Junio 2021

OBLIGATORIO PROBABILIDAD Y ESTADÍSTICA



```
1.1- \times N = (1-p)^{x-1}, p = p = (1-p)^{x-1}, p = p = p = p = p

Experior = 0

= (1-p)^{x-1}

= (
```

Ejercicio 1.2

```
# Muestra de tamaño N de una X ~ Geo(p=0.5)

N=1000
p=0.5
muestra = list(random.geometric(p, size=N))

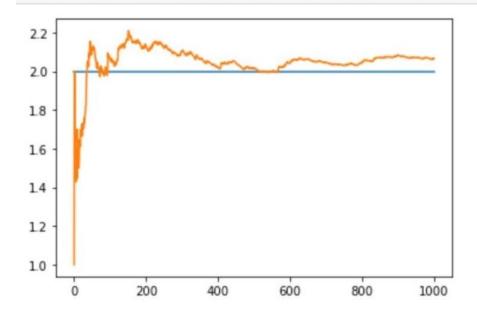
# Promedio en función de N

avg = []
for i in range(1,len(muestra)+1):
    primeros_i = muestra[:i]
    suma = sum(primeros_i)
    largo = len(primeros_i)
    promedio_i = suma/largo
    avg.append(promedio_i)
```

```
# Gráfica promedio en función de N

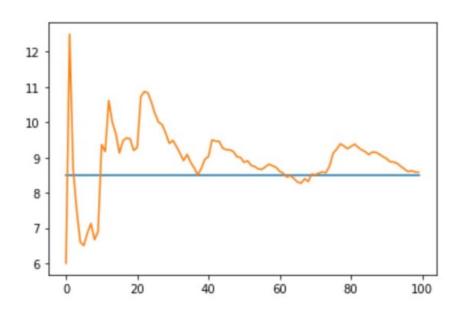
line = [2 for _ in range(0,1000)]
plt.plot(line,linestyle='solid')
plt.plot(avg)
plt.show()|

# Verifica la ley de los grandes números porque tiende a p=2
```



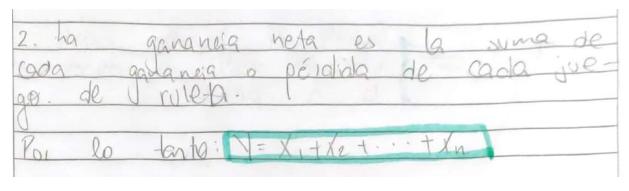
1.3- El experimento prede modelotre como una distribución geometrica donde
X: "contidod de tirodos hosto obtenel un 6" P: "probabilidad de obtenel un 6"
X~Geo(p) donde Plxipl= (1-p) P, x=1,7
Enfonces
E(p) = (1-p)x1.1 p(1-p)x2.1 p (1-p)x2.1
E(p): pn(1-p) = x1-n
In L(p): nlnp+(Ex1-n) In(1-p)
$\frac{d[\ln L[p]] = n - (\stackrel{\circ}{k}x_{i-n}) = 0}{dp} \qquad p \qquad (1-p)$
$P = \frac{n}{(2x)} = \frac{100}{6x}$
(£xì) & x.

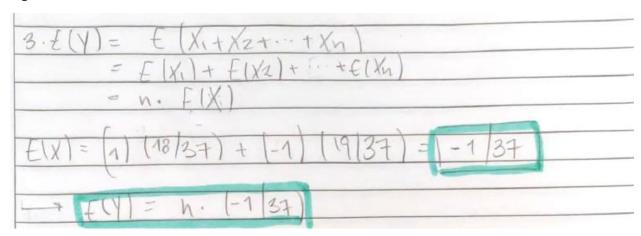
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
from numpy import random
import math
muestra = [6,19,1,4,3,6,9,9,3,9,34,7,28,2,5,1,15,11,9,3,11,41,14,10,4,2,4,8,3,1,12,4,3,
         2,15,1,3,1,16,19,12,29,8,9,1,7,9,7,2,8,2,11,2,7,5,8,13,13,6,6,2,5,2,11,3,2,
         6,17,3,23,7,12,11,7,23,36,18,21,5,3,14,15,2,3,5,2,15,9,4,3,4,1,8,5,1,2,1,11,
         5,8]
# Probabilidad de obtener un 6
   # x = "cantidad de tiradas hasta obtener un 6"
   # p = "probabilidad de obtener un 6"
   # n = "tamaño de la muestra"
for i in range(0,len(muestra)):
   primeros_i = muestra[:i]
   suma = sum(primeros_i)
p = 100/suma #Explicación en imagen adjunta
#Verificamos
     # Promedio en función de N
avg = []
for i in range(1,len(muestra)+1):
     primeros_i = muestra[:i]
     suma = sum(primeros i)
     largo = len(primeros i)
     promedio i = suma/largo
     avg.append(promedio i)
     # Gráfica promedio en función de N
line = [(1/p) for _ in range(0,100)]
plt.plot(line,linestyle='solid')
plt.plot(avg)
plt.show()
```



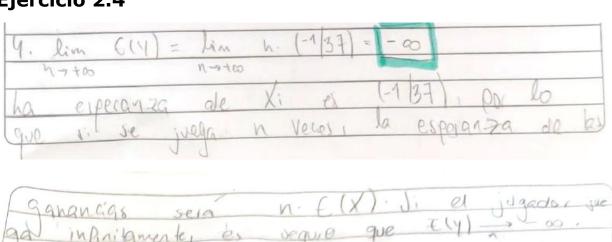
1. X X- {1,-19} Ju	q V.A	decete	probab	n ec	es le
P(x=1)= 18/37 -	-> Proba	bilidad	de	que	80.190
P(X=+1) = 19/37		billod nego		que	No

Ejercicio 2.2





Ejercicio 2.4

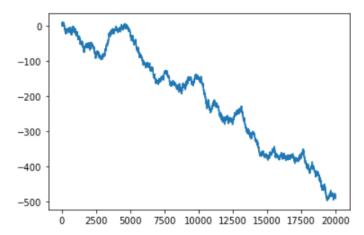


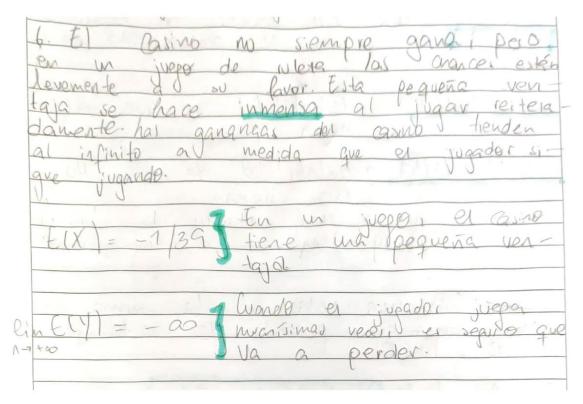
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
from numpy import random

muestra = list(random.choice([1,-1], p=[18/37,19/37], size=20000))
esperanzas = []
for i in range(1, len(muestra) + 1):
    primeros_i = muestra[:i]
    suma = sum(primeros_i)
    esperanzas.append(suma)

plt.plot(esperanzas)
```

[<matplotlib.lines.Line2D at 0x1d86bd35c40>]





Ejercicio 3.1

$$P(A0 \le X \le 50) = P(X \le 50) - P(X \le 40)$$

$$= \sum_{i=0}^{50} C_i^{100} (0,5)^i (1-0,5)^{00-i} - \sum_{i=0}^{40} C_i^{100} (0,5)^i (1-0,5)^{00-i}$$

$$= \sum_{i=0}^{50} C_i^{100} (0,5)^i (0,5)^{00-i}$$

$$= \sum_{i=40}^{50} C_i^{100} (0,5)^i (0,5)^{00-i}$$

Se aproxima mediante la Distribución Normal

$$N(n.p, \sqrt{n.p.(1-p)^{7}})$$

$$N(100, 0.5, \sqrt{100(0.5)(1-0.5)^{7}} = N(50.5) \Rightarrow \mu=50$$

$$0=5$$

$$P(40 \le X \le 50) = P(40-50 \le \frac{X-50}{5}) \le \frac{50-50}{5}$$

$$= P(-2 \le z \le 0) = P(z \le 0) - P(z \le -2)$$

Ejericio 3.3

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
from scipy.stats import binom
from numpy import random

n = 100 # Cantidad de veces que se lanza una moneda equilibrada.
p = 0.5 # Probabilidad de que salga cara al lanzar una moneda (éxito)

# P(40 <= z <= 50)
p = binom.cdf(50, n, p) - binom.cdf(40, n, p)
print (f"La probabilidad de que la cantidad de caras esté entre 40 y 50 de forma exacta es : {p}" )

# P(40 <= z <= 50) utilizando la aproximación a una Distribución Normal.
pn = norm.cdf(0) - norm.cdf(-2)
print (f"La probabilidad de que la cantidad de caras esté entre 40 y 50 utilizando la aproximación a una Normal es: {pn}")
print (f"La diferencia fue de: {abs(p - pn)}")</pre>
```

La probabilidad de que la cantidad de caras esté entre 40 y 50 de forma exacta es : 0.5113506518730992 La probabilidad de que la cantidad de caras esté entre 40 y 50 utilizando la aproximación a una Normal es: 0.4772498680518208 La diferencia fue de: 0.03410078382127846