: since both the histograms are exactly equal we could say that they are Ergodic but seeing the transition Matrix we halt our conclusion. But From the transition matrix and the simulation graphs we see that 3 is an absorption state. Therefore, it is reducible but aperiodic. For an Ergodic Markov Chain it must be Aperiodic and Irreducible. Since, this is not the case the Markov Chain with transition Matrix P3 is not Ergodic and Does not converge.

Hence, The Markov Chain with transition matrix P3 does not Converge and is not ergodic.