**Task 4**

**Problem Description:**

Estimate the parameters for a simple model of two-coin tosses using EM algorithm.

**Methodology:**

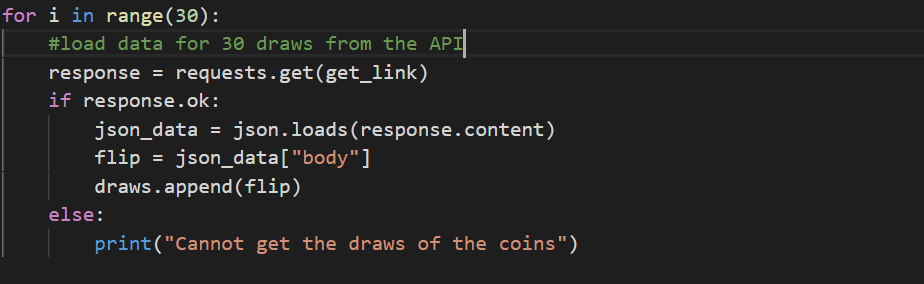
The EM algorithm is not an algorithm rather it is a framework. The basic idea of EM in this context is to *pretend* that we know the parameters of the model and then to infer the probability that each data point belongs to each component. In our problem, we assign random probabilities initially to a draw being of the either two coins and then based on the data from the draws estimate the probability that each flip belongs to which coin. After that, we refit the components to the data, where each component is fitted to the entire data set with draw weighted by the probability that it belongs to that coin. The process iterates until convergence.

In this problem, we use the EM algorithm to find the biases of the coins. The biases of the coins are indicated by theta\_1 and theta\_2 for both the coins respectively, where

To implement the EM algorithm in Python I did the following steps:

1. **Extract Data from API:**

Using the requests module in python, I extracted the data from the given API: [https://24zl01u3ff.execute-api.us-west-1.amazonaws.com/beta](https://urldefense.com/v3/__https:/24zl01u3ff.execute-api.us-west-1.amazonaws.com/beta__;!!LIr3w8kk_Xxm!5gxOGg5QvO_VbZ3Z27eyuV787mQrvcJXp9I30-5vg_YiwcYLoIGRKuGEa-0C-xc$). Each call to this API returns 20 flips of heads/tails indicated by “1”/”0” by choosing any one of the two coins. I make 30 calls to this API for 30 such draws. If the response code is 200 (OK) then I parse the json data into a python object using the json module of python.



1. Initialize thetas for both the coins:

Since the initial biases of the coins is not known, I use the random module of python and assign the initial biases as follows:



1. **Expectation Step:**

In the expectation step, we calculate the probability that a given draw belongs to coin 1 or coin 2. This is done by using Bayes’ conditional theorem.

We have to find and where this notation stands for Probability of the draw being from that particular coin.

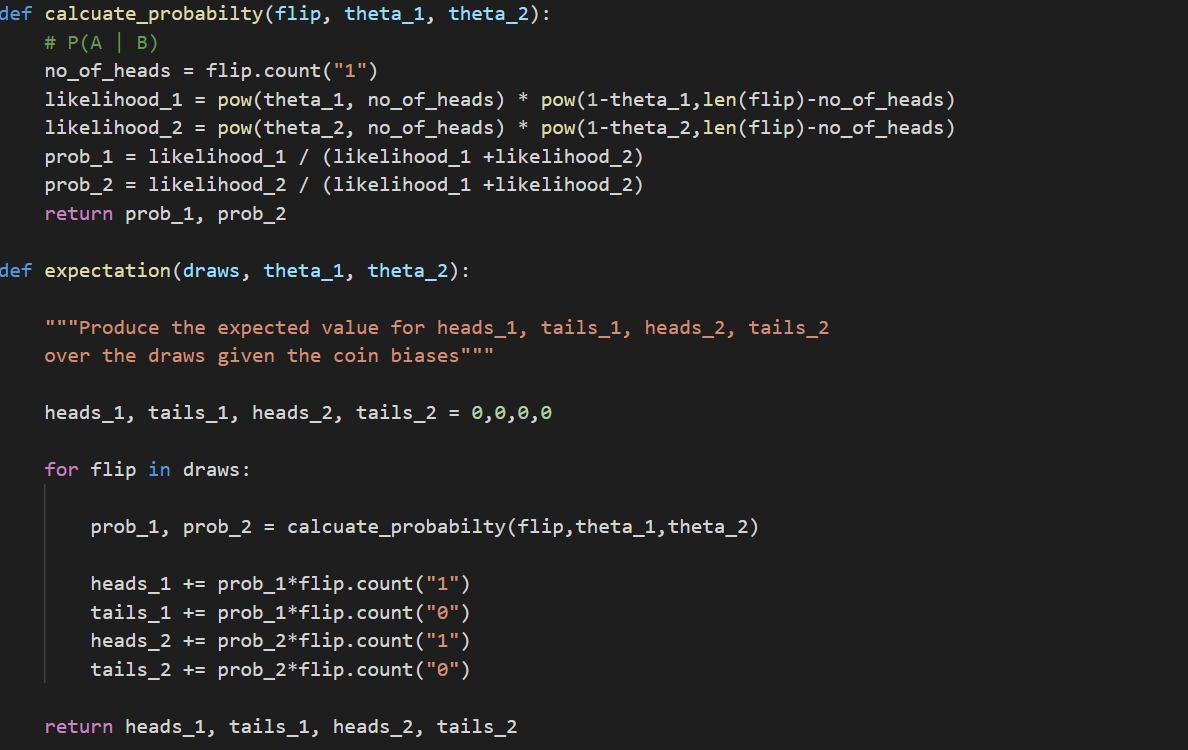
Given Bayes theorem and the law of total probability, we can partition all the flips in Coin or Coin 2 as we have to choose one or the other.

Now we know that is **theta\_1** or the initial bias. We can easily calculate the

and hence we can calculate .

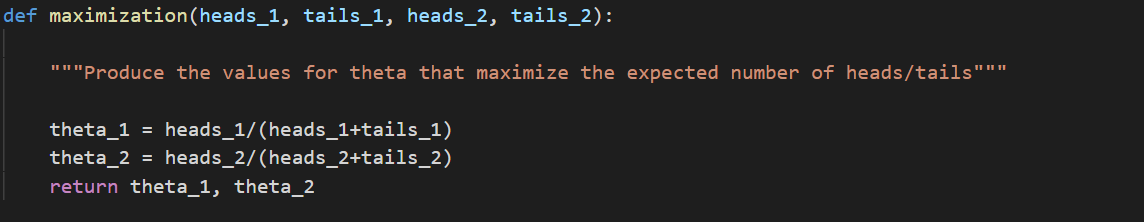
Now that I have gotten the probabilities of the flip being from the respective coins, I use the probabilities to calculate the number of heads and tails from that coin. These values of heads and tails are added up for each of the 30 draws to get an overall count of the heads and tails from each of the two coins.

As we can see in this snippet of code, the expectation function calculates the probability of the flip belonging to each of the two coins for all the flips. And for each flip, using the probability, it calculates the number of heads and tails for each coin and adds up the values for all 30 flips.



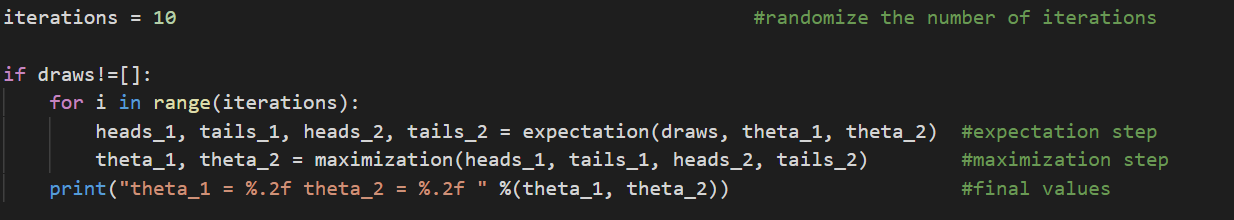
1. **Maximization Step:**

Now that we have obtained the number of heads and tails for both the coins after all the 30 flips, we re-calculate the biases based on the formula as shown in the snippet below. This value now gives a better estimate of biases of the coins, or in other words, maximizes the biases of each of coins.



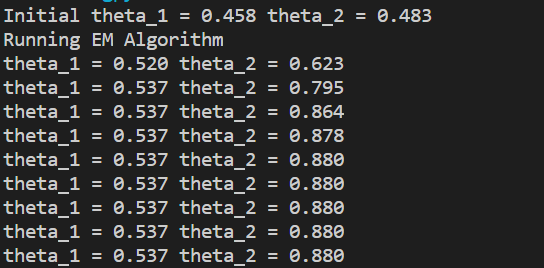
1. **Iteration:**

We do the Expectation and Maximization steps for several iterations to converge the biases.



**Output:**

As we can see here, the biases converge while running several iterations. In this example, I have run 10 iterations of the EM algorithm



**Solution File and Execution Steps:**

File: EM\_Algorithm.py

Execution Steps:

1. Open your terminal
2. Go the folder which contains EM\_Algorithm.py
3. Run the command: python EM\_Algorithm.py

**Dynamic Variation:**

In a user-friendly variation, the user can choose to provide the initial biases for the two coins and the number of iterations for which he/she wants to run the EM algorithm.

**Dynamic File and Execution Steps:**

File: EM\_Algorithm\_dynamic.py

Execution Steps:

1. Open your terminal
2. Go the folder which contains EM\_Algorithm\_dynamic.py
3. Run the command: python EM\_Algorithm\_dynamic.py <theta\_1 > <theta\_2> <iterations>

where the <theta\_1> , < theta\_2> and <iterations> are specified by the user.