Problem 2 - Report

For project repository, please check: <https://github.com/DariMe20/Game-Theory-Applications>

**Game Description**

In this problem, there are **N doors**, with a gift placed behind one of the doors. The player initially selects one door, and then all doors except for the one chosen and one other door are opened, revealing no gift. The player is then given two strategies to choose from for the final selection:

1. **Strategy 1 (Stay):** Keep the door initially chosen.
2. **Strategy 2 (Switch):** Switch to the remaining unopened door.

The task is to simulate the game multiple times and calculate the probabilities of winning under each strategy, for different values of **(n)** (number of doors). The game is repeated **(k)** times to get an accurate estimate of the probabilities.

**Pseudocode for the Probability Calculation Function**

**Function: simulate\_game(N, K, strategy)**

Initialize win\_count = 0

For each simulation (from 1 to K):

Randomly place the gift behind one of the doors (using random.randint(0, N - 1)).

The player randomly chooses an initial door (using random.randint(0, N - 1)).

If the strategy is "stay":

If initial door choice = = door with gift, win\_count ++.

If the strategy is "switch":

If initial door choice != door with gift, win\_count ++.

Return the win probability as **win\_count / K.**

**Function: evaluate\_probabilities(N, K\_values)**

For each value of K in the K\_values list:

Calculate the probabilities for both strategies ("stay" and "switch") using simulate\_game.

Store the results in a list.

Convert the list of results into a DataFrame and print it.

**Results**

The following results demonstrate that switching the door gives a higher change of winning the gift then staying with the initial choice, especially as the number of simulations increases.

**For N = 3 doors:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 10 | 20.00% | 70.00% |
| 100 | 33.00% | 61.00% |
| 1000 | 33.20% | 63.60% |
| 10000 | 33.15% | 66.80% |

**For N = 5 doors:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 10 | 60.00% | 70.00% |
| 100 | 17.00% | 81.00% |
| 1000 | 18.70% | 80.00% |
| 10000 | 19.52% | 80.68% |

**For N = 10 doors:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 10 | 20.00% | 90.00% |
| 100 | 9.00% | 91.00% |
| 1000 | 11.30% | 89.40% |
| 10000 | 10.71% | 90.12% |

**For N = 20 doors:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 10 | 10.00% | 90.00% |
| 100 | 8.00% | 95.00% |
| 1000 | 3.30% | 94.40% |
| 10000 | 4.97% | 94.92% |

**For N = 100 doors:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 10 | 0.00% | 90.00% |
| 100 | 2.00% | 97.00% |
| 1000 | 1.10% | 99.20% |
| 10000 | 0.91% | 98.89% |

**For N = 3 doors and custom K = 100,000 simulations:**

|  |  |  |
| --- | --- | --- |
| **k** | **Stay Prob** | **Switch Prob** |
| 100000 | 33.10% | 66.50% |