

1. Lleve a codigo funcional las funciones

$$y1 = \log(x)$$

Condiciones

- 1. $x > 0$
- 2. $x \neq 0$

```
from math import log

x=float(input('Ingrese x: '))

if (x<=0):
    print('Fuera del dominio de de la funcion')
else:
    z = log(x)
    print(f'y1 es: {z}')
```

Ingrese x: 589
y1 es: 6.3784261836515865

$$y2 = \log(\sqrt{x})$$

Condiciones

- 1. $x > 0$
- 2. $x \neq 0$

```
from math import log
from math import sqrt

x=float(input('Ingrese x: '))

if (x<=0):
    print('Fuera del dominio de de la funcion')
else:
    z = log(sqrt(x))
    print(f'y2 es: {z}')
```

Ingrese x: 16
y2 es: 1.3862943611198906

$$y3 = \log(\sqrt{(x^2 - 1)})$$

Condiciones

- 1. $x > 1$
- 2. $x \neq 1$

```
from math import log
from math import sqrt

x=float(input('Ingrese x: '))

if (x<=1):
    print('Fuera del dominio de de la funcion')
else:
    z = log (sqrt((x**2)-1))
    print(f'y3 es: {z}')
```

🔊 Ingrese x: 1
Fuera del dominio de de la funcion

$$y4 = \frac{\log(\sqrt{x-1})}{x^2+1}$$

Condiciones

- 1. x>1
- 2. x≠1

```
from math import log
from math import sqrt

x = float(input('Ingrese x: '))
if x<=1:
    print('Fuera del dominio de la funcion')
else:
    z = log (sqrt(x-1)) / (x**2 +1)
    print(f'y4 es: {z}')
```

Ingrese x: 10
y4 es: 0.010877349392753562

$$y5 = \frac{x-1}{\log(x)}$$

Condiciones

- 1. x>0
- 2. x≠0
- 3. x≠1

```
from math import log
from math import sqrt

x = float(input('Ingrese x: '))
if x<=0 or x==1:
    print('Fuera del dominio de la funcion')
else:
    z = (x-1) / log(x)
    print(f'y5 es: {z}')
```

Ingrese x: 0.5
y5 es: 0.7213475204444817

$$y6 = \frac{1}{x} + \sqrt{x-1}$$

Condiciones

- 1. x>1

```
from math import sqrt

x = float(input('Ingrese x: '))
if x<1:
    print('Fuera del dominio de la funcion')
else:
    z = 1/x + sqrt(x-1)
    print(f'y6 es: {z}')
```

Ingrese x: 5
y6 es: 2.2

$$y7 = \frac{1}{x} + \sqrt{x-1} - (\frac{1}{\log(\frac{1}{x+1})})$$

Condiciones

- 2. x>1

```
from math import sqrt
from math import log

x = float(input('Ingrese x: '))
if x<1:
    print('Fuera del dominio de la funcion')
else:
    z = (1/x) + (sqrt(x-1)) + (1/log(1/(x+1)))
    print(f'y7 es: {z}')
```

Ingrese x: 0
Fuera del dominio de la funcion

$$y8 = \frac{2}{x^2 + 1}$$

Condiciones

Ninguna

```
x = float(input('Ingrese x: '))

z = 2 / ((x**2)+1)
print(f'y8 es: {z}')
```

Ingrese x: 5
y8 es: 0.07692307692307693

$$y9 = \frac{1}{n} \sum_{i=1}^n x_i$$

```
x= float(input("Ingrese x : "))
n=int(input("Ingrese n : "))

if n >= 1:
    summatory = []
    for i in range(n):
        summatory.append(x)
        z =(1/n)*(sum(summatory))
        print( f'y9 es:{z}')
```

Ingrese x : 7
Ingrese n : 2
y9 es:3.5
y9 es:7.0
No se puede realizar la funcion con un n menor a 1

$$y10 = \frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})^2$$

```
x= float(input("Ingrese x "))
n=int(input("Ingrese n "))
if n >= 1:
    summatory = []
    for i in range(n):
        summatory.append((x-(x/n))**2)
        z=(1/(n-1))*(sum(summatory))
        print( f'y10 es:{z}')
```

Ingrese x 7
Ingrese n 8
y10 es:5.359375
y10 es:10.71875
y10 es:16.078125
y10 es:21.4375

```
y10 es:26.796875
y10 es:32.15625
y10 es:37.515625
y10 es:42.875
No se puede realizar la funcion con un n menor a 1
```

$$y_{11} = \frac{\sqrt{y_{10}}}{y_9}$$

```
x= float(input("Ingrese x : "))
n=int(input("Ingrese n : "))

if n >= 1:
    summatory = []
    for i in range(n):
        summatory.append(x)
        y9 =(1/n)*(sum(summatory))
        print( f'y9 es:{y9}')
```

```
else:
    print("No se puede realizar la funcion con un n menor a 1")

x= float(input("Ingrese x "))
n=int(input("Ingrese n "))
if n >= 1:
    summatory = []
    for i in range(n):
        summatory.append((x-(x/n))**2)
        y10=(1/(n-1))*(sum(summatory))
        print( f'y10 es:{y10}')
```

```
else:
    print("No se puede realizar la funcion con un n menor a 1")

z=(y10**1/2)/(y9)

print(f'y11 es:{z}')
```

```
Ingrese x : 5
Ingrese n : 8
y9 es:0.625
y9 es:1.25
y9 es:1.875
y9 es:2.5
y9 es:3.125
y9 es:3.75
y9 es:4.375
y9 es:5.0
No se puede realizar la funcion con un n menor a 1
Ingrese x 7
Ingrese n 8
y10 es:5.359375
y10 es:10.71875
y10 es:16.078125
y10 es:21.4375
y10 es:26.796875
y10 es:32.15625
y10 es:37.515625
y10 es:42.875
No se puede realizar la funcion con un n menor a 1
y11 es:4.2875
```

Construya una funcion que retorne la varianza de un conjunto de datos ($n \geq 2$) cuando se va incorporando dato a dato. Use como primer par de datos los que se generan de la distribuci3n normal con $\mu = 3$ y $s = 0,3$. Use la misma semilla y los mismos parametros de la distribucion para incorporar cada dato. Haga el proceso 50 veces y luego grafique la varianza como funcion de n.

```
def variance(n):
    random.seed(123)
    if n < 2:
        return f'El conjunto de datos debe tener 2 o m1s datos'
    else:
        var=np.random.normal(3,0.3,2)
        list_var=[var[0],var[1]]
        list_g=np.random.normal(3,0.3,n)
        for i in range (n):
```

```
list_var.append(list_g[i])
print(f'La varianza es {round(np.var(list_var),4)}, cuando n es {i+1}')
```



```
variance(50)
```

La cantidad de radiacion absorbida puede medirse por la transmitancia. La relacion entre transmitancia (T) y absorbancia (A) esta dada por la ley Beer-Lambert (ley de Beer). Cree una funcion para la expresion que aparece en <https://www.electrical4u.com/what-is-transmittance/> y genere unos datos de T para determinar A. Grafique la funcion. Finalmente lea la seccion titulada: ★★Why is Absorbance the Preferred Unit Over Transmittance?★★ (buscar el link desde el buscador)

FUNCION PARA LA LEY DE BEER-LAMBERT (LEY DE BEER)

```
from math import log10

porcentaje_de_T = float(input('Ingrese porcentaje_de_T: '))

z = 2 - (log10(porcentaje_de_T))

print(f'A es: {z}')
```



```
    Ingrese porcentaje_de_T: 100
    A es: 0.0
```

GRAFICA DE TRANSMITANCIA Y ABSORBANCIA

Con los porcentajes de transmitancias teoricas de 1%,25%,50%,75%,100% se calculan las respectivas absorbancias y se grafican

```
import matplotlib.pyplot as plt
x=[1,25,50,75,100]
y=[ 2,0.6020599913279623,0.30102999566398125,0.1249387366082999,0]
plt.plot(x,y)
plt.xlabel('% de transmitancia')
plt.ylabel('Absorbancia')
plt.show()
```

