

Ingeniería en Computación

Probabilidad y Estadística

Trabajo Práctico  
“Simulación de variables aleatorias”  
Anexo: Documentación del Código

Alumnos:

Araneda, Alejandro – eloscurodeefeso@hotmail.com  
Quinteros, Fernando - lordfers@gmail.com  
Speciale, Gastón - gasticai@hotmail.com

Práctica entregada:

2do. Cuatrimestre 2020  
Jueves, 3 de Diciembre

Docentes:

Dr. Ing. Néstor Rubén Barraza  
Dr. Lic. Verónica Moreno  
Ing. Gabriel Pena

---

## Índice general

---

**Índice de Módulos Python**

**5**



**class simulador.Normal**

Clase contenedora para la enumeración de métodos de simulación normal

**BoxMuller = 2**

Constante correspondiente al métodos de aproximación Box-Müller

**Rejection = 1**

Constante correspondiente al método de aceptación y rechazo

**simulador.bernoulli** (*size, p*)

Generador de una muestra con resultados de un experimento Bernoulli

**Parámetros**

- **size** (*int*) – tamaño de la muestra
- **p** (*float*) – probabilidad de obtener un éxito

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[bool]

**simulador.binomial** (*size, tries, p*)

Generador de una muestra con base en una distribución binomial

**Parámetros**

- **size** (*int*) – tamaño de la muestra
- **tries** (*int*) – cantidad de experimentos Bernoulli
- **p** (*float*) – probabilidad de obtener un éxito

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[int]

**simulador.empirica** (*size, sample*)

Generador de una muestra con base en la distribución empírica de otra.

**Parámetros**

- **size** (*int*) – tamaño de la muestra
- **sample** (*List*) – muestra que se utilizará como distribución empírica

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List

**simulador.exponencial** (*size, beta=1*)

Generador de una muestra con base en una distribución exponencial

**Parámetros**

- **size** (*int*) – tamaño de la muestra
- **beta** (*float, optional*) – media de la distribución exponencial, defaults to 1

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[float]

**simulador.normal** (*size, mu=0, sigma=1, method=2*)

Generador de muestra con base en una distribución normal.

**Parámetros**

- **size** (*int*) – tamaño de la muestra
- **mu** (*float, optional*) – media de la distribución normal de base, defaults to 0
- **sigma** (*float, optional*) – desviación estandar de la distribución de base, defaults to 1

- **method**(*int*, *optional*) – método de simulación entre *Normal.Rejection* o *Normal.BoxMuller*, defaults to *Normal.Rejection*

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[float]

**Raises** ValueError: si no es uno de los metodos definidos en la clase *simulador.Normal*

`simulador.normal_boxmuller(size, mu=0, sigma=1)`

Implementación de la simulación normal con la aproximación de Box-Müller.

**Parámetros**

- **size**(*int*) – tamaño de la muestra
- **mu**(*float*, *optional*) – media de la distribución normal de base, defaults to 0
- **sigma**(*float*, *optional*) – desviación estandar de la distribución de base, defaults to 1

**Tipo del valor devuelto** List[float]

`simulador.normal_rejection(size, mu=0, sigma=1)`

Implementación de la simulación normal con el método de aceptación y rechazo.

**Parámetros**

- **size**(*int*) – tamaño de la muestra
- **mu**(*float*, *optional*) – media de la distribución normal de base, defaults to 0
- **sigma**(*float*, *optional*) – desviación estandar de la distribución de base, defaults to 1

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[float]

`simulador.random()` → x in the interval [0, 1).

`simulador.uniform(size, min=0, max=1)`

Generador de muestra con base en una distribución uniforme continua.

**Parámetros**

- **size**(*int*) – tamaño de la muestra
- **min**(*float*, *optional*) – valor mínimo de la variable aleatoria, defaults to 0
- **max**(*float*, *optional*) – valor máximo de la variable aleatoria, defaults to 1

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[float]

`simulador.uniformdis(size, min, max)`

Generador de muestra con base en una distribución uniforme discreta.

**Parámetros**

- **size**(*int*) – tamaño de la muestra
- **min**(*int*) – valor mínimo de la variable aleatoria
- **max**(*int*) – valor máximo de la variable aleatoria

**Devuelve** una lista con posibles valores de la variable aleatoria

**Tipo del valor devuelto** List[int]

`analisis.desviacion_estandar_muestral(muestra)`

Calcula la desviación estandar o raíz de la varianza muestral insesgada

**Parámetros muestra** (List[float]) – listado de valores de la variable aleatoria

**Devuelve** devuelve la desviación estandar de la muestra

**Tipo del valor devuelto** float

`analisis.estadist (muestra)`

Funcion auxiliar para la presentacion de la media y varianza muestrales

**Parámetros muestra** (*List [float]*) – listado de valores de la variable aleatoria

`analisis.frecuencias (muestra, binsize)`

Contabiliza las frecuencias absolutas de los valores de una muestra agrupados por clases del mismo ancho de banda

**Parámetros**

- **muestra** (*List [float]*) – listado de valores de la variable aleatoria
- **binsize** (*float*) – tamaño del ancho de banda de las clases

**Devuelve** listado de frecuencias absolutas para cada clase incluyendo clases con frecuencia cero.

**Tipo del valor devuelto** List[float]

`analisis.media_muestral (muestra)`

Calcula la media muestral o promedio de valores de una muestra

**Parámetros muestra** (*List [float]*) – listado de valores de la variable aleatoria

**Devuelve** devuelve la media muestral

**Tipo del valor devuelto** float

`analisis.varianza_muestral (muestra)`

Calcula la varianza muestral insesgada

**Parámetros muestra** (*List [float]*) – listado de valores de la variable aleatoria

**Devuelve** devuelve la varianza muestral

**Tipo del valor devuelto** float

`graficas.histograma (sample, gaps, relative=False)`

Graficador de histogramas por ancho de bandas

**Parámetros**

- **sample** (*List*) – muestra a graficar
- **gaps** (*List [float]*) – listado de anchos de banda para las clases
- **relative** (*bool*) – transforma los pesos en relativos, default to False



### **a**

analisis (*Linux, Windows*), 2

### **g**

graficas, 3

### **s**

simulador (*Linux, Windows*), 1