

Gianluca Aguado 1556124

Steven Sanchez 1556027

Punto-uno

Usando el generador lineal congruente de la clase , cambiamos los parámetros para

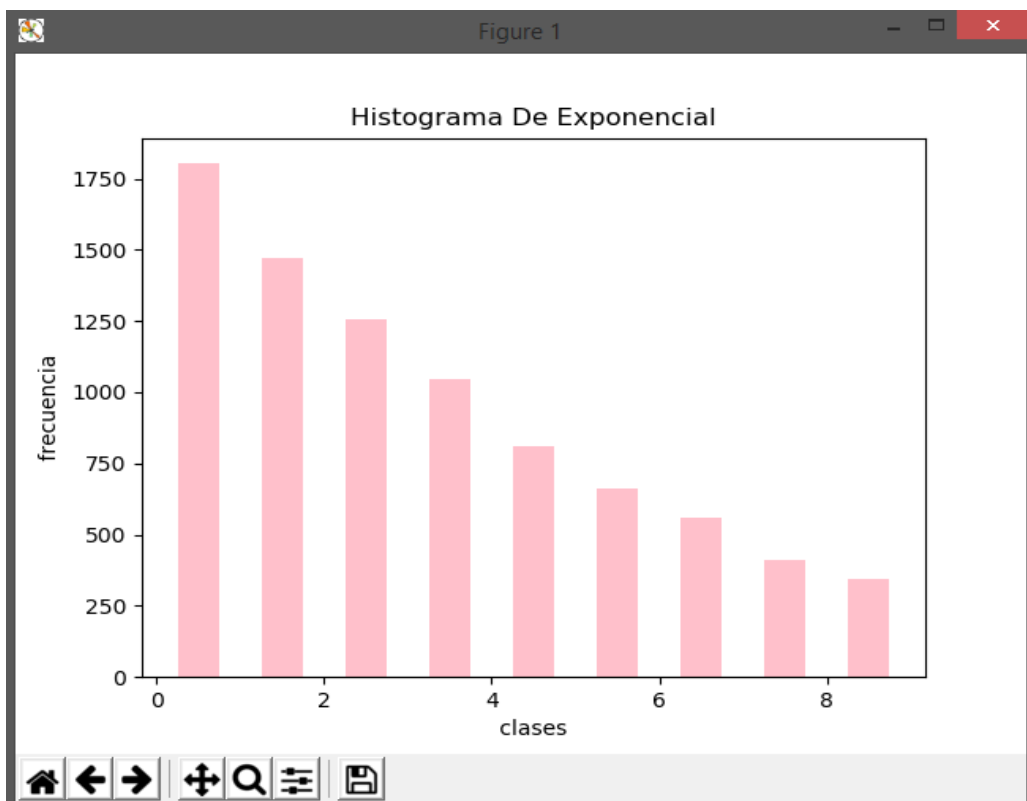
Convertir a generador estándar mínimo , importamos las librerías matplotlib.pyplot as plt

Y numpy as np; generaremos dos arrays , el primero se llenara con los datos del generador , el cual usa como semilla un numero random, el segundo array será las clases en las cuales se

Hubicaran los datos del array uno en el histograma.

La librería matplotlib nos permitirá graficar el histograma con los mencionados arrays .

En la función GEM haremos al tiempo la conversión a exponencial y estos son los que guardaremos en el primer array.



Punto-Dos

Implementamos un vector de rango que genera números pseudoaleatorios entre 1 y 99 de rango.

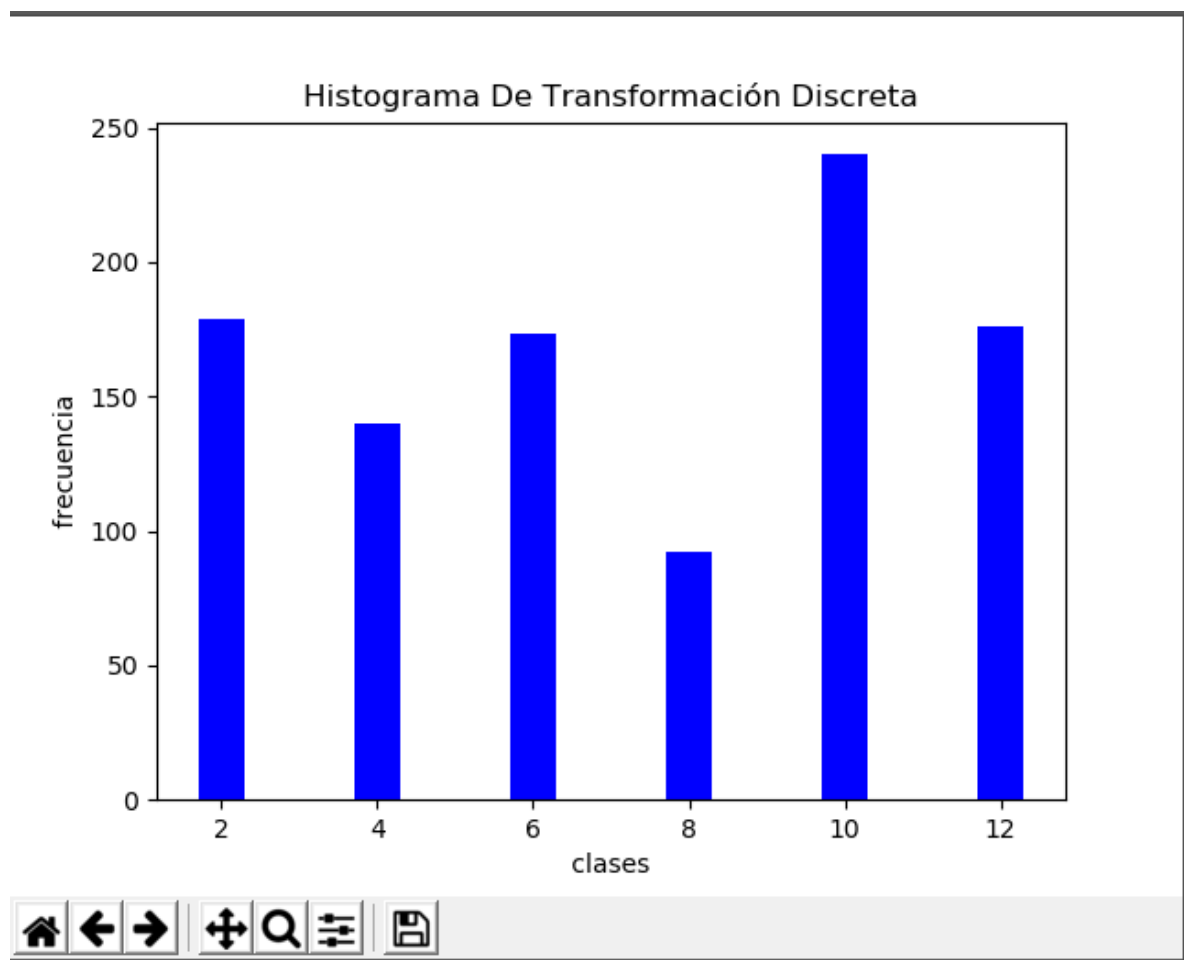
Dividimos los datos de este arreglo entre 100 para tener un rango de 0,1 a 1,0 .

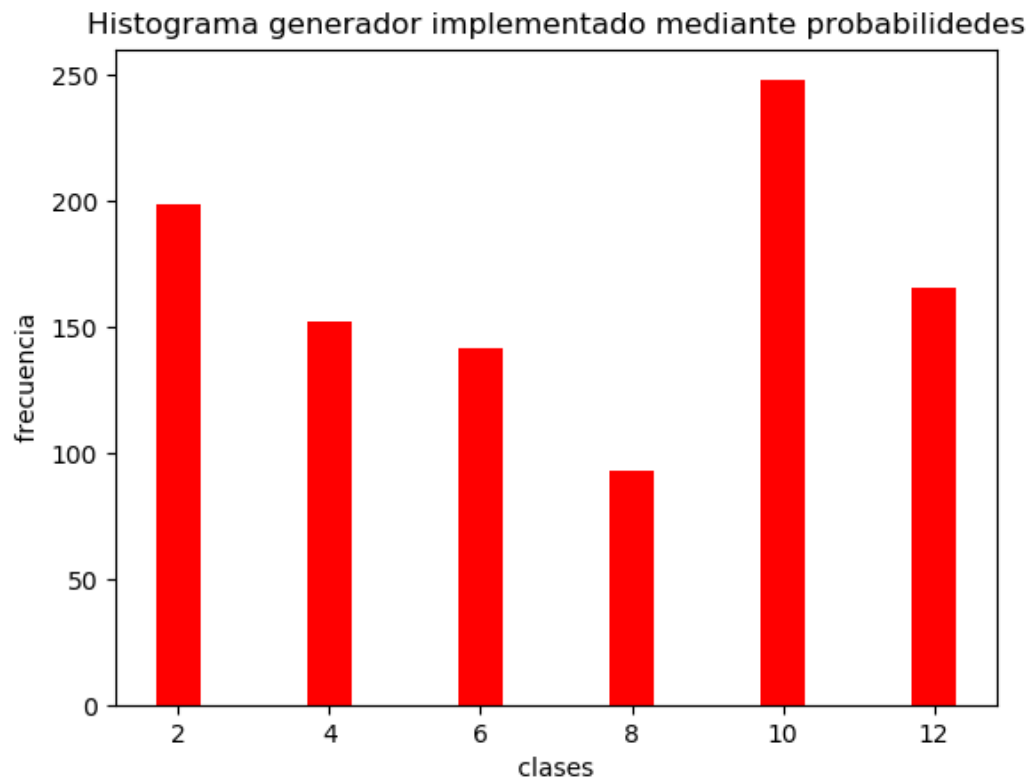
Con apoyo de dos bucles y condicionales preguntaremos entre que rangos de P se encuentran

Los datos de cada arreglo, paralelamente asignamos el A a un nuevo array dependiente de la condición en la que caiga.

Obteniendo esto pasaremos a sacar la tabla R y A on sus valores con la función tabla .

Luego usamos el código del taller para tener con que comparar el histograma , será el de color rojo y el de la transformación discreta de color azul .





x=11.4195 y=194.595

6 a 9 mas bien :	0.75 ----> 9	0.97 ----> 11
r ----> a	0.08 ----> 1	0.7 ----> 9
0.84 ----> 9	0.94 ----> 11	0.65 ----> 9
0.72 ----> 9	0.09 ----> 1	0.05 ----> 1
0.02 ----> 1	0.52 ----> 7	0.23 ----> 3
0.24 ----> 3	0.9 ----> 11	0.56 ----> 7
0.55 ----> 7	0.39 ----> 5	0.65 ----> 9
0.28 ----> 3	0.42 ----> 5	0.25 ----> 3
0.85 ----> 11	0.84 ----> 9	0.51 ----> 7
0.64 ----> 9	0.95 ----> 11	0.77 ----> 9
0.35 ----> 11	0.17 ----> 1	0.38 ----> 5
0.55 ----> 7	0.5 ----> 11	0.13 ----> 1
0.7 ----> 9	0.51 ----> 7	0.19 ----> 1
0.89 ----> 11	0.46 ----> 5	0.69 ----> 9
0.75 ----> 9	0.44 ----> 5	0.89 ----> 11
0.56 ----> 7	0.1 ----> 1	0.73 ----> 9
0.82 ----> 9	0.45 ----> 5	0.1 ----> 1
0.21 ----> 3	0.64 ----> 9	0.28 ----> 3
0.59 ----> 7	0.76 ----> 9	0.44 ----> 5
0.84 ----> 9	0.36 ----> 5	0.41 ----> 5
0.35 ----> 11	0.96 ----> 11	0.53 ----> 7
0.39 ----> 5	0.32 ----> 3	0.37 ----> 5
0.46 ----> 5	0.78 ----> 9	0.79 ----> 9
0.7 ----> 9	0.55 ----> 7	0.08 ----> 1
0.77 ----> 9	0.75 ----> 9	0.25 ----> 3
0.61 ----> 9	0.43 ----> 5	0.55 ----> 7
0.97 ----> 11	0.07 ----> 1	0.39 ----> 5
0.17 ----> 1	0.68 ----> 9	0.4 ----> 5
0.24 ----> 3	0.97 ----> 11	0.24 ----> 3
0.82 ----> 9	0.86 ----> 11	0.11 ----> 1
0.74 ----> 9	0.46 ----> 5	0.88 ----> 11
0.86 ----> 11	0.83 ----> 9	0.74 ----> 9
0.42 ----> 5	0.67 ----> 9	0.03 ----> 1
0.62 ----> 9	0.54 ----> 7	0.18 ----> 1
0.72 ----> 9	0.86 ----> 11	0.64 ----> 9
0.93 ----> 11	0.22 ----> 3	0.43 ----> 5
0.02 ----> 1	0.74 ----> 9	0.08 ----> 1
0.54 ----> 7	0.09 ----> 1	0.02 ----> 1
0.46 ----> 5	0.34 ----> 3	0.75 ----> 9
0.67 ----> 9	0.25 ----> 3	0.69 ----> 9
0.29 ----> 3	0.65 ----> 9	0.35 ----> 11
0.72 ----> 9	0.64 ----> 9	0.25 ----> 3
0.6 ----> 11	0.38 ----> 5	0.95 ----> 11
0.35 ----> 11	0.15 ----> 1	0.34 ----> 3
0.09 ----> 1		

TABLA R --> A del punto dos ...generada por discreta

Punto-tres – Kolmogorov

Python 3.7.3 Shell

File Edit Shell Debug Options Window Help

===== RESTART: C:\Users\gianlucca\Desktop\punto-tres.py =====

Clases		FO	FOA	POA	PEA	PEA - POA
0.0000	0.0100	93.0000	93.0000	0.0093	0.0100	0.0007
0.0100	0.0200	99.0000	192.0000	0.0192	0.0200	0.0008
0.0200	0.0300	103.0000	295.0000	0.0295	0.0300	0.0005
0.0300	0.0400	102.0000	397.0000	0.0397	0.0400	0.0003
0.0400	0.0500	85.0000	482.0000	0.0482	0.0500	0.0018
0.0500	0.0600	101.0000	583.0000	0.0583	0.0600	0.0017
0.0600	0.0700	110.0000	693.0000	0.0693	0.0700	0.0007
0.0700	0.0800	86.0000	779.0000	0.0779	0.0800	0.0021
0.0800	0.0900	113.0000	892.0000	0.0892	0.0900	0.0008
0.0900	0.1000	101.0000	993.0000	0.0993	0.1000	0.0007
0.1000	0.1100	105.0000	1098.0000	0.1098	0.1100	0.0002
0.1100	0.1200	100.0000	1198.0000	0.1198	0.1200	0.0002
0.1200	0.1300	114.0000	1312.0000	0.1312	0.1300	0.0012
0.1300	0.1400	93.0000	1405.0000	0.1405	0.1400	0.0005
0.1400	0.1500	99.0000	1504.0000	0.1504	0.1500	0.0004
0.1500	0.1600	103.0000	1607.0000	0.1607	0.1600	0.0007
0.1600	0.1700	95.0000	1702.0000	0.1702	0.1700	0.0002
0.1700	0.1800	100.0000	1802.0000	0.1802	0.1800	0.0002
0.1800	0.1900	107.0000	1909.0000	0.1909	0.1900	0.0009
0.1900	0.2000	91.0000	2000.0000	0.2000	0.2000	0.0000
0.2000	0.2100	92.0000	2092.0000	0.2092	0.2100	0.0008
0.2100	0.2200	110.0000	2202.0000	0.2202	0.2200	0.0002
0.2200	0.2300	111.0000	2313.0000	0.2313	0.2300	0.0013
0.2300	0.2400	94.0000	2407.0000	0.2407	0.2400	0.0007
0.2400	0.2500	100.0000	2507.0000	0.2507	0.2500	0.0007
0.2500	0.2600	102.0000	2609.0000	0.2609	0.2600	0.0009
0.2600	0.2700	100.0000	2709.0000	0.2709	0.2700	0.0009
0.2700	0.2800	94.0000	2803.0000	0.2803	0.2800	0.0003
0.2800	0.2900	110.0000	2913.0000	0.2913	0.2900	0.0013
0.2900	0.3000	99.0000	3012.0000	0.3012	0.3000	0.0012
0.3000	0.3100	99.0000	3111.0000	0.3111	0.3100	0.0011
0.3100	0.3200	95.0000	3206.0000	0.3206	0.3200	0.0006
0.3200	0.3300	82.0000	3288.0000	0.3288	0.3300	0.0012
0.3300	0.3400	107.0000	3395.0000	0.3395	0.3400	0.0005
0.3400	0.3500	100.0000	3495.0000	0.3495	0.3500	0.0005
0.3500	0.3600	95.0000	3590.0000	0.3590	0.3600	0.0010
0.3600	0.3700	101.0000	3691.0000	0.3691	0.3700	0.0009
0.3700	0.3800	101.0000	3792.0000	0.3792	0.3800	0.0008

Ella va hasta mas abajo pero bueno , ya usted la ejecuta -->

Comenzamos con el famoso GLC y los parametros especificados

Definimos: confianza, n, clases , DMcritico según la formula .

En la columna de clases comenzamos generando su lado inferior (a) y superior (b) del intervalo.

Si el dato-j del array generado por GLC esta en el intervalo , añada y sumele el valor tantas veces encuentre numeros en ese rango en un nuevo array .

De nuevo : miraremos si el dato en el array de GLC en la posicion j se encuentra en el primer intervalo del array de Clases , lo añada a FO . busque todos los datos de GLC que esten en dicho rango para tener la FO

Luego sacamos la FOA

Con operaciones aritmeticas la POa ,PEA y PEA-Poa

Sacaremos el mayor dato de estos para asignarselo a DMcalculado

Con una condicion comparamos los DM y eso sera todo .

Punto –cuatro: POKER

Usando operaciones aritmeticas sacaremos los digitos del flotante ,ejemplo:

0.256 = a

Entonces

$a * 10$ y de este cojemos la parte entera.

Lo mismo con los demas los cuales seran b y c .

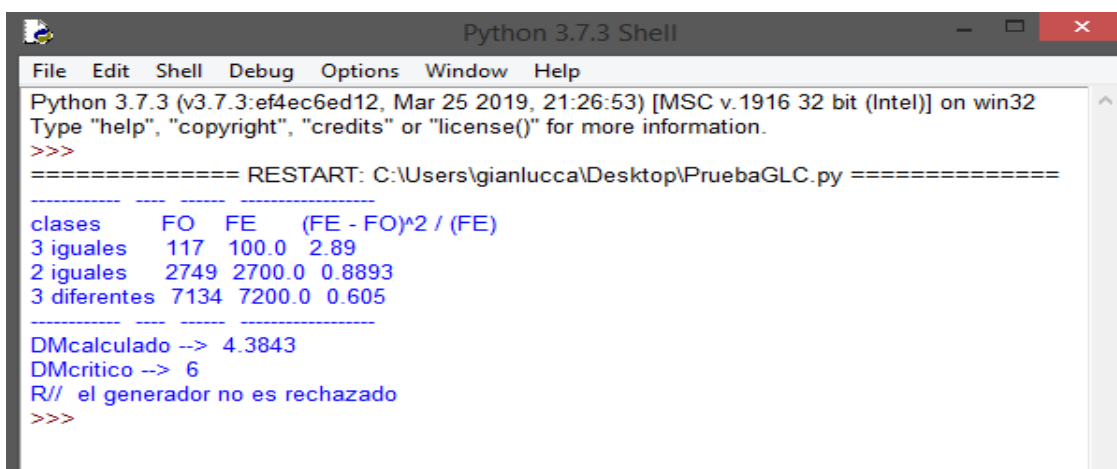
Teniendo estos enteros los guardaremos en un array .

El siguiente paso sera usar un bucke y varias condiciones para preguntar si un numero se encuentra repetido y asi saber en que caso de poker entrara.

calculamos el DM para la clase 1 teniendo en cuenta las probabilidades para un poker de con la formula $(FE - FO)^2 / (FE)$.

Lo mismo para cada clase con sus respectivos datos .

despues de hacer y generar la tabla , pasaremos a decidir si la prueba lo rechaza o no .



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\gianlucca\Desktop\PruebaGLC.py =====
clases  FO  FE  (FE - FO)^2 / (FE)
3 iguales  117  100.0  2.89
2 iguales  2749  2700.0  0.8893
3 diferentes  7134  7200.0  0.605
DMcalculado --> 4.3843
DMcritico --> 6
R// el generador no es rechazado
>>>
```

