**DOCUMENTATION**

**Variables :-**

Theta\_X = Probability of getting heads with coin X

L\_X = Likelihood of coin X

P\_X = Probability of coin X

LH\_X, LT\_X = Likelihood of getting heads and tails respectively with coin X

Heads\_count , tails\_count = Number of heads and tails in a draw

**Formulae :-**

L\_X =

P\_X =

LH\_X = P\_X \* heads\_count

LT\_X = P\_X \* tails\_count

Theta\_X =

**Assumptions :-**

Before beginning EM algorithm, we have to assume a suitable values of theta\_A and theta\_B which represent the individual probabilities of getting heads with coin A and coin B respectively. These values are independent of each other i.e there is no relationship between theta\_A and theta\_B

**EM Algorithm :-**

When your data is incomplete, has missing data points, or has unobserved (hidden) latent variables, the Expectation-Maximization (EM) procedure is a means to find maximum-likelihood estimates for model parameters. It is a method for approximating the maximum likelihood function that is done iteratively. Although maximum likelihood estimation can discover the "best fit" model for a collection of data, it struggles with missing data sets. Even if you have missing data, the more sophisticated EM technique can find model parameters. It works by guessing random values for missing data points and then estimating a second set of data using those guesses. The new values are utilized to improve the original set's guess, and the process is repeated until the algorithm converges on a point. Steps of Em algorithm are mention below :-

1. A probability distribution is constructed based on an initial guess for the model's parameters. The "Expected" distribution is sometimes referred to as the "E-Step."
2. The model is updated with new observations.
3. To account for the additional data, the probability distribution from the E-step is modified. This is also known as the "M-step."
4. Steps 2–4 are continued until the distribution is stable (i.e., it does not change from the E-step to the M-step) or maximum number of iterations are completed.

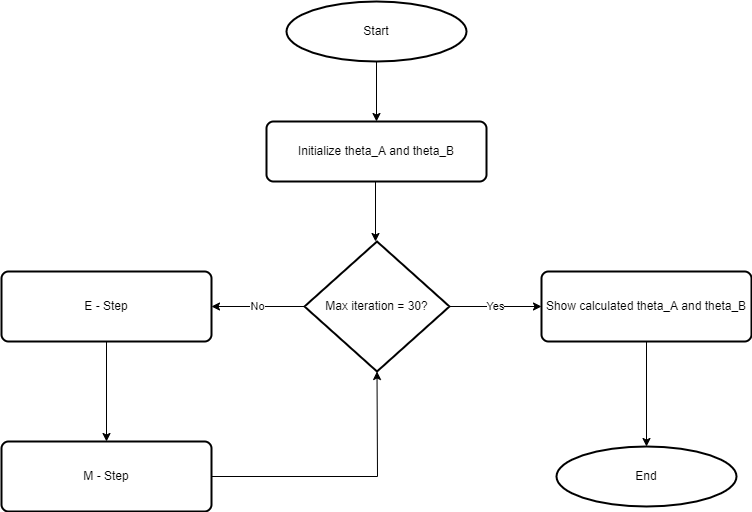
Choices -

While implementing the python code for EM algorithm, there were two options to choose from. Implement steps manually or use in built libraries.

In my opinion the first option was more suitable in the given example as the data fetched from the api can be used directly without any modifications required. Whereas while using the in built libraries like scikit learn or scipy, the GMM methods required some modifications on the api data. This made using those libraries complex as compared to manually implementing the algorithm.

Hence, the algorithm was implemented manually step by step with enhanced visibility of each iteration.

**Flowchart :-**



Output :-

Initial -

Theta\_A = 0.6 Theta\_B = 0.5

Final -

Theta\_A = 0.75 Theta\_B = 0.29