

PruebaHLeon

July 22, 2020

```
[22]: ##cuadrados medios
import matplotlib.pyplot as plt
import math
import pandas as pd

iteraciones = 100

x0 = 1232374564#8874 ##1000 - 9999

ui = ''
xn = x0
iteraciones_dict = []
semillas = []

def divisor(numero):
    resultado = '1'
    for i in range(len(str(numero))):
        resultado+='0'
    return int(resultado)

def cuadradosMedios(semilla):
    xn=semilla
    i = 0;
    global semillas
    global semillasgeneral
    for j in range(iteraciones):

        while True:

            xn2 = str(int(math.pow(int(xn),2)))
            longitud = len(xn2)

            digitos = len(str(xn))
            ui = xn2[int((longitud/2)-digitos/2):int((longitud/2)+digitos/2)]

            if ui in semillas:
```

```

        #print("Encontro el valor repetido",i)
        iteraciones_dict.append(i);
        i=0
        xn=semillas[0]
        semillas=[]
        break
    semillas.append(ui)
    rn = int(ui)/divisor(xn)
    xn=int(ui)
    x0=xn
    i+=1

```

cuadradosMedios(x0)

```

#Ver los repetidos
lista_aux=[]
lista_repetido = []
unico = []
lista_result=[]
print(iteraciones_dict)
def frecuencia(lista,valor):
    lista_aux=0
    for i in valor:
        for j in lista:
            if i == j:
                lista_aux=lista_aux+1
        lista_result.append(lista_aux)
        lista_aux=0
    return lista_result

for x in iteraciones_dict:
    del lista_aux[:]
    if x not in unico:
        unico.append(x)
    else:
        if x not in lista_repetido:
            lista_repetido.append(x)

fra=frecuencia(iteraciones_dict,lista_repetido)
print('')
print('----- HISTOGRAMA -----')
print('')

```

```

print('lista_repetidos:', lista_repetido)

plt.bar(range(len(lista_repetido)), lista_result, edgecolor='red')
plt.xticks(range(len(lista_repetido)), lista_repetido, rotation=60)
plt.title("Cuadrados medios")
plt.ylim(min(lista_result)-0.5, max(lista_result)+0.5)
plt.show()

print('{:~3}{:~20}'.format('Numero', 'Frecuencia Repeticion'))
for lista_repetido, lista_result in zip(lista_repetido, lista_result):
    print('{:~4}{:~23}'.format(lista_repetido, lista_result))

```

```

[104, 103, 102, 29, 28, 27, 26, 25, 24, 64, 63, 62, 61, 60, 50, 49, 48, 47, 83,
82, 81, 80, 79, 78, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24,
23, 22, 54, 53, 114, 113, 112, 111, 110, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32,
31, 30, 29, 28, 27, 26, 25, 74, 73, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61,
60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 54, 53, 88, 87, 86, 85, 84, 83, 82,
81, 80]

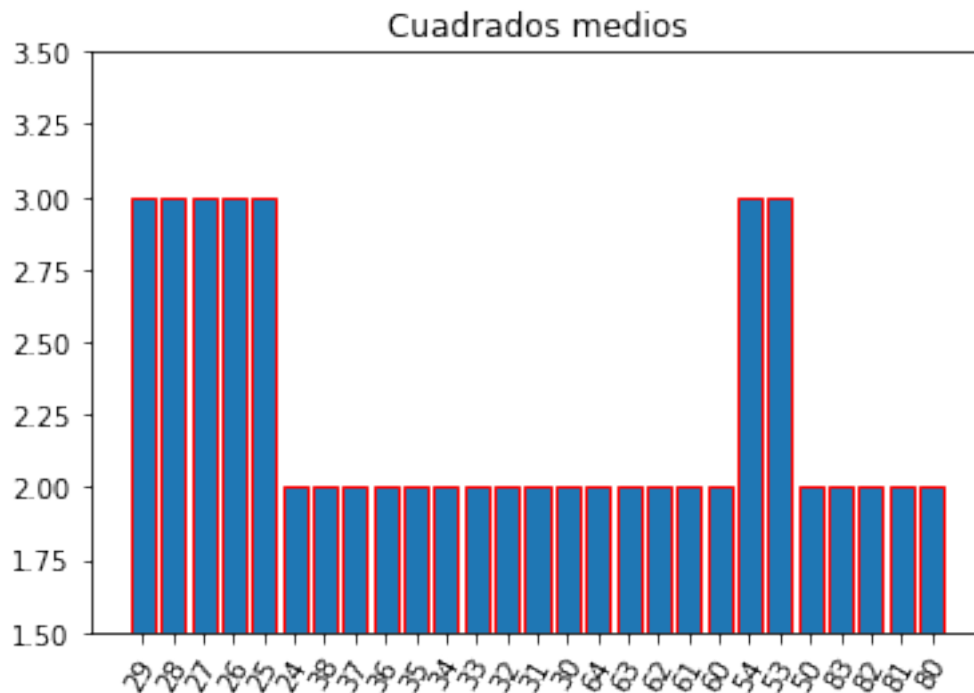
```

----- HISTOGRAMA -----

```

lista_repetidos: [29, 28, 27, 26, 25, 24, 38, 37, 36, 35, 34, 33, 32, 31, 30,
64, 63, 62, 61, 60, 54, 53, 50, 83, 82, 81, 80]

```



Numero	Frecuencia	Repeticion
29	3	
28	3	
27	3	
26	3	
25	3	
24	2	
38	2	
37	2	
36	2	
35	2	
34	2	
33	2	
32	2	
31	2	
30	2	
64	2	
63	2	
62	2	
61	2	
60	2	
54	3	
53	3	
50	2	
83	2	
82	2	
81	2	
80	2	

```
[9]: import matplotlib.pyplot as plt
      #Valores iniciales
      a=3
      b=2
      semilla=345634
      m=6
      inte=100

      iteraciones_congruencia=[]

      for j in range(inte):
          i=1
          lista_ui=[]
          xn=0
          ui=0.0
          lista_aux=0
          bandera= True
```

```

xn=semilla
while bandera:
    lista_aux=((a*xn)+b)%(m)
    ui=lista_aux/m
    xn=lista_aux
    if (i==1):
        lista_ui.append(ui)
        i=i+1
    else:
        for x in range(0,len(lista_ui)):
            if (lista_ui[x]==ui):
                iteraciones_congruencia.append(i)
                semilla=int(xn)
                bandera=False
                a=a+1
                b=b+3
                m=m+1

        lista_ui.append(ui)
        i=i+1

#Calacular los lista_repetidos
lista_aux=[]
lista_repetido = []
unico = []
lista_result=[]
print(iteraciones_congruencia)
def frecuencia(lista,valor):
    lista_aux=0
    for i in valor:
        for j in lista:
            if i == j:
                lista_aux=lista_aux+1
        lista_result.append(lista_aux)
        lista_aux=0
    return lista_result

for x in iteraciones_congruencia:
    del lista_aux[:]
    if x not in unico:
        unico.append(x)
    else:
        if x not in lista_repetido:
            lista_repetido.append(x)

```

```

fra=frecuencia(iteraciones_congruencia,lista_repetido)

print('lista_repetidos con frecuencia:', lista_repetido)

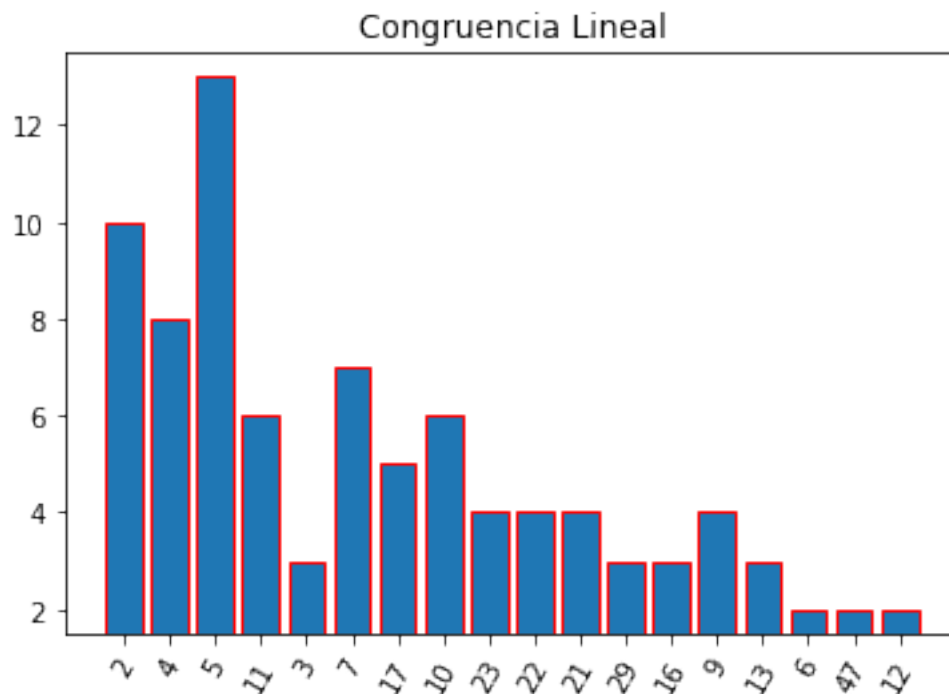
plt.bar(range(len(lista_repetido)), lista_result, edgecolor='red')
plt.xticks(range(len(lista_repetido)), lista_repetido, rotation=60)
plt.title("Congruencia Lineal")
plt.ylim(min(lista_result)-0.5, max(lista_result)+0.5)
plt.show()

print('{:~3}{:~20}'.format('Numero', 'Frecuencia'))
for lista_repetido, lista_result in zip(lista_repetido, lista_result):
    print('{:~4}{:~23}'.format(lista_repetido, lista_result))

```

[2, 4, 2, 3, 5, 11, 2, 7, 4, 5, 2, 17, 2, 10, 5, 2, 11, 23, 3, 21, 7, 4, 4, 29, 5, 16, 5, 11, 17, 13, 2, 10, 10, 7, 5, 9, 4, 22, 11, 5, 23, 47, 5, 22, 21, 17, 7, 53, 4, 21, 4, 10, 29, 59, 5, 6, 16, 4, 9, 7, 11, 12, 17, 23, 13, 71, 3, 13, 10, 5, 10, 31, 2, 40, 5, 5, 9, 83, 2, 17, 22, 29, 11, 89, 6, 7, 23, 16, 47, 37, 9, 49, 22, 12, 21, 101, 2, 18, 7, 5]

lista_repetidos con frecuencia: [2, 4, 5, 11, 3, 7, 17, 10, 23, 22, 21, 29, 16, 9, 13, 6, 47, 12]



Numero	Frecuencia
2	10

4	8
5	13
11	6
3	3
7	7
17	5
10	6
23	4
22	4
21	4
29	3
16	3
9	4
13	3
6	2
47	2
12	2

0.0.1 Conclusiones

- En conclusion se puede apreciar que el metodo de congruencia lineal ofrece una mayor aleatoriedad con respecto al metodo de los cuadrados medios ya que la frecuencia de encontrar un dato repetido es menos probable en dicho metodo. Es por esto que las librerias de numeros randomicos utilizan este metodo para la generacion de dichos numeros.
- Aunque el metodo de cuadrados medios no ofrezca la misma aleatoriedad que el de congruencia lineal, se puede dar una semilla mas grande para que tratar de solucionar este error.

0.0.2 Opiniones

- Estos metodos son muy utiles para entender como trabajan internamente las diferentes librerias de los lenguajes de programacion y asi poder replicarlos en caso de que no se cuente con la documentacion respectiva en un momento determinado.

0.0.3 Recomendaciones

- La semilla debe ser una optima para que los numeros sean lo mas aleatorios posible. Ademas es importante que el metodo considere los diferentes casos que se pueden dar en la modificacion de las semillas.