Prueba_Segundo_Interciclo

January 27, 2021

0.1 NOMBRE: ALEX BENAVIDEZ

0.2 MATERIA: SIMULACION

0.3 PROFESOR: DIEGO QUISI

[48]:
```
import numpy as np
import math as mt
import matplotlib.pyplot as plt
```

[49]:
```
def MetodoConLineales(a,b,c,m,iteraciones):
    lst=[]
    for i in range(iteraciones):
        a = (a*b+c) % m
        lst.append(round(a/m,2))
    return lst
```

[50]:
```
def obtenerPosicion(dgs):
    vl = 0
    vl2 = 0
    if dgs%2 !=0:
        vl = int(dgs/2)
        vl2 = int(dgs/2)+1
    else:
        vl = int(dgs/2)
        vl2 = vl
    return vl,vl2
```

[51]:
```
def cuadradosMedios(iteraciones,v,dgs):
    sm = int(v)
    lst=[]
    m = obtenerPosicion(dgs)
    for i in range(iteraciones):
        n = sm**2
        longuitud = len(str(n))
        u_i = str(n)[int(longuitud/2)-m[0]:int(longuitud/2)+m[1]]
        #print(u_i)
        lst.append(round(int(u_i)/10**digs,2))
        sm = int(u_i)
```

```
    return lst
```

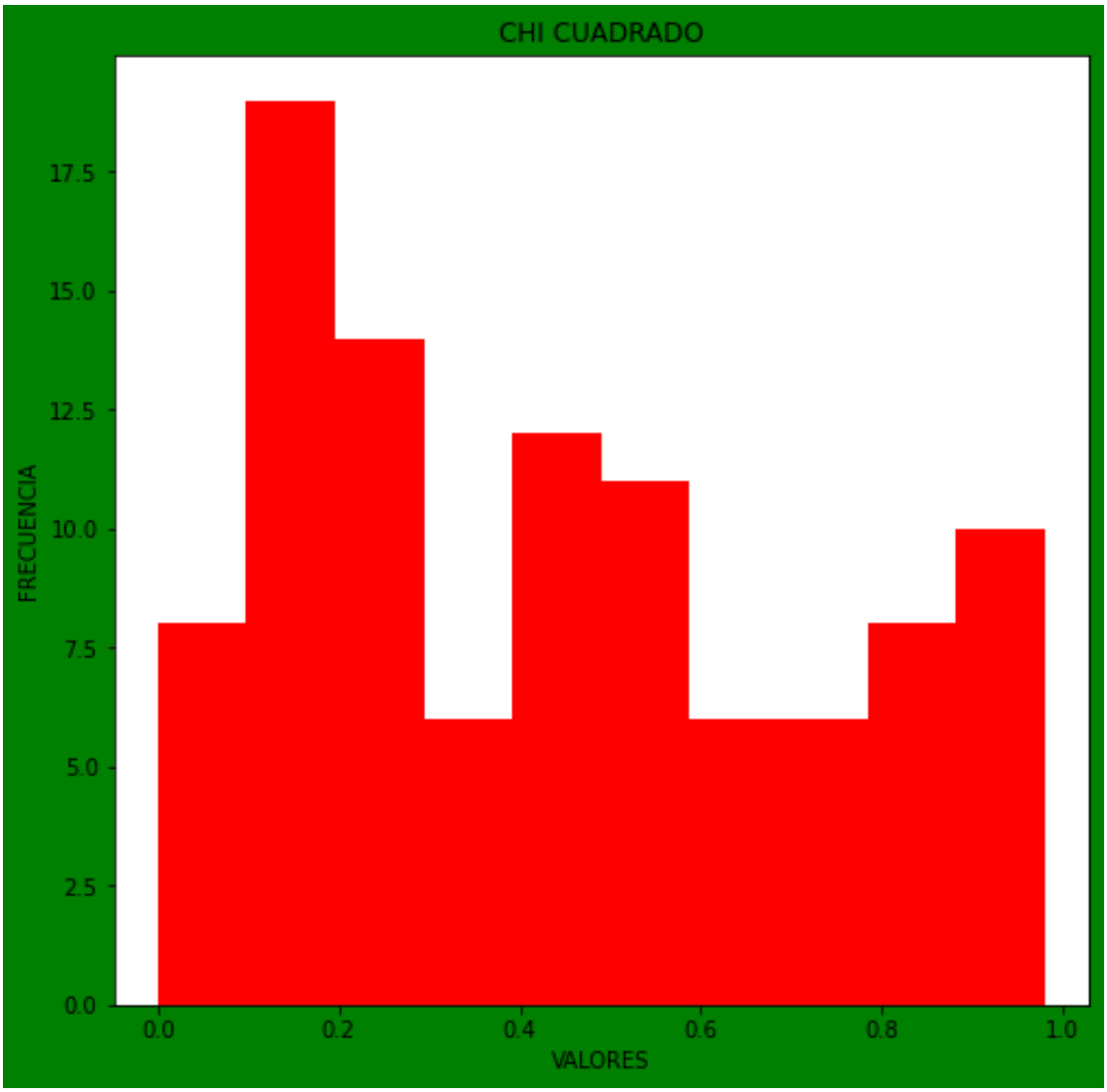
[52]:
```
def get_list(n_gr,aum,lst):
    variable = 0
    g = []
    ran = n_gr+1
    num1 = 0
    num2 = 1
    rgs = {}
    for i in range(ran):
        g.append(round(variable,2))
        variable = variable+aum

    ran2 = len(g)-1
    for i in range(ran2):
        f = g[num1]
        s = g[num2]
        rgs.update({str(f)+","+str(s):[]})
        for i in lst:
            if i!= 0:
                if i >f and i <=s:
                    rgs[str(f)+","+str(s)].append(i)
            else:
                 if i >=f and i <=s:
                    rgs[str(f)+","+str(s)].append(i)
        num1=num2
        num2=num1+1
    return rgs
```

[59]:
```
def metodo_chi(lst,v):
    n_gr = int(mt.sqrt(len(lst)))
    aum = 1/n_gr
    sumatoria = 0
    diccionario = get_list(n_gr,aum,lst)
    print(" El intervalo: ", "      Ei: ", "       Oi: ", " (Oi-Ei)^2/Ei:")
    for i, itr in enumerate(diccionario.items()):
        oper = ((len(itr[1])-n_gr)**2)/n_gr
        sumatoria+= oper
        txt = itr[0].split(',')
        print(i, "            ", str(n_gr)+"("+txt[0]+"-"+txt[1]+")     ",␣
↪len(itr[1]),"     ", oper)

    plt.figure(figsize=(8,8),facecolor='green',edgecolor='yellow')
    plt.hist(lst,color='red')
    plt.ylabel('FRECUENCIA')
    plt.xlabel('VALORES')
    plt.title('CHI CUADRADO')
```

```
    plt.show()

    if sumatoria < v:
        return True
    else:
        return False
```

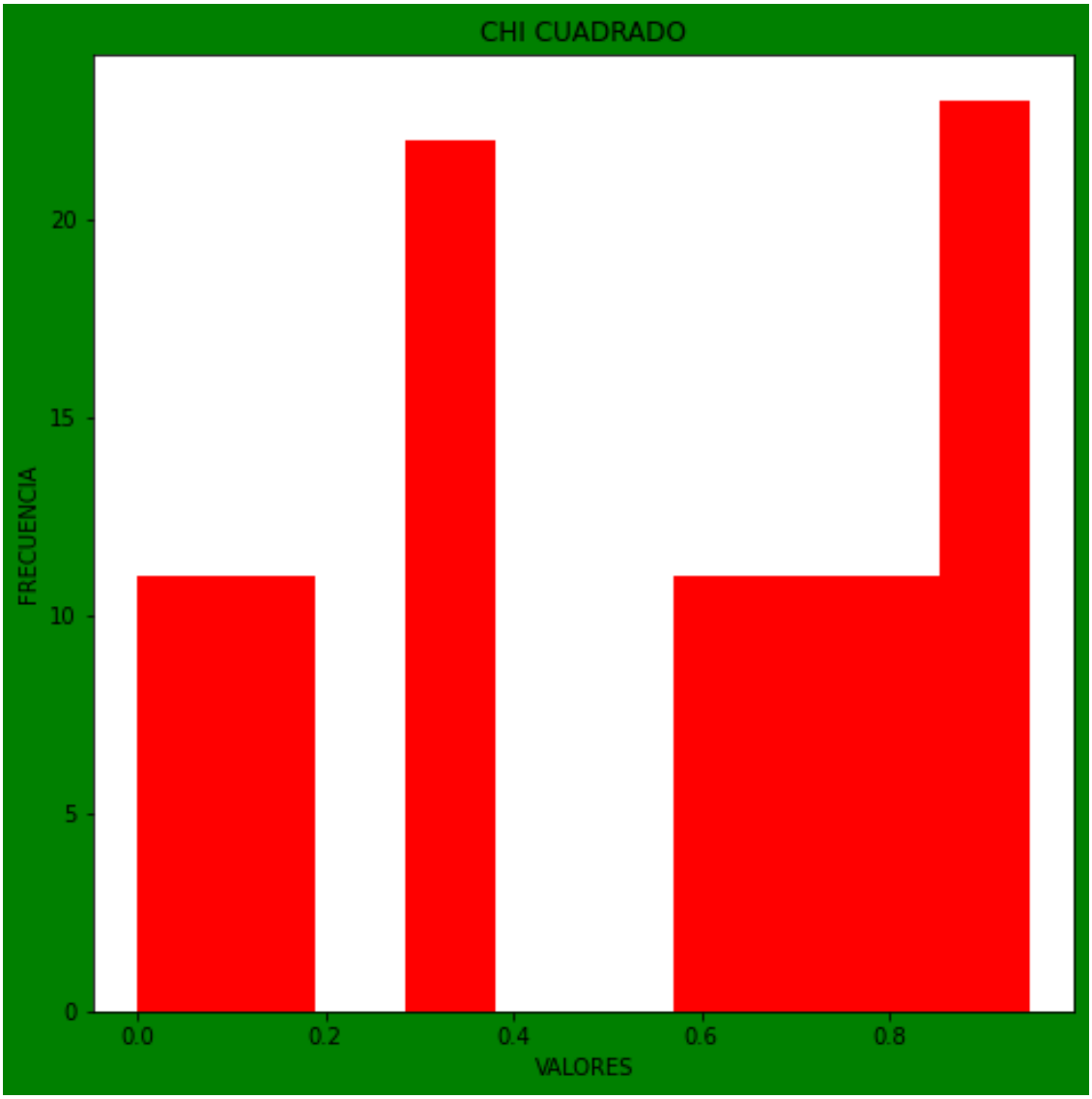
[60]:
```
if __name__ == "__main__":
    iters = 100
    v_obtenido= 16.9
    D = 7
    Xo = 74731897457
    b = 37747318974
    M = 19

    print("PRIMERA PARTE")
    lista = cuadradosMedios(iters, Xo, 7)
    res=metodo_chi(lista,v_obtenido)
    print("")
    print("SEGUNDA PARTE")
    lista2 = MetodoConLineales(D,Xo,b,M,iters)
    res2 = metodo_chi(lista2,v_obtenido)
```

```
PRIMERA PARTE
 El intervalo:        Ei:         Oi:   (Oi-Ei)^2/Ei:
0              10(0-0.1)      11       0.1
1              10(0.1-0.2)      18       6.4
2              10(0.2-0.3)      12       0.4
3              10(0.3-0.4)      7       0.9
4              10(0.4-0.5)      13       0.9
5              10(0.5-0.6)      9       0.1
6              10(0.6-0.7)      7       0.9
7              10(0.7-0.8)      5       2.5
8              10(0.8-0.9)      8       0.4
9              10(0.9-1.0)      10       0.0
```



SEGUNDA PARTE

	El intervalo:	Ei:	Oi:	(Oi-Ei)^2/Ei:
0	10(0-0.1)	11		0.1
1	10(0.1-0.2)	11		0.1
2	10(0.2-0.3)	0		10.0
3	10(0.3-0.4)	22		14.4
4	10(0.4-0.5)	0		10.0
5	10(0.5-0.6)	0		10.0
6	10(0.6-0.7)	11		0.1
7	10(0.7-0.8)	11		0.1
8	10(0.8-0.9)	23		16.9
9	10(0.9-1.0)	11		0.1


CHI CUADRADO
FRECUENCIA
VALORES
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8

[ ]: