Project 10: Monty Hall, Fall, and Crawl Problems

The Monte Hall Problem

You should switch door. This is because, at first, the chance of picking the correct door is 1/3. When the host open the door with the goat and allow you to pick again, you now have a 1/2 chance of picking the correct door, increasing your chance of winning. This is the same for the problem with 100 doors. The chance of picking the correct door is slim, 1/100. However, as 98 of the doors are revealed to have a goat behind them and the host allow you to re-pick the door, you now have a 1/2 chance of winning. Therefore, choosing to switch will increase your chances in winning the car. If you choose to switch, the probability of winning is 2/3. Whereas, if you choose to stay with your original choice, the probability of winning is 1/3.

In[343]:=
```
montyHall[n_] := Module[{car, myguess, ifStay, ifSwitch, stayprob, switchprob},
  car := RandomInteger[{1, 3}, n];
  myguess := RandomInteger[{1, 3}, n];
  ifStay = Count[Transpose[{car, myguess}], {match_, match_}];
  ifSwitch = n - ifStay;
  stayprob = N[ifStay / n];
  switchprob = N[ifSwitch / n];
  Print["Probability of Wins by Method"];
  Print["Stay Wins = ", stayprob];
  Print["Switch Wins = ", switchprob]
 ]
```

In[344]:=
```
montyHall[1000]
```

```
Probability of Wins by Method

Stay Wins = 0.336

Switch Wins = 0.664
```

The Monte Fall Problem

In the case of the Monte Fall Problem, if the host accidentally open the door to the car, the game is over. However, when the host fall and open the door of a goat, the game continues. In this case, the probability of winning will be the same regardless of switching. There is a probability of 1/2 of winning when switching door and staying at the same door.

In[345]:=
```
montyFall[n_] :=
 Module[{car, myguess, open, ifStay, ifSwitch, stayprob, switchprob, openCar, m},
  car := RandomInteger[{1, 3}, n];
  myguess := RandomInteger[{1, 3}, n];
  open := RandomInteger[{1, 3}, n];
  openCar = Count[Transpose[{car, open}], Drop[{match_, match_}]];
  Transpose[{car, open}];
  {match_, match_};
  m = n - openCar;
  ifStay = Count[Transpose[{car, myguess}], {match_, match_}];
  ifSwitch = m - ifStay;
  stayprob = N[ifStay / m];
  switchprob = N[ifSwitch / m];
  Print["Probability of Wins by Method"];
  Print["Stay Wins = ", stayprob];
  Print["Switch Wins = ", switchprob]
 ]
```

In[346]:=
```
montyFall[1000]
```

```
Probability of Wins by Method

Stay Wins = 0.504559

Switch Wins = 0.495441
```

In[347]:=

The Monte Crawl Problem

This problem is dependent on which number door the host will open. There are two events that can occur for this problem.

The first event that can occur is when the host passes the first door he could have open. This extra information insinuate that the prize is behind the door, and that the door of the current guess is a goat. Therefore, there is a probability of 1 of winning if you switch door and 0 if you stay. This occurs when the host open a high number door (Door #3), when you select one of the low number door (Door #1 or #2). This also occur when the host open Door #2 when you initially choose Door #3, meaning the car is being Door #1. The list of this event would be {1, 2, 3}, {2, 1, 3}, and {1, 3, 2}. (in the form {car,initial guess, open})

In[348]:=
```
montyCrawlsit1[n_] :=
 Module[{car, myguess, ifStay, ifSwitch, stayprob, switchprob, m, game, open},
  car := RandomInteger[{1, 3}, n];
  myguess := RandomInteger[{1, 3}, n] ;
  open := RandomInteger[{1, 3}, n];
  ifSwitch = m = Count[Transpose[{car, myguess, open}], {1, 2, 3}] +
     Count[Transpose[{car, myguess, open}], {2, 1, 3}] +
     Count[Transpose[{car, myguess, open}], {1, 3, 2}];
  (*Since you cannot win unless you switch, the number of trials
   that win is the same as the number of ifSwitch*)
  ifStay = m - ifSwitch;
  stayprob = N[ifStay / m];
  switchprob = N[ifSwitch / m];
  Print["Probability of Wins by Method"];
  Print["Stay Wins = ", stayprob];
  Print["Switch Wins = ", switchprob]
 ]
```

In[349]:=
```
montyCrawlsit1[1000]
```

```
Probability of Wins by Method
Stay Wins = 0.
Switch Wins = 1.
```

In the second situation, the host goes to the first door and does not skip. During this game, there is no extra information given by the host. Since the host does not provide any extra information, the only event where you win by staying with your initial pick is when you pick the door with the car. The list of this event would be {1, 1, 2}, {2, 2, 1}, {3, 3, 1}. (in the form {car,initial guess, open}) Otherwise, you would have to switch from your initial pick to win. The list of this event would be {3, 1, 2}, {3, 2, 1}, {2, 3, 1}. (in the form {car,initial guess, open}) Therefore, the probability of choosing the door with the car is 1/2 and choosing the door with a goat is 1/2.

In[350]:=
```
montyCrawlsit2[n_] := Module[{car, myguess, open, openCar, openMyguess, m},
  car := RandomInteger[{1, 3}, n];
  myguess := RandomInteger[{1, 3}, n]; open := RandomInteger[{1, 3}, n];
  m = Count[Transpose[{car, myguess, open}], {1, 1, 2}] +
    Count[Transpose[{car, myguess, open}], {3, 1, 2}] +
    Count[Transpose[{car, myguess, open}], {2, 2, 1}] +
    Count[Transpose[{car, myguess, open}], {3, 2, 1}] +
    Count[Transpose[{car, myguess, open}], {2, 3, 1}] +
    Count[Transpose[{car, myguess, open}], {3, 3, 1}];
  ifStay = Count[Transpose[{car, myguess, open}], {1, 1, 2}] +
    Count[Transpose[{car, myguess, open}], {2, 2, 1}] +
    Count[Transpose[{car, myguess, open}], {3, 3, 1}];
  ifSwitch = m - ifStay;
  stayprob = N[ifStay / m];
  switchprob = N[ifSwitch / m];
  Print["Probability of Wins by Method"];
  Print["Stay Wins = ", stayprob];
  Print["Switch Wins = ", switchprob]
 ]
```

In[351]:=
```
montyCrawlsit2[1000]
```

```
Probability of Wins by Method

Stay Wins = 0.483871

Switch Wins = 0.516129
```