Reto 2 Nodel

El problema de Monty Hall

> ***En este concurso, el concursante escoge una puerta entre tres, y su premio consiste en lo que se encuentra detrás. Una de ellas oculta un coche, y tras las otras dos hay una cabra. Sin embargo, antes de abrirla, el presentador, que sabe dónde está el premio, abre una de las otras dos puertas y muestra que detrás de ella hay una cabra. Ahora tiene el concursante una última oportunidad de cambiar la puerta escogida ¿Debe el concursante mantener su elección original o escoger la otra puerta? ¿Hay alguna diferencia?***

Simule el problema de Monty Hall descrito anteriormente, usando Python. Conteste las preguntas planteadas en el problema y justifique su respuesta en base a las probabilidades obtenidas al cambiar o no de puerta. Es decir, **demostrar** que al cambiar de puerta mejora sus probabilidades de ganar a un 66% en lugar del 33% aprox. si no cambia de puerta, para ello debe simular el problema de Monty Hall al menos 100000 ó 1000000 veces. Adicionalmente **analice qué pasaría** si se agrega una puerta más (sin premio) al problema original y que el presentador no sepa en qué puerta está el premio.

Autor: Johao Villarroel

Fecha de Entrega: 05 diciembre de 2022

Índice de contenido

Acrónimos

Caso de estudio

Se desea validar los conocimientos con una serie de retos, en el presente reto se validan los conocimientos con la simulación de un problema llamado Monty hall.

Requerimientos

- Python 3.9
- Librería numpy
- Librería pandas

Procedimientos

Luego de importar las librerías, crearemos una función para que esta pueda ser repetida las veces que se necesite.

```
def MontyHallSimulation (N):
```

```
MontyHallSimulation(N=100000)
```

Aquí haremos dos listas las cual almanezaran cuantas veces gana la persona cuando cambia y no cambia su eleccion, asi generaremos las 3 puertas donde una de estas sera la ganadora, ademas de la eleccion de la persona, luego de esto se quitara una de las puertas que no tengan nada y la otra puerta quedara guardada en una variable, esto con el objetivo de comparar la puerta ganadora con la eleccion del jugador y con la otra puerta, el segundo haciendo referencia las veces que se gana cuando cambia de puerta.

```
NoCambia=[]
Cambia=[]
for i in range(0,N):
    puertaGanadora=random.choice(['Puerta 1', 'Puerta 2', 'Puerta 3'])
    primeraEleccion=random.choice(['Puerta 1', 'Puerta 2', 'Puerta 3'])
    puertaAbierta=list(set(['Puerta 1', 'Puerta 2', 'Puerta 3'])-set([primeraEleccion,puertaGanadora]))[0]
    otraPuerta=list(set(['Puerta 1', 'Puerta 2', 'Puerta 3'])-set([primeraEleccion,puertaAbierta]))[0]
    NoCambia.append(primeraEleccion==puertaGanadora)
    Cambia.append(otraPuerta==puertaGanadora)
```

Con esto podemos generar la simulación para las veces que se dese y se obtendrá las probabilidades que ino tiene depende la estrategia que se ha elegido.

```
    100,000 simulaciones realizadas
    Porcentaje de ganar basado en estrategia:
    Mantener eleccion: 33.5%
    Cambiar puerta: 66.5%

Tiempo total:: 3.08 Segundos
```