

Prueba Practica 2

Materia:

SIMULACION

Docentes:

Ing. Diego Quisi

Estudiante:

Ricardo Jara

```
In [59]: #Importamos Librerias
import matplotlib.pyplot as plt
from tabulate import tabulate
from prettytable import PrettyTable
import pandas as pd
```

- Método para indica cuantas posiciones hacia adelante o atrás puedo correr en base a la mitad de un numero para sacar una cantidad de números

```
In [7]: def medios(digs):
        if digs % 2 != 0:
            return int(digs / 2), int(digs / 2) + 1
        else:
            return int(digs / 2), int(digs / 2)
```

- Método que calculara el cuadrado medio y me retorna una lista de números entre [0-1] en base a la a una semilla y la cantidad de números medios que deseo sacar.

```
In [8]: def cuadrados_medios(iters, semilla, dig):
        lista = []
        posI, posF = medios(dig)
        num = int(semilla)
        for i in range(iters):
            num = num ** 2
            tam = len(str(num))
            centro = str(num)[int(tam / 2) - posI: int(tam / 2) + posF]
            rd = round(int(centro) / 10 ** dig, 2)
            lista.append(rd)
```

```

        num = int(centro)
    return lista

```

- Método que calculara la congruencia lineal y me retorna una lista de números entre [0-1] en base a la a una semilla y la cantidad de números medios que deseo sacar.

```

In [127... def congruencia(semilla,iteraciones,a,c,m, dig):
    lista = []
    for i in range(1, iteraciones):
        xn = (a*semilla + c) % m
        semilla = xn
        lista.append(round(xn/((dig*2)+(dig/2)+1),2))
    return lista

```

- Método que retorna un diccionario con la cantidad de veces que un número se encuentra en una lista, para esto se recorre la lista que se necesita como parámetro

```

In [9]: def cantidad_lista(lista):
    dic = {}
    p1 = 0
    for i in range(0, 10):
        name = str(float(i) / 10) + "-" + str(float(i + 1) / 10)
        dic[name] = len(list(filter(lambda x: p1 < x < p1 + 0.1, lista)))
        p1 += 0.1
    return dic

```

- Método que grafica un diccionario en base a un diccionario, obtenido las claves y valores del mimos

```

In [10]: def plot_histrograma(dic):
    plt.figure(figsize=(15, 5))
    keys = dic.keys()
    values = dic.values()
    plt.bar(keys, values)
    plt.title("Histograma de Rango")
    plt.show()

```

- Método que realiza los distintos procesos del chi cuadrado en base a un diccionario que cuenta con el rango y el numero de repeticiones, los cálculos a realizar necesario como $(O_i - E_i)^2/E_i$. Retorna un dataframe para mejor trabajo, la suma total del chi cuadrado y la validacion de la diferencia entre la distribución de ji cuadrado.

```

In [49]: def chi_cuadrado(dic, val_vali):
    ei = []
    oi = []
    to = []
    for i in list(dic.keys()):
        ei.append(i)
        oi.append(dic[i])
        to.append((len(dic) - dic[i]) ** 2 / len(dic))
    d = {'Ei': ei, 'Oi': oi, "(Oi - Ei)^2/Ei": to}
    df = pd.DataFrame(data=d)
    total = df['(Oi - Ei)^2/Ei'].sum()
    validacion = total < val_vali
    return df, total, validacion

```

Cálculo del chi cuadrado usando una lista de números de Cuadrados Medios

```

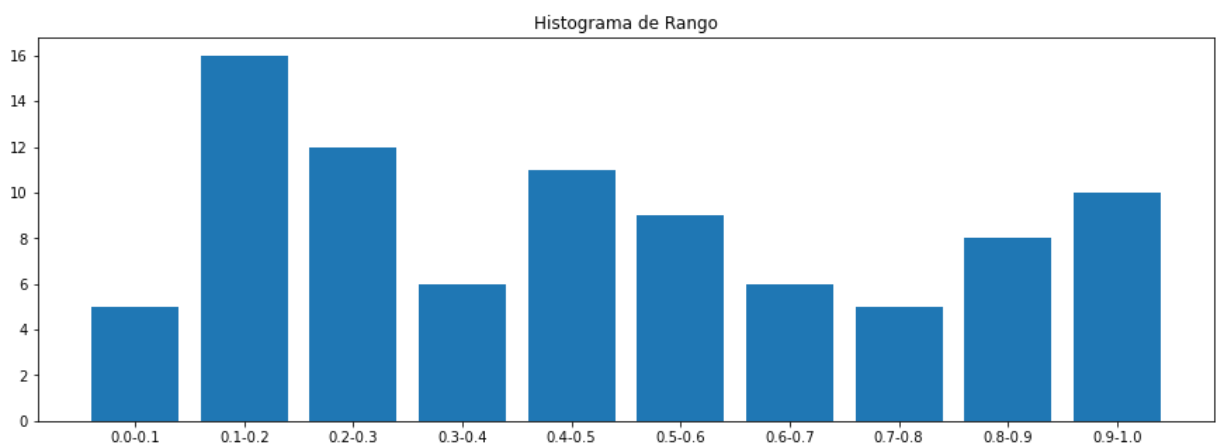
In [130... #Valor de semilla para cuadrados medios
semilla = 74731897457

```

```

#Cantidad de numeros de cuadrados medios a obtener
cantidad = 100
#numero de digitos medios a obtener en cuadrados medios
digs = 7
#numero a validar de ji cuadrado 0.005 y 9
val_ji = 16.9
#Obtencion de una lista de @cantidad de numeros aleatorios con cuadrados medios
lista = cuadrados_medios(cantidad, semilla, digs)
#Calculo de repeticion de los numeros en el rango [0.0 - 0.1, . . 0.9 - 1]
dic = cantidad_lista(lista)
#Graficamos la distribucion de los rangos
plot_histograma(dic)
#Calculamos el chi cuadrado y obtenemos el df con los calculos y el total de eso
df, total, val = chi_cuadrado(dic, val_ji)
print("Total de  $(O_i - E_i)^2/E_i$ ", total, "\n \n")
print("La distribución uniforme SE acepta " if val < val_ji else "La distribución uniforme NO")
print("Valores calculados del Chi Cuadrado")
df

```



Total de $(O_i - E_i)^2/E_i$ 12.799999999999999

La distribución uniforme SE acepta
Valores calculados del Chi Cuadrado

Out[130...

	Ei	Oi	$(O_i - E_i)^2/E_i$
0	0.0-0.1	5	2.5
1	0.1-0.2	16	3.6
2	0.2-0.3	12	0.4
3	0.3-0.4	6	1.6
4	0.4-0.5	11	0.1
5	0.5-0.6	9	0.1
6	0.6-0.7	6	1.6
7	0.7-0.8	5	2.5
8	0.8-0.9	8	0.4
9	0.9-1.0	10	0.0

Cálculo del chi cuadrado usando una lista de números de Congruencia Lineal

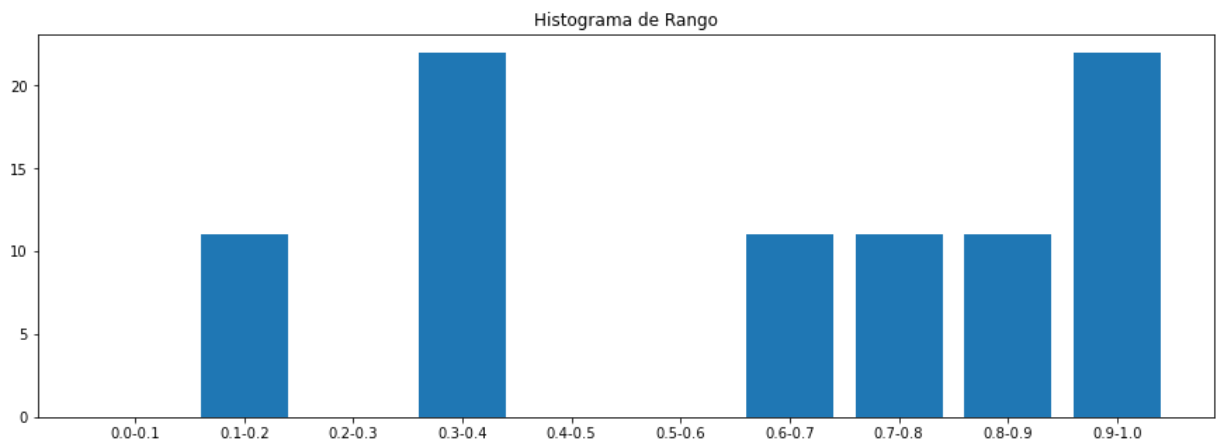
In [134... #Valor de semilla para congruencia lieal
semilla = 74731897457

```

#Cantidad de numeros de congruencia lieal a obtener
cantidad = 100
#numero para calculo de rango entre 0 - 1
digs = 7
#numero a validar de ji cuadrado 0.005 y 9
val_ji = 16.9
#Valores inicial de congruencia lieal
a=74731897457
c=37747318974
M=19 #mod == %

#Obtencion de una lista de @cantidad de numeros aleatorios con congruencia lieal
lista = congruencia(semilla, cantidad,a,c,M,digs)
dic = cantidad_lista(lista)
#Calculo de repeticion de los numeros en el rango [0.0 - 0.1, . . 0.9 - 1]
dic = cantidad_lista(lista)
#Graficamos la distribucion de los rangos
plot_histograma(dic)
#Calculamos el chi cuadrado y obtenemos el df con los calculos y el total de eso
df, total, val = chi_cuadrado(dic, val_ji)
print("Total de (Oi - Ei)^2/Ei", total, "\n \n")
print("\t \t La distribución uniforme SE acepta \n" if val else "\t \t La distribuci
print("Valores calculados del Chi Cuadrado")
df

```



Total de $(O_i - E_i)^2/E_i$ 69.2

La distribución uniforme NO SE acepta

Valores calculados del Chi Cuadrado

Out[134...

	Ei	Oi	$(O_i - E_i)^2/E_i$
0	0.0-0.1	0	10.0
1	0.1-0.2	11	0.1
2	0.2-0.3	0	10.0
3	0.3-0.4	22	14.4
4	0.4-0.5	0	10.0
5	0.5-0.6	0	10.0
6	0.6-0.7	11	0.1
7	0.7-0.8	11	0.1
8	0.8-0.9	11	0.1
9	0.9-1.0	22	14.4

Culcluciones

- Aplicando el chi cuadrado podemos apreciar si el rango de números son uniformidad e independencia. Como podemos ver usando una semilla de 74731897457 y sacando 7 números medios en la generación de números pseudoaleatorios usando el método de cuadrados medios. Tenemos que los números obtenidos tiene un distribución uniforme, Mientras que al usar el método de Congruencia lineal con la misma semilla obtenemos que los numero no están distribuidos uniformemente.

Referencias:

[1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>

[2] <https://www.mdirector.com/marketing-automation/estrategias-de-marketing-automation-para-fidelizar.html>

[3] <https://blog.inconcertcc.com/automatiza-el-marketing-digital-en-tus-redes-sociales/>