README File

Generalized Monty Hall Problem Simulation

This ***Python script*** simulates the Generalized Monty Hall problem and provides both theoretical and simulated results for the probabilities of winning the game with and without switching doors.

● Overview

It is a generalized form of famous Monty Hall Problem. Here, we are given ‘***n’*** doors and ‘***k’*** cars, and we have to find out probability for getting a car with shifting the door and without shifting the door. We have also made a ‘***Surface Plot***’ for n, k and their probability ratio.

● Features

1. **Simulations:** The script performs simulations to estimate the probabilities of winning the game both with and without switching doors. We have made an array to function as doors, and stored the value as ‘**1’** for having a car behind the door and ‘**0**’ otherwise. Later we choose a random index for the array, and break it into cases.

2. **Theoretical Calculation:** Theoretical probabilities are calculated based on the mathematical analysis of the problem.

3. **3D Surface Plot:** The script generates a 3D surface plot visualizing the probability ratio of winning with switching to winning without switching, varying the number of doors and the number of doors with prizes.

● Usage

1. **Install Dependencies:** Make sure you have Python installed on your system along with the required libraries - **NumPy** and **Matplotlib.**

```
*******************************************************
pip  install  numpy  matplotlib
*******************************************************
```

2. **Run the Script:** Execute the Python script `***monty_hall_simulation.py***` using a Python interpreter.

```
*******************************************************
python  monty_hall_simulation.py
*******************************************************
```

3. **Input Parameters:** Enter the number of doors `*n*` and the number of doors with car `*k*` when prompted.

4. **View Results:** The script will display both theoretical and simulated probabilities of winning with and without switching doors. Additionally, it will generate a 3D surface plot illustrating the probability ratio.

- ## Inputs

1. **n** : Number of doors in the game.
2. **k** : Number of doors with car behind them.

- ## Outputs

1. Theoretical and simulated probabilities of winning the game with and without switching doors.
2. 3D surface plot depicting the probability ratio vs. the number of doors vs. the number of doors with cars.

- ## What do I infer from the plot?

1. **Trends in Probability Ratio:** The surface plot shows how the probability ratio, which represents the likelihood of winning with switching versus winning without switching, varies with the number of doors `*n*` and the number of doors with cars `*k*`. For **low values** of '***n***' and '**k**' the **ratio is larger** whereas **for higher values it decreases**. After a certain value of '***n'*** and '***k***' the **ratio becomes near constant, depicting a very slight change**. But it is **always grater than '*1*'** it implies that it is **beneficial to switch the door in all scenarios (probabilistically)**.

2. **Generalization of results:** The surface plot offers a visual representation of the complex relationship between the number of doors, the number of doors with cars, and the probability ratio.One may look at curve and see that **it is always beneficial to switch the door**.

Thank You :)

Submitted by -

Purav Jangir
220837