## **Prueba Adriana Castillo**

Se buscar encontrar la eficiencia de la generación de números pseudo-aletorios a través de los métodos de cuadrados medios y congruencia lineal, para ello se debe seguir el siguiente proceso:

- 1. A traves de la misma api generar una semilla diferente.
- 2. Encontrar el numero de iteraciones hasta que se repita uno de sus datos.
- 3. Generar 100 simulaciones con diferentes semillas.
- 4. Generar un histograma con el resultado obtenidos por cada método. 5. Agregar sus conclusiones, opiniones y recomendaciones

## Método de cuadrados medios

1. A traves de la misma api generar una semilla diferente.

```
In [1]: import pandas as pd, math
        cuadrados={}
        repetidos=[]
        unicos=[]
        iteraciones_s = 100
        semilla_s = 910932#1029 #int(input ("Semilla: "))
        digitos_s = 4
        val_xn_s=[]
        val_xn_m_s=[]
        val_ui_semillas=[]
        val_rn_s=[]
        div_s=[int(str(num).ljust(digitos_s+1, "0")) for num in [1]]
        ui_s=""
        izq_s=int(digitos_s/2)
        der_s=digitos_s-izq_s
        for i in range(iteraciones_s):
            if(len(str(semilla_s))>=3):
                if(len(str(semilla_s*semilla_s))>digitos_s):
                    val_xn_s.append(semilla_s)
                    val_xn_m_s.append(semilla_s*semilla_s)
                    numero_s = str(semilla_s*semilla_s)
                    mitad_s=math.floor(len(numero_s)/2)
                    if(int(numero_s)>0):
                        for p in range(mitad_s-izq_s, mitad_s+der_s, 1): ui_s+=numero_s[p]
                    val_ui_semillas.append(ui_s)
                    val_rn_s.append(int(ui_s)/div_s[0])
                    semilla_s=int(val_ui_semillas[-1])
                    ui_s=''
                    print('\n* Número de dígitos es muy alto')
                    break
            else:
                print('\n* * La semilla debe ser mayor a 0 y de 3 dígitos')
        print('Semillas generadas: \n', val_ui_semillas)
        Semillas generadas:
         ['9710', '2841', '0712', '0694', '8163', '6345', '2590', '7081', '1405', '9740', '8676', '27
        29', '4474', '0166', '2755', '5900', '8100', '6100', '2100', '4100', '8100', '6100', '2100',
        '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '210
        0', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100',
        '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '610
        0', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100',
        '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '810
        0', '6100', '2100', '4100', '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100',
        '8100', '6100', '2100', '4100', '8100', '6100', '2100', '4100']
```

1. Encontrar el numero de iteraciones hasta que se repita uno de sus datos.

```
In [2]: def cuadrados_medios(semilla):
            rep=1
            iteraciones = 1000
            digitos = 4
            val_xn=[]
            val_xn_m=[]
            val_ui=[]
            val_rn=[]
            div=[int(str(num).ljust(digitos+1, "0")) for num in [1]]
            izq=int(digitos/2)
            der=digitos-izq
            for i in range(iteraciones):
                val_xn.append(semilla)
                val_xn_m.append(semilla*semilla)
                numero = str(semilla*semilla)
                mitad=math.floor(len(numero)/2)
                for p in range(mitad-izq, mitad+der, 1): ui+=numero[p]
                if ui in val_ui:
                    #print('Repetido', ui, ' pos: ',rep)
                    cuadrados[rep] = cuadrados[rep] + 1 if rep in cuadrados else 1
                unicos.append(ui)
                val_ui.append(ui)
                val_rn.append(int(ui)/div[0])
                semilla=int(val_ui[-1])
                ui=''
                rep+=1
                #print(val_ui)
```

1. Generar 100 simulaciones con diferentes semillas.

```
In [3]: cuadrados={}
        for x in val_ui_semillas:
            #print(' ',ite,'o --> Semilla:', x)
            cuadrados_medios(int(x))
        print(cuadrados)
        {20: 1, 19: 1, 18: 1, 17: 1, 16: 1, 15: 1, 14: 1, 13: 1, 12: 1, 11: 1, 10: 1, 9: 1, 8: 1, 7:
        1, 6: 1, 5: 85}
```

1. Generar un histograma con el resultado.

```
In [9]: import matplotlib.pyplot as plt
         D = cuadrados
         plt.bar(range(len(D)), list(D.values()), align='center')
         plt.xticks(range(len(D)), list(D.keys()))
         plt.show()
          80
          70
          60
          50
          30
          20
         10
              20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
```

## Método de congruencias lineales 1. A traves de la misma api generar una semilla diferente.

```
In [5]: import pandas as pd, math
        congruencias={}
        iteraciones_cl =100
        semilla_cl= 1 ##Semilla inicial
        a_cl=3
        b_cl=5
        m_cl=19
        val_xn_cl=[semilla_cl]
        val_un_cl=['']
        for i in range(iteraciones_cl): val_xn_cl.append((a_cl*val_xn_cl[-1]+b_cl)%m_cl); val_un_c
        l.append(val_xn_cl[-1]/m_cl)
        print('Semillas generadas: \n', val_xn_cl)
        Semillas generadas:
         [1, 8, 10, 16, 15, 12, 3, 14, 9, 13, 6, 4, 17, 18, 2, 11, 0, 5, 1, 8, 10, 16, 15, 12, 3, 14,
        9, 13, 6, 4, 17, 18, 2, 11, 0, 5, 1, 8, 10, 16, 15, 12, 3, 14, 9, 13, 6, 4, 17, 18, 2, 11, 0,
        5, 1, 8, 10, 16, 15, 12, 3, 14, 9, 13, 6, 4, 17, 18, 2, 11, 0, 5, 1, 8, 10, 16, 15, 12, 3, 1
        4, 9, 13, 6, 4, 17, 18, 2, 11, 0, 5, 1, 8, 10, 16, 15, 12, 3, 14, 9, 13, 6]
         1. Encontrar el numero de iteraciones hasta que se repita uno de sus datos.
```

In [6]: def congurencias\_lineales(semillas):

```
semilla=semillas
  rep_c=1
  iteraciones = 1000
  a=3
  b=9
  val_xn=[semilla]
  val_un=['']
  for i in range(iteraciones):
      xn=(a*val_xn[-1]+b)%m
      if xn in val_xn:
           congruencias[rep_c] = congruencias[rep_c] + 1 if rep_c in congruencias else 1
           break
      val_xn.append(xn)
      val_un.append(val_xn[-1]/m)
      rep_c+=1
  #print(val_xn)
1. Generar 100 simulaciones con diferentes semillas.
```

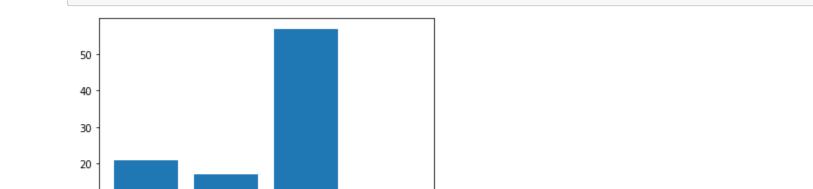
In [7]: for x in val\_xn\_cl:

```
congurencias_lineales(int(x))
print(congruencias)
{4: 21, 2: 17, 5: 57, 1: 6}

    Generar un histograma con el resultado.
```

In [8]: import matplotlib.pyplot as plt D = congruenciasplt.bar(range(len(D)), list(D.values()), align='center')

```
plt.xticks(range(len(D)), list(D.keys()))
plt.show()
50
```



## 10

Conclusiones, opiniones y recomendaciones

Como conclusión podemos observar que en el metodo de cuadrados medios en la iteración 5 ocurre el mayor numero de repeticiones y en las demas solo se repite una vez, mientras que en el metodo de congruencia lineal donde aparecen las repeticiones va variando y no unicamente en la iteración número 5.