

Álbum de figuritas del mundial

Julián Zylber

2022

Las figuritas del mundial

El objetivo de esta actividad será realizar un programa en **Python** que responda la pregunta:
¿Cuántos paquetes hay que comprar para completar el álbum del Mundial?

Datos

- Álbum con 669 figuritas.
- Cada figurita se imprime en cantidades iguales y se distribuye aleatoriamente.
- Cada paquete trae cinco figuritas.

El modelo del álbum de figuritas

Vamos a representar un álbum de n figuritas utilizando una lista de n posiciones numeradas de 0 a $n - 1$. Cada posición representa el estado de una figurita con dos valores :

- 0 para indicar que no la tenemos
- 1 para indicar que sí

Por ejemplo: Si tuviéramos un álbum de seis figuritas vacío vamos a representar como $[0, 0, 0, 0, 0, 0]$. Con consigamos la figurita 4 tendremos que indicar poniendo un 1 en el cuarto lugar de la lista, es decir $\text{album}[3] = 1$ y el álbum nos va a quedar $[0, 0, 0, 1, 0, 0]$

Modelo Preliminar

Suponga que las figuritas se compran **individualmente** (de a una, no en un paquete con varias). En este caso, la dinámica del llenado es la siguiente:

1. Iniciamos con un álbum vacío y sin haber comprado ninguna figurita
2. Compramos figuritas (de a una) hasta llenar el álbum; es decir, se repite la acción (*el paso*) de comprar figuritas *mientras* el álbum este incompleto
3. Al terminar nos interesa saber cuántas figuritas tuvimos que comprar para llenar el álbum

Ejercicios

1. Implemente la función `comprar_figu(figus_total)` que recibe el número total de figuritas que tiene el álbum (dado por el parámetro `figus_total`) y devuelve un número entero aleatorio que representa la figurita que nos tocó. **Recordar las funciones vistas en la presentación**
2. Implementen la función `esta_lleno(album)` que dado un álbum representado por una lista (`album`), devuelva `True` si está lleno, `False` si no¹. Hay muchas maneras de hacer esto, ¿Pueden pensar otra? **Responder en la hoja de respuestas, no dando una implementación sino otra/s idea/s**
3. Ahora, implementen la función `cuantas_figus(figus_total)` que, dado el tamaño del álbum (`figus_total`), simule su llenado y devuelva la cantidad de figuritas que se debieron adquirir para completarlo.
4. Para el punto que sigue, necesitamos calcular promedios. Para eso, crearemos la función `promedio(lista)` que dada una lista de números, devuelve la media aritmética. **Recordemos: la media aritmética calcula sumando todos los valores y dividiéndolos por la cantidad.**
5. Calculen `n_repeticiones=1000` veces la función `cuantas_figus` utilizando `figusTotal=6` y guarden en una lista los resultados obtenidos en cada repetición. Con los resultados obtenidos estimen cuántas figuritas hay que comprar, en promedio, para completar el álbum (de seis figuritas) **Responder en la hoja de respuestas.**
6. Calculen `n_repeticiones=100` veces la función `cuantas_figus(figus_total=669)` y guarden los resultados obtenidos en cada repetición en una lista. Con los resultados obtenidos estimen cuántas figuritas hay que comprar, en promedio, para completar el álbum (de 669 figuritas). **Responder en la hoja de respuestas**

¹Los valores booleanos, no texto

Modelo avanzado: ahora con paquetes

Recordar:

- Cada paquete tiene 5 figuritas
- Puede tener repetidos, igual que en la vida real.

Ejercicios

1. Pensar y describir una forma de representar en Python los paquetes descritos anteriormente.
2. Implementen una función `generar_paquete(figus_total, figus_paquete)` que, dado el tamaño del álbum (`figus_total`) y la cantidad de figuritas por paquete (`figus_paquete`), genere un paquete de figuritas al azar.
3. Implementen una función `cuantos_paquetes(figus_total, figus_paquete)` que dado el tamaño del álbum simule el llenado del álbum y devuelva cuántos paquetes se debieron adquirir para completarlo.
4. Calculen `n_repeticiones=100` veces la función `cuantos_paquetes`, utilizando `figus_total=669` y `figus_paquete=5` y guarden los resultados obtenidos en una lista. Luego, calculen su promedio.
Responder en la hoja de respuestas
5. Utilizando lo implementado en el ítem anterior, estimen la probabilidad de completar el álbum con 850 paquetes o menos. Recordar que la probabilidad de un evento se estima como:

$$\frac{\text{veces que ocurrió evento}}{\text{total de eventos}}$$

Responder en la hoja de respuestas

Extras: por más nota

Las siguientes consignas dan puntos extras, que se suman a la nota básica por entregar el trabajo. No se puede superar el 10

- **(1 punto)** Utilizando funciones de `numpy` y lo obtenido en el ítem 4, estime cuántos paquetes debe comprar para tener una chance del 90% de completar el álbum.
- **(1 punto)** Modificar la generación de paquetes, de forma que no haya figuritas repetidas por paquete, y no haya más de un escudo por paquete (hay 32 escudos, es lo mismo que 32 figuritas representan a dichos escudos).
- **(2 puntos)** Complejizar aún más el modelo, incluyendo la idea de que haya otro coleccionista y se guarden las repetidas. Cada vez que ambos compran un paquete, ven si pueden intercambiar figuritas repetidas, es decir, si jugador A tiene una figurita repetida que B no tiene, y viceversa, intercambian, sino no. No se pueden regalar figuritas. Recalcular el punto 5 con este modelo. **(1 punto más)** Aumentar ahora la cantidad de coleccionistas, y que esta sea ahora un parámetro de la función. Armen el intercambio de figuritas como para que el jugador que arranque a intercambiar sea elegido al azar, y con quien arranque también.
- **(2 puntos)** Complejizar aún más el modelo, haciendo que haya figuritas mas comunes que otras. Fíjense como esto afecta al promedio de paquetes necesarios para completar el álbum. **TIP:** Usen otra función que no sea `randint`, o adapten a `randint` para que tome en cuenta rarezas. **(1 punto más)** Busquen alguna distribución conocida mediante `scipy` para las figuritas.
- **(2 puntos)** Sin usar `sort`, u otras funciones de ordenamiento de librerías, calcular la mediana obtenida en el ítem 4. **Tip:** Para calcular la mediana necesitamos ordenar los casos que obtuvimos. ¿Como hacemos eso? Consultarme si no se les ocurre nada (no me pueden consultar sin haberlo discutido un rato antes).

Modo de evaluación

- La entrega mínima son los ejercicios del modelo preliminar y el modelo avanzado. Con eso se alcanza al 6-7.
- Los ejercicios extras permiten subir esta nota. Solo serán considerados si la entrega mínima está completa y correcta.
- No solo se considerará que el código funcione, sino que también se considerarán las buenas prácticas de programación:
 - Nombres de variables representativos y apropiados
 - Nombres de funciones representativos y apropiados
 - Código comentado **solo cuando sea necesario**. No hagan un *text wall* de comentarios, el código debería ser autoexplicativo si los nombres son apropiados. Solo en situaciones mas confusas comentenlo.
- Muchos items tienen consignas para ser respondidas. Recuerden responderlas en un archivo de texto aparte.
- Este trabajo es **INDIVIDUAL**. Se pueden consultar ideas, pero no código. ¡Miren que voy a correr un software de plagio!

Entrega

Se entrega el código realizado en un archivo `.py`, y una hoja de respuestas en un archivo de texto. Los mismos deben ser subidos en el campus.

Conocimientos Mínimos

Para hacer esta trabajo, necesitan saber los siguientes conceptos y su sintaxis de `python`.

- Variables, declaración y modificación
- Condicionales, sintaxis `if,elif,else`
- Ciclos, ya tanto `for` como `while`
- Listas: acceso, iteración, declaración, `append` y `pop`
- Importación y uso de bibliotecas.

¡A programar!