Jesús Uriel Guzmán Mendoza

Distribución binomial

$$b(x;n;p) = \binom{n}{x} P^x (1-p)^{n-x})$$

En estadística, la distribución binomial es una distribución de probabilidad discreta que cuenta el número de éxitos en una secuencia de n ensayos de Bernoulli independientes entre sí, con una probabilidad fija p de ocurrencia del éxito entre los ensayos.

In [54]:
```
from math import *
from scipy.integrate import quad
import numpy as np

def choose(n, k):
    # escoger k objetos en n
    return factorial(n) / (factorial(n - k) * factorial(k))

def binom(x, n, p):
    # n: cantidad de veces que se repite el experimento
    # p: probabilidad de tener un éxito o fracaso
    # x: probabiliad de queremos calcular
    return choose(n, x) * p**x * (1 - p)**(n - x)
```

Ejemplo:

Un jugador encesta con probabilidad 0.55. Calcula la probabilidad de que al tirar 6 veces enceste:

a) 4 veces

b) todas las veces

c) ninguna vez

In [56]:
```
p = 0.55 # probabilidad de encestar una vez
n = 6    # número de experimentos a realizar

print("La probabilidad de encestar 4 veces: %.4f" % binom(4, n, p)) # P(x = 4)
print("La probabilidad de encestar todas las veces: %.4f" % binom(n, n, p)) # P(x = n)
print("La probabilidad de encestar ninguna vez: %.4f" % binom(0, n, p)) # P(x = 0)
```

```
La probabilidad de encestar 4 veces: 0.2780
La probabilidad de encestar todas las veces: 0.0277
La probabilidad de encestar ninguna vez: 0.0083
```

Distribución acumulada

$$F(X \leq x) = B(x,n,p) = \sum_{y=0}^{x} B(y,n,p) = \sum_{y=0}^{x} \binom{n}{y} p^y (1-p)^{(n-y)}$$

La función de distribución acumulada (CDF) calcula la probabilidad acumulada de un valor dado de x. Utilice la CDF para determinar la probabilidad de que una observación aleatoria que se toma de la población sea menor que o igual a cierto valor. También puede usar esta información para determinar la probabilidad de que una observación sea mayor que cierto valor o se encuentre entre dos valores.

In [57]:
```
def binomAcum(x1, x2, n, p):
    #x1: principio de la distribución acumulada
    #x2: fin de la distribución acumulada
    #n: número de experimentos realizados
    #p: probabilidad de éxito o fracaso
    tot = 0
    for x in range(x1, x2 + 1):
        tot = tot + binom(x, n, p) # se acumulan todas las distribuciones binomiales con el ciclo
    return tot
```

Ejemplo

La probabilidad de que el comprador de un osciloscopio haga uso del service dentro del plazo de garantía es 0.2. Para los 5 osciloscopios que cierta empresa ha vendido independientemente a 5 compradores este mes:

a) ¿Cuál es la probabilidad de que 3 o más compradores hagan uso de la garantía?

b) ¿Cuál es la probabilidad de que entre 2 y 4 compradores hagan uso de la garantía?

In [58]:
```
n = 5 # cantidad de osciloscopios que la empresa ha vendido
p = 0.2 # probabilidad de que el comprador haga uso del servicio durante el plazo de garantía

print("La probabilidad de que 3 o más compradores hagan uso de la garantía es de: %.4f" % (1 - binomAcum(0, 2
, n, p))) # P(x >= 2)
print("La probabilidad de que entre 2 y 4 compradores hagan uso de la garantía es de: %.4f" % binomAcum(2, 4,
n, p)) # P(2 <= x <= 4)
```

```
La probabilidad de que 3 o más compradores hagan uso de la garantía es de: 0.0579
La probabilidad de que entre 2 y 4 compradores hagan uso de la garantía es de: 0.2624
```

Distribución hipergeométrica

$$h(x,N,M,n) = \frac{\binom{M}{x}\binom{N-M}{n-M}}{\binom{N}{n}}$$

En teoría de la probabilidad la distribución hipergeométrica es una distribución discreta relacionada con muestreos aleatorios y sin reemplazo. Suponga que se tiene una población de N elementos de los cuales, d pertenecen a la categoría A y N-d a la B. La distribución hipergeométrica mide la probabilidad de obtener 0 ≤ x ≤ d elementos de la categoría A en una muestra sin reemplazo de n elementos de la población original.

In [59]:
```
def hiper(x, N, M, n):
    #x: número de éxitos en una muestra
    #N: número total de objetos
    #M: número de objetos seleccionados de N
    #n: tamaño de la muestra
    return (choose(M, x) * (choose(N - M, n - M))) / choose(N, n)

def hiperAcum(x1, x2, N, M, n):
    #x1: mínima probabilidad
    #x2: máxima probabilidad
    tot = 0
    for x in range(x1, x2 + 1):
        tot = tot + hiper(x, N, M, n)
    return tot
```

Ejemplo:

Diez refrigeradores de cierto tipo han sido devueltos a un distribuidor debido a la presencia de un ruido oscilante agudo cuando el refrigerador está funcionando. Supongamos que 4 de estos 10 refrigeradores tienen compresores defectuosos y los otros 6 tienen problemas más leves. Si se examinan al azar 5 de estos 10 refrigeradores, y se define la variable aleatoria X: “el número entre los 5 examinados que tienen un compresor defectuoso”. Indicar:

a) La probabilidad de que no todos tengan fallas leves

b) La probabilidad de que a lo sumo dos tengan fallas de compresor

c) La probabilidad de que entre 1 y 3 tengan fallas en el compresor

In [60]:
```
N = 10 # número total de refrigeradores
M = 4 # número de refrigeradores que tienen componentes defectuosos
n = 5 # tamaño de la muestra

print("La probabilidad de que no todos tengan fallas leves es: %.4f" % (1 - hiper(0, N, M, n))) # P(x >= 1)
print("La probabilidad de que a lo sumo tres tengan fallas de compresor es: %.4f" % hiperAcum(0, 2, N, M, n))
# P(x <= 2)
print("La probabilidad de que entre 1 y 3 tengan fallas en el compresor es: %.4f" % hiperAcum(1, 3, N, M, n))
# P(1 <= x <= 3)
```

```
La probabilidad de que no todos tengan fallas leves es: 0.9762
La probabilidad de que a lo sumo tres tengan fallas de compresor es: 0.2619
La probabilidad de que entre 1 y 3 tengan fallas en el compresor es: 0.3333
```

Distribución binomial negativa

$$nB(x,n,p) = \binom{x-1}{k-1} p^k (1-p)^(x-k)$$

Todo exponente que cumpla una de las siguientes 4 condiciones obedece a una distribución binomial negativa:

1) El experimento consiste en una secuencia de ensayos individuales

2) El ensayo da resultado de éxito o falla

3) La probabilidad de éxito es constante de un ensayo a otro

4) El experimento continúa hasta que un total de r éxitos hayan sido obtenidos, donde r es un número entero positivo esperado

In [42]:
```
def binNeg(x, p, k):
    #x: cantidad de veces que la probabilidad de éxitos es mayor a n
    #p: probabilidad de tener un éxito o fracaso
    #k: cantidad de éxitos
    return choose(x - 1, k - 1) * p**k * (1 - p)**(x - k)

def binNegAcum(x1, x2, n, p):
    #x1: mínima probabilidad
    #x2: máxima probabilidad
    tot = 0
    for x in range(x1, x2 + 1):
        tot = tot + binNeg(x, n, p)
    return tot
```

Ejemplo:

Los registros de una compañía constructora de pozos, indican que la probabilidad de que uno de sus pozos nuevos, requiera de reparaciones en el término de un año es de 0.20.

a) ¿Cuál es la probabilidad de que el sexto pozo construido por esta compañía en un año dado sea el segundo en requerir reparaciones en un año?

b) ¿Cuál es la probabilidad de que el octavo pozo construido por esta compañía en un año dado sea el tercero en requerir reparaciones en un año?

In [61]:
```
x = 6 # número de pozos construidos
p = 0.2 # probabilidad de que uno de sus pozos requiera reparación al término del año
k = 2 # que sea el segundo en requerir reparaciones

print("La probabilidad de que el sexto pozo construido por esta compañía en un año dado sea el segundo en req
uerir reparaciones en un año: %.4f" % binNeg(x, p, k))

x = 8 # número de pozos construidos
k = 3 # que sea el segundo en requerir reparaciones

print("La probabilidad de que el octavo pozo construido por esta compañía en un año dado sea el tercero en re
querir reparaciones en un año: %.4f" % binNeg(x, p, k))
```

```
La probabilidad de que el sexto pozo construido por esta compañía en un año dado sea el segundo en requerir
reparaciones en un año: 0.0819
La probabilidad de que el octavo pozo construido por esta compañía en un año dado sea el tercero en requerir
reparaciones en un año: 0.0551
```

Distribución de Poisson

$$\lambda = \frac{n}{p} = np$$

$$P(x,\lambda) = \frac{e^(-\lambda) \lambda^x}{x!}$$

En teoría de probabilidad y estadística, la distribución de Poisson es una distribución de probabilidad discreta que expresa, a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto período de tiempo. Concretamente, se especializa en la probabilidad de ocurrencia de sucesos con probabilidades muy pequeñas, o sucesos.

In [62]:
```
def poisson(x, lam):
    #x: variable que define el número de éxitos
    #lam: relación que existe entre el número de éxitos y la frequencia en que estos ocurren
    return (exp(-lam) * lam**x) / factorial(x)
```

Ejemplo

Si un banco recibe en promedio 6 cheques sin fondo por día, ¿cuáles son las probabilidades de que reciba,

a) cuatro cheques sin fondo en un día dado,

b) 10 cheques sin fondos en cualquiera de dos días consecutivos?

In [63]:
```
# a)
x = 4 # cantidad de cheques sin fondo en un día dado
lam = 6 # cheques sin fondo que llegan por día

print("La probabilidad de cuatro sin fondo en un día dado es: %.4f" % poisson(x, lam))

# b)
x = 10 # cantidad de cheques sin fondo en un día dado
lam = 2 * 6 # cheques sin fondo que llegan en dos días

print("La probabilidad de 10 cheques sin fondos en cualquiera de dos días consecutivos: %.4f" % poisson(x, la
m))
```

```
La probabilidad de cuatro sin fondo en un día dado es: 0.1339
La probabilidad de 10 cheques sin fondos en cualquiera de dos días consecutivos: 0.1048
```

Distribución Gausseana

$$f(x,\mu,\sigma^2) = \frac{e^{(\frac{-(x-\mu)^2}{2\sigma^2})}}{\sqrt{2\pi\sigma^2}}$$

En estadística y probabilidad se llama distribución normal, distribución de Gauss, distribución gaussiana o distribución de Laplace-Gauss, a una de las distribuciones de probabilidad de variable continua que con más frecuencia aparece en estadística y en la teoría de probabilidades.

La gráfica de su función de densidad tiene una forma acampanada y es simétrica respecto de un determinado parámetro estadístico. Esta curva se conoce como campana de Gauss y es el gráfico de una función gaussiana.

La importancia de esta distribución radica en que permite modelar numerosos fenómenos naturales, sociales y psicológicos. Mientras que los mecanismos que subyacen a gran parte de este tipo de fenómenos son desconocidos, por la enorme cantidad de variables incontrolables que en ellos intervienen, el uso del modelo normal puede justificarse asumiendo que cada observación se obtiene como la suma de unas pocas causas independientes.

In [68]:
```
def gauss(x, mu, sd):
    # x: variable aleatoria
    # mu: media
    # sd: desviación estándar
    return math.exp(-0.5 * (((x - mu) / sd)**2)) / (sd * math.sqrt(2 * math.pi))

def integ(x, mu, sd):
    # integral definida en el rango [-inf, x] tomando en cuenta la función gauss(x, mu, sd)
    return quad(gauss, -np.inf, x, args=(mu, sd))[0]

def gaussAcum(x1, x2, mu, sd):
    # calcula la integral desde [a, b] de la distribución gausseana
    minProb = integ(x1, mu, sd) # integral de [-inf, x1] de gauss(x, mu, sd)
    maxProb = integ(x2, mu, sd) # integral de [-inf, x2] de gauss(x, mu, sd)
    return maxProb - minProb
```

Ejemplo

Un grupo grande de estudiantes tomaron un examen en física y los resultados finales tienen un promedio de 70 y una desviación estándar de 10. Si podemos aproximar la distribución de estas calificiaciones a una distribución normal, qué porcentaje de los estudiantes que:

a) sacaron arriba de 80

b) aprobaron (calificación >= 60)

c) reprobaron (calificación menores a 60)

d) sacaron entre 60 y 80

In [67]:
```
mu = 70 # promedio de las calificaciones de todos los estudiantes
sd = 10 # desviación estándar de las calificaciones

print("Porcentaje de estudiantes que sacaron arriba de 80: %.4f" % (1 - gaussAcum(0, 80, mu, sd))) # P(x > 80
)
print("Porcentaje de estudiantes que sacaron arriba o igual de 60: %.4f" % (1 - gaussAcum(0, 60, mu, sd))) #
P(x >= 60)
print("Porcentaje de estudiantes que sacaron abajo de 60: %.4f" % gaussAcum(0, 60, mu, sd)) # P(x < 60)
print("Porcentaje de estudiantes que sacaron entre 60 y 80: %.4f" % gaussAcum(60, 80, mu, sd)) # P(60 <= x <=
80)
```

```
Porcentaje de estudiantes que sacaron arriba de 80: 0.1587
Porcentaje de estudiantes que sacaron arriba o igual de 60: 0.8413
Porcentaje de estudiantes que sacaron abajo de 60: 0.1587
Porcentaje de estudiantes que sacaron entre 60 y 80: 0.6827
```

In [ ]: