Monty Hall

Sebastián Martínez, Juan Darracq, Diego Handalian

PROBABILIDAD Y ESTADÍSTICA APLICADA

05/04/2024

Sebastián Martínez, Juan Darracq, Diego Handalian

Introducción

El problema de Monty Hall es un problema de probabilidad inspirado en un viejo concurso de televisión llamado "Let 's Make a Deal". El problema consiste en que un concursante tiene delante tres puertas, detrás de una de ellas hay un auto nuevo y en las otras dos hay cabras, el concursante debe adivinar en cuál puerta está el vehículo.
La pregunta que surge en este problema es ¿Cambiar de puerta puede influir en las chances del concursante de ganar el auto?
En este informe, vamos a profundizar en este problema clásico. En primer lugar, explicaremos el marco teórico detrás de este problema. Además, presentaremos una solución mediante una simulación del problema de Monty Hall utilizando un programa en Python. Esta herramienta nos permitirá no solo comprender mejor el problema, sino también responder a las preguntas establecidas y analizar las estrategias de cambio y no cambio de puerta.
Por último, realizaremos una conclusión de acuerdo a los resultados obtenidos en la simulación.

Marco teórico

La teoría de la probabilidad es esencial para entender el problema de Monty Hall. Algunos conceptos clave incluyen el espacio muestral, eventos y eventos aleatorios . El espacio muestral representa el conjunto de todos los posibles resultados de un experimento aleatorio. Un evento es un subconjunto del espacio muestral. Un evento aleatorio es un suceso cuyo resultado no puede ser predicho con certeza, como el lanzamiento de una moneda o un dado.
Según la definición axiomática de probabilidad trabajada en clase, dada un espacio muestral, una probabilidad es una asignación de números entre 0 y 1 para cada evento del espacio muestral. Estas asignaciones deben cumplir con dos propiedades fundamentales:

1) La probabilidad del espacio muestral es igual a 1.
2) Si S1,S2,... es una colección infinita de eventos incompatibles dos a dos entonces:

$$P\left(\bigcup_{n=1}^{\infty} S_n\right) = \sum_{n=1}^{\infty} P(S_n)$$

De esa definición también sacamos las siguientes propiedades:

- $P(\emptyset) = 0$
- $P(A \cup B) = P(A) + P(B)$ cuando A y B son incompatibles
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(A^c) = 1 - P(A)$

En este problema nos gustaría medir cuál es el porcentaje de veces que ocurre un evento comparado con el total, para calcular eso tenemos la frecuencia relativa que es la proporción de veces que ocurre un evento en relación con el total de eventos observados.

La fórmula para calcular la frecuencia relativa de un evento específico es:

> **Frecuencia Relativa** = *número de veces que ocurre el evento / Total de observaciones en el conjunto de datos*

"Total de observaciones en el conjunto de datos" es el número total de datos u observaciones en el conjunto.

Funcionamiento del juego

Al concursante se le ofrece la posibilidad de escoger una entre tres puertas. Tras una de ellas se encuentra un coche, y tras las otras dos hay sendas cabras. El concursante gana el premio que se oculta detrás de la puerta que escoja.

Después de que **el concursante escoja una puerta**, **el presentador abre una de las otras dos puertas, mostrando una cabra.** Siempre puede hacerlo, ya que incluso si el concursante ha escogido una cabra, queda otra entre las puertas que ha descartado y el presentador conoce lo que hay detrás de cada puerta.

Entonces, ofrece al concursante la posibilidad de cambiar su elección inicial y escoger la otra puerta que descartó originalmente, que continúa cerrada o continuar con su elección inicial.

Explicación Del Código

Al comienzo del script se le pide al usuario que ingrese si quiere simularlo cambiando de puerta, simularlo sin cambiar de puerta o jugar manualmente. Además, se tiene un array o lista llamado puertas que representa las puertas 1, 2 y 3

```python
import random
def monty_hall():
    print("Ingrese:\n- 'True' para simular 100000 intentos cambiando de puerta\n- 'False' para simular 100000 intentos sin cambiar de puerta\n- 0 para jugar tú")
    answer = input()
    doors = [1, 2, 3]

    if answer == "0":
        chosen_door = input("Elige una puerta (1, 2, o 3): ")
        print("Has elegido la puerta " + chosen_door)
    else:
        answer = bool(answer)
        simulate_monty_hall(answer)
        exit()
```

Imagen 1

Jugar Manualmente

En caso de decidir jugar manualmente, el algoritmo utiliza la librería random para generar un número entre 1 y 3 que será la puerta donde se encuentra el coche. En el caso donde el usuario eligió la puerta correcta, se elimina del array el número de puerta. Si no eligió la puerta correcta se quita del array la puerta elegida y la puerta ganadora. Esto se hace para que el presentador elija la puerta donde no hay auto y la puerta que el jugador no eligió. Si el jugador eligió la puerta correcta, en el array estarían las otras dos puertas con las cabras, entonces el presentador elegirá entre una de esas dos, Si el jugador no tomo la puerta correcta, en el array solo estaría la puerta con cabra que no fue elegida y esa sería la puerta del presentador

```python
correct_door = random.randint(1, 3)

if int(chosen_door) == correct_door:
    doors.remove(int(chosen_door))
else:
    doors.remove(int(chosen_door))
    doors.remove(correct_door)
presenters_door = random.choice(doors)
print("El presentador muestra la puerta " + str(presenters_door))
```

Imagen 2

Se le pregunta si quiere cambiar de puerta al jugador, si decide no cambiar de puerta sé compara la puerta elegida con la puerta correcta. En caso de querer cambiar, el array de doors se resetea y se quita la puerta elegida inicialmente y la puerta que mostró el presentador, entonces la puerta nueva del jugador sería la puerta que quedo en la lista. Una vez cambiado o no, se comparan la puerta elegida con la puerta correcta, si son las mismas el jugador gano y si son distintas perdió.

```python
change_door = input("Decide cambiar de puerta (y/n): ")
if change_door == "y" or change_door == 1:
    doors = [1, 2, 3]
    doors.remove(int(chosen_door))
    doors.remove(presenters_door)
    chosen_door = doors[0]
    print("Has decidido cambiar a la puerta " + str(chosen_door))
print("El auto se encuentra en la puerta " + str(correct_door))
if int(chosen_door) == correct_door:
    print("GANASTE EL AUTO!")
else:
    print("PERDISTE!")
```

Imagen3

Simulación

Se guarda en variables la cantidad de victorias, derrotas e intentos, mientras que el número de intentos sea mayor a 0, se simula. Igual que cuando el jugador juega manualmente, se guarda en una lista las puertas y se randomiza entre 1 y 3 la puerta correcta y en este caso también la puerta elegida. Si la puerta elegida es distinta a la puerta correcta, se quitan ambas del array y si son iguales se quita la puerta elegida por el jugador de la lista, quedando las puertas con cabras. Luego el presentador elige una puerta de la lista de puertas, donde solo están las puertas con cabras.

```python
def simulate_monty_hall(decision):
    wins = 0
    losses = 0
    attempts = 100000
    while attempts > 0:
        doors = [1, 2, 3]
        chosen_door = random.randint(1, 3)
        correct_door = random.randint(1, 3)

        if chosen_door != correct_door:
            doors.remove(chosen_door)
            doors.remove(correct_door)
        else:
            doors.remove(chosen_door)

        presenters_door = random.choice(doors)
```

Imagen 4

Si se decidió simular cambiando de puerta, se pone en la lista de puertas las 3 y se quita la puerta elegida y la puerta que mostró el presentador. Luego se le asigna al jugador la puerta que quedo en la lista. En caso de decidir no cambiar, esto no se realiza y se compara entre la puerta elegida y la puerta correcta

```python
if decision == True:
    doors = [1, 2, 3]
    doors.remove(chosen_door)
    doors.remove(presenters_door)
    chosen_door = doors[0]

if chosen_door == correct_door:
    wins += 1
else:
    losses += 1
attempts -= 1
```

Imagen 5

Analisis de la solución:

Elección de simular 1000,10000 y 100000 sin cambiar de puerta:

```
- 'False' para simular 100000 intentos sin cambiar de puerta
- 0 para jugar tú
False
Cantidad de veces ganadas sin cambiar de puerta en 1000 intentos: 343
Frecuencia Relativa: 0.343
Cantidad de veces ganadas sin cambiar de puerta en 10000 intentos: 3290
Frecuencia Relativa: 0.329
Cantidad de veces ganadas sin cambiar de puerta en 100000 intentos: 33276
Frecuencia Relativa: 0.33276
```

Imagen 6

Podemos visualizar una frecuencia relativa en todas las simulaciones de alrededor de 0,33.

Elección de simular 1000,10000 y 100000 cambiando de puerta :

```
Ingrese:
- 'True' para simular 100000 intentos cambiando de puerta
- 'False' para simular 100000 intentos sin cambiar de puerta
- 0 para jugar tú
True
Cantidad de veces ganadas cambiando de puerta en 1000 intentos: 677
Frecuencia Relativa: 0.677
Cantidad de veces ganadas cambiando de puerta en 10000 intentos: 6674
Frecuencia Relativa: 0.6674
Cantidad de veces ganadas cambiando de puerta en 100000 intentos: 66565
Frecuencia Relativa: 0.66565
```

Imagen 7

Podemos visualizar una frecuencia relativa en todas las simulaciones de alrededor de 0,66.

Explicación de la Solución

¿Cuánto vale aproximadamente la probabilidad de que el participante gane el auto con la estrategia de cambio de puerta? ¿Cuánto vale la misma probabilidad con la estrategia donde no se cambia la puerta?

Para explicar esta solución vamos a definir el espacio muestral:

coche, cabra u otra cabra

Eventos:

A: Participante gana con la estrategia donde no se cambia la puerta

B : Participante gane el auto con la estrategia de cambio de puerta

Ganar sin cambiar de puerta:

Como hay tres puertas y solo detrás de una se encuentra el coche la

P[A] = 1/3

Ganar cambiando de puerta:

Como el presentador siempre abre una puerta la cual tiene una cabra podemos definir todos los casos posibles que ocurren cambiando de puerta. Para ejemplificar este problema vamos a decir que siempre elige en primer lugar la puerta 1 y después cambia a la puerta que no abre el presentador.

Presentador abre la puerta = subrayado

Situación	puerta 1	puerta 2	puerta 3	respuesta
1)	cabra	cabra	coche	**gana**
2)	cabra	coche	cabra	**gana**
3)	coche	cabra	cabra	**pierde**

- En la situación 1 el concursante elige en primer lugar la puerta 1, el presentador abre la puerta 2 que tiene una cabra, como esta situación simula todos los casos que el participante cambia puerta, el participante elige la puerta 3, y gana el auto.

- En la situación 2 el concursante elige en primer lugar la puerta 1, el presentador abre la puerta 3 que tiene una cabra, como esta situación simula todos los casos que el participante cambia puerta, el participante elige la puerta 2 , y gana el auto.

- En la situación 3 el concursante elige en primer lugar la puerta 1, el presentador abre la puerta 3 que tiene una cabra, como esta situación simula todos los casos que el participante cambia puerta, el participante elige la puerta 2, y pierde.

Se ve claramente que el participante gana dos de tres veces entonces P[B]=2/3
Comparando las dos estrategias vemos que si el participante no cambia de puerta tiene un 33,3% de ganar, mientras que sí cambia de puerta un 66,6% de ganar.
Para comprender esto desde una forma más Matemática.
Vamos a plantear la situación A: abrir la puerta tras la cual está el coche.En este problema contamos con 3 puertas.
Para eso utilizamos esta fórmula:

$$P[A] = 100 \times \frac{n+l}{t}\%$$

Donde:

- P[A] la probabilidad de abrir la puerta tras la cual está el coche.
- n, es el número de puertas elegidas al principio.
- l, es el número de puertas en las que se ha enseñado que dentro no hay nada.
- t, es el número total de puertas.

Entonces para 3 puertas
tenemos que si la probabilidad de abrir la puerta la cual está el coche sin cambiar de puerta es:

n=1
I=0
t=3

$$100 \times \frac{1+0}{3}$$

= 33,3%

Mientras que la probabilidad de abrir la puerta la cual está el coche cambiando de puerta es:
n=1
I=1
t=3

$$100 \; * \; \frac{1+1}{3} \; = \; 100 \; * \frac{2}{3} = 66,6\%$$

Conclusión

En conclusión, para la respuesta de si cambiar de puerta puede influir en las chances del concursante de ganar el auto, la respuesta es sí, las chances de ganar cambiando de puerta son mayores que las chances de ganar no cambiando de puerta.
A lo largo de este informe fuimos desarrollando conceptos y definiciones así como la aplicación de una simulación de este problema. Una de las enseñanzas clave que hemos internalizado es la comprensión de los conceptos probabilísticos y cómo se aplican a los problemas de la vida real.
Además, el proceso de fabricación del código de simulación nos ha enseñado cómo entender y modelar un problema de manera efectiva. Al descomponer el problema en sus componentes básicos, identificar las variables relevantes y diseñar un algoritmo para simularlo, hemos adquirido habilidades importantes en la resolución de problemas y la modelización matemática. Sin embargo, alguno de los conceptos que tuvimos más problemas en abarcar era de cómo explicar la solución de manera clara. Si bien elegimos una explicación de la solución de tipo elemental y también mediante una fórmula fácil de entender , existen más soluciones que incluyen otros conceptos de probabilidad como la probabilidad condicional, sin embargo como no la hemos trabajado en clase decidimos no ir por ese lado.

Referencias

https://es.wikipedia.org/wiki/Problema_de_Monty_Hall

https://www.math4all.es/el-problema-de-monty-hall/

https://docs.python.org/es/3/library/index.html