SIGMA WEDGE – CODING CHALLENGE

- KRITHIKA L (21PD19)

The goal of this problem statement is to get familiarized with the Quantrocket platform and also apply algorithms to build a simple model to make decision based on the stock prices.

1. Data collection through Quantrocket platform

I obtained the closing prices data for each day from 01-01-2023 to 31-12-2023 of the AAPL stock from Quantrocket platform. Using the platform, universities and filtered universities were created to retrieve the desired stock prices. And now, I will be applying my algorithm to this historical data to get portfolio value and buy indices.

2. My approach using the pre-specified logic

I designed a **Markov chain model** (statistical model) for this problem statement. A Markov chain is a stochastic process that involves a sequence of random variable with discrete or continuous index set and state space where the probability of moving to the next state depends only on the current state and not the previous state.

There are 3 measure we need to be well aware of:

- **States**: All the states (occurrences) within the state-space 's' of the dynamical system
- **The initial state distribution**: The initial probability distribution of the starting state. It is encoded into a column vector denoted as 'q'
- **State transition probabilities**: transition probability of moving from one state to another. It is encoded into an s-by-s matrix denoted as 'P'

We want to model the stock markets trend. To do so we must first recall our assumption that a stocks market movement is random.

Therefore there is a **dynamical system** we want to examine — the stock markets trend.

We can identify that this system transitions **randomly,** between a set number of states.

All possible collections of all these states is called a **state-space**. For our case, we can identify that a stock markets movement can only take on three states (the state-space):

Bull: the price is increasing

Bear: the price is decreasing

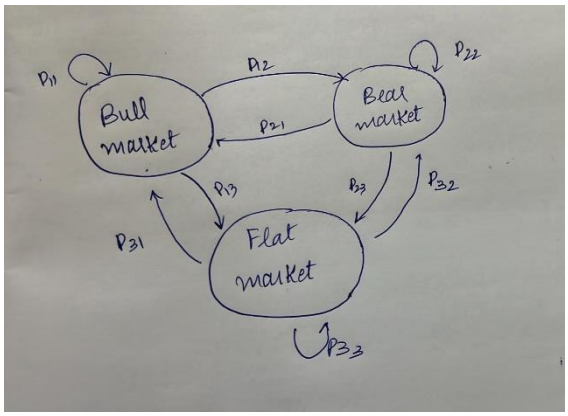
Flat: there is no change in price, neither increasing nor decreasing

Each of these states are **unique** occurrences. The total number of occurrences of a state is the **frequency**, and it is reflected by the number arrows pointing to it on a transition state diagram.

In our case we shall adopt the premise that stock market trends are independent of past events and only the current state can determine the future state.

Since the system contains states, is random, and satisfies Markov's property — we may therefore model our system as **a Markov chain**. Additionally, we will choose to model in **discrete-time**.

a) <u>STATE TRANSITION DIAGRAM:</u>



Since we have three states we will henceforth have a three-state Markov chain.

Let us draft the transition state diagram,

b) <u>CALCULATE RETURNS AND STATES:</u>

The returns[r(d)] here is calculated by subtracting the previous state price from current state and diving it by previous state price (i.e)

R(d) = [p(d) – p(d-1)] / p(d-1)

Therefore, let us define X(n) is a stochastic process where n>=0 as

X(n) = 1 (bull state) if R(d) >=0.0.1

0 (flat state) if R(d) >-0.01

-1 (bear state), otherwise.

As the returns for the stochastic process X(n) is calculated solely based on its previous state's return and not the one's before, therefore we can say that X(n) is a Markov chain.

Next we compute the transition probability matrix P_{ij} where i and j are the states that the Markov chain will be transitioning to. Therefore P_{ij} is computed as :

```
Transition Probabilities:
To           -1         0         1
From
-1     0.138889  0.722222  0.138889
 0     0.146497  0.598726  0.254777
 1     0.125000  0.678571  0.196429
```

That is, if the current state is -1, the probability of transitioning -1 on the next step is 0.138889. This is also called as **1-step transition probability**.

Similarly we can also compute 2-step transition probability.

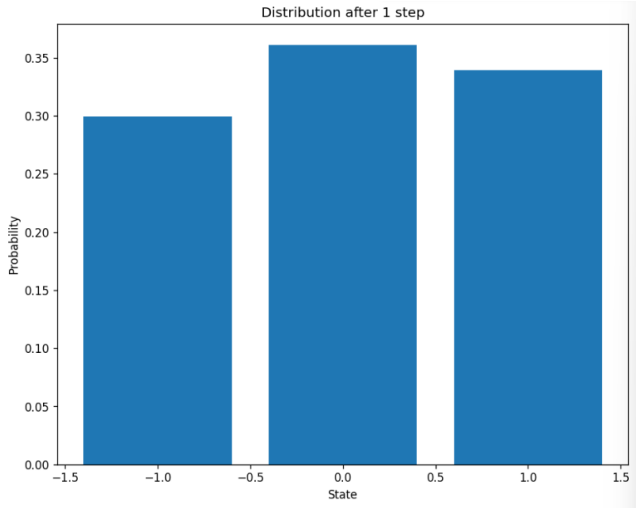
$P_{ij}(2)$ =

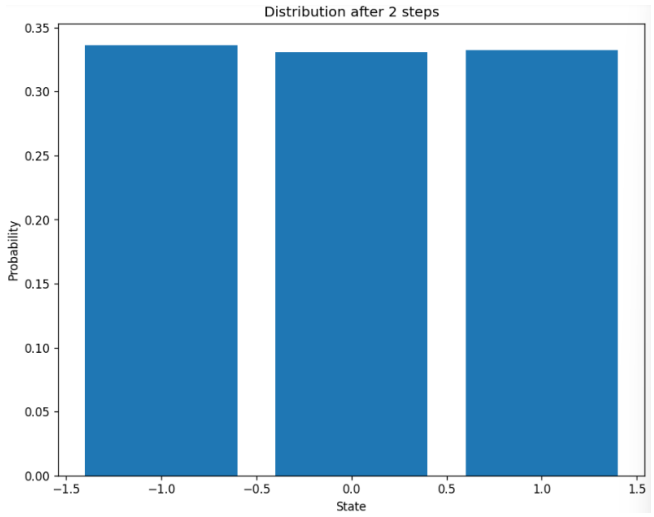
```
[[0.14245464 0.62696783 0.23057754]
 [0.13990551 0.63716046 0.22293403]
 [0.14132337 0.62984687 0.22882976]]
```

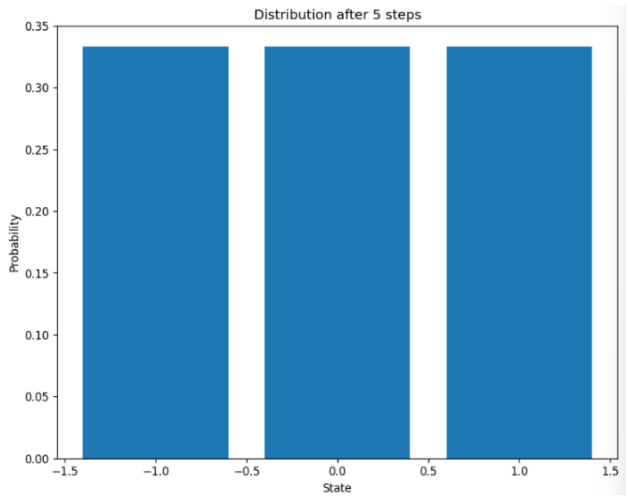
That is if the current state is -1, then say for example 2 days/steps later, the probability of me landing state -1 is 0.142454. This result is obtained by squaring the 1-step transition probability. Similarly we can also find n-step transition probability by multiplying the matrix n times with itself.

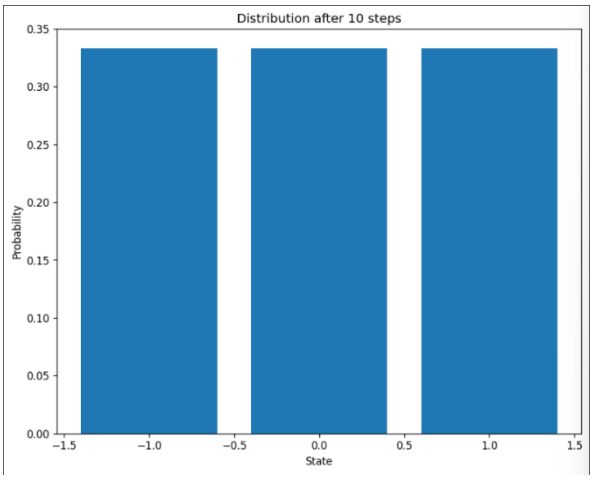
But if X(n) is a Markov chain, then it has a property called as **stationary distributions,**

In the COLAB notebook I have proved this both mathematically and visually.









Here we would have noticed that, as the number of simulation increased, we get an interesting phenomenon. This is because as we increased the number o simulations, we some fluctuations in the first step but eventually they will stabilize to what is called as stationary distribution.

c) CALCULATE PORTFOLIO VALUE AND BUY INDICES

Based on the value function (i.e) if the current state if bull state and previous is flat, then portfolio value increases by 1 and if the current state is bear state and previous is flat, then portfolio value decrease by 1, or else the value remains the same.

Based on this logic, I got the output of the portfolio value and buy_indices as :

```
Portfolio value:  17
Buy indices:  [6, 8, 12, 16, 21, 28, 30, 33, 36, 39, 41, 45, 50, 52,
57, 59, 61, 64, 66, 69, 79, 85, 88, 94, 97, 100, 103, 108, 110, 113,
117, 120, 123, 128, 133, 136, 142, 145, 147, 154, 156, 160, 164, 169,
173, 177, 179, 183, 187, 191, 196, 201, 204, 207, 209, 212, 216, 218,
232, 234, 236, 238, 243]
```

which is stored in output.txt file.

(task completed)

3) ARIMA model

Further, to complete the entire stock price forecasting, I used a technique called ARIMA model which is a combination of two models Auto Regression (AR) and Moving average (MA) that is predominantly used for time series analysis.

Differencing was done to convert the non-stationery data to stationary data and remove any patterns and trends in the data. The hyper-parameter values p,q and d were found based on auto correlation function and partial auto correlation function. Finally the model was fit and the predicted results were found to be very close to the actual value. Hence the model is a good model with higher accuracy.