

Actividad Extra 1 - *En los tiempos del Coronavirus*

La ruina del Jugador

Exactas Programa

Verano 2020

1. La Ruina del Jugador

El problema de la ruina del jugador: Se trata de un desafío entre un jugador y el casino que consta de un indeterminado número de jugadas. Tanto el jugador como el casino cuentan con una cantidad inicial de monedas. En cada jugada, se lanza un dado (que se usa para definir el resultado) y quién pierde la jugada entrega una moneda al otro jugador. La partida concluye cuando uno de los dos participantes ha perdido todas sus monedas. De Santos et al. (2008)

1.1. Ejemplo

Supongamos que inicialmente Juan dispone de $j = 3$ monedas, mientras que el Casino tiene $c = 2$ monedas. Es decir, hay un total de $m = j + c$ monedas en juego. Cada jugada consiste en lanzar un dado; para este ejemplo, usaremos que si sale un cuatro, gana Juan; caso contrario, gana el Casino. Es decir, en cada jugada, con probabilidad $p = 1/6$ Juan gana una moneda del Casino, mientras que con probabilidad $q = 1 - p = 5/6$ el Casino gana una moneda de Juan. La partida continúa hasta que uno de los actores se queda sin monedas (esa es su ruina), mientras que su contrincante es quien gana la partida.

Lo que nos interesa saber es ¿cuál es la probabilidad de que Juan gane la partida?

2. Ejercicio

Proponemos este problema para explorar diferentes aspectos de la programación, como se sugiere a lo largo de la siguiente lista (tenga presente la diferencia entre *partida* y *jugada*):

1. Implemente una función `una_jugada` que dado p simule una jugada y devuelva `True` si ganó Juan o `False` en caso contrario.
2. Simule una jugada con $p = 1/6$. Repita $n_rep = 1000$ veces. ¿Qué proporción de veces gana Juan la jugada?
3. Calcule $n_rep = 1000$ veces la función `una_jugada` para los siguientes valores de p : 0.2, 0.5, 0.8. ¿Qué proporción de veces gana Juan en cada caso?
4. Implemente una función `juan_se_arruina` que dados j (cantidad de monedas de Juan), m (cantidad total de monedas) y p (probabilidad de que Juan gane una jugada) simule una partida completa y devuelva `True` si Juan se arruina (pierde todas las monedas) y `False`, caso contrario.
5. Simule $n_rep=1000$ partidas para $j=3$, $m=5$ y $p=1/6$. Calcule la proporción de veces en las que Juan gana.
6. Implemente una función `estimacion_juan_gana` que dados j , m , p , n_rep , simule n_rep partidas con los parámetros que se le pasen y devuelva la proporción de veces en que gana Juan.
7. Fije $m = 5$ y $p = 1/6$. Para todos los posibles valores de j , es decir $0, 1, \dots, 5$, simule $n_rep=1000$ partidas y, en cada caso, calcule la proporción de veces en que gana Juan.

8. Grafique la proporción de veces en que Juan gana (eje y) cuando $p = 1/6$ y $m = 5$ en función de j (eje x), para $0 \leq j \leq m = 5$. Repita para $p = 0,5$ y $p = 0,8$.
9. Fije $p = 0,5$. Grafique la probabilidad estimada de que Juan gane en función de j cuando $m = 10$, para todos los posibles valores $0 \leq j \leq m$.
10. Repita para $m = 20$, $m = 30$, $m = 50$.
11. Para los valores de $p = 1/6$, $1/2$, $4/5$ y los valores de $m = 10$, 20 , 30 , 50 , combine estos valores y grafique la probabilidad estimada de que Juan gane en función de j (para todos los j que cumplan con $0 \leq j \leq m$).
12. (*Optativo*) Extender el análisis para además calcular el largo de las partidas con los diferentes parámetros, ¿se puede observar algún patrón en las partidas que gana Juan? ¿Y en las que gana el Casino?

Comandos útiles

Módulo random

- `random.random()`: da un número aleatorio con distribución uniforme entre el 0 y 1.

Módulo matplotlib

- `import matplotlib.pyplot as plt` para cargar el módulo
- `plt.plot(v1, v2)` para agregar el gráfico los datos de la lista `v1` contra los de `v2`
- `plt.show()` para mostrar el gráfico

Referencias

Santos, J. B., Ruiz, J. A. C., & Hidalgo, M. D. P. (2008). El problema de la ruina del jugador. Suma: Revista sobre Enseñanza y Aprendizaje de las Matemáticas, (59), 23-30.