Ejercicio 1.1

$$1.1 - X \sim Geo(p) \iff P(X=k) = (1-p)^{k-1} \cdot p, \quad k \in \mathbb{Z}^+$$

Esperanza:

$$q = (1-p)$$

$$E(X) = \sum_{i=1}^{+\infty} i(1-p)^{i-1} p = p \sum_{i=1}^{+\infty} i q^{i-1} = \circledast$$

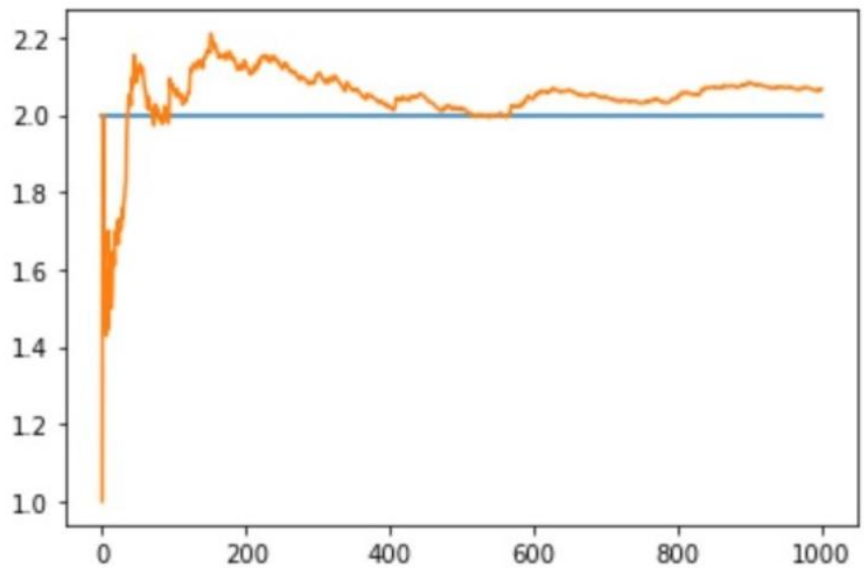
$$\circledast \; p \sum_{i=1}^{+\infty} \frac{dq^i}{dq} = p \frac{d}{dq} \sum_{i=1}^{+\infty} q^i = p \cdot \frac{d}{dq}\left(\frac{q}{1-q}\right) = p \cdot \frac{1}{(1-q)^2} = \frac{p}{p^2}$$

Entonces queda demostrado que: $\boxed{E(X) = p/p^2 = 1/p}$

Ejercicio 1.2

```python
# 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)
```

```python
# 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
```



Ejercicio 1.3

1.3- El experimento puede modelarse como una distribución geométrica donde

X = "cantidad de tiradas hasta obtener un 6"
p = "probabilidad de obtener un 6"

$X \sim Geo(p)$ donde $P(x;p) = (1-p)^{x-1} p$, $x = 1, 2, \ldots$

Entonces:

$$L(p) = (1-p)^{x_1 - 1} p (1-p)^{x_2 - 1} p \ldots (1-p)^{x_n - 1}$$

$$L(p) = p^n (1-p)^{\sum_i^n x_i - n}$$

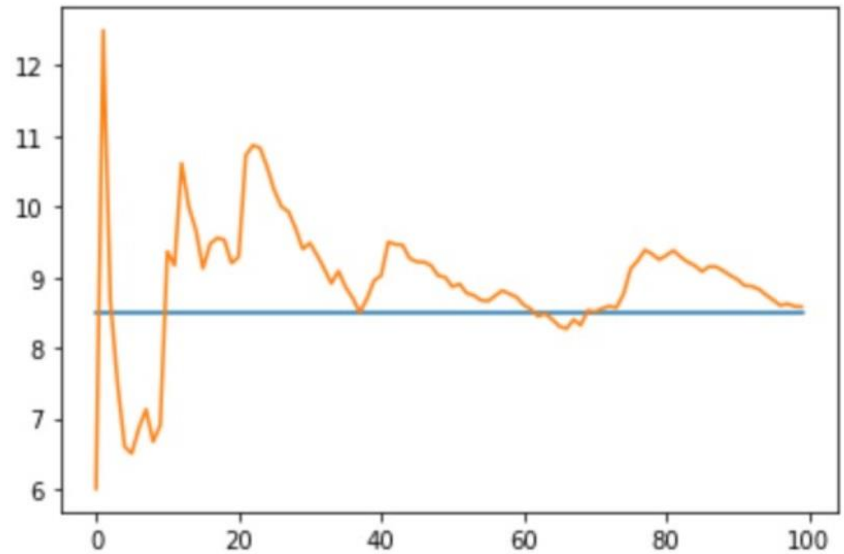
$$\ln L(p) = n \ln p + \left(\sum_i^n x_i - n\right) \ln(1-p)$$

$$\frac{d[\ln L(p)]}{dp} = \frac{n}{p} - \frac{\left(\sum_i^n x_i - n\right)}{(1-p)} = 0$$

$$p = \frac{n}{\left(\sum_i^n x_i\right)} = \frac{100}{\sum_i x_i}$$

```python
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
```

```python
#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()
```



Ejercicio 2.1

1. X es una V.A discreta con recorrido $\boxed{X=\{1,-1\}}$. Su función de probabilidad es la siguiente.

$P(X=1) = 18/37$ → Probabilidad de que salga negra.

$P(X=-1) = 19/37$ → Probabilidad de que NO salga negra.

Ejercicio 2.2

2. La ganancia neta es la suma de cada ganancia o pérdida de cada juego de ruleta.

Por lo tanto: $\boxed{Y = X_1 + X_2 + \cdots + X_n}$

Ejercicio 2.3

$$3.\ E(Y) = E(X_1 + X_2 + \cdots + X_n)$$
$$= E(X_1) + E(X_2) + \cdots + E(X_n)$$
$$= n \cdot E(X_i)$$

$$E(X) = (1)(18/37) + (-1)(19/37) = \boxed{-1/37}$$

$$\longrightarrow \boxed{E(Y) = n \cdot (-1/37)}$$

Ejercicio 2.4

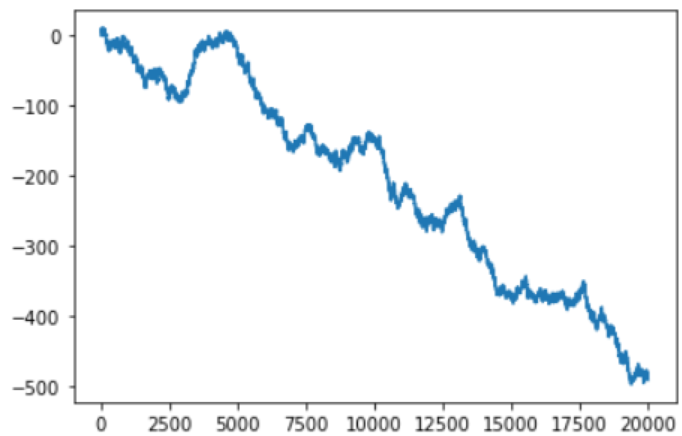
$$4.\ \lim_{n \to +\infty} E(Y) = \lim_{n \to +\infty} n \cdot (-1/37) = \boxed{-\infty}$$

La esperanza de X_i es $(-1/37)$, por lo que si se juega n veces, la esperanza de las ganancias será $n \cdot E(X)$. Si el jugador juega infinitamente, es seguro que $E(Y) \xrightarrow[n]{} -\infty$.

Ejercicio 2.5

```
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 2.6

6. El casino no siempre gana, pero en un juego de ruleta las chances están levemente a su favor. Esta pequeña ventaja se hace inmensa al jugar reiteradamente. Las ganancias del casino tienden al infinito a medida que el jugador sigue jugando.

$E(X) = -1/39$ } En un juego, el casino tiene una pequeña ventaja

$\lim_{n \to +\infty} E(Y) = -\infty$ } Cuando el jugador juega muchísimas veces, es seguro que va a perder.

Ejercicio 3.1

$$P(40 \leq X \leq 50) = P(X \leq 50) - P(X \leq 40)$$

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

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

Ejercicio 3.2

Se aproxima mediante la Distribución Normal

$$N\left(n.p,\ \sqrt{n.p.(1-p)}\right)$$

$$N\left(100.0,5\ ,\ \sqrt{100(0,5)(1-0,5)}\right) = N(50,5) \Rightarrow \begin{cases}\mu = 50\\ \theta = 5\end{cases}$$

$$P(40 \leq X \leq 50) = P\left(\frac{40-50}{5} \leq \overbrace{\frac{X-50}{5}}^{Z} \leq \frac{50-50}{5}\right)$$

$$= P(-2 \leq Z \leq 0) = P(Z \leq 0) - P(Z \leq -2)$$

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)}")
```

```
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
```