BITCOIN PRICE PREDICTION

Data Modeling Project

By

Abhilasha Jain

MATH 7241 – Probability I

Supervisor: Professor Christopher King

APPLIED MATHEMATICS, MS 2022

Contents

1. Introduction

A Markov chain is a stochastic process where the distribution of a future state of a random sequence depends only on the distribution of the current state and not the past states.

$$P(X_{n+1} = j | X_0 = i_0, \dots, X_n = i_n) = P(X_{n+1} = j | X_n = i_n) \qquad (1)$$

for all $n \geq 0$ and all states $j, i_0, \dots, i_n \in \Omega$.

The objective of this project is to analyze a time-series and find out if the Markov chain method produces a good model for the time series.

I will be analyzing the everyday prices of bitcoin for this project.

2. Data Description

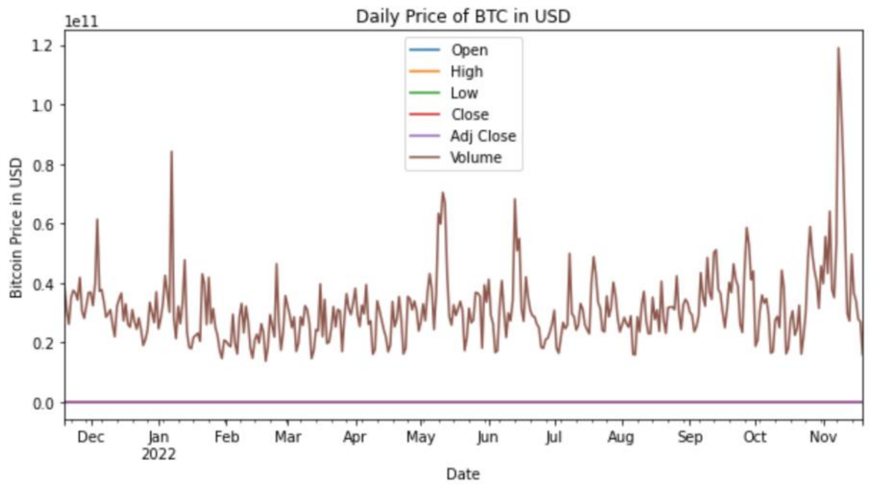
Bitcoin employs peer-to-peer technology to operate without a central authority or banks; the network as a whole is responsible for handling transactions and issuing bitcoins. Since Bitcoin is an open-source project, anyone can participate, no one owns or controls it. Due to a number of its special characteristics, Bitcoin enables novel applications that were not possible with earlier payment systems.

I have downloaded the daily bitcoin prices from Yahoo Finance. Daily bitcoin prices are available from September 2014. I will be analyzing the daily prices observed in the past one year.

Summary statistics of daily bitcoin prices (2021-2022),

Statistics	***Value***
Count	366
Mean	32355.950374
Std	11820.176548
Min	15883.158203
25%	20594.497071
50%	30244.466797
75%	42198.630860
Max	59730.507813

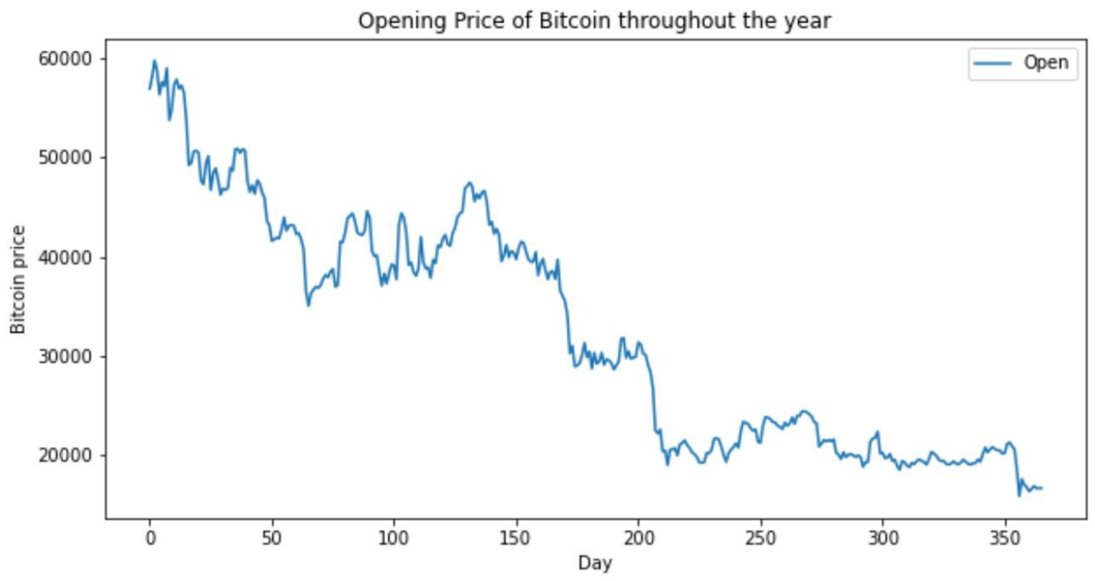
Here's a plot of the timeseries of raw data



3. Data Cleaning

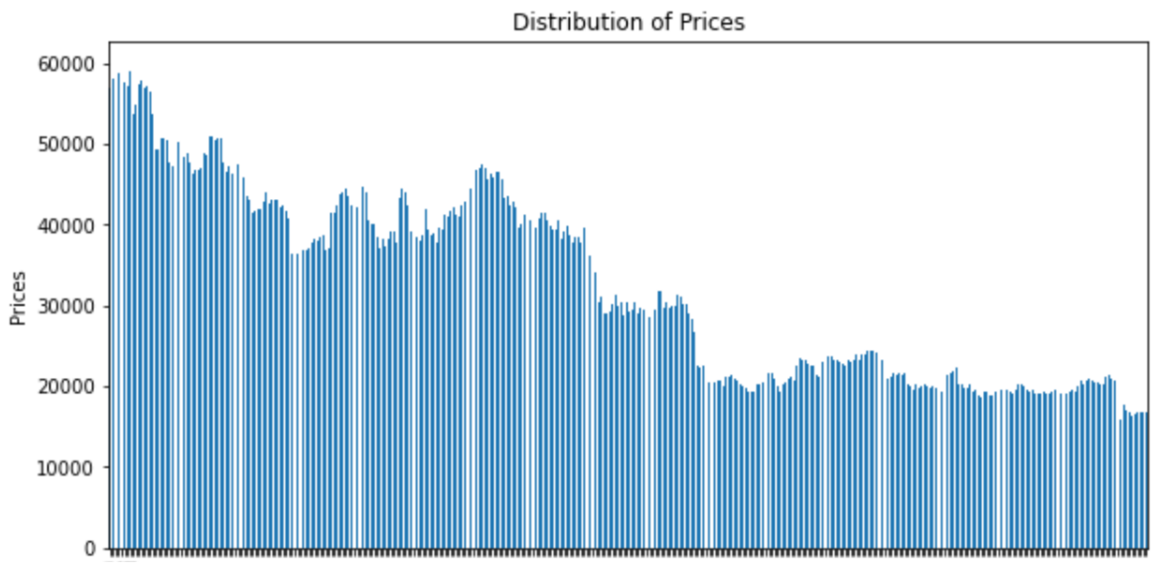
For the purposes of this project, I have chosen to use daily opening prices of Bitcoins along with the timestamp to predict the opening price of next day using Markov Chain.

Here's the time series plot of cleaned data:



4. Data Analysis & Modeling

The price of bitcoin ranged between 10,000 USD to 60,000 USD in past 366 days. Here's the distribution of prices:

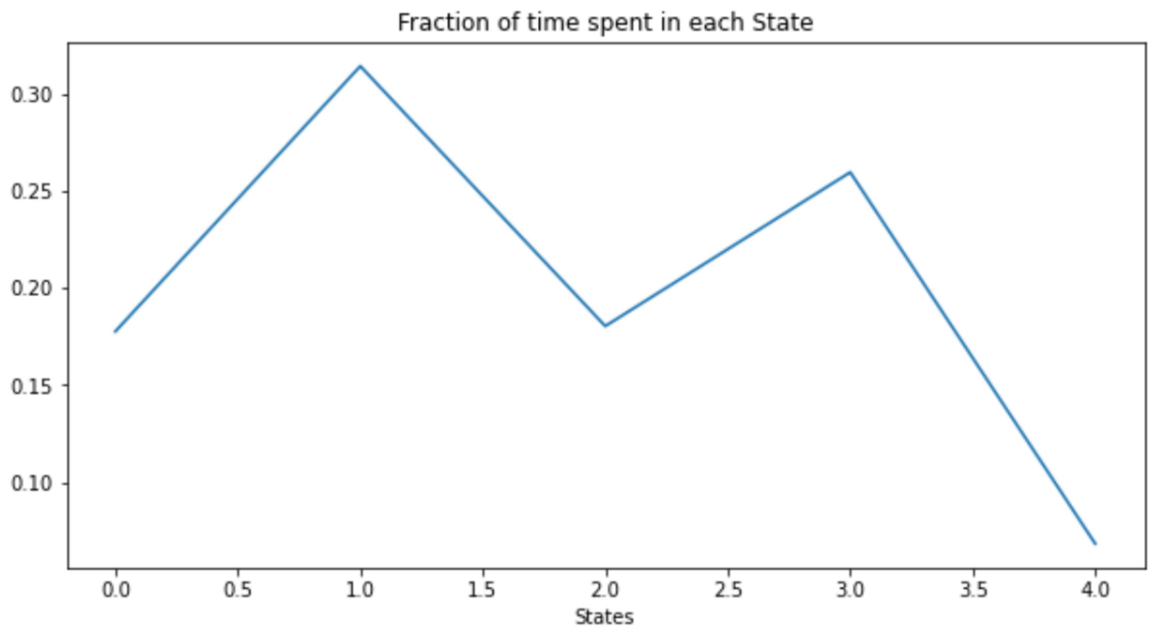


Based on the above distribution, I have divided the data into groups of 10,000 USD, starting 20,000 USD until 60,000 USD and all the counts ≤ 20,000 USD and ≥ 60,000 USD are in a single group. This gives 5 Markov states.

Price Range	***Markov States***
≤ 20,000	0
20,001 – 30,000	1
30,001 – 40,000	2
40,001 – 50,000	3
≥ 50,001	4

Note: I have used Python for my analysis

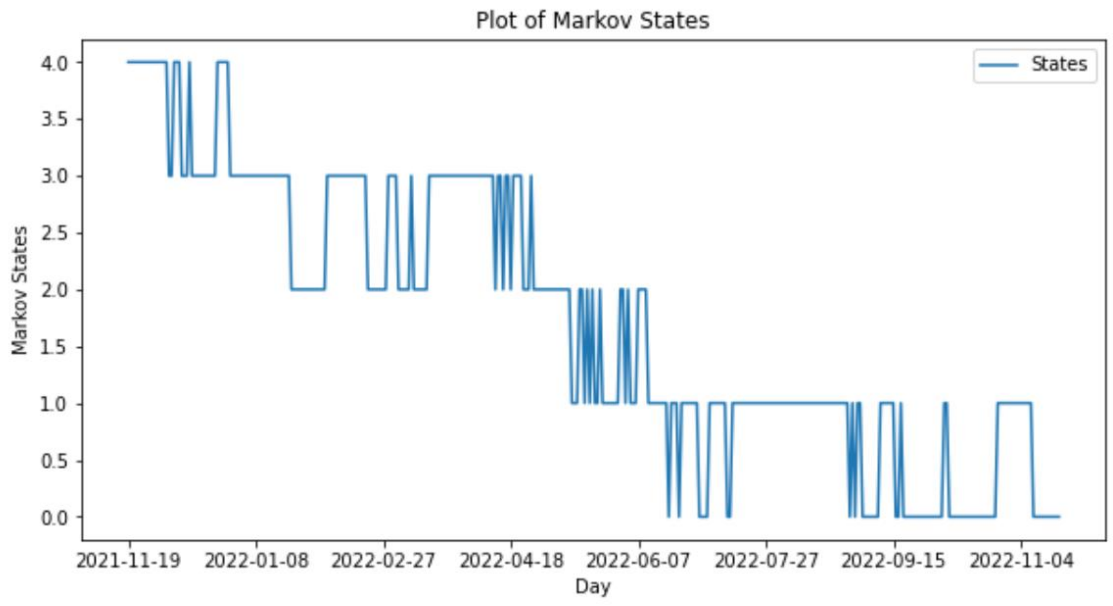
4.1 Empirical Distribution



Statistics	***Value***
Count	366
Mean	1.726776
Std	1.219339
Min	0
25%	1
50%	2
75%	3
Max	4

The empirical distribution of the states is Bimodal. The statistical summary did not suggest that the data follow a bimodal distribution. The first and third states are modes. In the descriptive statistics, we notice how the mean and median (both near 2) lie between modes where there are relatively few observations. Typically, these measures find where most values fall, but that's not the case here, reducing their usefulness in bimodal distributions. Bitcoin prices stayed between 10,000 USD to 20,000 USD and 30,000 USD to 40,000 USD throughout the year.

We could also visualize the Markov states as a time series



4.2 Transition Matrix

The transition probabilities for the different states are as follows,

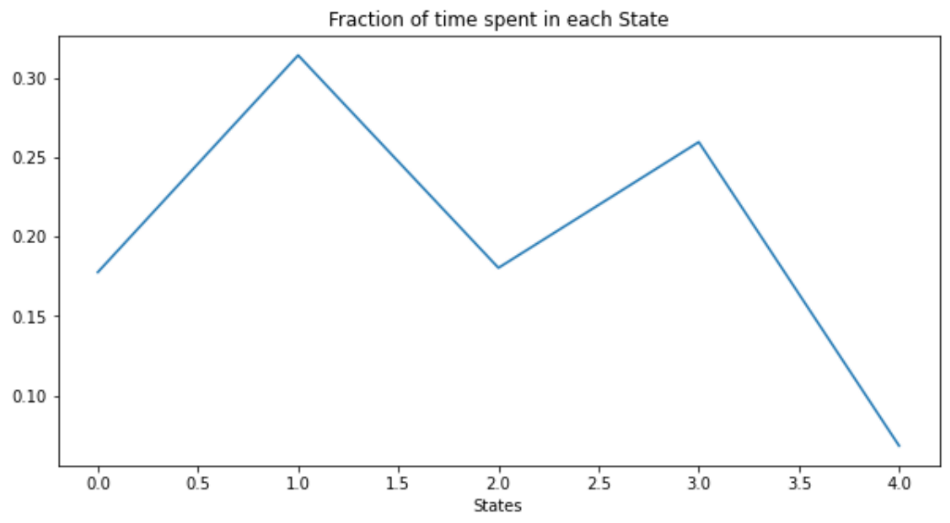
$$\begin{array}{ccccc|c}
0.844 & 0.156 & 0 & 0 & 0 & 0 \\
0.096 & 0.843 & 0.061 & 0 & 0 & 1 \\
0 & 0.121 & 0.758 & 0.121 & 0 & 2 \\
0 & 0 & 0.095 & 0.874 & 0.032 & 3 \\
0 & 0 & 0 & 0.16 & 0.84 & 4 \\
\hline
0 & 1 & 2 & 3 & 4 &
\end{array}$$

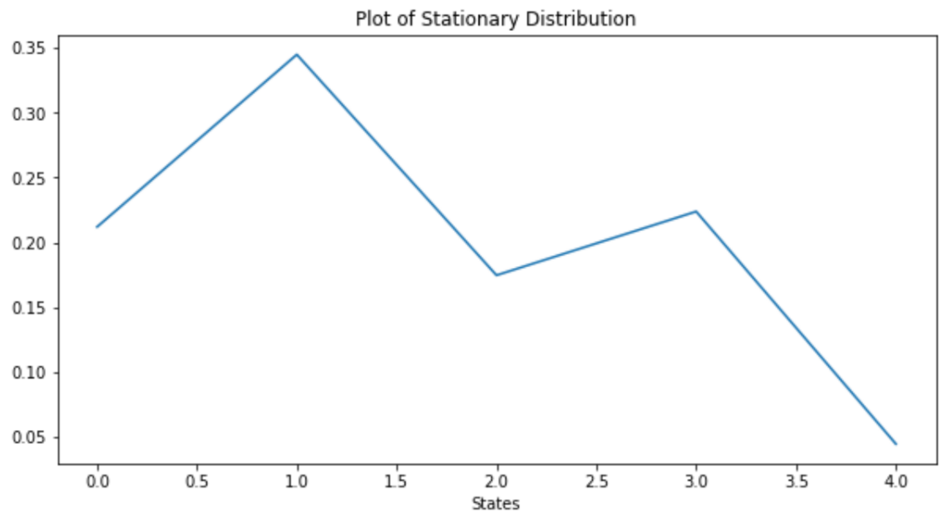
Probabilities along the diagonal are higher than the others. Bitcoin prices change gradually every day. As I have grouped 10,000 counts in a state, the probabilities for jumping to the same state is higher than others.

To compute the Transition matrix, I wrote a function in Python available in the Appendix of the report.

4.3 Stationary Distribution

This is the stationary distribution of the chain,
W = [0.21201313, 0.3449842, 0.17465264, 0.22385965, 0.04449037]
From the stationary distribution, we can see that the long run probabilities are higher for states 1 and 3, but for 1 is the highest. For the period I have selected for analysis, we can expect most of the days to have the price in the range that corresponds to the states 1 and 3 which is not moving towards the increment in the bitcoin prices.
Function to compute the stationary distribution is included in the Appendix of the report.
Comparison of empirical distribution from original distribution to the stationary distribution of the states,



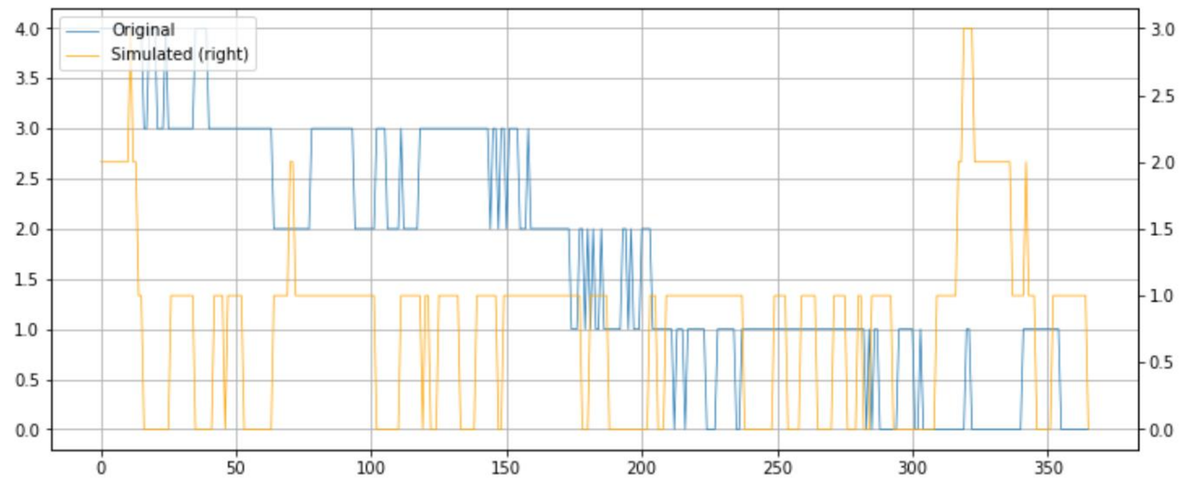


From the above plots we can see that stationary distribution and the distribution of states from the time series are similar.

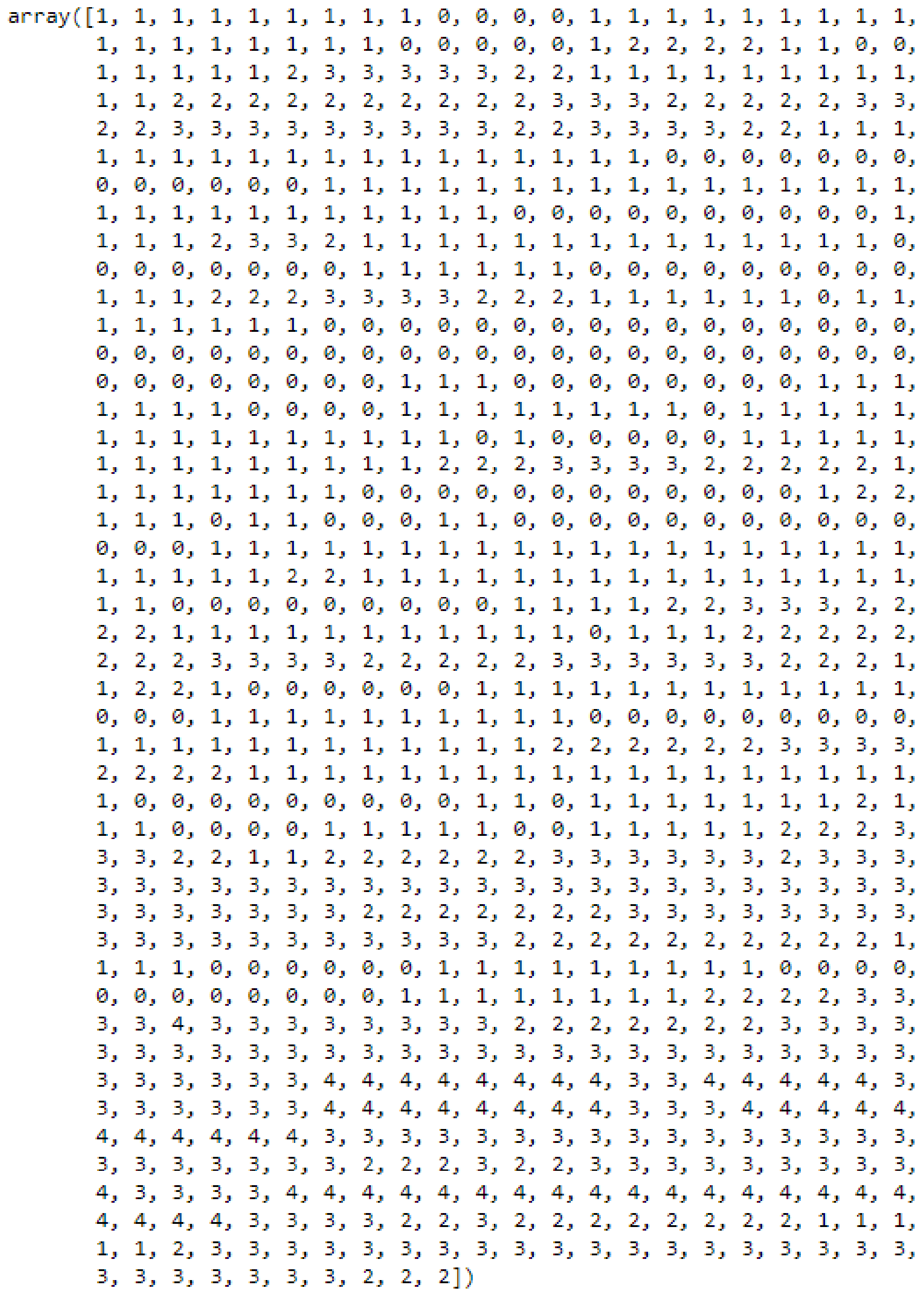
4.4 Simulated Time Series

I wrote a function to generate a time series based on the transition matrix of the original time series. First state is generated randomly. The next state is selected based on the transition probabilities of the first state. I have converted the transition probabilities of each state into a cumulative distribution. Then I select the next state based on a random number generated between 0 and 1.
For example, if the first state is 3, the transition probabilities for state 3 are [0 0 0.095 0.874 0.031]. The cumulative distribution is [0 0 0.095 0.969 1]. Now a random number between 0 and 1 is generated, say 0.46. Now, the next state is chosen as 3 based on above distribution. This is iterated and a new series is generated with 1000 values.

Here's a comparison of transition of states of the original series and the simulated series for first 365 timesteps.



Markov Chain Simulation for next 1000 days.



```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 2, 2, 2, 2, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2, 3, 3,
       2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 2, 2, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 2, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2,
       1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 3, 3, 2, 2,
       2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 2, 2, 2, 2, 2,
       2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 2, 2, 2, 1,
       1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
       2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 2, 1,
       1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 3,
       3, 3, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3,
       3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 4, 4, 4, 4, 4, 3,
       3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       4, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 3, 3, 3, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1,
       1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 2, 2, 2])
```

4.5 Goodness of fit Test

We further analyze our original time series using goodness of fit test for the 2-step transition probabilities. We compare the observed 2-step jumps in our original time series with the expected jumps which is obtained by multiplying the 2-step jumps by the 2-step transition probabilities (square of our original single step transition matrix) for each state. We are testing our hypothesis that the observed frequencies follow the expected frequencies for two-step jumps.
We compute the test statistic (Pearson's) and the chi-squared distribution for each state and reject the null hypothesis if TS > chi-squared. We are using 5% significance level.

We get the following results:

States	***Test statistics (TS)***	***Chi Squared test***
0	1.25175601	5.99146455
1	3.49838054	7.8147279
2	3.69751162	9.48772904
3	1.37026407	7.8147279
4	0.38113277	5.99146455

Based on the above results, we don't have enough evidence to reject the null hypothesis for all the states. So, the Markov chain is a good model for 2-step transitions of the original time series.

5. Conclusions

-The randomly simulated time series based on the transition probabilities of the original series also has a similar behavior.
-The goodness of fit test doesn't have enough evidence to reject the null hypothesis.
-From the above analyses, I am concluding that the Markov chain method produces a good model for the daily prices of bitcoin observed in last one year. The everyday prices over a longer period can be analyzed to understand their pattern over the years.

6. Appendix

6.1 Function to Compute Transition Matrix

```
def TransitionMatrix(states):
    pd.value_counts(states)
    y = len(pd.unique(states))
    P = np.zeros([y,y])

    for i in range(y):
        for j in range(y):
            for x in range(len(states)-1):
                if states[x] == i and states[x+1] == j:
                    P[i][j] += 1


    for row in P:
        s = sum(row)
        if s > 0:
            row[:] = [round(f/s,3) for f in row]
    return P
```

6.2 Function to Compute Stationary Vector

```
def StationaryDistribution(P):
    A = P.T-np.identity(P.shape[0])
    A = np.vstack([A,np.ones((P.shape[0]))])
    b = np.zeros((P.shape[0])).T
    b = np.zeros((P.shape[0]+1,1))
    b[-1] = 1
    W = np.linalg.lstsq(A,b,rcond=None)[0]
    return W
```

6.3 Function to Compute Two step Transition Frequencies

```
def TwoStepFrequency(states):
    pd.value_counts(states)
    y = len(pd.unique(states))
    P = np.zeros([y,y])

    for i in range(y):
        for j in range(y):
            for x in range(len(states)-2):
                if states[x] == i and states[x+2] == j:
                    P[i][j] += 1
    for row in P:
        s = sum(row)
        if s > 0:
            row[:] = [round(f/1,3) for f in row]
    return P
```

6.4 Function to Compute Test Statistic and Chi Square Value

```
def GoodnessofFit(N,Q):
    n = 0
    TS = np.zeros((N.shape[0]))
    chi2 = np.zeros(N.shape[0])
    p_value = np.zeros(N.shape[0])
    for i in range(N.shape[0]):
        N1 = N[i][Q[i]>0]
        n = N1.sum()
        chi2[i] = stats.chi2.ppf(q = 0.95, df = len(N1) - 1)
        for j in range(N.shape[0]):
            if Q[i][j] > 0:
                O = N[i][j]
                E = n * Q[i][j]
                TS[i] += ((O - E)**2 / E)
    return TS, chi2
```

7. References

Bitcoin USD (BTC- USD). (n.d.). Retrieved from Yahoo Finance: https://finance.yahoo.com/quote/BTC-USD/history?p=BTC-USD

Jain, A. (n.d.). *Markov Chain Data Modeling*. Retrieved from thetechgirl14 Github: https://github.com/thetechgirl14/Markov-Chain-Data-Modeling