
TP 2.2 - GENERADORES DE NÚMEROS PSEUDOALEATORIOS DE DISTINTAS DISTRIBUCIONES DE PROBABILIDAD

Juan I. Torres
Universidad Tecnológica Nacional - FRRO
Zeballos 1341, S2000, Argentina
orkuan@gmail.com

26 de abril, 2022

ABSTRACT

El siguiente documento tiene por objetivo detallar el trabajo que debe realizarse para finalizar con nuestro estudio sobre los números pseudoaleatorios del cual van a depender nuestros próximos experimentos.

Keywords Simulación · generadores pseudoaleatorios · distribuciones de probabilidad

1 Introducción

El estudio anterior a este, demostró que la generación de números pseudoaleatorios no es una tarea sencilla. Por suerte para nosotros es un tema felizmente resuelto para nuestras necesidades. Ahora que tenemos un generador testeado, es hora de empezar a utilizarlo. El problema surge como hacerlo. Si lo empleamos tal y como está, este solo genera valores uniformes continuos entre 0 y 1.

Como nosotros hemos aprendido en la materia Probabilidad y Estadística, hay otras distribuciones de probabilidad tanto continuas como discretas. La pregunta que debemos responder es: ¿cómo hacemos para generar valores de distintas distribuciones?. La buena noticia es que la solución ya ha sido inventada y es nuestra tarea como ingenieros redescubrir el mecanismo para implementarlo en nuestros experimentos.

Por lo tanto el objetivo de este trabajo es construir generadores de números pseudoaleatorios de distintas probabilidades por medio de un material antiguo pero sumamente práctico que es el elaborado por el autor Thomas Naylor[1].

2 Descripción del trabajo

El trabajo consiste en hacer un estudio de las distintas distribuciones de probabilidad, que aunque no son todas las que existen, las mismas abarcan una gran cantidad de los fenómenos de interés para la simulación. De cada una se deben realizar distintas tareas como se detalla a continuación:

- Introducir la distribución de probabilidad en forma teórica; se deben incluir su/s parámetro/s de entrada, función de probabilidad, gráficas, etc.
- Describir matemáticamente el paso a paso para la obtención de su función acumulada e inversa.
- Elaborar un programa por cada distribución de probabilidad en lenguaje Python 3.x.
- Testear la generación de valores de la forma más conveniente para cada caso (queda a criterio del grupo el como testear).

En este trabajo es necesario mostrar el código en el informe, por lo tanto se brinda el siguiente ejemplo de formato para presentarlo en Latex (hay mucha información disponible de como realizar esto):

```

1 from math import *
2
3 # define function
4 def analytic_fibonacci(n):
5     sqrt_5 = sqrt(5);
6     p = (1 + sqrt_5) / 2;
7     q = 1/p;
8     return int( (p**n + q**n) / sqrt_5 + 0.5 )
9
10 # define range
11 for i in range(1,31):
12     print analytic_fibonacci(i)

```

Código 1: Secuencia de Fibonacci calculada analíticamente como ejemplo de inserción de código Python en el informe.

Finalmente para resumir y teniendo en cuenta el orden de temas en la bibliografía seleccionada, se sugiere repasar la siguiente tabla para incorporar el contenido que debe estar en el informe:

Distribución de Probabilidad	Tipo	T. Inversa	M. Rechazo	Código	Testeo
UNIFORME	continua	si	si	si	si
EXPONENCIAL	continua	si	si	no	si
GAMMA	continua	no	si	no	no
NORMAL	continua	si	si	si	si
PASCAL	discreta	no	si	no	no
BINOMIAL	discreta	no	si	no	si
HIPERGEOMÉTRICA	discreta	no	si	no	no
POISSON	discreta	no	si	no	si
EMPÍRICA DISCRETA	discreta	no	si	no	si

Tabla 1: Listado de distribuciones y tareas a realizar con cada una.

2.1 Presentación del trabajo y entrega

Este trabajo de investigación y al igual que los anteriores se debe presentar obligatoriamente en formato \LaTeX .

El contenido a entregar es:

- Informe en formato Latex con introducción, gráficas, fórmulas empleadas, conclusiones, referencias (hay un apartado para esto) y cualquier otra información que se quiera agregar relevante a la investigación.

3 Recursos online obligatorios

Volveremos a usar el template \LaTeX de la Cornell University, pero se adjunta un documento como ejemplo de inserción de código Python en el informe y el capítulo 4 de el libro Técnicas de Simulación en Computadoras[1]:

<https://es.overleaf.com/latex/templates/style-and-template-for-preprints-arxiv-bio-arxiv/pkzcrhzcxdxmc>

References

- [1] Naylor, T.H. Técnicas de simulación en computadoras, 1982.