FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
BHARAT SWAMI	INDIA	bharatswami1299@gmail.com	
WYCLIFFE KIPKOECH CHERUIYOT	KENYA	cherukipkoech@gmail.com	
CHRIS ENNY OFIKWU	NIGERIA	c.ofikwu@outlook.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an “X” above).	
Team member 1	**BHARAT SWAMI**
Team member 2	**WYCLIFFE KIPKOECH CHERUIYOT**
Team member 3	**CHRISTOPHER ENNY OFIKWU**

Objective

In this group assignment we are going to test and model regime switching models for time series data. Our regime switching models will be based on the Hidden Markov Model (HMM) .
Below are key highlights of our procedure -

- We are going to start with downloading the data from yahoo finance python package "yfinance", which will be the combination of two or more data time series data.
- Then we will clean the data as we require i.e, removing infinite values (after dividing by 0), NaN values or other unwanted data.
- And finally we are going to run our HMM model on time series data with different input parameters or combinations of different input parameters and record the changes.

Data

For the purpose of this group we are going to use time series data. These Time series are stock, index, credits, cryptocurrency or combination of these (mean of every day percentage change for different types).

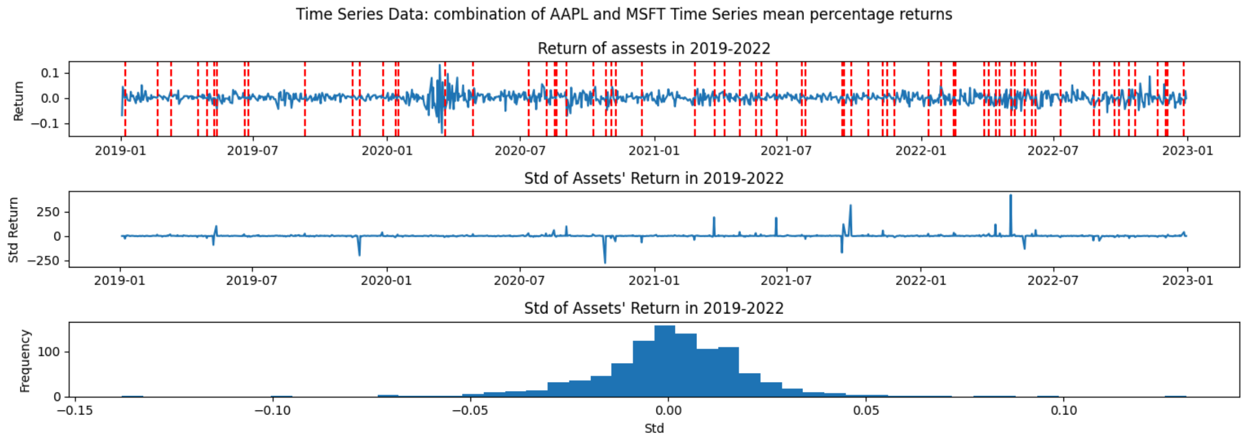
S.no.	Team Member	Data Tickers
1	A	AAPL, MSFT, BTC-USD
2	B	SPY, QQQ
3	C	^DJI, AUDUSD=X

Table 1

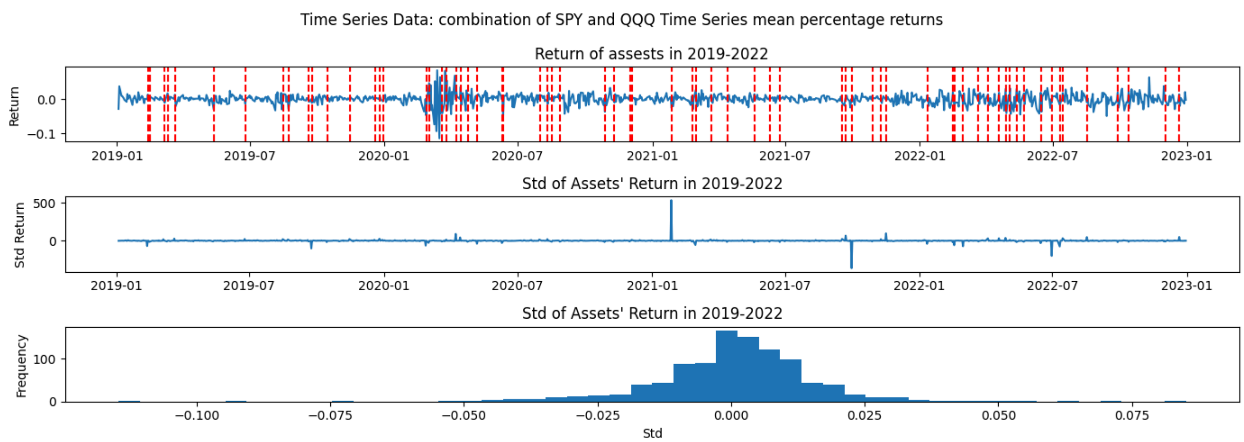
Data Tickers in table 1 represents the ticker which we are going to use in our calculation for the Markov-Regime Change Model. We are going to download the time series data from yahoo finance using mentions ticker.

Now, we are going to use the following steps to visualize our 3 time series data and its regimes where it may change drastically.

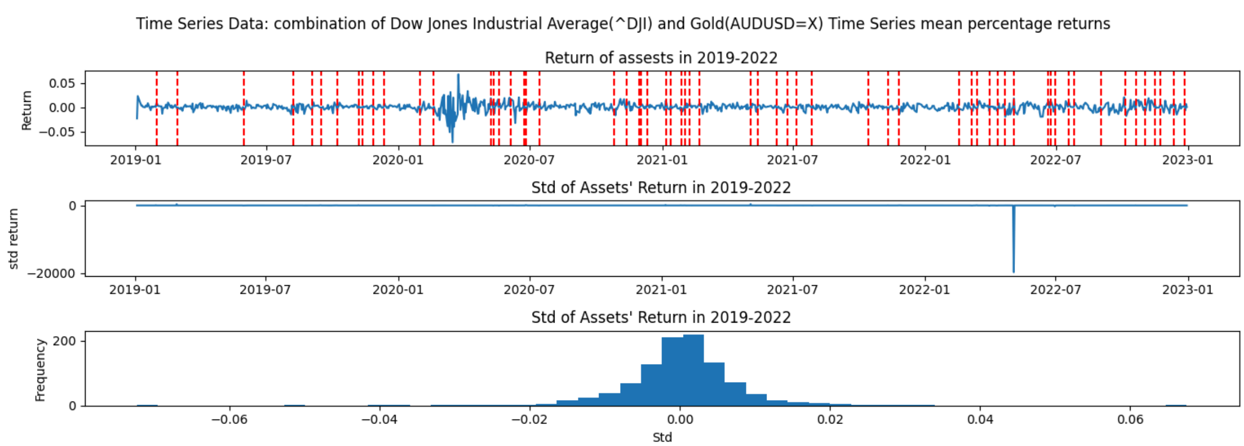
- After downloading the data, we will use adjusted close for all sub time series in our each time series
- Then we are going to calculate the percentage change in adjusted close with a lag of 1 day.
- Next step will be cleaning of unwanted or distorted data like nan and infinite values from data.
- Once we cleaned the data we will take the mean of percentage adjusted close change for all the sub time series for each row. Part (a) of graphs 1,2 and 3 shows the data for time series 1,2,and 3, respectively.
- Now, we are going to choose the threshold for our regime change. Which in our case is 10(arbitrarily chosen). We are again taking the percentage change in data with 1 day lag but this time for time series (mean) data. After comparison with threshold, Part (b) of graphs 1, 2, and 3 shows the standard deviation for our data series and red vertical lines in Part(a) of graphs 1,2, and 3 shows regime change in series.
- Part(c) of graphs 1, 2, and 3 shows the distribution of our standard deviation in histogram bins.



Graph 1



Graph 2



Graph 3

Markov-regime Switching Model / HMM

HMM is a Markov Process or has an observable Markovian component in the process. We only know the observable state data in the form of time series and we like to find out whether a given time series has a HMM or not. These initial parameters are P (Transition matrix), μ initial expectations (constant or varying), σ initial variance (constant or varying) and stationary probabilities π.

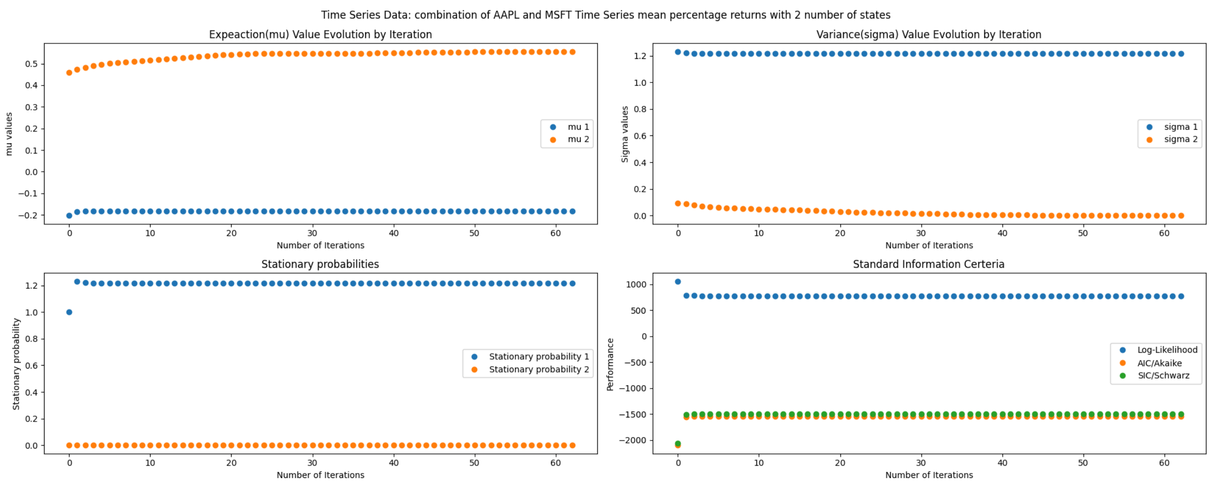
These initial parameters are hard to define and mostly we have to use a hit and trial guessing method for the model to run properly. (For a non-expert).

Markov-regime Switching Model Estimation

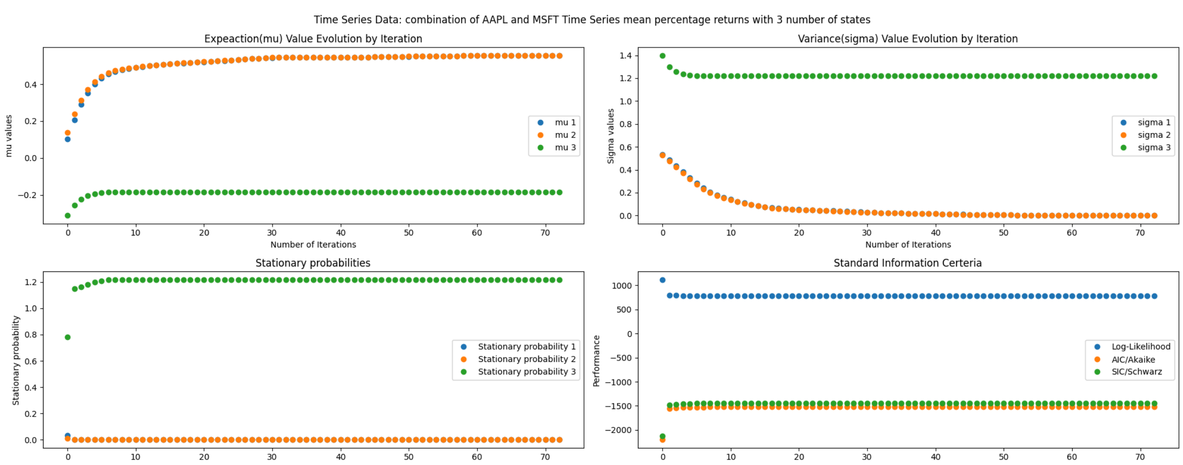
From now on we are going to use the Time Series 1 for our further calculator in the HHM model, unless it's stated otherwise.
Now, The time series 1 is made up of AAPL and MSFT stock adjusted close time series with some modification of percentage change and mean, as mentioned under data heading above.
We are going to use HHM model as described above with different input parameter and combination of different parameters.
Estimating the HHM model with -

Different number of states (N)

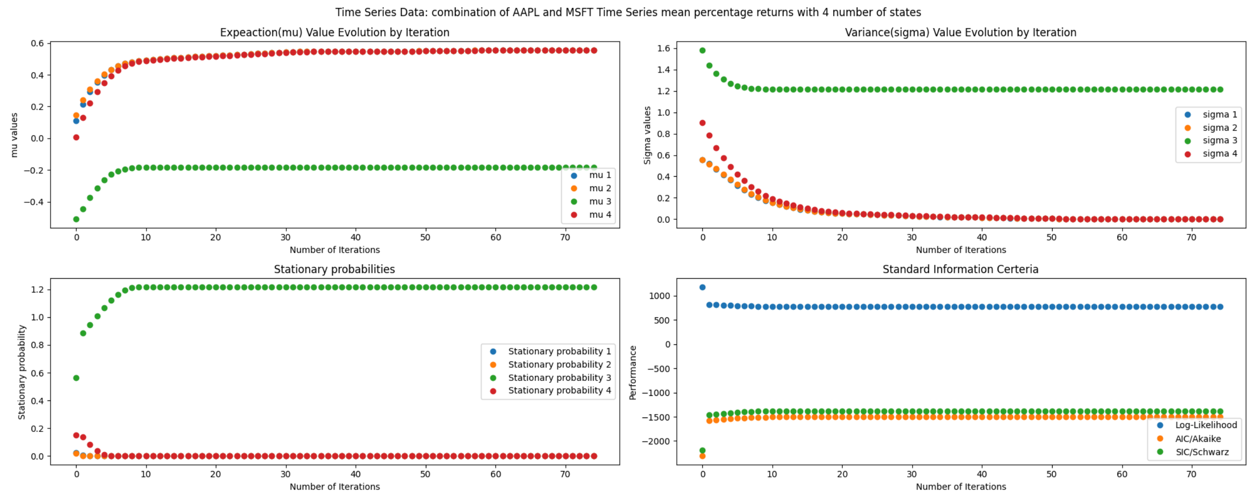
We are going to use $N = \{2, 3, 4\}$ for our calculation.Below are the results from after running the script.



Graph 4



Graph 5



Graph 6

Special Case: N = 1=> Graph for n=1 state shows no results that's why we are not including those here. Reason why they are empty is because n=1 does not make any sense regarding changing state from one state to another, because n=1 says we have only one hidden state which only transitions into itself with probability one, which does fulfill our requirement.

- **Results**

Graphs 4, 5 and 6 are Expectation, variance, Pi and Standard information criteria for different numbers of states (N) with having similar initial parameters with assumptions as mentioned above.
Graphs in this section provide us some insights about underlying hidden chain, which are as follows-

1. Graphs shows the Expectation, variance, Pi and Standard information criteria converges to values as model iterates.
2. We don't need high number of iteration to come up values for these parameters, we can approximated them with low iterations

Allowing mu's to vary, while sigma's are constant

In this section we are going to vary the mu or expectation values while making sure sigma remains constant throughout the HHM process.

Initial Conditions which will be constant throughout the process are-

$$Number\ of\ stages:\ N\ = 3$$

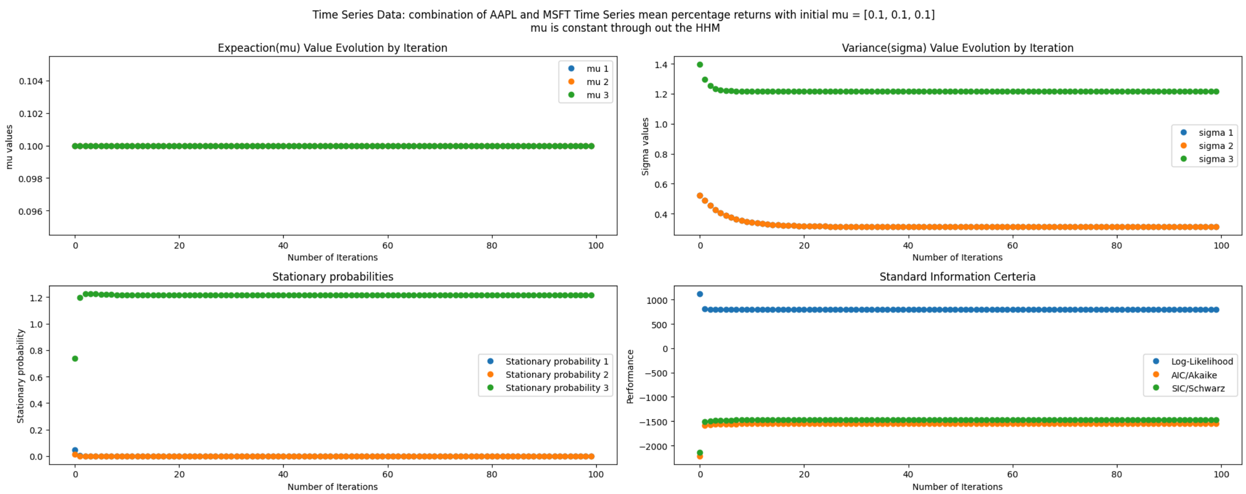
$$Transition\ Matrix:\ P\ = [[0.55, 0.11, 0.34], [0.4, 0.2, 0.4], [0.4, 0.1, 0.5]]$$

$$Initial\ Probabilities:\ \pi\ = [0.5,\ 0.1,\ 0.4]$$

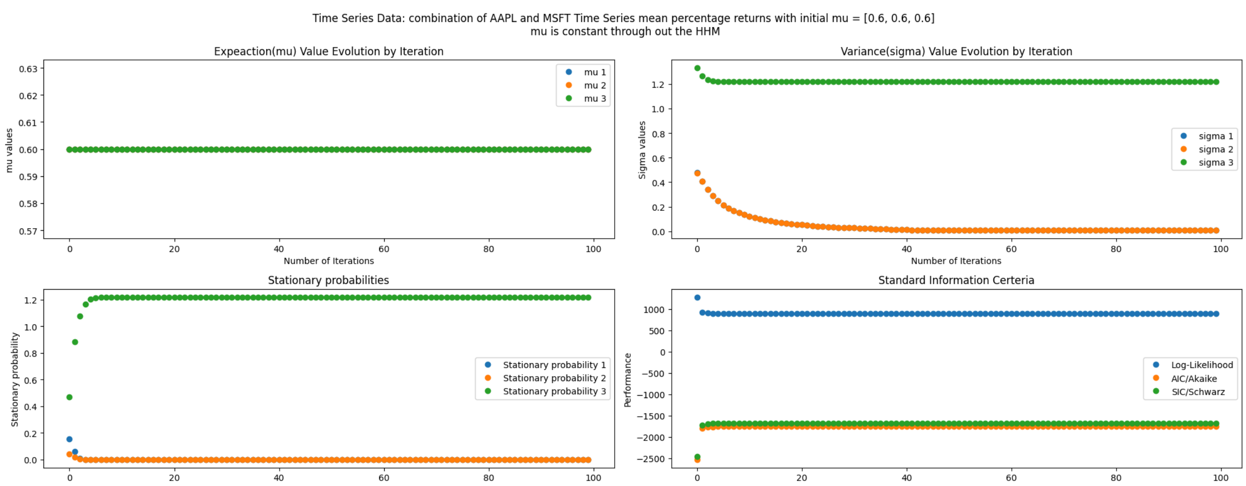
$$Exceptation:\ \mu\ \in\ \{0.1, 0.6, 1.5, -\ 0.3\}$$

$$Variance:\ \sigma\ = [0.5,\ 0.5,\ 1]$$

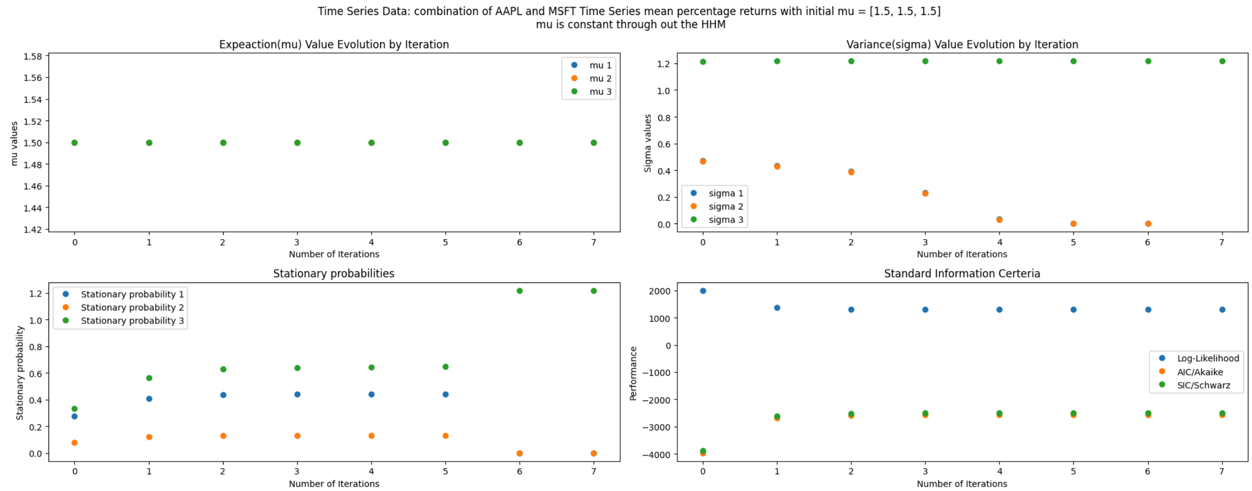
Below are the results we get after running the python script.



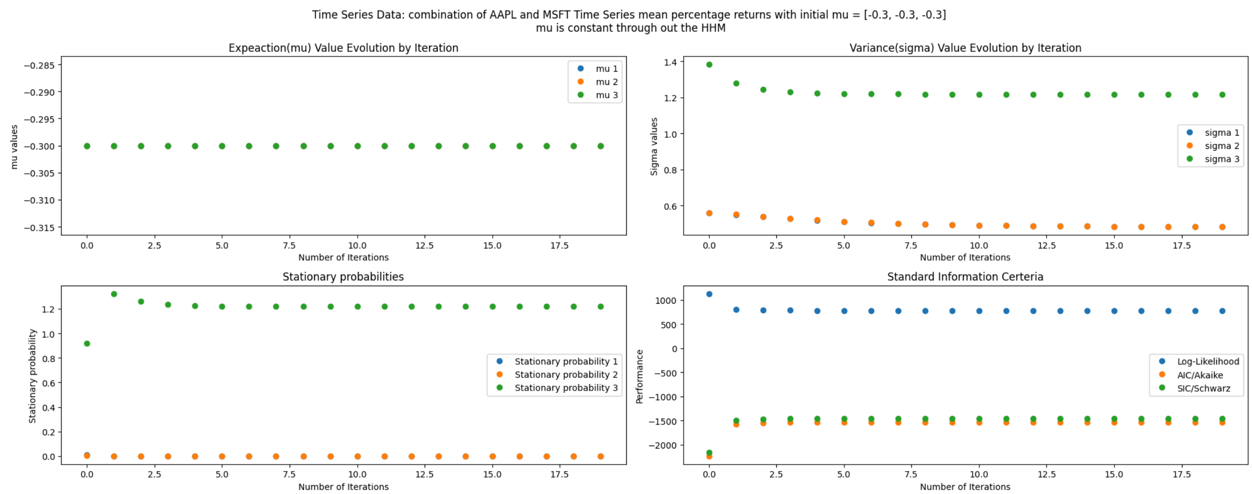
Graph 7



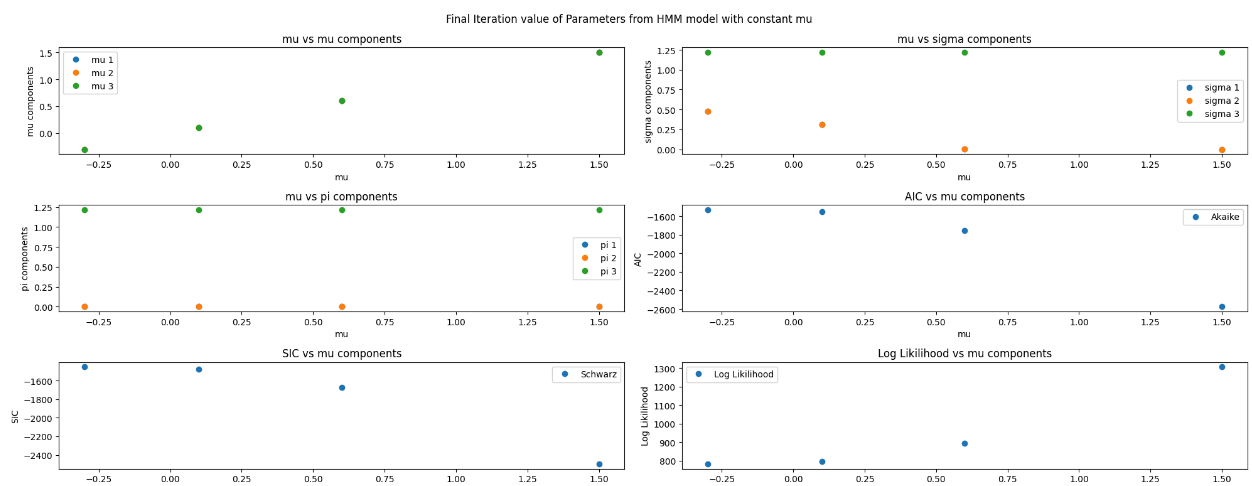
Graph 8



Graph 9



Graph 10



Graph 11

- **Results**

Graphs 7, 8, 9 and 10 are Expectation (contant), variance, Pi and Standard information criteria for same numbers of states (N = 3) with constant Expectation throughout the one model run with similar initial parameters with assumptions as mentioned above. Graphs in this section provide us some insights about underlying hidden chain, which are as follows-

1. Graphs show the Expectation, variance, Pi and Standard information criteria converges to values as model iterates.
2. We don’t need high number of iteration to come up values for these parameters, we can approximated them with low iterations
3. Graph 11 shows the final values (conversing values) from all the model with constant expectation.

Allowing Sigma's to vary, while mu's are constant

Now, in this section we are going to vary the sigma or variance values while making sure expectation remains constant throughout the HHM process.
Initial Conditions which will be constant throughout the process are-

$$Number\ of\ stages:\ N\ = 4$$

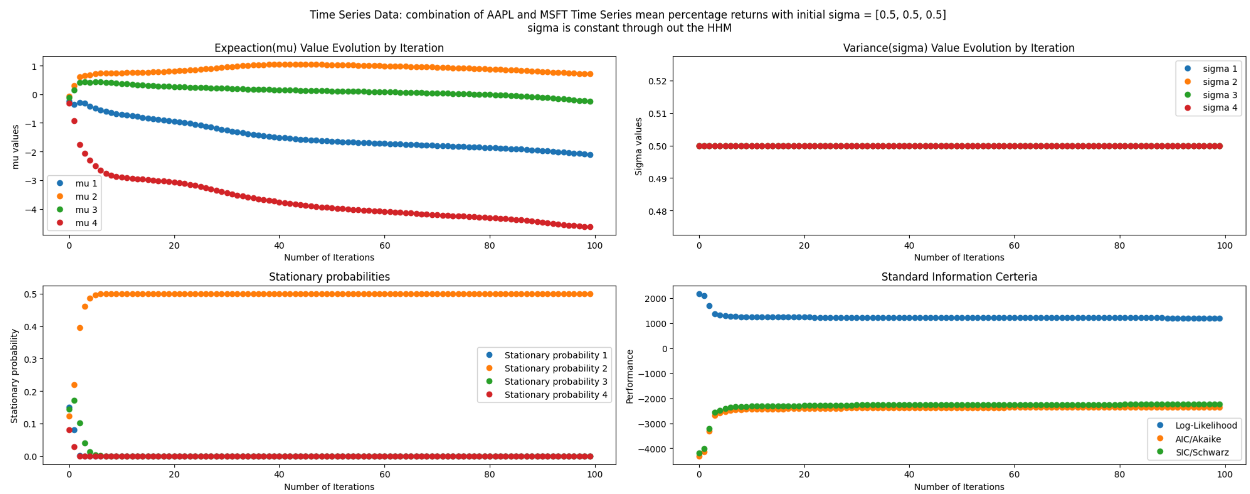
$$Transition\ Matrix:\ P\ = [[0.33, 0.11, 0.34, 0.22], [0.4, 0.2, 0.3, 0.1], [0.4, 0.1, 0.23, 0.27], [0.09, 0.61, 0.2, 0.1]]$$

$$Initial\ Probabilities:\ \pi\ = [0.4,\ 0.1,\ 0.4, 0.1]$$

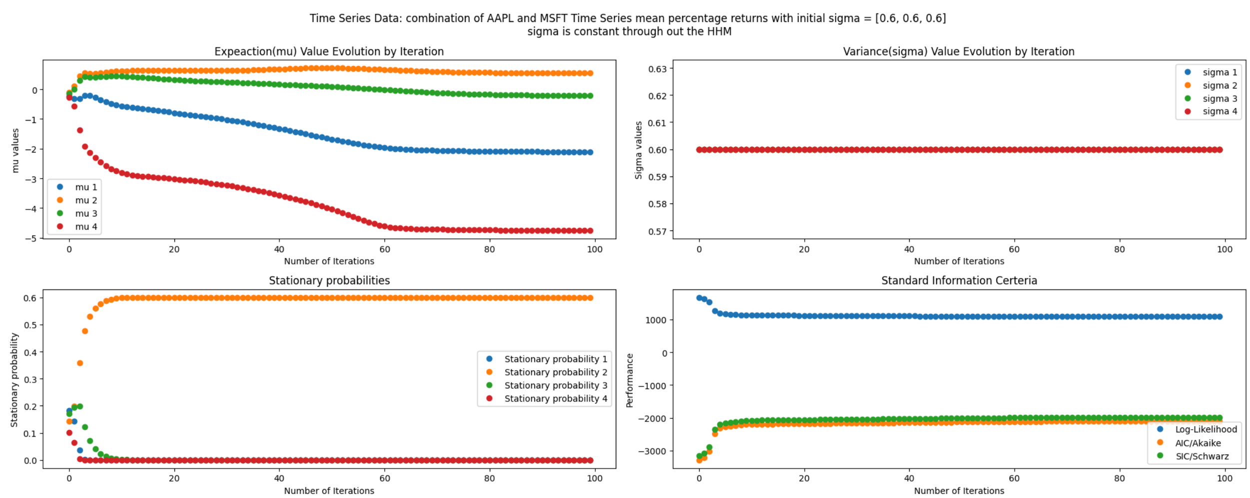
$$Exceptation:\ \mu\ =\ [0.012,\ 0.044,\ 0.033, -\ 0.001]$$

$$Variance:\ \sigma\ \in \{0.5, 0.6, 1.5, 2.3\}$$

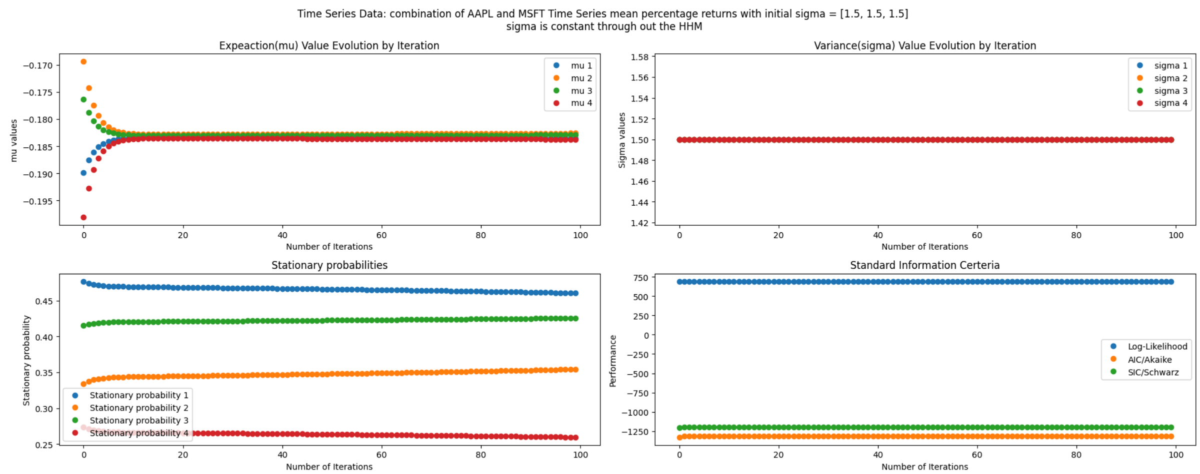
Below are the results we get after running the python script.



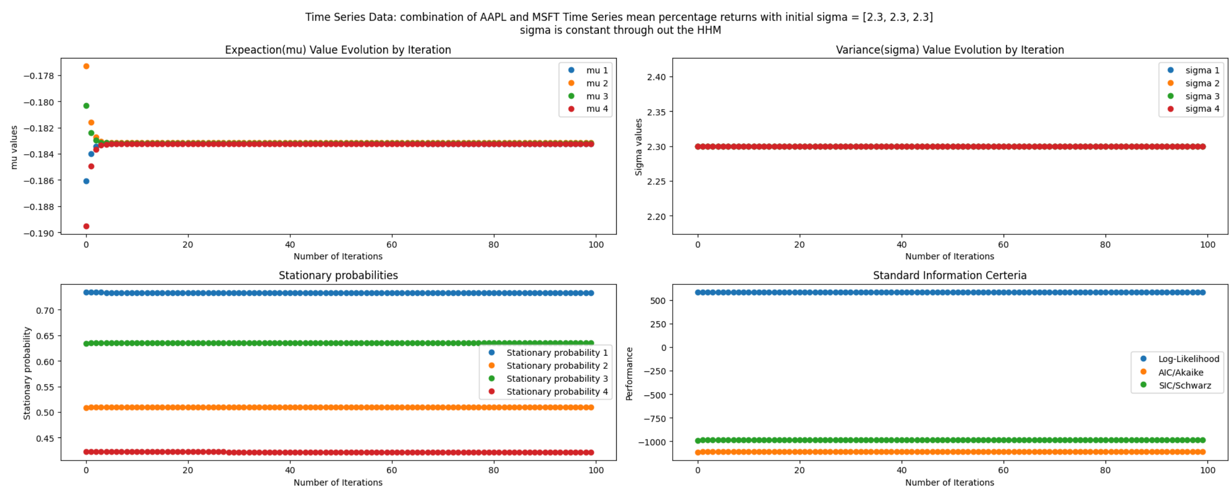
Graph 12



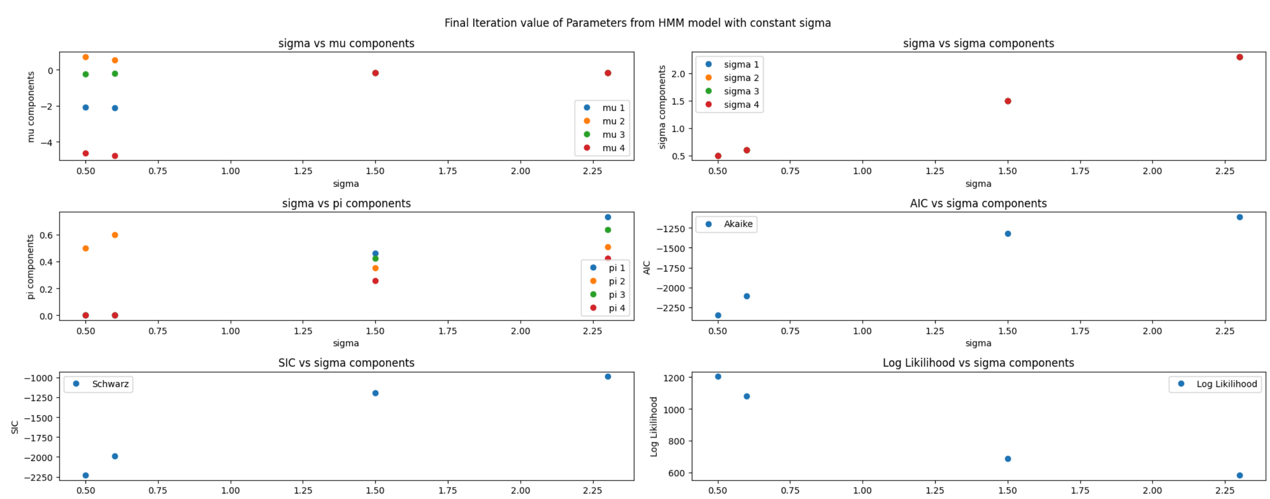
Graph 13



Graph 14



Graph 15



Graph 16

- **Results**

Graphs 12, 13, 14 and 15 are Expectation, variance (constant) , Pi and Standard information criteria for same numbers of states (N =4) with constant Variance throughout the one model run with similar initial parameters with assumptions as mentioned above.

Graphs in this section provide us some insights about underlying hidden chain, which are as follows-

2. Graphs show the Expectation, variance, Pi and Standard information criteria converges to values as model iterates.
4. We don't need high number of iteration to come up values for these parameters, we can approximated them with low iterations
5. Graph 16 shows the final values (conversing values) from all the models with constant Variance.

Allowing Sigma’s and Mu’s to vary

And, Finally we are going to vary both expectation and variance throughout the HHM process (at each state or iteration)

Initial Conditions which will be constant throughout the process are-

For example, (3 more initial conditions in python script)

$Number\ of\ stages:\ N\ = 3$

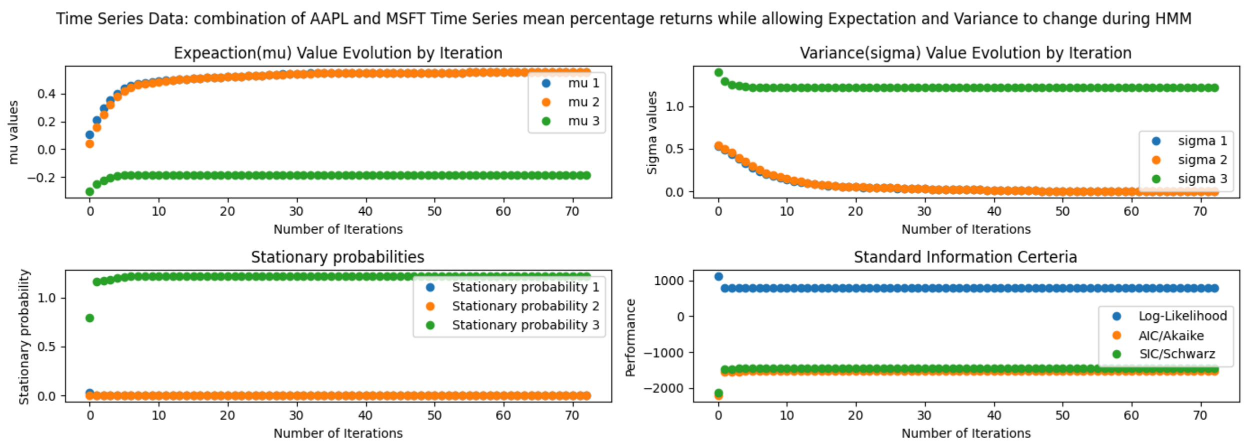
$Transition\ Matrix:\ \ P\ = [[0.55, 0.11, 0.34], [0.4, 0.2, 0.4], [0.4, 0.1, 0.5]]$

$Initial\ Probabilities:\ \pi\ = [0.5,\ 0.1,\ 0.4]$

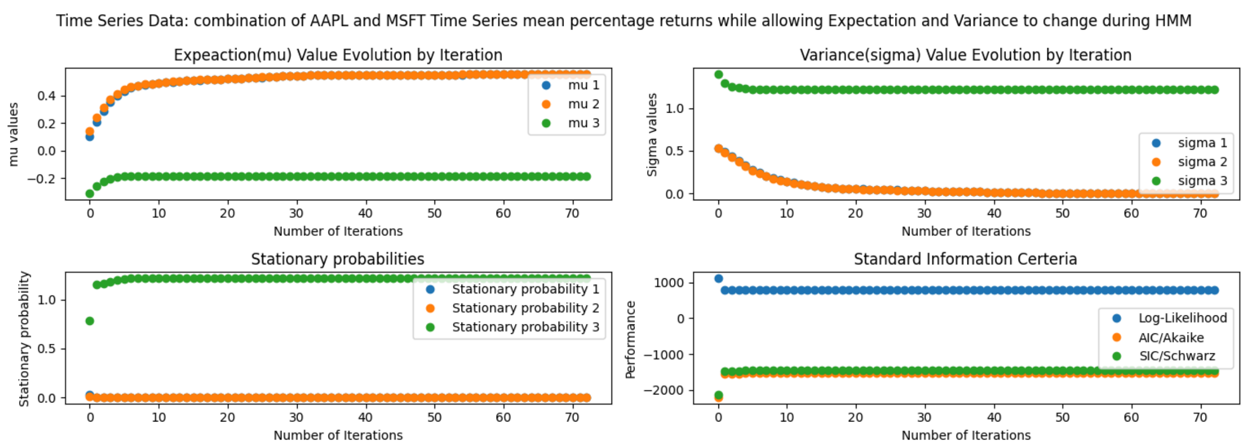
$Exceptation:\ \mu\ = [0.012,\ \ -0.044,\ 0.033]$

$Variance:\ \sigma\ = [0.5,\ 0.5,\ 1]$

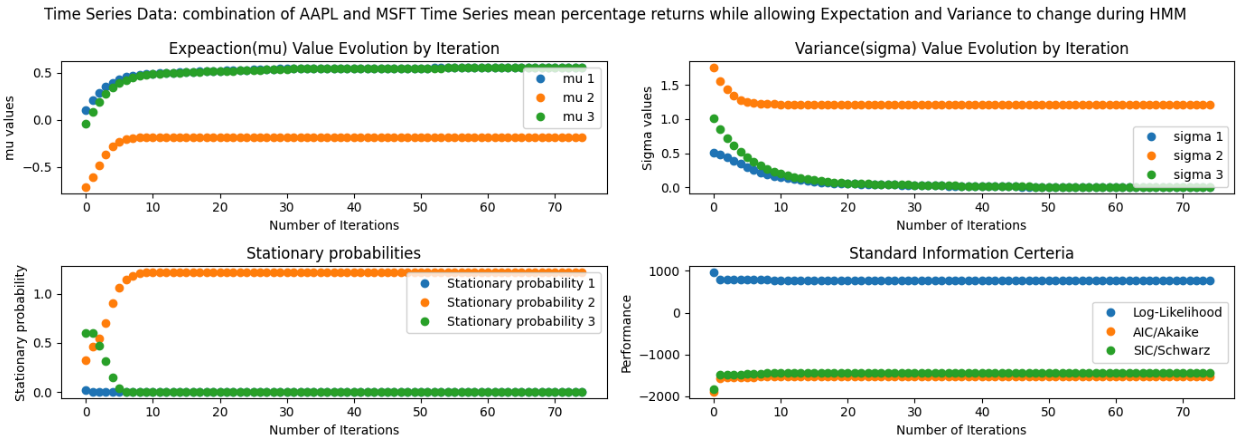
Below are the results we get after running the python script.



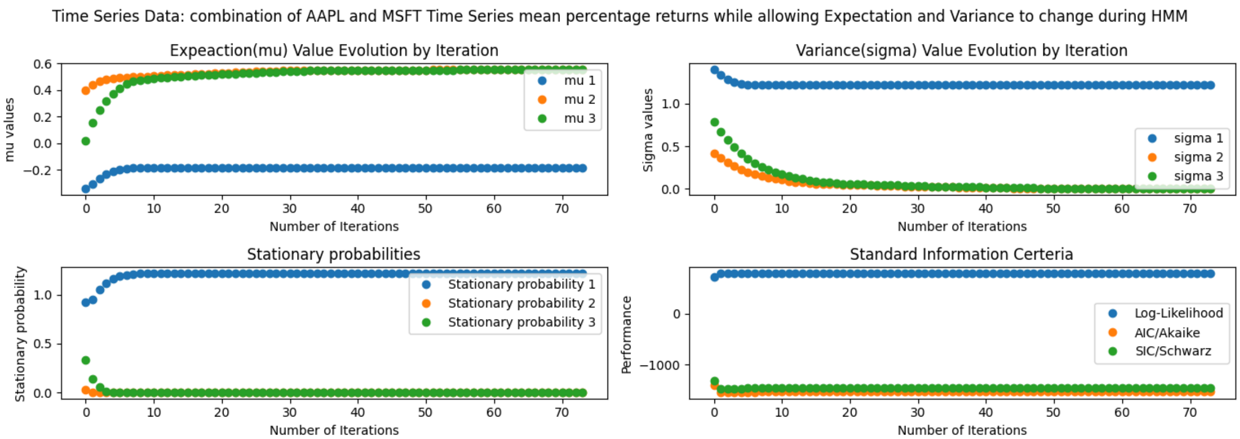
Graph 17



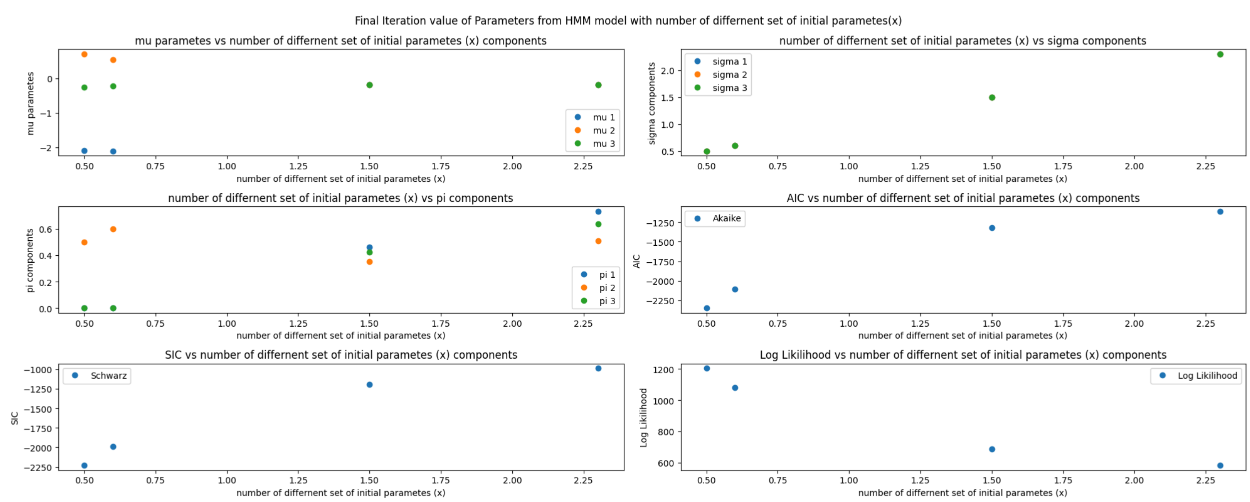
Graph 18



Graph 19



Graph 20



Graph 21

- **Results**

Graphs 17, 18,1 9 and 20 are Expectation (varying), variance(varying), Pi and Standard information criteria for same numbers of states (N = 3) with varying Expectation and Variance throughout the one model run with similar initial parameters with assumptions as mentioned above.

Graphs in this section provide us some insights about underlying hidden chain, which are as follows-

3. Graphs show the Expectation, variance, Pi and Standard information criteria converges to values as model iterates.
6. We don't need high number of iteration to come up values for these parameters, we can approximated them with low iterations
7. Graph 21 shows the final values (conversing values) from all the models with varying Expectation and Variance.

Best Model from above results

- AIC, SIC and Log-likelihood values are plotted in the part(d) of above graphs for each model
- For our chosen data we can use any of the above models but the best one is one which has the closest AIC, SIC or highest Log-likelihood, uses less number of iterations, less overall computational time, and is less complex.
- And the best model for our time series is- Model with varying Expectation and Variance, and have a number of states equal to 4 with initial parameters as follows (model result drawn in Graph 20).

  $Number\ of\ stages:\ N\ =\ 3$

  $Transition\ Matrix:\ \ P\ =\ [[0.55, 0.11, 0.34], [0.4, 0.2, 0.4], [0.4, 0.1, 0.5]]$

  $Initial\ Probabilities:\ \pi\ =\ [0.5,\ 0.1,\ 0.4]$

  $Exceptation:\ \mu\ =\ [-\ 0.012,\ 0.4,\ \ -\ 0.033]$

  $Variance:\ \sigma\ =\ [2,\ 0.5,\ 1]$

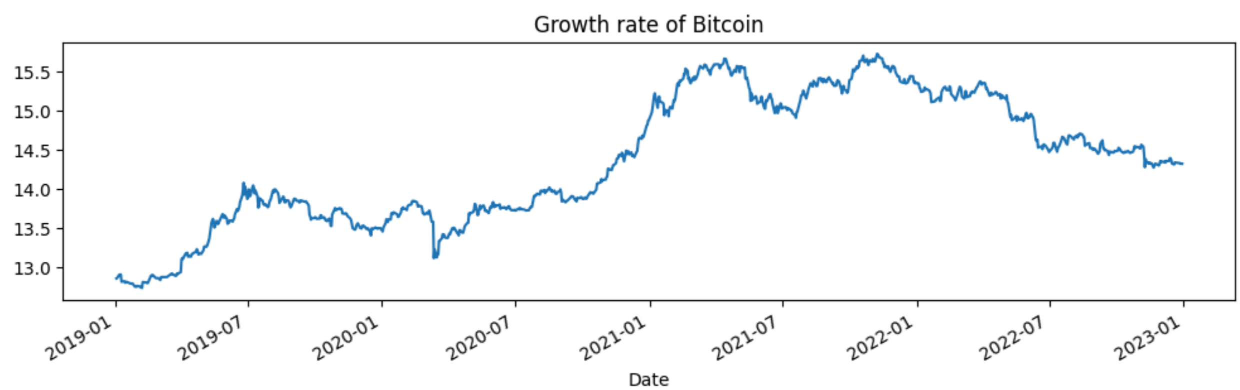
AutoRegressive Process Time Series

We are going to assume that our observable data follows a AutoRegressive model with the one term dependent upon a hidden time series which is Markovian in nature.
For simplicity we are going to assume the underlying series can have two possibilities with some probability which is given by initial parameters.

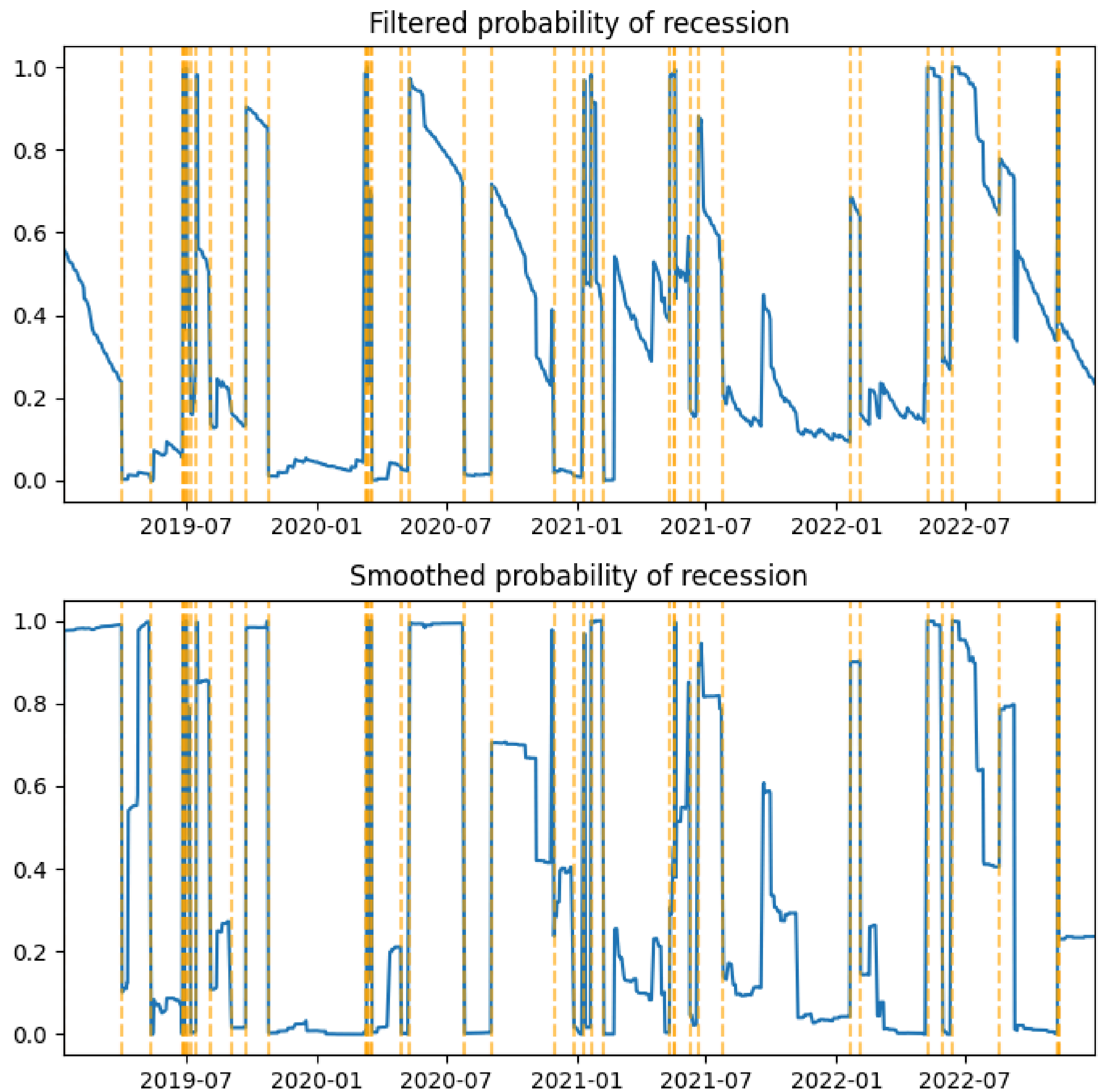
Now, Here we are going to change our primary time series from a combination of Apple and Microsoft stocks to Bitcoin cryptocurrency (Ticker : BTC-USD).
Reason behind changing the time series, we need to look for assets which have a high number of regime changes in our instreated period.

We are going to assume our mode follows AR(3), which implies our current state of time series depends upon 3 lays behind.

Result from running python script:



Graph 22



Graph 23

- **Results**

Graphs 22 and 23 are showing results from the hamilton autoregressive markov regime module. And the orange vertical line shows the actual regime change in real time with the threshold of 0.7% change in value in a date (which is huge for Bitcoin).

As we can see from the graph, orange vertical lines pretty much overlap the regime change from the real world.

We used AR(3) for our calculation and number of states equal to (N=2).

Reference

1. Hamilton, James D. (1994) “Time Series” Analysis, chapter 22, Princeton University Press, Princeton, New Jersey.
2. Kole, Eric (May 2020), “Markov Switching Models: An Example for a Stock Market Index”