Worksheet 5 – Bayes Rule

Bayes Rule

Recursive Bayesian estimation is used to estimate a unknown probability density function by using incoming data as evidence to update a systems prior beliefs. By performing this process interactively one can become less or more confident of a given event occurring given some condition

I apply the recursive Bayes method to a coin flipping scenario that will get more complex with each task.

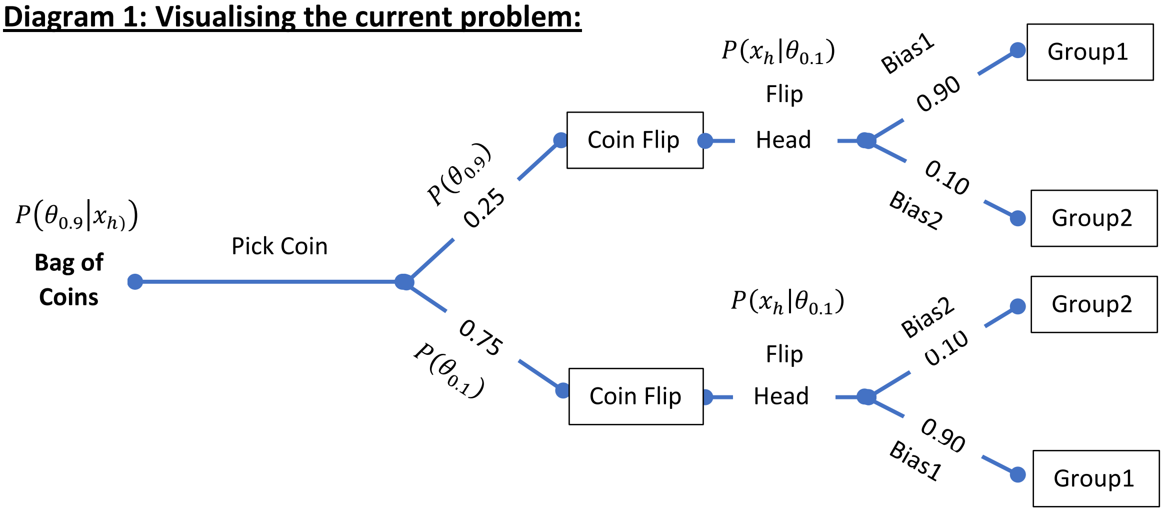
Task 1

Parameters: A Bag of 100 coins, 25 biased to flip heads 90% of the time (group 1), 75 biased to flip heads 10% of the time (group 2).

Aim: Task 1 requires the utilisation of MATLAB to simulate the process of selecting a coin at random and flipping it several times to determine the probability of the coin, being flipped, belonging to group 1.

Solution: Implement the Bayes rule equation iteratively using a for loop.

Diagram 1: Visualising the current problem:



The above diagram shows the chain of events that can occur during the coin flip simulation. $P(\theta_{0.9}|x_h)$ Represents the question at hand or hypothesis, it reads, what is the probability that the coin being flipped belongs to group 1($\theta_{0.9}$) given some data or evidence x_h which in this case would be the result of coin flip (coin lands on heads). $P(\theta_{0.9-0.1})$ is defined as the prior which represents the probability of choosing a coin from group 1 or 2 irrespective of one another. In this simulation the coin is set to always land on heads, yet the manner in which the coin flips will differ relative the possible biased weights of the coin (90%, 10%). The parameters that populate This conditional tree can now be substituted into the Bayes equation.

Please continue to the next page

Bayes Equation

Probability of landing a heads given the hypothesis.

Probability of the coin belonging to group1(hypothesis) given the coin lands on heads.

The prior probability of the hypothesis occurring

$$P(\theta_{0.9}|x_h) = \frac{P(x_h|\theta_{0.9})P(\theta_{0.9})}{(P(x_h|\theta_{0.9})P(\theta_{0.9})) + P(x_h|\theta_{0.1})P(\theta_{0.1})}$$

The prior probability of the evidence/data in general.

The output or posterior of the Bayes equation represents the probability of the coin belonging to group1 given it lands on heads after one coin flip. We can become certain of this probability the more we flip the current coin. This requires recursive Bayes, with each coin flip we will assign a new set of prior probabilities to be equal to the posterior of the last coin flip. This will allow the current belief of coins identify group to be updated iteratively.

Programmatic Implementation

```
%% task 1.a

group1 = 25;
bias1 = 0.9;
prior1 = 0.25;

group2 = 75;
bias2 = 0.1;
prior2 = 0.75;

bayes = [];

bayes(1) = prior1
```

This section of code defines the parameters to be used in the Bayes equation. Define an empty array called Bayes that will store the posterior values outputted by Bayes equation for each coin flip.

```
for i = 2: 6

    num = (bias1 * prior1);
    dinom = (bias1 * prior1) + (bias2 * prior2);

    bayes(i) = num / dinom;

    prior1 = bayes(i);
    prior2 = 1 - bayes(i);

end

disp(bayes(i))
plot(bayes)
```

A for loop is used to simulate 4 coin flips.

The variable "num" defines the numerator parameters and operations of the Bayes equation, whilst the variable "dinom" defines the denominator parameters and operations of the Bayes equation.

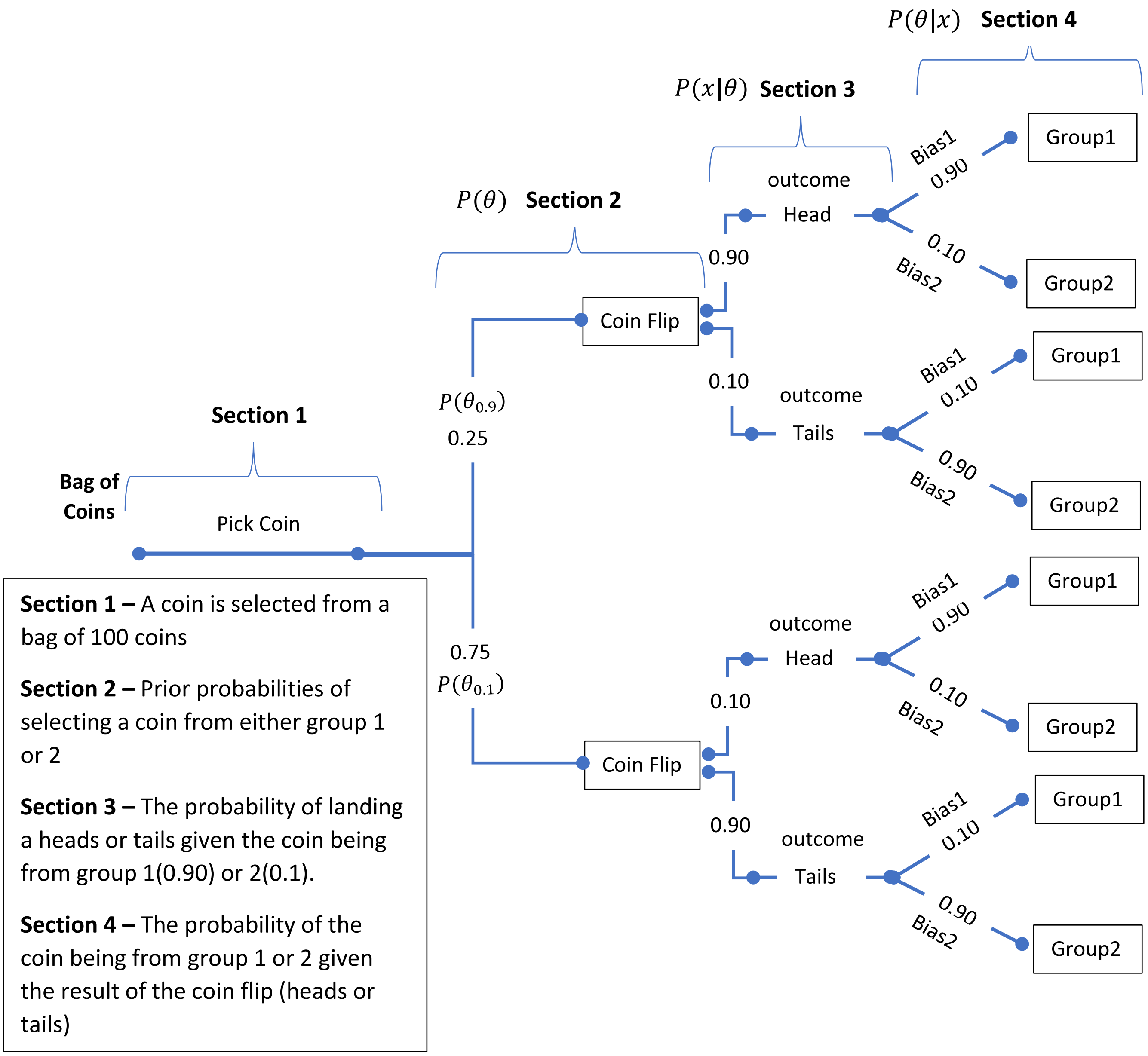
The Bayes(i) variable executes the current posterior probability value for hypothesis $P(\theta_{0.9}|x_h)$. The current output value relative to the iteration number is then stored in the Bayes array.

Finally, the prior probabilities are then updated based on the posterior output of the bayes equation.

<u>Task 2:</u>

Task 2 requires the above program to take into consideration the possibility of the evidence taking on two possible states, x_h(heads) and x_t(tails). Therefore, our conditional tree must be updated, as follows:

<u>Diagram 2: Visualising the current problem</u>



The Above Diagram demonstrates causal chain of probabilities relative to the possible outcomes that can occur when simulating a coin flip, with 2 different biased coins, both heads and tails outcomes being possible. The next section will detail the programmatic implementation of the above intuition followed by a comparison and discussion of the results produced by programs 1(diagram 1) and 2(diagram 2).

Please continue to the next page

Program 2: Implementation of Diagram 2

```
  %% task 1.b

  group1 = 25;
bias1 = 0.9;
prior1 = 0.25;

group2 = 75;
bias2 = 0.1;
prior2 = 0.75;
```

Sets the priors and bias values for each group of coins

```
flip_store(1, :) = [prior1; prior2];
post1 = prior1;
post2 = prior2;
total_count = [];
```

Flip_store variable Stores the initial priors and posterior values for each iteration in a matrix to allow for graphical analysis. "Total count" function tracks the outcomes of each coin flip (1 = heads, 0 = tails)

```
for i = 2:5

prior1 = post1;
prior2 = post2;

if (rand < 0.9)
     % Heads
       post1 = (bias1 * prior1) / (bias1 * prior1 + bias2 * prior2);
       post2 = (bias2 * prior2) / (bias1 * prior1 + bias2 * prior2);
       total_count(i) = 1;
else
    % tails

     post1 = ((1 - bias1) * prior1) / ((1 - bias1) * prior1 + (1 - bias2) * prior2);
     post2 = ((1 - bias2) * prior2) / ((1 - bias1) * prior1 + ( 1- bias2) * prior2);
     total_count(i) = 0;
end

flip_store(i, :) = [post1; post2];


end
disp(total_count)
plot(flip_store)
```

Rand returns a uniformly distributed random number between 0-1, conditional logic is used to simulate flipping a coin. Therefore if the value of rand is smaller than 0.9 then the outcome of the coin flip is heads, if the value is greater than 0.9 then the inverse outcome probability is assumed and the coin lands on tails.

Executes Bayes rule for each group. Calculates the probability of each coin to group hypothesis being true given the coin landing on **heads**. The bias values causally connected to the outcome of landing heads for each group are also taken into consideration for the calculation of each posterior.

Executes Bayes rule for each group. Calculates the probability of each coin to group hypothesis being true given the coin landing on **tails**. The bias values casually connected to the outcome of landing tails for each group are considered by inversing the heads bias.

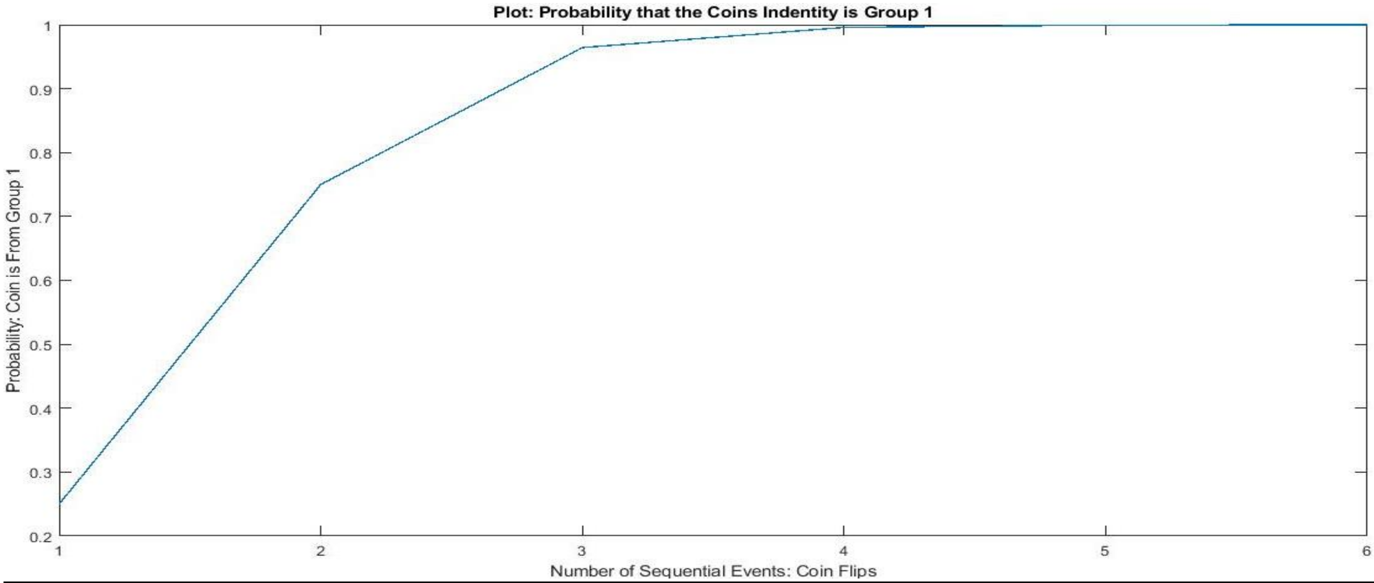
Sets the posterior values for each hypothesis to be the prior for the next recursive Bayes iteration

Please continue to the next page

Task 1 Analysis of Results

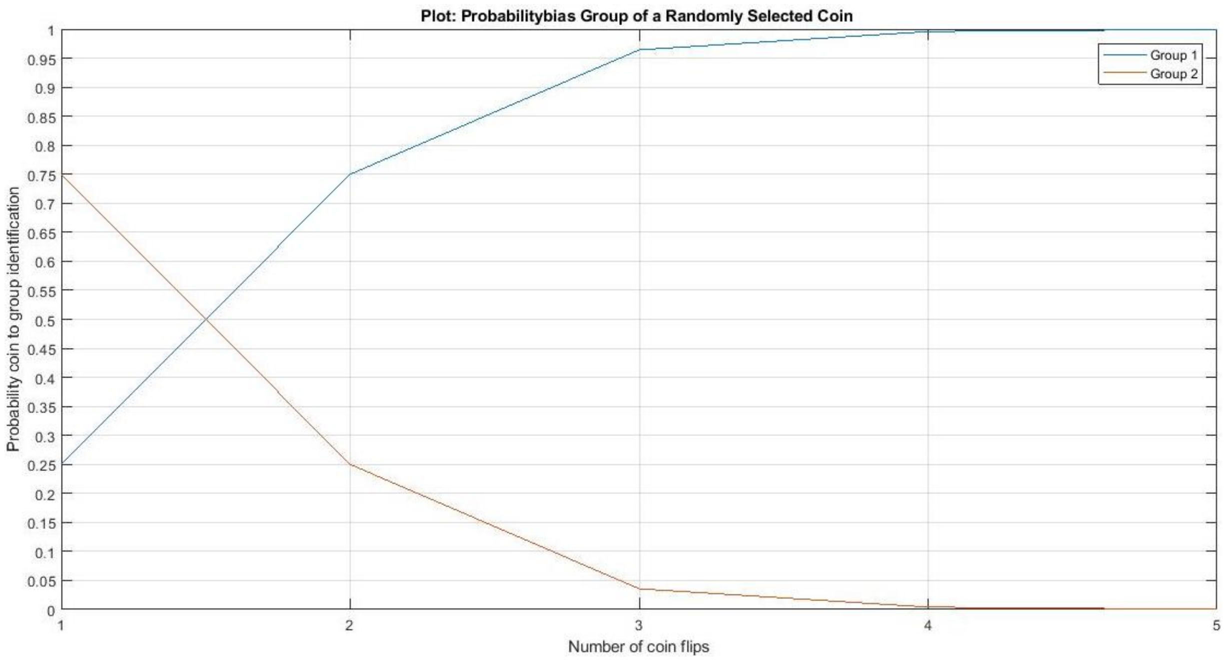
Results for task 1.a show that the Bayes Model becomes increasingly confident that the coin belongs to group one the more the coin is flipped. It takes a total of 4-coin flips to stably converge to a 100% certainty. This makes logical sense when following our Bayesian Tree (diagram 1), As the coin can only land on a head outcome we expect the probability of the coin belonging to group 1 to increase, as group 1 is biased to produce a heads outcome 90% of the time.

Results Task 1.a



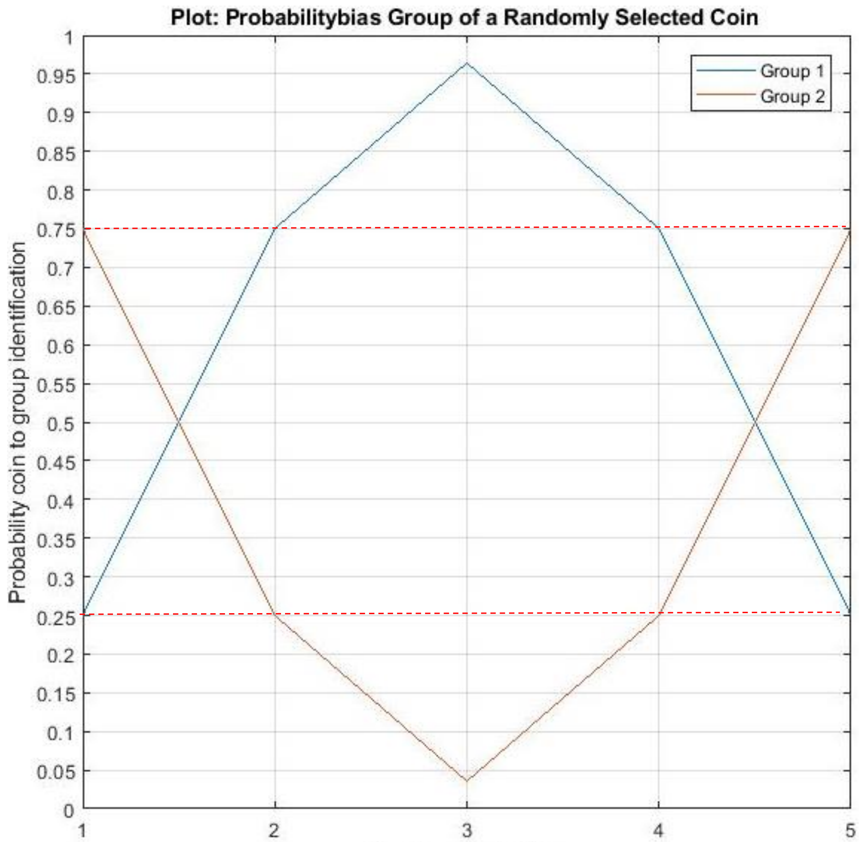
Results Task 1.b

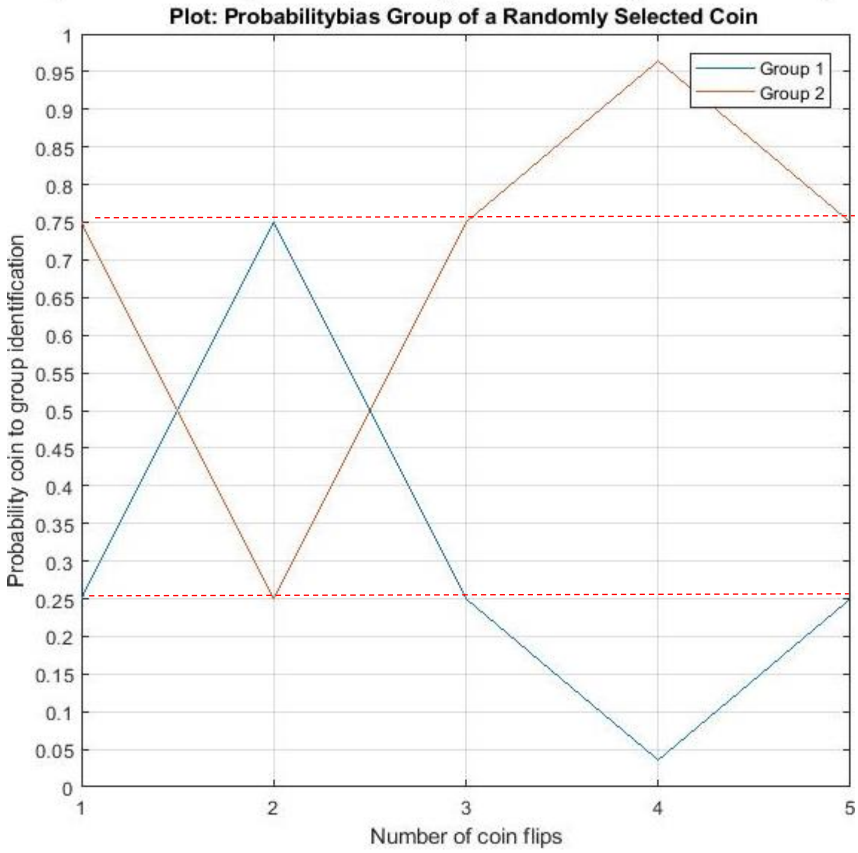
Results from task 1.b Shows the change in posterior probability from the prospective of both hypotheses ($P(\theta_{0.9}|x_{ht})$, $P(\theta_{0.9}|x_{ht})$). Group 1s probability curve is represented by the blue line and it follows the same trajectory as task 1.a. The final outcome set for this particular trial is [1111] or [heads, heads, heads, heads]. Group 1 & 2s probability lines behave symmetrically, overall as the probability of the coin belonging to group 1 increases as the probability of the coin belonging to group2 decreases by the same amount.



Analysis of results Task 1.b: *For the Programmatic Implementation of this task please see Appendix A-B*

Interestingly when running several coin simulations the results occasionally yield unexpected probability shapes that deviate from the norm, over the course of 4 iterations. 2 examples are shown as follows:





After analysing the change in probability across the number of iterations, for each probability shape, one can identify a common pattern where the final posterior value of the model is equal to the prior value of the first iteration. Therefore despite the system gaining new evidence to update its current belief on the identity of the coin, one finds that in the end the model does not provide any more probabilistic insight than before recursive Bayes was implemented. In the case of the coin flip sequence the possible group associated bias of the coin cancels out, which prohibits either hypothesis of coin to group identity from converging to a stable probability across the number of iterations. For the purposes of this report I will refer to this effect as Bayesian Cancellation with both groups probability lines, as a whole, being termed a Cancelation oscillation.

Investigation and Explanation of the Data

In the given situation Bayes rule is being implemented recursively upon a discrete random variable that has two possible states (heads = 1 or tails = 0). Further investigation reveals that the occurrence of Bayesian cancellation is mathematically linked to the number of sequential events that occur. Therefore, the number of sequential events is equal to the number of recursive iterations of the Bayes equation. For example with the given coin flipping scenario each flip of the coin represents a single event occurring with two possible outcomes, or two possible states, the event collapses towards. Interestingly for the given "coin flipping scenario" the effects of Bayesian cancelation only occur when the number of sequential events is even. This is due to the coin being restricted to an integer state value, as it is not possible to obtain the outcome of half a coin flip. However, an odd number of sequential events may cause significant Bayesian Cancelation when applied to a different scenario, that isn't restricted to an integer value sequence. Consider the following equation:

Equation 1:

$$C_N = S^N$$

Where N is the number of sequential events that occur and S being the number of possible states each N can form . Thus the total number of possible combinations of an Nth set (C_N) can be calculated. The value of C_N when N = 2 can be illustrated below:

$$\xrightarrow{\text{Time}}$$

$$\begin{array}{c} 1.) \\ 2.) \\ 3.) \\ 4.) \end{array} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Notice that two of a total of four combinations are equal state combinations, which will be denoted as E_C . An equal state combination can be defined as a sequence of events in which an equal number of the events states occur across time, in this case this is the 3rd and 4th combination, thus $E_C = 2$. It is these equal state combinations that cause Bayesian Cancellation to occur within a recursive Bayesian loop, and will ultimately generate a final Nth posterior value equal to the N_0 prior probability value at beginning of the Bayesian loop. Via a programmatic simulation table 1.1 shows the Total number of possible combinations for the first ten even values of N, with two states, along with the total number of equal state combinations that can occur in a given C_N set. Finally, the Table shows the probability of generating an equal state combination relative to the Number of sequential events. (Appendix A – programmatic Implementation)

Table 1.1:

States = 2				
Number of Sequential Events (N)	**Number of Possible Event Combinations (C_N)**	**Number of Possible Equal State Combinations (E_C)**	**Probability of an Even Combination Occurring ($P(E_C)$)**	**Probability as a percentage (%)**
2	4	2	0.50	50
4	16	6	0.375	37.5
6	64	20	0.3125	31.25
8	256	70	0.2734	27.34
10	1024	252	0.2460	24.60

Mathematical Expression

I derived A 4th degree polynomial function to map the association between the number of sequential events and the number of equal state combinations that can occur within the given event chain, shown below:

Equation 2: Equal state Combination Function

$$E_c = 18(\frac{N-6}{3.1623})^4 + 40(\frac{N-6}{3.1623})^3 + 38(\frac{N-6}{3.1623})^2 + 35\left(\frac{N-6}{3.1623}\right) + 20$$

The fractional terms represent a z score transformation on the data, This allows the polynomial model to be centred and scaled to curvature changes of the data points, which in turn allows for a more precise representation of the original data. Finally the value is rounded to the nearest multiple of 1, to account for any error that may occur. Thus the equation can be simplified as follows:

$$z = (\frac{N-6}{3.1623})$$

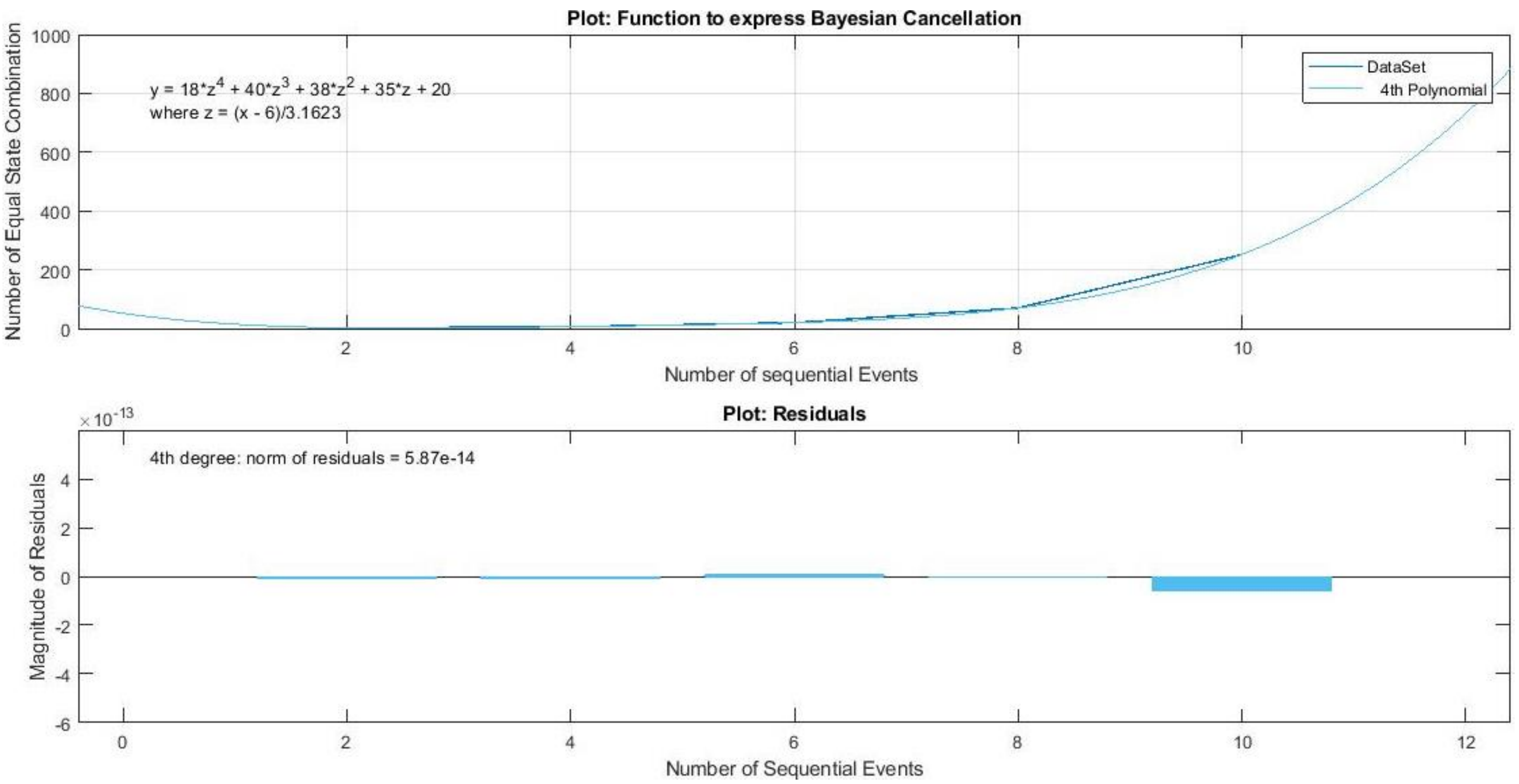
$$E_c = 1\lfloor 18z^4 + 40z^3 + 38z^2 + 35z + 20 \rceil$$

The probability of an equal state combination occurring relative the number of sequential events can be expressed by combining equations 1 and 2 respectively, as shown below:

Equation 3: Probability of Bayesian Model

$$P(E_c|N) = \frac{1\lfloor 18z^4 + 40z^3 + 38z^2 + 35z + 20 \rceil}{S^N}$$

The below plot indicates that the above model fits the data very well as results reveal the norm of residuals (difference between the sample and the estimated function value) to be extremely low at 5.87e-14. Therefore, one can be confident that the above equation describes equal state combinations relative to their respective N, along with their probability of occurrence for a given C_N.



Considering Outcome Biases

Thus far Equation 3 does not consider the possibility of the flipped coin being biased. Consider again, the value of C_N when N = 2:

$$\begin{array}{l} 1.) \\ 2.) \\ 3.) \\ 4.) \end{array} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Applying the bias values of the groups from task 1, the probability for combination 1 occurring is significantly higher than all other combinations when there is a 90% bias towards the coin landing heads (group 1s bias). Likewise, the probability of combination 2 occurring will be higher than all other combinations when there is a 10% bias towards landing heads (group 2). Combinations 3 & 4 are equal state combinations and thus their probabilities of occurrence are equivalent, as a change in biasing from group 1 to 2 will only rearrange the numbers being multiplied when calculating the joint probability of independent outcomes. Therefore, the total biased probability ($P(B_{Ec})$)of any equal state combination occurring from all possible event combinations (C_N) relative to the number of sequential events that will take place (N), can be found by calculating the biased probability for 1 equal state combination and scaling the value by the total number of equal state combinations that can occur. This can be expressed mathematically as follows:

We can calculate a single joint probability for any equal state combination via the following equation:

Equation 3: joint probability

$$P(B_{Ec}) = (B_1)^L (B_2)^L$$

B represents the head outcome Bias for each group in task 1 (0.9 group 1, 0.1 group 2), you can see that both biases are the inverse of one another and together sum to 1, thus the bias of group 2 represents the probability of landing tails in the group 1 condition and vice versa. L represents the required number of event outcomes needed to form an equal state combination. If one considers N = 4, the general form for an equal state combination would be as follows:

$$[1\,1 \;\; 0\,0\,]$$

Both are Equivalent:

$$[1\;0\;\;10]$$

As shown all other possible equal state combinations for N = 4 would contain the same ratio of outcomes, but simply occurring in different re-arranged orders. Thus their joint probabilities will be equivalent, thus both L values will be equivalent. Therefore, L represents number of outcomes that satisfy a given bias and its inverse to formulate all possible equal state combinations for a given Nth set. The Value of L can be calculated as follows:

$$L = \frac{N}{S}$$

We can then scale this up to calculate the total probability ($P(T_{Ec}^{B})$) of any equal state combination given N occurring in the Bayes model, relative to a given bias. This is achieved by implementing equation 2 as the scaling factor, shown below:

$$P(T_{Ec}^{B}) = (1\lfloor 18z^4 + 40z^3 + 38z^2 + 35z + 20 \rfloor)\; P(B_{Ec})$$

This can be simplified to produce the final equation:

$$P(T_{Ec}^{B}) = E_c\; P(B_{Ec})$$

This can be proven using an example via a fair coin simulation, N = 2 with the S = 2 and B1 and B2 = 0.5:

$$L = \frac{2}{2} = 1$$

$$P(B_{Ec}) = (0.5)^1(0.5)^1 = 0.25$$

Z score transformation of N:

$$z = \left(\frac{2-6}{3.1623}\right) = -1.2649021$$

$$18 * -1.2649021^4 + 40 * -1.2649021^3 + 38 * -1.2649021^2 + 35 - 1.2649021 + 20 = 1.65545494223637$$

$$E_c = 1\lfloor 1.65545494223637 \rfloor = 2$$

$$P(T_{Ec}^{B}) = 2 * 0.25 = 0.5$$

$$\begin{array}{l} 1.) \\ 2.) \\ 3.) \\ 4.) \end{array} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Using a N = 2 we can see all possible combinations easily, here one can see that when there is no bias to the coin each combination has a percentage probability of occurring 25% of the time, thus the total probability of an equal state combination occurring is 50%. Adding the remaining probabilities for each combination with the cumulative probability of the equal state combinations will equal 1.

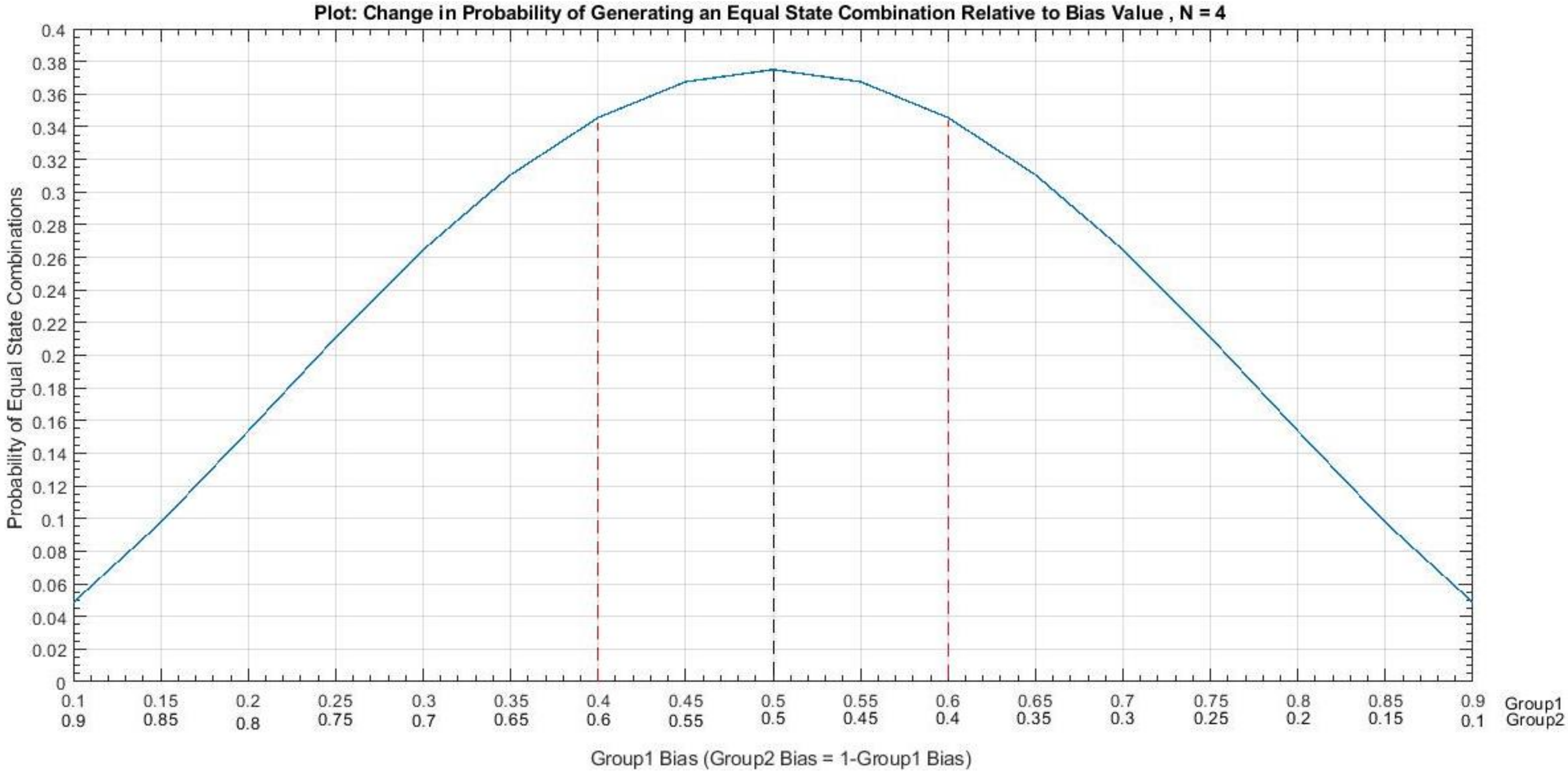
Extrapolating Beyond the Equations

We can now apply above equations to the parameters of task 1 and generate a revised table of Bayesian Cancelation:

States = 2	**Bias(group 1) = 0.9, Bias(group 2) = 0.1**			
Number of Sequential Events (N)	**Number of Possible Event Combinations (C_N)**	**Number of Possible Equal State Combinations (E_C)**	**Probability of an Even Combination Occurring ($P(E_C)$)**	**Probability as a percentage (%)**
2	4	2	0.18	18
4	16	6	0.486	4.86
6	64	20	0.01458	1.458
8	256	70	0.0045927	0.45927
10	1024	252	0.0014880348	0.14880348

Task 1 – Likelihood of generating Bayesian Cancelation

Finally, one can plot the probability of generating an equal state combination when N=4 as the ratio of bias values varies across both Groups.

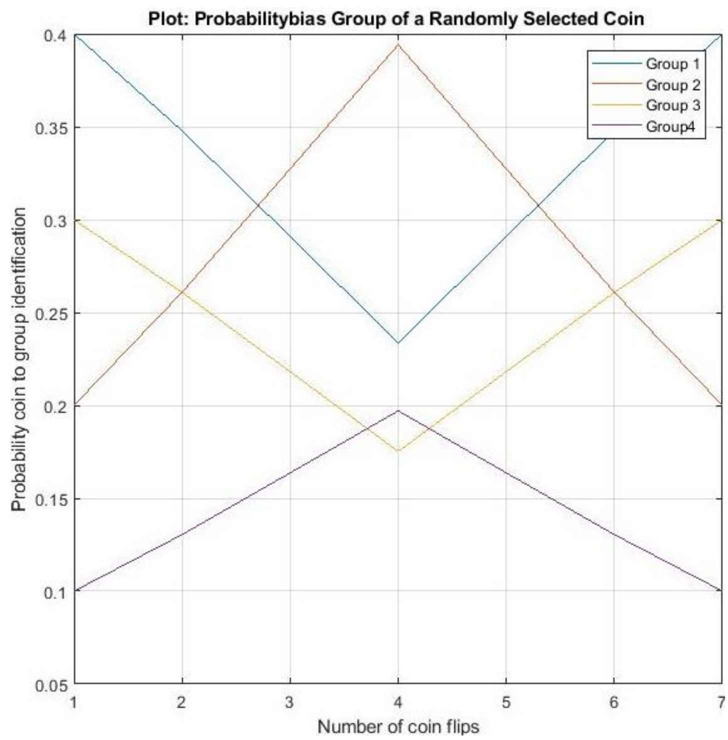


Results reveal a normal distribution of the probability of generating equal state combinations. This is due to the fact that as the group biases becomes more extreme the outcomes become more biased towards producing one of the two possible outcomes (E.g. 1111 or 0000). Logically it becomes less probable that an equal state combination will be generated as it requires outcomes from both sides of the distribution. Overall the graph shows that as the biases of groups 1 & 2 approach 0.5 the probability of generating Bayesian Cancelation effects increases. We also know that the probability of Bayesian cancellation is also dependent upon the value of N(see table 1.1), thus the height (peak) of the above distribution will vary relative to the value of N, N will also effect the frequency of the cancelation oscillations. The spread of the distribution and the amplitude of the cancelation oscillations will be dictated by the bias values for each group, biases that tend towards 0.5 yield smaller oscillations and vice versa. The plot below demonstrates that Bayesian Cancelation is possible when more than two groups are implemented in recursive Bayes.

The plot shows 2 cancelation oscillations and thus Bayesian Cancelation across 4 groups. It should be noted that the prior values can vary across all groups.

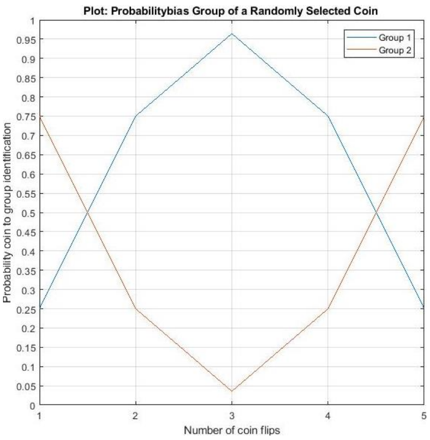
However, there are certain Conditions that must be met to produce Bayesian Cancelation effects.

- Number of groups must be Even
- Bias values set for two groups must be mirrored across all other groups, E.g. if group1 = 0.4/group2 = 0.6 then this biasing must be mirrored onto groups 2 & 3.

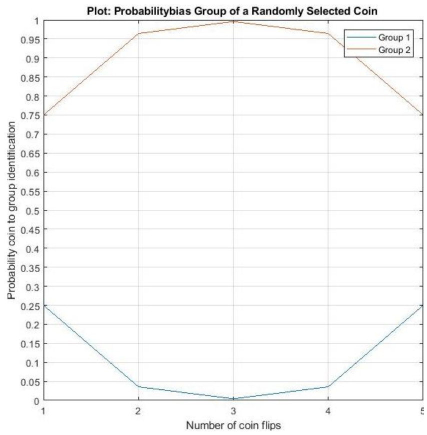


Thus far it is clear that the effects of Bayesian Cancellation found when applying recursive Bayes to the coin flipping problem is directly linked to the occurrence of equal state combinations (equal number of heads and tails). The mathematical relationship between equal combinations, the number of sequential events (number of coin flips) and the bias has also been detailed. Referring back task 1, it is now possible to, categorise, calculate the probability of occurrence and explain the cause of the deviations in the probability shapes, and why they resulted in prior to posterior equivalence when implementing a recursive Bayes model. Task 1.b with N=4, Biases = 0.9/0.1 and states = 2, the probability of Bayesian Cancelation is 4.86% and can be categorised as follows:

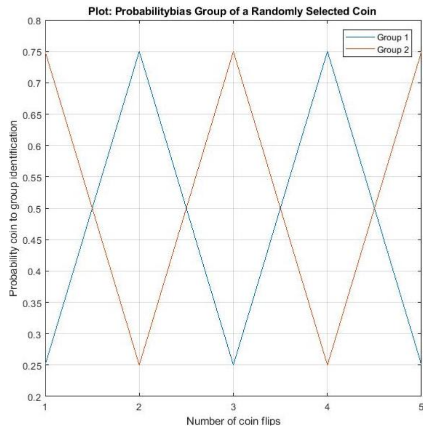
$E_C = [1\ 1\ 0\ 0]$



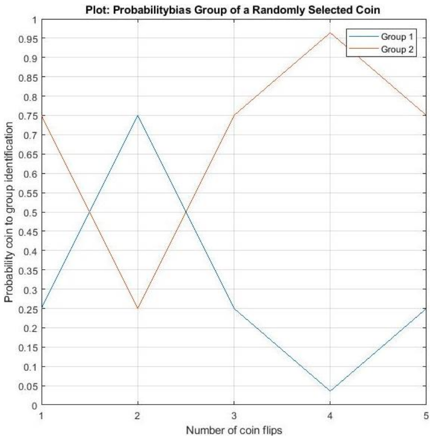
$E_C = [0\ 0\ 1\ 1]$



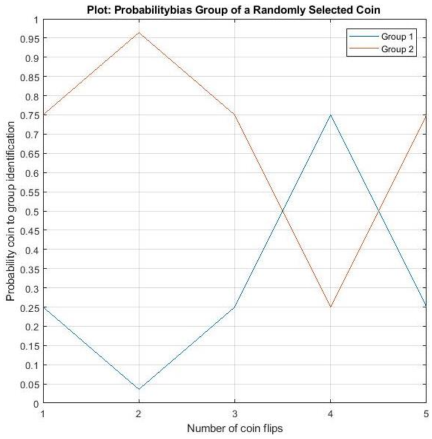
$E_C = [1\ 0\ 1\ 0]$



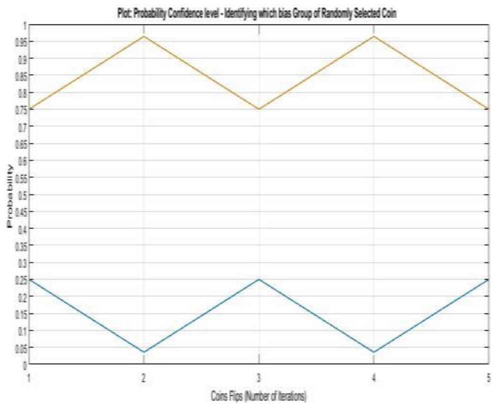
$E_C = [1\ 0\ 0\ 1]$



$E_C = [0\ 1\ 1\ 0]$



$E_C = [0\ 1\ 0\ 1]$



In conclusion it can be shown that flipping the coin more does not always lead a recursive Bayesian model to a stable convergence of relative certainty, but rather it is dependent upon the specific number of times more one flips the coin, and the respective bias placed upon the coin. As a Result there is a probability that Bayesian Cancelation can cause Recursive Bayes to be ineffective.(Appendix A -B)

<u>Conditions that must be Satisfied to Produce Bayesian Cancelation</u>

Only two groups:

- Cumulative Bias values across groups must equal to 1.
- N must be Even

More than two Groups:

- Cumulative Bias values per pair of groups (group1 – group 2 // group3 – group 4) must equal to 1.
- N must be Even
- Number of groups must be even.
- A primary group pair must be selected, its bias values must be mirrored across all other group pairs.

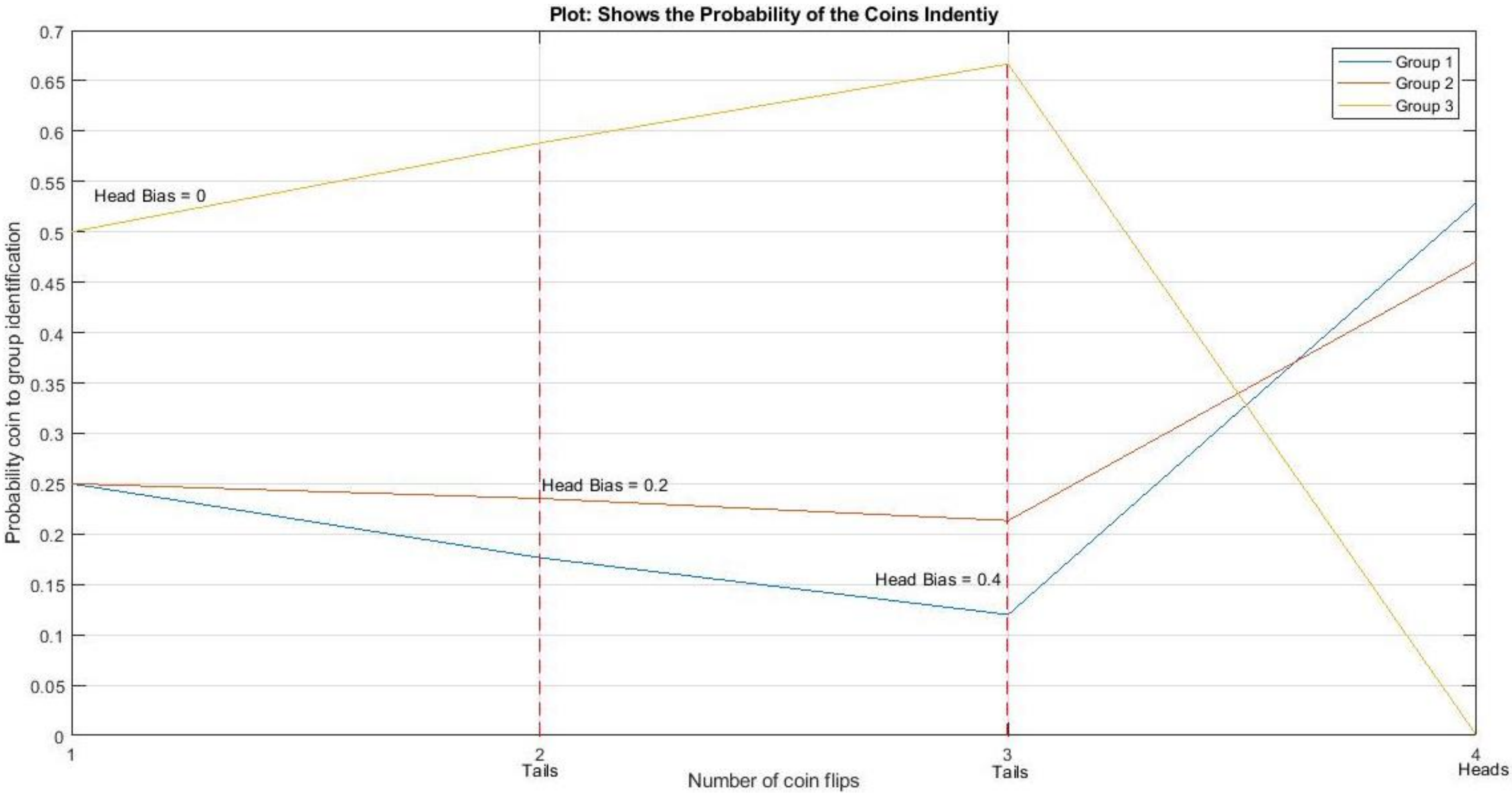
Task 2

2.a - Group1 = Group a, Group2= Group b, Group3 =Group c

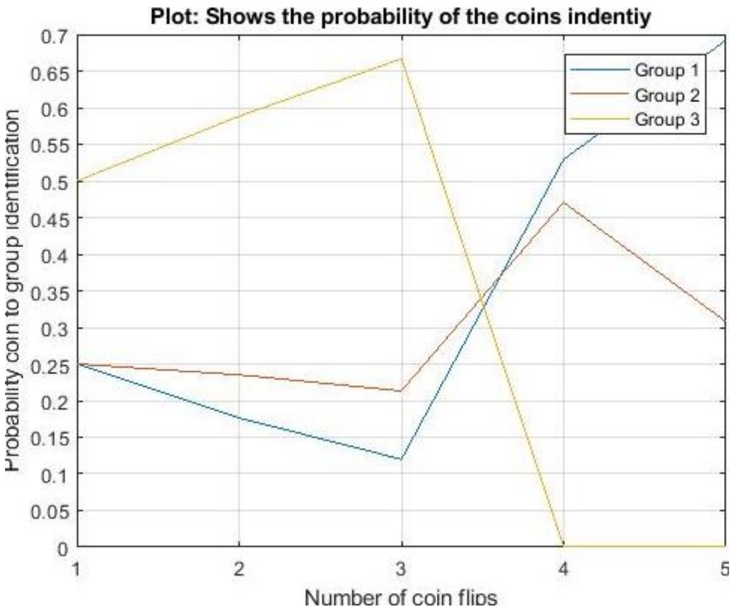
There is a total of 120 coins that can be selected, 60 of them are group3 coins, 30 from group2 and 30 from group 1. Consider all 120 coins to represent 100%, Group 3 accounts for half of the total coins, thus the probability of selecting a group three coin is 0.5. there is remaining to 50% to be a accounted for, if 60 coins is equal to 50% then 30 coins is equal to 25%. Thus selecting group 2 must have prior probability of 0.25.

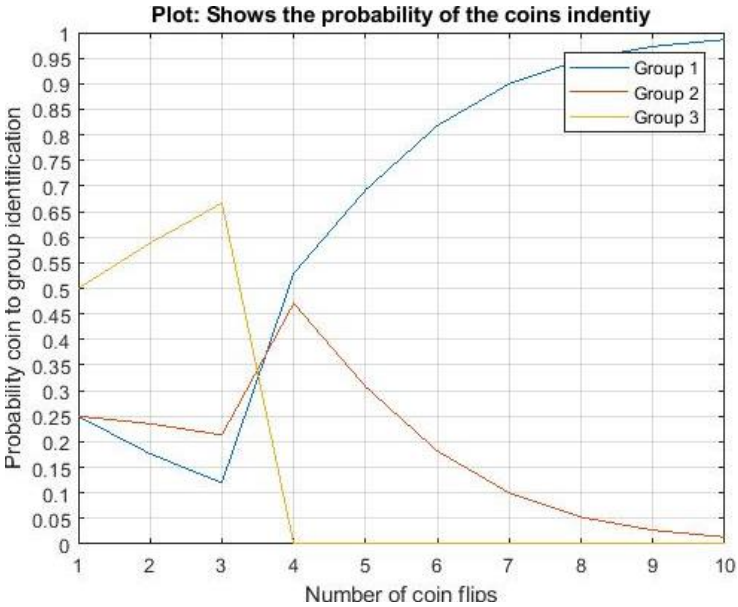
Task 2.b, 2.c & 2.d

Please see Appendix (C) for the programmatic implementation of this task.



In this task there were 3 groups, as stated in task 2.a the following prior probabilities of selecting a coin from each group are as follows; Group 1 prior = 0.25, group 2 prior = 0.25, group 3 prior = 0.5. Group 3 has a 0 probability head bias as it is trick coin with tails being the only outcome it can produce (tails on both faces). The head bias of group 1 will produce a head outcome 40% of the time, thus bias = 0.4. Group2 to will produce a head outcome 20% of the time thus the bias = 0.2. A predetermined outcome per coin flip was set to [tail , tail, heads]. Based on the stated parameters the above plot produces probability results as expected. First considering group 3 one would expect the probability that the coin belongs to group 3 to increase the more tail outcomes are produced with every flip. However with the introduction of a heads outcome the probability of group 3 drops to 0, this is expected due to the fact that if the current coin being flipped produces a heads it is impossible for the coin to belong to group 3, which only can produce tails outcomes. Now considering group 2 we expect the probability to decrease on flips 2 and 3 as it becomes more probable that the coin belongs to group 3. However such a decrease is expected to be relatively small due to the fact that group 2 still is more weighted to producing a tails outcome, 80% of the time. Alternatively when considering group 1 it is expected that its probability reduce more so than group 2, as group 1 is twice as less likely, respective to their head biases, in producing a tail outcome. On the third flip we expect the probabilities of groups 1 and 2 to increase significantly as group 3 is eliminated from the possible identity of the coin. Group 1 gains a slightly higher probability of 52% due to its bias being more likely to produce a heads result. Nonetheless, the model at flip 3 provides a relatively equal likelihood that the coin belongs to either group 1 or 2, with a percentage likelihood difference of only 6%. Although the Bayesian model has followed expected trends given the output set, and has been able to reduce the identity of the coin down to two possible groups. However, at this stage the model is unable to make any reliable inferences when distinguishing between groups 1 and 2. Therefore, this indicates that the model requires more iterations (coin flips) to converge to a more confident estimation. This can be evidence as follows:





The left plot implemented a 4^{th} heads outcome. As stated above increasing the iterations has increased the model's ability to distinguish between the two groups with a much larger percentage likelihood difference of 40% as opposed to the previous 6% difference. There is a 0.7 probability that the coin now belongs to group 1 and a 0.3 probability the coin belongs to group 2, which given the outcome is expected. The right plot shows that given group 1s optimal outcome set [coin landing continuous 1s or heads] past flip 3 requires 10 coin flips for the model to begin to converge at relative certainty of which group the flipped coin belongs too. Both groups increase and decrease symmetrically with an average 4% change per coin flip. Nonetheless the number of required flips for relatively convergence will vary due to a optimal outcome set being relatively unlikely to occur for each group.

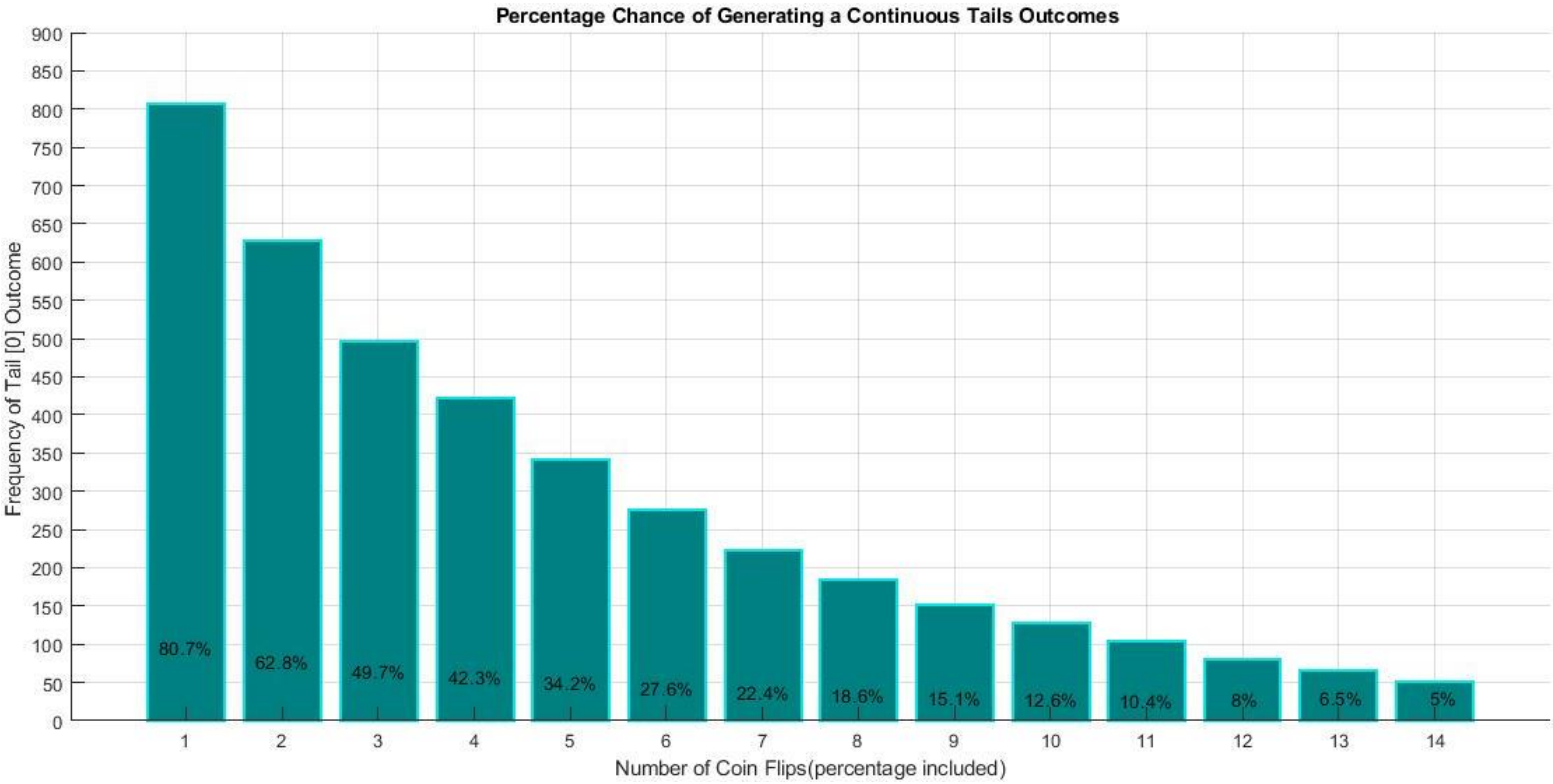
Task 2.3

For the programmatic implementation Of this task please see Appendix C

Assuming that a coin from group 2 has been selected, one can investigate how many iterations it takes the Bayesian Model to reach to a 95% probability of the coin belonging to group two, whilst considering the presence of group 1s & group 3s probabilities. To assume a group 2 coin has been selected I changed the head outcome conditional statement to [if rand < 0.2 = heads, else = tails]. Therefore now considering conceptually one has selected a group 2 coin, there is a 20% chance a heads outcome will occur or a 80% chance a tails outcome will occur when the coin is flipped. Interestingly when we simulate flipping a group 2 coin multiple times, over serval different instances, one finds that each instance requires a different number of coin flips or iterations of the Bayes model to reach a 95% confidence.

Why Does the number of Flips Vary?

The first component that causes the number of coin flips for group 2 to vary is dependent upon the number of coin flips required to eliminate group 3. During Task 2.b it was shown that increases in the probability of group3 causes decreases in probability of groups 1 & 2. This is due to the outcome of tails being more heavily biased towards group3(tail bias = 100%) than group 2(tails bias 80%) or group1(tails bias = 60%). Therefore, Group 2s posterior probability will only begin to increase once the possibility of group 3 is eliminated. The number of flips required for group 3 to reduce to a probability of 0 will also vary, but because it requires a continuous tails outcome sequence to remain probable one can calculate the probability of group3 becoming impossible relative to the number of flips of the coin, as shown below:

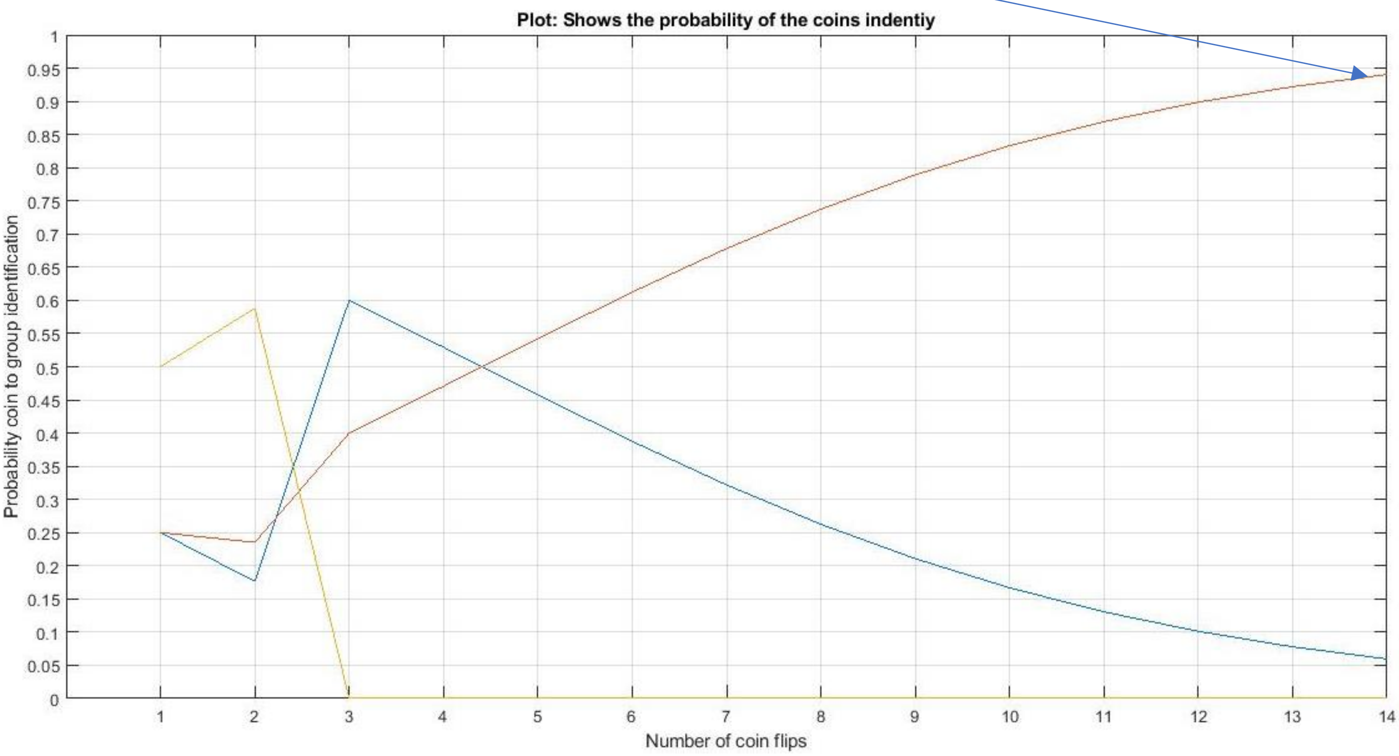


The above plot shows the probability of obtaining a set of continuous zeros as the number of coin flips increases. This was performed over a 1000 randomly generated outcome sequences. The probability of generating a continuous outcome set of tails decreases as the number of coin flips increases, results are also are very close to the analytically expected probability of occurrence (E.g. 0.8 * 0.8 = 0.64, group 2, tail bias = 0.8) . Consequently, one can expect group 3 to converge to a stable probability of 0 around 2-6-coin flips. At this point any tail outcomes that occur will increase the posterior probability of group2 whilst any head outcomes will cause a decrease, due to group1 being more biased towards a head outcome. After several simulations of manually implementing different outcomes, it was found that the optimal outcome with the least number of coin flips required is as follows:

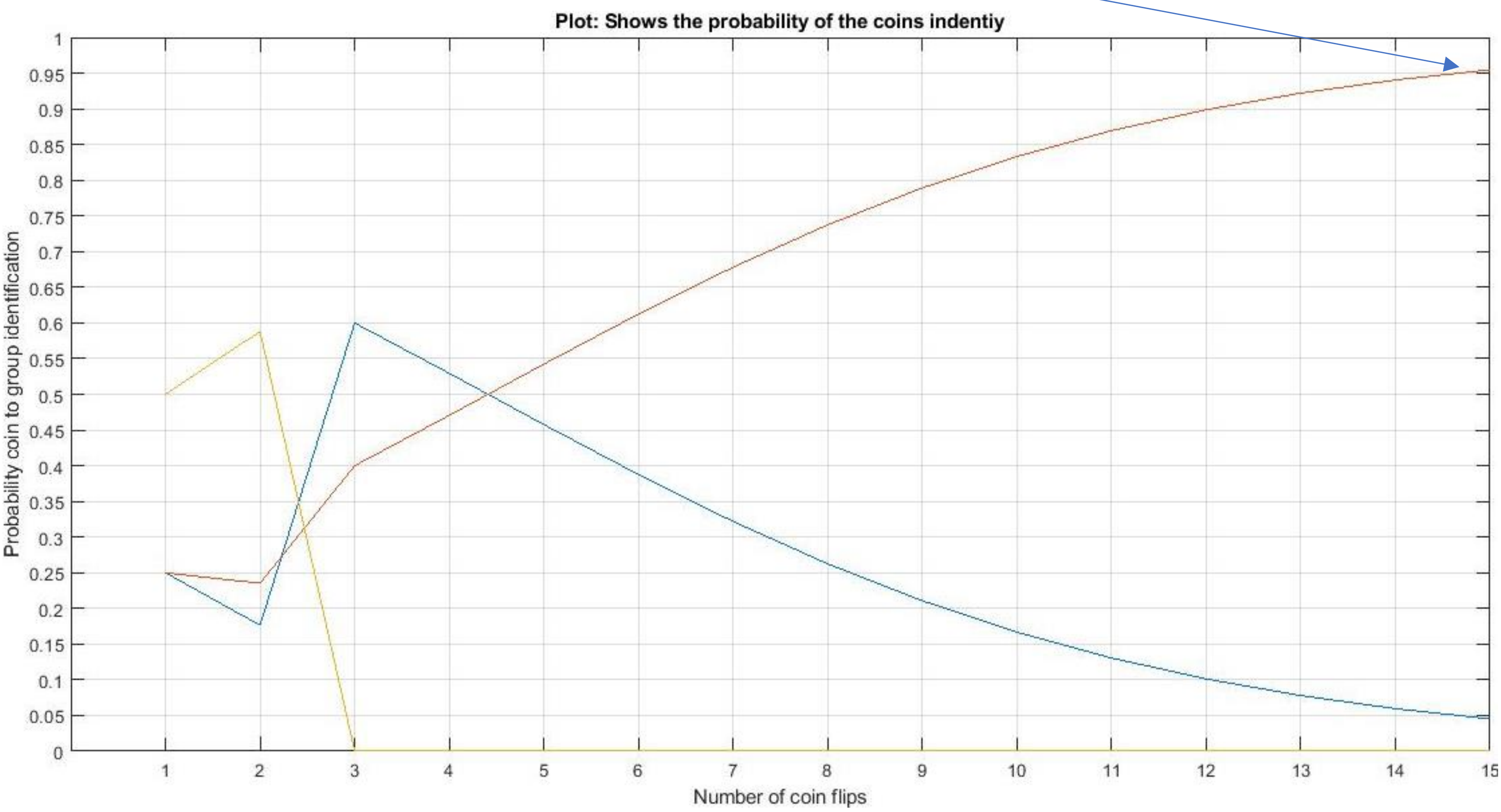
[0 1 0 0 0 0 0 0 0 0 0 0 0 0]

The heads or [1] outcome can occur anywhere in the 14 long sequence thus we can simplify to a 13:1 ratio of tails to heads outcome sequence. This optimal ratio can be evidenced below. Group 2 cannot reach probability of 0.95 with a ratio less than the optimal.

<u>Outcome Ratio = 12:1 (tails : heads)</u>

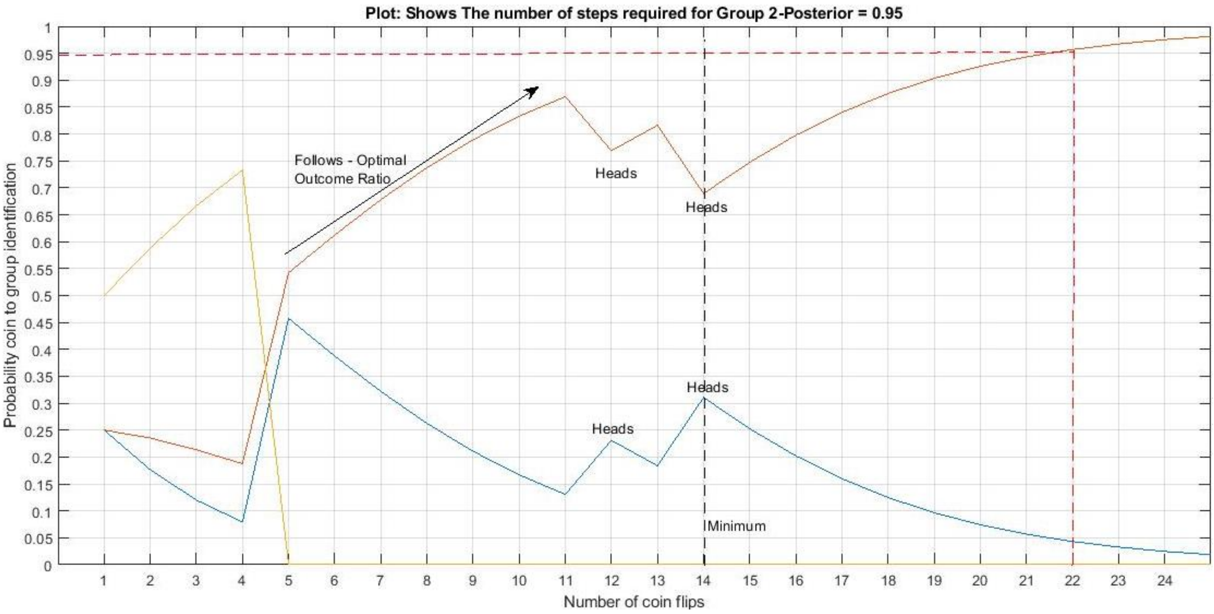


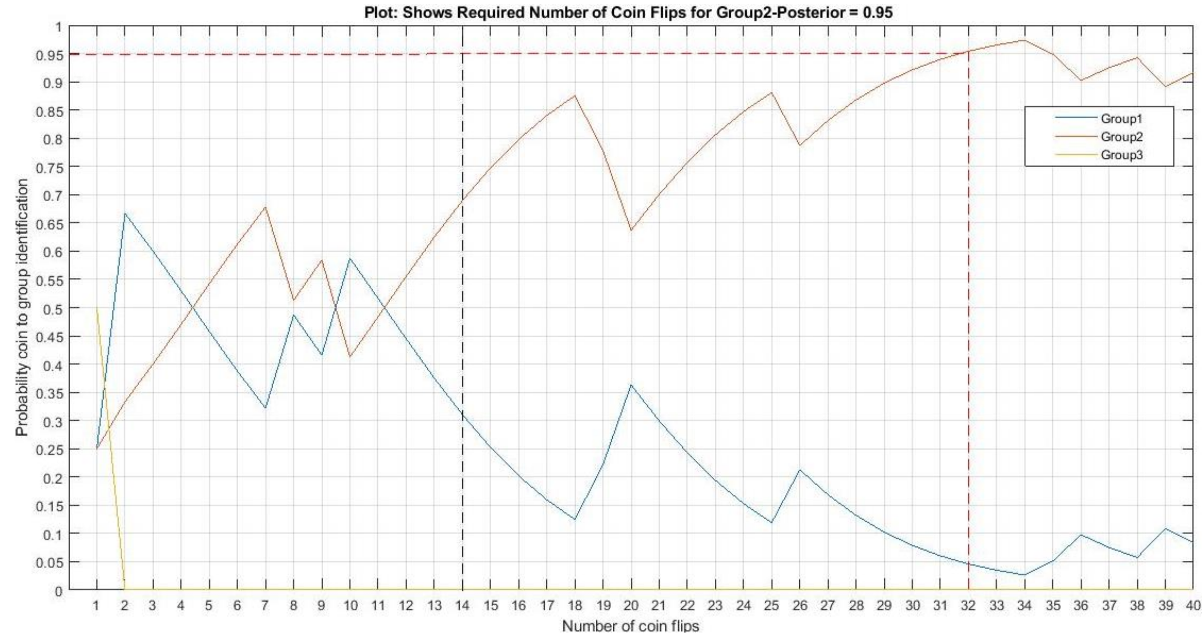
<u>(Optimal)Outcome Ratio = 13:1 (tails : heads)</u>

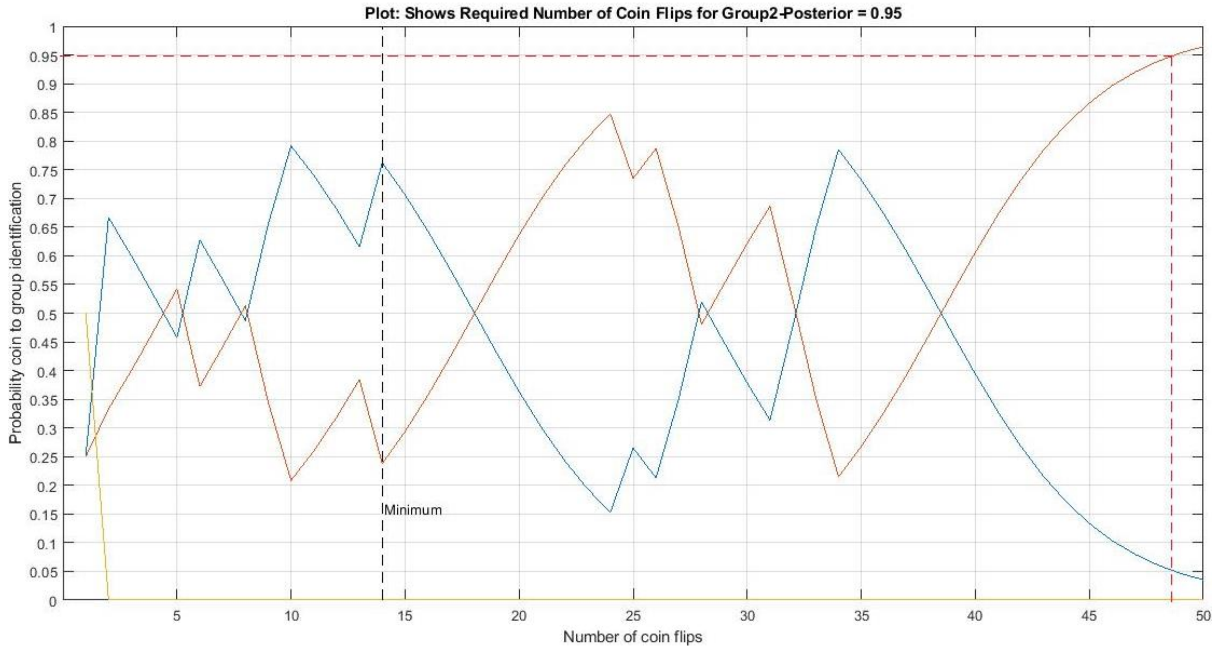


Group 2 requires a minimum of 14-coin flips in order to reach a probability of 0.95. However, this can only be achieved by randomly obtaining the optimal ratio which is very unlikely given its probability of [0.8^13 * 0.2 = 0.0110]. As a result, it is more probable to generate probability ratios that contain more head outcomes (E.g. 10:4, 8:6).

Therefore, In summary one head outcome is required to eliminate the possibility of group3, after which point any further occurrence of a heads outcome will cause a deviation from group 2s optimal outcome ratio of 13:1, and thus require more coin flips to reach a posterior probability of 0.95. The specific number of heads that occur in the deviant ratios can vary and thus the number of coin flips required for a posterior = 0.95 will also vary. As shown as a final note below.







Appendix A

```
% Array Stores
switches = [0 1];
store = [];
prob_store = [];

% Initial Parameters
c1 = 1;
width = 4;
head_bias = 0.2;
tail_bias = 0.8;

% Iterates through All possible Combinations of N = 4, Increase Number of For loops as N
increases(Number of Flips)
for i = switches:1
        for j = switches:1
                for k = switches:1
                    for p = switches:1

                        %Stores the Calculated Combination in Array
                        a = [i, j, k, p];
                        store(c1, :, :, :) = a;
                        c1 = c1 +1;
                        disp([i,j,k,p])
                    end
                end
         end
end

disp('NUMBER OF COMBINATIONS = ')
disp(length(store))

% Once all Combinations for N are Computed another For loop iterates
%through the Matrix "Store" and assesses whether each combination is an
%equal state combination

num_even = 0;

%(Validation)Informs on the location of all Equal state Combins in the matrix
even_location = [];
c2 = 1;


for d = 1:length(store)
    %How many ones and Zeros in the Combination
   a1_one_count = 0;
   a1_zero_count = 0;

   a1 = store(d, :, :, :); %The Current Combination being assesed
   a =  sum(store(d, :, :, :)); %Sum of current Combination

   %Calculation For Validating Equal State Combinations & location
   if ( a == width / 2)
       num_even = num_even + 1;
       even_location(c2, 1) = d;
       c2 = c2 + 1;
   end

   %Below Counts 0:1 ratio per combination to find value of L and thus
   %Allow for probability of the equal state occuring relative to the
   %current bias.
       for h = 1:length(a1)
           if (a1(h) == 1)
               a1_one_count = a1_one_count + 1;
           else
               a1_zero_count = a1_zero_count + 1;
           end
       end
       constant1 = head_bias^a1_one_count;
       constant2 = tail_bias^a1_zero_count;
        comb_probability = constant1 * constant2;
        prob_store(d,:) = comb_probability;

end
```

Appendix B

```
%% Equal State Combination Function & Probability of Occurrence Function

%Used to Iterate through different bias values for task 1.b - Creates
%Normal Distribution

group1_targetbox = [0.9 0.85 0.8 0.75 0.7 0.65 0.6 0.55 0.5 0.45 0.4 0.35 0.3 0.25 0.2
0.15 0.1];
group2_targetbox = [0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8
0.85 0.9];
store = [];
for i = 1:length(group1_targetbox)

% n = Number of sequential events
% S = Number of States
n = 6;
S = 2;

%Calculates probability of an equal State occurring Relative to N
L = n/S;
head_bias = group1_targetbox(i);
tail_bias = group2_targetbox(i);

Q = (head_bias)^L * (tail_bias)^L;

% Scaling Function - Calculates the Total Number of Equal state
% Combinations that can occur within the given Nth Set

z = (n - 6)/3.162;
Ec = round(18*z^4 + 40*z^3 + 38*z^2 + 35*z + 20);

%Scales up the Probability to a Total Probability of any equal State
%Combination occuring relative to N
Ec_prob = (Ec * Q);

store(i, :) = Ec_prob;

end

plot(group1_targetbox, store)
```

Appendix C – Part 1

```
%% Program: Task 2b, c, d & 2.3

% For task 2b, c and d, The Manual outcome Controller was set to [0 0 1] - [tails,
tails, heads].
% For task 2.3 - It is assumed that group 2 coin has been selected thus if (rand < 0.2
= heads)

%Defining the Variables

%group1
group1 = 30;
bias1 = 0.4;
prior1 = 0.25;

%group2
group2 = 30;
bias2 = 0.2;
prior2 = 0.25;

%group3
group3 = 60;
bias3 = 0;
prior3 = 0.50;

%The chance of landing a tails for each biased coin(inverses head chance)
bias_tails1 = (1-bias1);
bias_tails2 = (1-bias2);
bias_tails3 = (1-bias3);

%Stores the initial priors transfers them into the loop
flip_store(1, :, :) = [prior1; prior2; prior3];
post1 = prior1;
post2 = prior2;
post3 = prior3;
```

Appendix C – Part 2

```
%Manual outcome Control - Stores the results of each flip - 1 = heads, 0 = tails
total_count = [];
target = [0, 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 1 1 1 1 1
1 1 1 1 1 1 1 ];
target1 = [0, 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0];

%For loop performs flips of the same coin
for i = 2:50

    prior1 = post1;
    prior2 = post2;
    prior3 = post3;

%Tweak rand values to manually control which coin has been selected or change to target for
control of outcome.

if (rand < 0.4) or if (target(i) == 1)
       %heads

    % Bayes Group 1 - 40%
    num1  = bias1 * prior1;
    dim1 = (prior1 * bias1) + (prior3 * bias3) +  (prior2 * bias2);
    post1 = num1 / dim1;

    %Bayes Group 2 - 20%

    num2 = bias2 * prior2;
    dim2 = (prior2 * bias2) + (prior3 * bias3) +  (prior1 * bias1);
    post2 = num2 / dim2;

    %Bayes group 3 - 0%
    num3 = bias3 * prior3;
    dim3 = (prior3 * bias3) + (prior2 * bias2) +  (prior1 * bias1);
    post3 = num3 / dim3;

    flip_store(i, :, :) = [post1; post2; post3];

    total_count(i) = 1;
else
    %tails

    % Bayes Group 1 - 60%
    num1  = bias_tails1 * prior1;
    dim1 = (prior1 * bias_tails1) + (prior3 * bias_tails3) +  (prior2 * bias_tails2);
    post1 = num1 / dim1;


    % Bayes Group 2 - 80%
    num2 = bias_tails2 * prior2;
    dim2 = (prior2 * bias_tails2) + (prior3 * bias_tails3) +  (prior1 * bias_tails1);
    post2 = num2 / dim2;


    % Bayes Group 3 - 100%
    num3 = bias_tails3 * prior3;
    dim3 = (prior3 * bias_tails3) + (prior2 * bias_tails2) +  (prior1 * bias_tails1);
    post3 = num3 / dim3;

    flip_store(i, :, :) = [post1; post2; post3];
    total_count(i) = 0;

end
end

plot(flip_store, '-')
title('Plot: Shows the probability of the coins indentiy')
xlabel('Number of coin flips')
xticks([5 10 15 20 25 30 35 40 45 50])
ylabel('Probability coin to group identification')
yticks([0 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0
grid on
```

Appendix D