3. Implementar la función `cuantos_paquetes(figus_total, figus_paquete)` que dado el tamaño del álbum (`figus_total`) y la cantidad de figuritas por paquete (`figus_paquete`) simule el llenado del álbum y retorne cuántos paquetes se debieron adquirir para completarlo.
4. Ejecutar `n_rep=100` veces la función `cuantos_paquetes`, utilizando `figus_total=640`, `figus_paquete=5` y guardando los resultados obtenidos en un vector y calcular el promedio.
5. *Optativo*: Utilizando lo implementado estimar la probabilidad de completar el álbum con 850 paquetes o menos.
6. *Optativo*: Utilizando lo implementado en estimar cuántos paquetes se deben comprar para tener una chance del 90% de completar el álbum. ‘

5 Modelado: La ruina del jugador

Esta es la guía de ejercicios correspondiente a la clase 05. Deberá entregar un archivo .R con las funciones implementadas y un informe sobre la exploración del problema en formato .Rmd y .pdf.

En esta oportunidad presentamos la última actividad del curso. Es un nuevo problema para seguir programando y aprovechamos la oportunidad para invitar a explorar *algunas bondades* de R, a la hora de escribir un informe. Por un lado, queremos que resuelvas la guía armando, en un archivo .R, todas las funciones que vas a utilizar a la hora de escribir un informe sobre el problema, pensando en un estudiante de la escuela media. Incluímos algunas pautas para ayudar en la elaboración del informe:

- Contar el problema.
- Mostrar resultados correspondientes a diferentes experimentaciones incluyendo gráficos y especificando que se representa en cada caso.
- Explorar en detalle el caso $p = 0.5$
- Conjeturar una fórmula en función de las pruebas realizadas.
- Incluir referencias.

5.1 Comandos útiles

- `sample(x, size, replace = FALSE, prob = NULL)`
- `runif(1)`
- `rep(x, ...)`

5.2 Recursos para elaborar el informe

Algunos recursos que pueden explorar:

- Tutorial interactivo online
- Introducción
- QuickTour
- Hoja de trucos

5.3 La ruina del jugador

Se trata de un desafío entre un jugador y el casino que consta de un indeterminado número de jugadas. En cada jugada, se lanza una moneda 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).

A modo de 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; 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.

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 `FALSE` si Juan gana y `TRUE`, 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 y `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 $j \in \{0, 1, \ldots, m\}$, 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 $0 \leq j \leq m$. Repita para $m = 20$, $m = 30$, $m = 50$. ¿Qué forma observa en cada uno de los gráficos?
10. Conjeturar una fórmula para la probabilidad de que Juan gane cuando $p = 0.5$, en función de j y de m, la cantidad inicial y total de monedas.
11. Quiero más: 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 $0 \leq j \leq m$).

5.3.1 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. ‘