Stochastic Processes and Applications

Assignment 2

Sahil Goyal

2020326

Report

Q1)

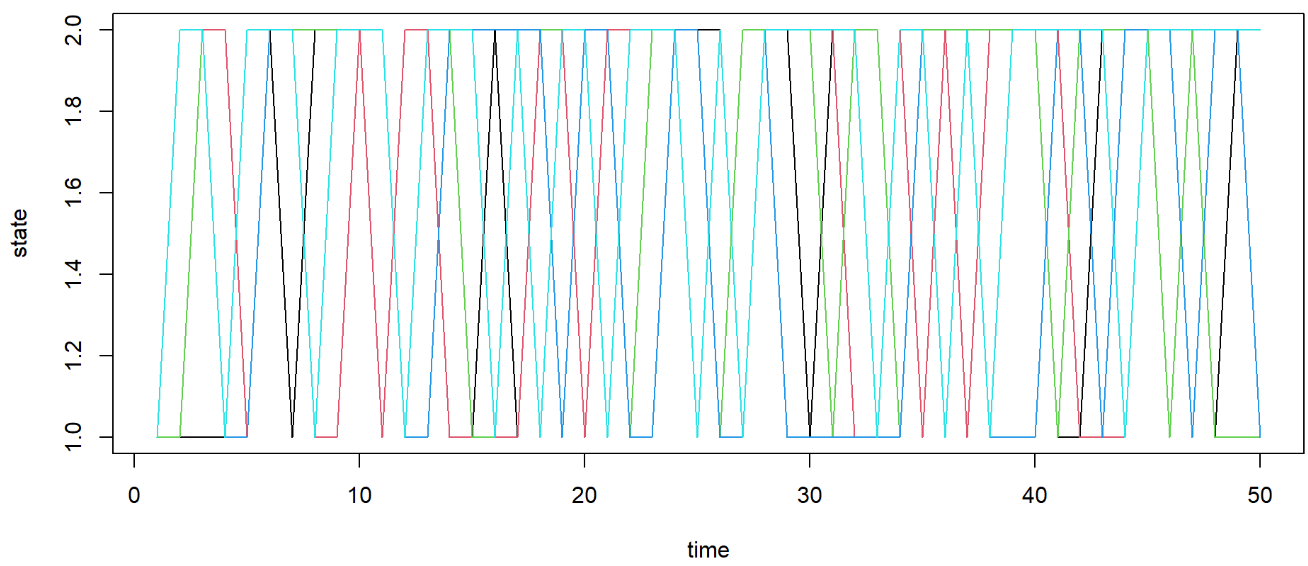
The State Space of the Matrix S is {1,2}

a)

In my code, we have begun by first creating variables for storing the current state, the probabilities and creating an empty matrix that will be used later. After that, we create a for loop that will run 5 times. Inside it, we create an empty vector y and append the current state to it. Then, in a loop that runs 50 times, we append the state to y, and then finally, we assign a column of the empty matrix to y. In this way, we have successfully simulated 5 times a 50 step Markov chain. After this, we plot it using matplot function. In this way, we have successfully plotted a state vs time graph.

To find the next state, we have used sample function. In the output of sample function, if we get a success, the state has changed otherwise the state remains the same.

Plot:



As we can see, we have plotted 5 lines of 50 step Markov chains.

b)

In my code, we have created the matrix P, and then a diagonal matrix ans. Then, we run a loop from 1 to 50 in which we multiply ans by P, and then use this loop to find the values of P_{10},P_{20} and P_{50}.

Their values are:

```
[1] "P 10 is:"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166666 0.5833334
[1] "P 20 is:"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166667 0.5833333
[1] "P 50 is:"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166667 0.5833333
```

We can see that we can reach state 1 from state 2, and we can reach state 2 from state 1. Therefore, we can see that the given Markov chain is irreducible. We can see that both states have a time period of 1, and therefore the Markov chain is also aperiodic and all states are recurrent. Also, we can see that P_{10}, P_{20}, and P_{50} are all the same(i.e., all the elements in the matrices are the same), we have reached a limiting distribution.

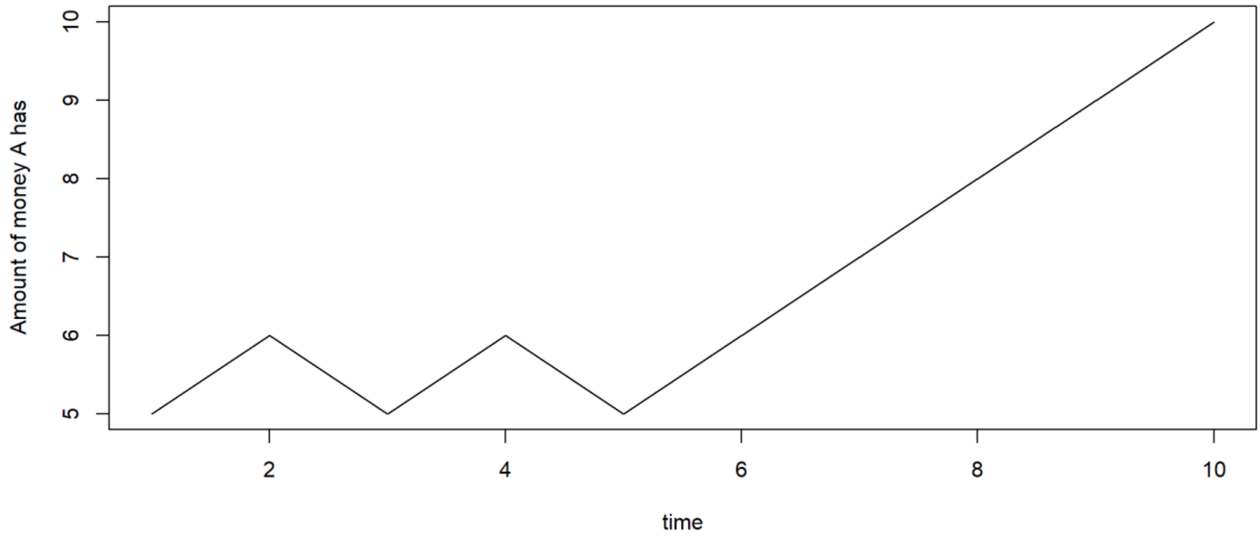
Q2)

In this question, we had to simulate a situation where A and B both initially have 5 dollars, and the probability of A winning is 0.8, and B winning is 0.2, and they keep playing until either one of them runs out of money. This is like the Gamblers problem.

In my code, first we store the amount of money A has. We also store the probability of A winning and make an empty vector y which we will use to store the amount of money A has. After that, we make a while loop. Inside the loop we use the sample function in order to find out who one, and then update the amount of money A has accordingly, and store it in y. After someone runs out of money, we plot the amount of money A has had throughout the game.

We do not need to plot the amount of money B had, as the total money is constant (10 dollars) because of which if the amount of money A has increases, the amount of money B has will decrease.

Plot:



In this particular simulation, we can see that A lost the game twice, and won the game 7 times, eventually getting the entirety of the 10 dollars, and B had no money left.