Monty Hall Problem

1 Introduction

The Monty Hall Problem is loosely based on the famous television game show *Let's Make a Deal* originally hosted by Monty Hall. The problem is a variation of the final round of the classic television game show. It was originally asked by Steve Selvin on the American Statistician in 1975 and became famous as a question from a reader's letter quoted in Parade magazine in 1990. The problem goes as follows:

> Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice? [2]

The rules to this fictional game show are as follows:

1. Only one door of the three randomly holds a car
2. You may choose any of the three doors with your first choice
3. Monty Hall reveals one of the doors that meet both these criteria:
   - the door contains a goat
   - the door is not the one you chose with your first choice
4. You are then offered a second choice to either:
   - stick with the door you originally chose
   - switch your choice to the other remaining door
5. The door you chose with your second choice is revealed:
   - if there is a car behind the door, you win a car
   - if there is a goat behind the car, you win a goat

It is assumed someone would much rather win a car than a goat. In order to help people win cars rather than goats, I will answer the question of whether or not it is better to switch doors or stay with the original door when offered the second choice. Our intuition tells us that it would not matter whether or not you switched your choice. If there are two doors to chose from after a goat is revealed, and one has a car, there is a 50% chance the car is in your door and a 50% chance the car is in the other. However, as we will see, our intuition is wrong. I used a Monte Carlo simulation to arrive at my conclusion.

2 Algorithm

Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot be easily predicted due to the existence of random variables. The Monte Carlo method relies on a large number of realizations or trials of a particular scenario to calculate the probabilities of certain events. The more realizations there are, the more accurate the probabilities become. I performed three simulations using 1,000, 10,000, and 100,000 realizations.

2.1 Using the Monte Carlo Method

Monte Carlo methods vary, but here is a general pattern [1]:

1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs.
4. Aggregate the results

2.1.1 My Algorithm

For my simulation I used this specific algorithm:

1. Initialize three doors all holding goats, a player first choice of null, and a player second choice of null.
2. Randomly choose one of the three doors to hold the car and randomly assign the first choice a value of one of the three doors using the `MatLab` "randi" method.
3. Reveal one of the doors that meet both these criteria:
   - the door contains a goat
   - the door is not the door assigned to first choice
4. Deliberately assign second choice the value of the remaining door that **is not contained** in first choice or deliberately assign second choice the value of the door that **is contained** in first choice.
5. Repeat steps one through four using **only one of** step four's options many times counting the number of times a car is won.
6. Divide the number of times a car is won by the total number of trials ran to obtain the probability of winning a car if you choose to switch doors or not to switch doors.
7. Repeat steps one through six using step four's other option.
8. Display both probabilities side by side to see whether or not it is smarter to switch doors or stick with the original pick.

3 Results

In `MatLab`, my simulation code is invoked at the prompt by calling the function `MontyHallProblem()`. The command

```
>> probabilities=MontyHallProblem();
```

will compute the probabilities of winning a car based on whether or not you choose to switch doors and store the results as a string in a variable called probabilities.

The results of my simulations show that it is better to switch doors rather than to stick with your original pick.

```
>> MontyHallProblem()

ans =

Running statistics for 1,000, 10,000, and 100,000 realizations.

1000 Realizations:
Probability of winning a car if you choose to:
pick the other door:  66.4%
stay with the same door:  37.3%

10000 Realizations:
Probability of winning a car if you choose to:
pick the other door:  67.53%
stay with the same door:  32.49%

100000 Realizations:
Probability of winning a car if you choose to:
pick the other door:  66.825%
stay with the same door:  33.253%


>>
```

4 Conclusion

The probability of winning a car if you choose to change doors when offered a second choice is roughly $\frac{2}{3}$ and the probability of winning a car if you choose to stick with your original pick when offered a second choice is roughly $\frac{1}{3}$. Therefore, it is better to change doors when offered

a second choice.

This goes against our intuition that the chance each door holds a car should be split 50/50. We should think about the problem differently than we did in the introduction. At the beginning, one of the three doors has a car and you guess which door has one. Based on random chance, there is a $\frac{1}{3}$ chance your pick has a car behind it and a $\frac{2}{3}$ chance the car is not behind your door. When Monty Hall reveals one of the goats, those odds do not change. There is still a $\frac{2}{3}$ chance that your door does not have the car, so you should switch doors.

A Implementation in MatLab

The MatLab implementation with comments is given here:

DoesSwitch.m:

```
% FUNCTION THAT RETURNS A BOOLEAN TRUE OR FALSE VALUE DEPENDING ON WHETHER
% OR NOT THE CAR WAS WON WHEN THE PLAYER DECIDED TO SWITCH DOORS. IF true
% IS RETURNED, THE CAR WAS WON. IF false WAS RETURNED, A GOAT WAS WON.

function gotCar = DoesSwitch()

% INITIALIZE VARIABLES AND DECIDE WHICH DOOR HOLDS THE CAR
%   AND WHICH DOOR THE PLAYER CHOSE
    door1 = false;
    door2 = false;
    door3 = false;
    carDoor = randi(3);
    OriginalDoorPick = randi(3);
    SecondDoorPick = 0;

% SET THE VALUE OF WHICHEVER DOOR HOLDS THE CAR TO TRUE
    if (carDoor == 1)
        door1 = true;
    elseif (carDoor == 2)
        door2 = true;
    else
        door3 = true;
    end

% REVEAL ONE OF THE DOORS THAT CONTAINS A GOAT
% THEN CHOOSE WHETHER OR NOT TO SWITCH DOORS
%   IN THIS CASE THE PLAYER CHOOSES TO SWITCH DOORS
% FINALLY RETURN WETHER OR NOT THE PLAYER CHOSE THE DOOR WITH THE CAR

% REVEAL DOOR 1 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 2 OR 3)
    if (door1 == false && OriginalDoorPick ~= 1)

        if (OriginalDoorPick == 2)%
            SecondDoorPick = 3;   %
        else                      % CHOOSE TO SWITCH DOORS
            SecondDoorPick = 2;   %
        end                       %
```

```
        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 2)
            gotCar = door2;
        else
            gotCar = door3;
        end
    end

% REVEAL DOOR 2 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 1 OR 3)
    if (door2 == false && OriginalDoorPick ~= 2)

        if (OriginalDoorPick == 1)%
            SecondDoorPick = 3;   %
        else                      % CHOOSE TO SWITCH DOORS
            SecondDoorPick = 1;   %
        end                       %

        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 1)
            gotCar = door1;
        else
            gotCar = door3;
        end
    end

% REVEAL DOOR 3 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 1 OR 2)
    if (door3 == false && OriginalDoorPick ~= 3)

        if (OriginalDoorPick == 2)%
            SecondDoorPick = 1;   %
        else                      % CHOOSE TO SWITCH DOORS
            SecondDoorPick = 2;   %
        end                       %

        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 1)
            gotCar = door1;
        else
            gotCar = door2;
        end
    end
```

```
end
```

DontSwitch.m:

```
% FUNCTION THAT RETURNS A BOOLEAN TRUE OR FALSE VALUE DEPENDING ON WHETHER
% OR NOT THE CAR WAS WON WHEN THE PLAYER DECIDED NOT TO SWITCH DOORS. IF
% true IS RETURNED, THE CAR WAS WON. IF false WAS RETURNED, A GOAT WAS WON.

function gotCar = DontSwitch()

% INITIALIZE VARIABLES AND DECIDE WHICH DOOR HOLDS THE CAR
%   AND WHICH DOOR THE PLAYER CHOSE
    door1 = false;
    door2 = false;
    door3 = false;
    carDoor = randi(3);
    OriginalDoorPick = randi(3);
    SecondDoorPick = 0;

% SET THE VALUE OF WHICHEVER DOOR HOLDS THE CAR TO TRUE
    if (carDoor == 1)
        door1 = true;
    elseif (carDoor == 2)
        door2 = true;
    else
        door3 = true;
    end

% REVEAL ONE OF THE DOORS THAT CONTAINS A GOAT
% THEN CHOOSE WHETHER OR NOT TO SWITCH DOORS
%   IN THIS CASE THE PLAYER CHOOSES NOT TO SWITCH DOORS
% FINALLY RETURN WETHER OR NOT THE PLAYER CHOSE THE DOOR WITH THE CAR

% REVEAL DOOR 1 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 2 OR 3)
    if (door1 == false && OriginalDoorPick ~= 1)

        SecondDoorPick = OriginalDoorPick;%CHOOSE NOT TO SWITCH DOORS

        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 2)
            gotCar = door2;
        else
```

```
            gotCar = door3;
        end
    end

% REVEAL DOOR 2 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 1 OR 3)
    if (door2 == false && OriginalDoorPick ~= 2)

        SecondDoorPick = OriginalDoorPick;%CHOOSE NOT TO SWITCH DOORS

        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 1)
            gotCar = door1;
        else
            gotCar = door3;
        end
    end

% REVEAL DOOR 3 TO BE A GOAT (PLAYER ORIGINALLY CHOSE DOOR 1 OR 2)
    if (door3 == false && OriginalDoorPick ~= 3)

        SecondDoorPick = OriginalDoorPick;%CHOOSE NOT TO SWITCH DOORS

        %CHECK TO SEE IF PLAYER CHOSE DOOR WITH CAR
        if (SecondDoorPick == 1)
            gotCar = door1;
        else
            gotCar = door2;
        end
    end

end
```

probabilitiesAfter.m:

```
% FUNCTION THAT CALCULATES THE PROBABILITIES OF WINNING A CAR DEPENDING ON
% WHETHER OR NOT YOU CHOOSE TO SWITCH OR NOT TO SWITCH DOORS USING n
% REALIZATIONS. RESULTS ARE GIVEN AS A STRING.

function statistics = probabilitiesAfter(n)%trials

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % WILL BE USED TO HOLD THE TOTAL NUMBER OF PEOPLE WHO WIN CARS OUT OF
    % n PLAYERS FOR A GIVEN RUN THROUGH OF REALIZATIONS
```

```
    numWinners         = 0;
% RUN A PROBABILITY ANALYSIS ON THE ODDS OF WINNING A CAR DEPENDING ON
% WHETHER OR NOT YOU CHOOSE TO SWITCH DOORS BASED ON n TRIALS

    % FORMATTING OUTPUT
    statistics = num2str(n);
    statistics = strcat(statistics,sprintf(' Realizations:'));
    statistics = strcat(statistics,sprintf('\nProbability of winning'));
    statistics = strcat(statistics,sprintf(' a car if you choose to:'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for iterator = 1:n                  %                                      %
        if (DoesSwitch())               % COUNT NUMBER OF WINNERS OUT OF       %
            numWinners = numWinners + 1;% n PEOPLE WHO MAKE THE CHOICE         %
        end                             % TO SWITCH DOORS                      %
    end                                 %                                      %
                                                                               %
    % TURN THE NUMBER OF WINNERS INTO A PROBABILITY REPRESENTED AS A           %
    probability = (numWinners/n) * 100;                        % PERCENTAGE %
    numWinners  = 0; % SET numWinners BACK TO ZERO FOR FUTURE USE              %
                                                                               %
    % FORMATTING OUTPUT                                                       %
    statistics  = strcat(statistics,sprintf('\npick the other door'));         %
    statistics  = strcat(statistics,sprintf(':      \10'));                    %
    statistics  = strcat(statistics,num2str(probability),'%');                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for iterator = 1:n                  %                                      %
        if (DontSwitch())               % COUNT NUMBER OF WINNERS OUT OF       %
            numWinners = numWinners + 1;% n PEOPLE WHO MAKE THE CHOICE         %
        end                             % NOT TO SWITCH DOORS                  %
    end                                 %                                      %
                                                                               %
    % TURN THE NUMBER OF WINNERS INTO A PROBABILITY REPRESENTED AS A           %
    probability = (numWinners/n) * 100;                        % PERCENTAGE %
                                                                               %
    % FORMATTING OUTPUT                                                       %
    statistics  = strcat(statistics,sprintf('\nstay with the same door'));%
    statistics  = strcat(statistics,sprintf(':  \10'));                        %
    statistics  = strcat(statistics,num2str(probability),'%');                 %
    statistics  = strcat(statistics,sprintf('\n\n\n\10'));                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

MontyHallProblem.m:

```
% FUNCTION THAT RUNS MY THREE SIMULATIONS OF THE MONTY HALL PROBLEM USING
% 1,000, 10,000, AND 100,000 SIMULATIONS RESPECTIVELY AND GIVES RESULTS AS
% A STRING

function results = MontyHallProblem()

    % INITIALIZE VARIABLES THAT WILL BE USED IN STATISTICAL ANALYSIS
    numRealizationsMin = 1000;
    numRealizationsMid = 10000;
    numRealizationsMax = 100000;

    % FORMAT OUTPUT
    results = 'Running statistics for 1,000, 10,000, and 100,000';
    results = strcat(results,' realizations.',sprintf('\n\n\n\10'));

    % CALULATE RESULTS BY USING probabilitiesAfter(n)"trials" FUNCTION
    results = strcat(results,probabilitiesAfter(numRealizationsMin));
    results = strcat(results,probabilitiesAfter(numRealizationsMid));
    results = strcat(results,probabilitiesAfter(numRealizationsMax));
end
```

References

[1] Monte carlo method. https://en.wikipedia.org/wiki/Monte_Carlo_method. Accessed 2016-10-11.

[2] Monty hall problem. https://en.wikipedia.org/wiki/Monty_Hall_problem. Accessed 2016-10-11.