2.1 - GENERADORES PSEUDOALEATORIOS

Daniela Ramirez
danielaramirez.ros@gmail.com

Juan Franco Petrelli
jfpetrelli@gmail.com

Joan Romero
joanromerosfc@gmail.com

Guido Ojeda
guidociclon07@gmail.com

Fernando Gómez
ferg2595@gmail.com

7 de mayo de 2022

ABSTRACT

El siguiente informe tiene como finalidad introducirmos en el mundo de los generadores de números pseudoaleatorios. Para ello analizaremos las características de distintos generadores y a través de diversos test estadísticos comprobaremos la calidad de los mismos.

Keywords generadores pseudoaleatorios simulación GCL

Índice

1. Introducción **3**

2. Verdaderos números aleatorios **3**

3. Números pseudoaleatorios **3**

3.1. Generadores 4

3.1.1. Generador congruencial lineal 4

3.1.2. Generador media del cuadrado 4

4. Pruebas de aleatoriedad **4**

4.0.1. Prueba Chi-Cuadrado 4

4.0.2. Prueba de independencia corridas de arriba y abajo de la media 5

4.0.3. Prueba de autocorrelación 5

4.0.4. Prueba Póquer 5

5. Comprobación de resultados **6**

5.1. Método de la parte media del cuadrado 6

5.2. Generador congruencial lineal 7

5.3. Generador python 9

6. Conclusiones **10**

7. Anexo **10**

7.1. Valores obtenidos 10

7.2. Codigo Python 10

1. Introducción

Un número aleatorio es aquel obtenido al azar, es decir, que todo número tenga la misma probabilidad de ser elegido y que la elección de uno no dependa de la elección del otro. El ejemplo clásico más utilizado para generarlos es el lanzamiento repetitivo de una moneda o dado ideal no trucado.
Los números aleatorios permiten a los modelos matemáticos representar la realidad. En general cuando se requiere una impredecibilidad en unos determinados datos, se utilizan números aleatorios
Los seres humanos vivimos en un medio aleatorio y nuestro comportamiento lo es también. Si deseamos predecir el comportamiento de un material, de un fenómeno climatológico o de un grupo humano podemos inferir a partir de datos estadísticos. Para lograr una mejor aproximación a la realidad nuestra herramienta predictiva debe funcionar de manera similar: aleatoriamente. De esa necesidad surgieron los modelos de simulación.
En la vida cotidiana se utilizan números aleatorios en situaciones tan dispares como pueden ser los juegos de azar, en el diseño de la caída de los copos de nieve, en una animación por ordenador, en tests para localización de errores en chips, en la transmisión de datos desde un satélite o en las finanzas.

Un experimento aleatorio de este tipo tiene ciertas caracteristicas como:

- No poder determinar el resultado particular que ocurrirá, pero si describir el conjunto de los resultados posibles.

- Después de un gran numero de repericiones de la experiencia aleatoria, existe una distribución regular de los resultados. Es decir, a medida que el experimento se repite, los resultados parecen ocurrir de manera incierta, sin embargo, ante un gran número de repeticiones aparece un modelo definido de regularidad. Esta regularidad hace posible la construcción de un modelo matemático que permite el análisis del experimento.

2. Verdaderos números aleatorios

En general los números aleatorios se basan en alguna fuente de aleatoreidad física que puede ser teóricamente impredecible (cuántica) o prácticamente impredecible (caótica). Por ejemplo:

- random.org genera aleatoreidad a través de ruido atmosférico (el paquete random contiene funciones para obtener números de random.org),

- ERNIE, usa ruido térmico en transistores y se utiliza en la lotería de bonos de Reino Unido.

- RAND Corporation En 1955 publicó una tabla de un millón de números aleatorios que fue ampliamente utilizada. Los números en la tabla se obtuvieron de una ruleta electrónica.
La desventaja de éstos métodos es que son costosos, tardados y no reproducibles.

3. Números pseudoaleatorios

Los números pseudoaleatorios son generados por medio de una función (determinista, no aleatoria) y que aparentan ser aleatorios. Estos números pseudoaleatorios se generan a partir de un valor inicial aplicando iterativamente la función. La sucesión de números pseudoaleatorios es sometida a diversos tests para medir hasta qué punto se asemeja a una sucesión aleatoria .

3.1. Generadores

3.1.1. Generador congruencial lineal

Un generador congruencial lineal (GCL) es un algoritmo que permite obtener una secuencia de números pseudoaleatorios calculados con una función lineal definida a trozos discontinua. Es uno de los métodos más antiguos y conocidos para la generación de números pseudoaleatorios. La teoría que sustenta el proceso es relativamente fácil de entender, el algoritmo en sí es de fácil implementación y su ejecución es rápida.

Los GCL no deberían ser usados en aplicaciones para las que se requiera aleatoriedad de alta calidad. Por ejemplo, esta técnica es inadecuada para el uso en una simulación de Monte Carlo debido a la correlación serial de la secuencia (entre otros motivos). Tampoco deberían usarse para aplicaciones criptográficas; ejemplos de generadores más adecuados para esta función se pueden encontrar en Generador de números pseudoaleatorios criptográficamente seguro. Si un GLC es sembrado con un carácter e iterado una única vez, el resultado es un sencillo cifrado afín clásico, el cual puede ser descifrado con un análisis de frecuencia estándar.

Los Generadores Congruenciales Lineales (GCL) tienen la forma

$$X_{n+1} = (aX_n + c)mod(m) \tag{1}$$

Están determinados por los parámetros:

- $Mdulo : m > 0$
- $Multiplicador : 0 \leq a < m$
- $Incremento : c \leq m$
- $Semilla : 0 \leq X_0 < m$

3.1.2. Generador media del cuadrado

Este método fue propuesto en los años 40 por los matemáticos John von Neumann y Nicholas Metropolis. El método comienza tomando un número al azar, x_0, de $2n$ cifras (originalmente los autores proponían 4 cifras) que al elevarlo al cuadrado resultara un número de hasta $4n$ cifras. Si es necesario se añaden ceros a la izquierda para que el número resultante tenga exactamente $4n$ cifras. Sea x_1 el número resultante de seleccionar las $2n$ cifras centrales de x_0^2; el primer número aleatorio u_1 se obtiene poniendo un punto decimal delante las $2n$ cifras de x_1. A continuación x_2 y u_2 se generan a partir de x_1 del mismo modo y así sucesivamente. Este método tiene dos inconvenientes principales:

- tiene una fuerte tendencia a degenerar a cero r apidamente (probar por ejemplo con x0 = 1009).
- los números generados pueden repetirse cíclicamente después de una secuencia corta.

4. Pruebas de aleatoriedad

Existen algunos métodos disponibles para verificar varios aspectos de la calidad de los números pseudoaleatorios. Las dos propiedades mas importantes esperadas en los números aleatorios son uniformidad e independencia. La prueba de uniformidad puede ser realizada usando las pruebas de ajuste de bondad disponibles.

4.0.1. Prueba Chi-Cuadrado

La Prueba de Bondad de Ajuste Chi Cuadrado es el test de bondad de ajuste más utilizado. En general un test de bondad de ajuste se utiliza para discriminar si una colección de datos o muestra se ajusta a una distribución teórica de una determinada población. En otras palabras, nos dice si la muestra disponible representa (ajusta) razonablemente los datos que uno espera encontrar en la población.

El test de bondad de ajuste chi cuadrado puede ser utilizado para trabajar tanto con distribuciones discretas como, por ejemplo, la Distribución de Poisson o la Distribución Binomial como así también con distribuciones continuas (por ejemplo, Distribución Normal, Distribución Exponencial, etc).

La aplicación de la prueba de bondad de ajuste chi cuadrado requiere que los datos estén agrupados en categorías o clases. Si los datos originalmente no se encuentran agrupados será necesario agruparlos antes de aplicar el test de chi cuadrado para lo cual sería necesario construir una tabla de frecuencia o histograma.

4.0.2. Prueba de independencia corridas de arriba y abajo de la media

Este procedimiento consiste en determinar una secuencia de unos y ceros de acuerdo a la comparación de cada número r_i que cumpla con la condición de ser mayor a 0.5 (en el caso de los unos) o ser menor a 0.5 (en el caso de los ceros).Luego se determina el número de corridas C_0 y los valores de n_0 y n_1

Valores que se emplean:

C_0= Número de corridas en la secuencia.

n_0= Cantidad de ceros en la secuencia S.

n_1= Cantidad de unos en la secuencia de S.

n = Cantidad de números

$n = n_0 + n_1$

El n se halla de la siguiente manera:

$n = n_0 + n_1$

Posteriormente se calcula el valor esperado, la varianza del número de corridas y el estadístico Z_0 con las siguientes ecuaciones:

Valor esperado:

$\mu_{c0} = \frac{(2n_0n_1)}{n} + \frac{1}{2}$

Varianza del número de corridas:

$\sigma^2_{c0} = \frac{(2n_0n_1-n)}{n^2(n-1)}$

El estadístico:

$Z_0 = \frac{C_0-\mu_{c0}}{\sigma_{c0}}$

Para saber si el estadístico Z_0 está fuera del intervalo se emplea la siguiente fórmula:

$-Z_{\frac{\infty}{2}} \leq Z_0 \leq Z_\infty 2$

Si la condiciónanterior se cumple, entonces se concluye que los números evaluados son independientes, de lo contrario se rechaza al conjunto.

4.0.3. Prueba de autocorrelación

La prueba de autocorrelación valida la correlación entre números aleatorios y los compara con la deseable correlación de cero.

4.0.4. Prueba Póquer

Esta prueba examina en forma individual los dígitos del número pseudoaleatorio generado. La forma como esta prueba se realiza es tomando 5 dígitos a al vez y clasificándolos como : Par, dos pares, tercia, póker quintilla full y todos diferentes. Las probabilidades para cada una de las manos del póker diferentes se muestran enseguida:

Todos diferentes = 0.3024

Un par = 0.504

Dos pares = 0.108

Tercia = 0.072

Full = 0.009

Quintilla = 0.0001

Con las probabilidades anteriores y con el número de números pseudoaleatorios generados, se puede calcular la frecuencia esperada de cada posible resultado.

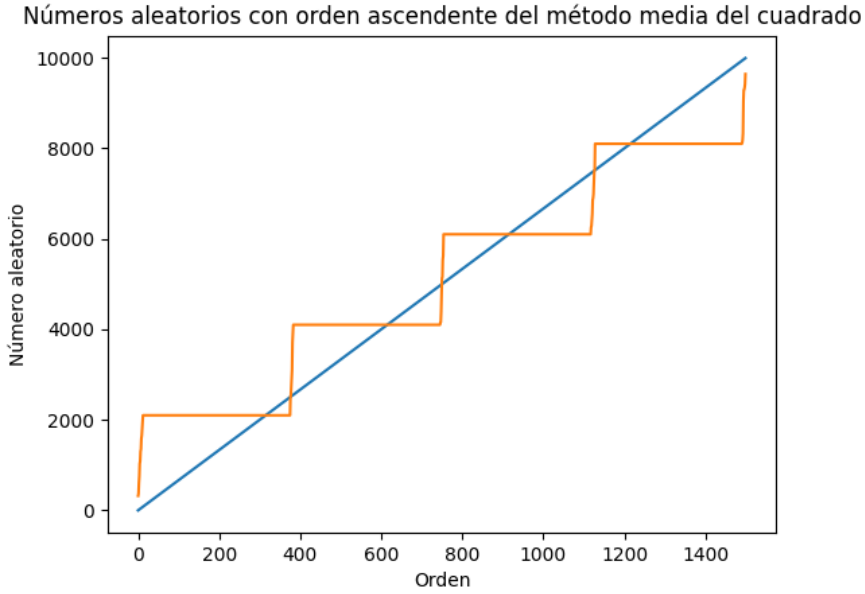
5. Comprobación de resultados

Mediante la aplicación de los distintas pruebas a los distintos generadores obtuvimos los siguientes resultados.

5.1. Método de la parte media del cuadrado

Se define la semilla = 1991 y se generan 1500 números aleatorios.

En la figura 1 graficamos los números generados por el generador ordenados en forma ascendente (Naranja) y lo comparamos con el valor esperado (Azul) lo que nos indicaría que la distribución no es uniforme.



Mediante el la figura 2 podemos observar que tampoco parecen tener una distribución uniforme. Resultados de tests generador parte media del cuadrado
Test chicuadrados: Rechazado
Test de corridas: Rechazado
Test prueba de series: Rechazado

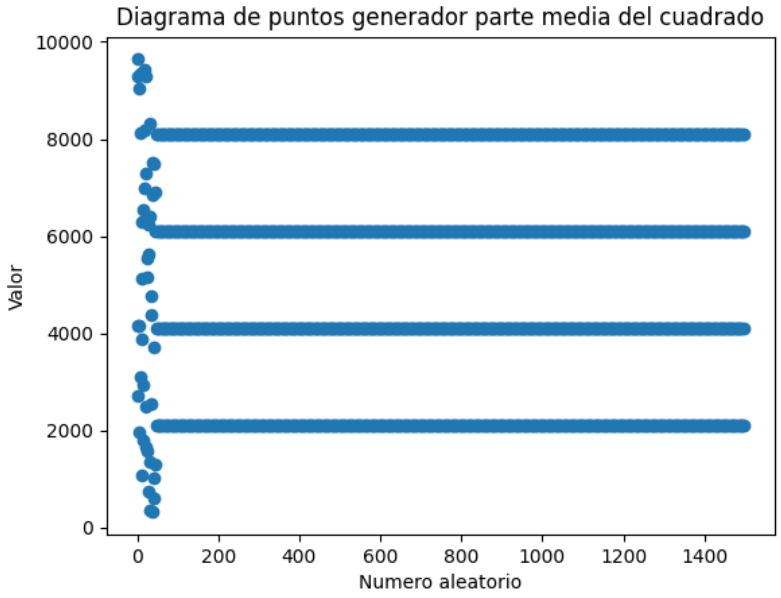
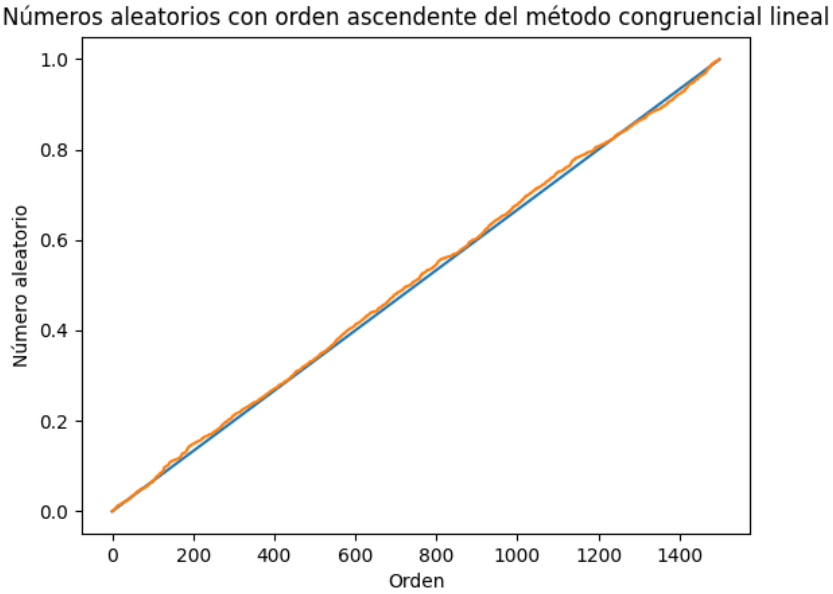


Diagrama de puntos

5.2. Generador congruencial lineal

Se define la semilla = 1991, módulo = 32768, multiplicador = 26765, incremento = 21001 y se generan 1500 números aleatorios.

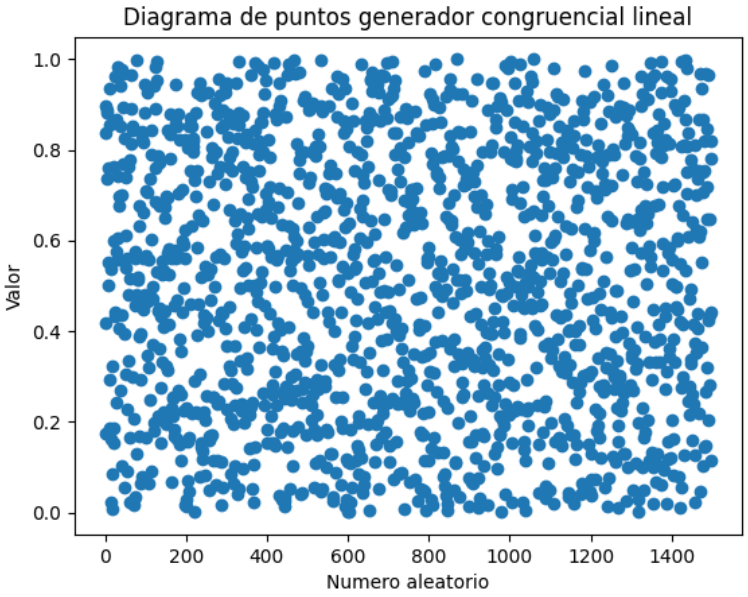
En la figura 3 graficamos los números generados por el generador ordenados en forma ascendente (Naranja) y lo comparamos con el valor esperado (Azul) lo que nos indicaría que la distribución podría ser uniforme.



Números pseudoaleatorios

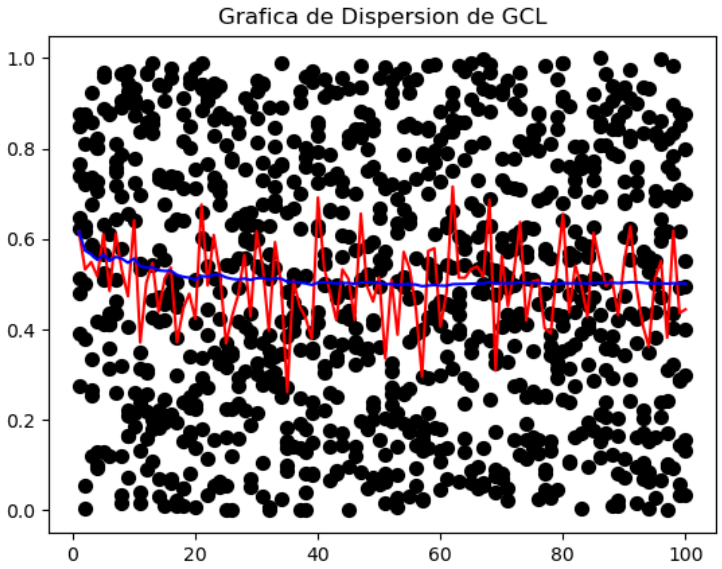
Mediante la figura 4 podemos observar que los números parecen tener una distribución uniforme.

Resultados de tests generador congruencial lineal
Test chicuadrados: Aceptado
Test de corridas: Aceptado
Test prueba de series: Aceptado



[h]

Diagrama de puntos

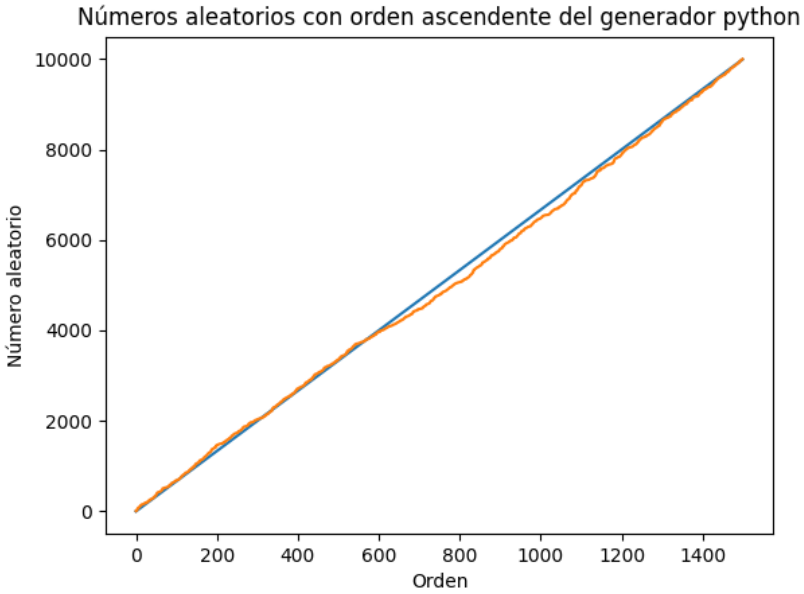


[h]

Diagrama de puntos

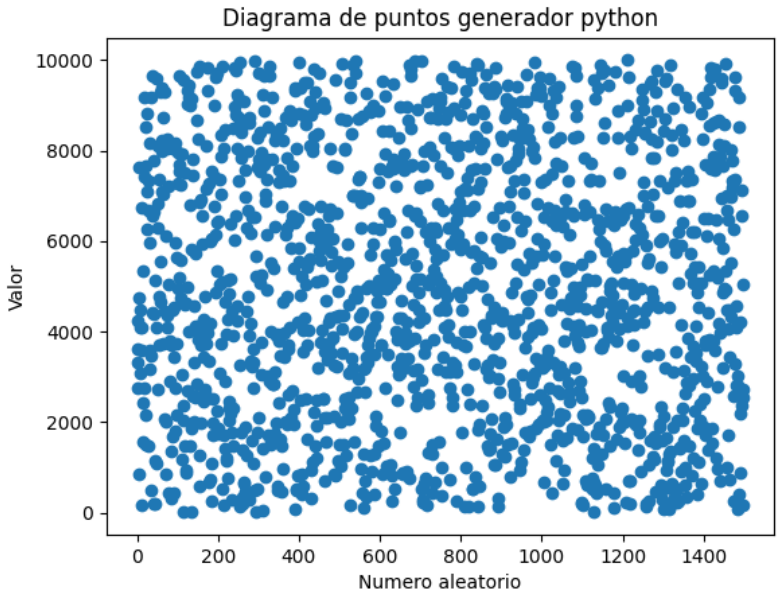
5.3. Generador python

En la figura 5 graficamos los números generados por el generador ordenados en forma ascendente (Naranja) y lo comparamos con el valor esperado (Azul) lo que nos indicaría que la distribución podría ser uniforme.



[h] Números pseudoaleatorios

Mediante la figura 6 podemos observar que los números parecen tener una distribución uniforme.



[h] Diagrama de puntos

Resultados de tests generador python
Test chicuadrados: Aceptado
Test de corridas: Aceptado
Test prueba de series: Aceptado

6. Conclusiones

En base a los experimentos realizados, llegamos a la conclusión que hay generadores con mejor adaptación que otros. La capacidad de diseñar algoritmos que producen numeros aleatorios con caractertsticas tales que pueden representar adecuadamente los verdaderos numerosos aleatorios es lo que permite resolver problemas complejos con el uso de la simulación. Sin embargo. dada la aproximación, es realmente imposible modelar con precisión las caracteristicas de la distribución uniforme mediante un buen generador. Esto quiere decir que algunas propiedades no se cumplirán, las cuales pueden no influir mucho en los resultados de determinado estudio, dando lugar a que el criterio de aceptación de un generador dado debe basarse en la aplicación que se le vaya a dar. Es precisamente responsabilidad del analista realizar las pruebas pertinentes.

7. Anexo

7.1. Valores obtenidos

`https://github.com/jfpetrelli/Simulacion/blob/main/generadores/Resultados.txt`

7.2. Codigo Python

`https://github.com/jfpetrelli/Simulacion/blob/main/generadores/generadores.py`

`https://github.com/jfpetrelli/Simulacion/blob/main/generadores/generadores1.py`

Referencias

[1] Latex - Documentacion
`https://es.overleaf.com/learn`

[2] PyE - Definiciones.
`https://economipedia.com/definiciones`

[3] Graficos en Python.
`https://python-para-impacientes.blogspot.com/2014/08/graficos-en-ipython.html`

[4] MatPlotLib Documentacion.
`https://matplotlib.org/stable/api/index`

[5] Números aleatorios
`https://www.estadisticaparatodos.es/taller/aleatorios/aleatorios.html`

[6] Números pseudoaleatorios
`https://tereom.github.io/est-computacional-2018/numeros-pseudoaleatorios.html`

[7] Video explicativo
`https://www.youtube.com/watch?v=FtEe0IlK6Hc`