

Crea una funció que donat un Array d'una dimensió, et faci un resum estadístic bàsic de les dades. Si detecta que l'array té més d'una dimensió, ha de mostrar un missatge d'error.

```
In [4]: import numpy as np

def calcular(array):
    if array.ndim != 1:
        print("Error: El array debe de ser de una dimension")
        return None

    # calculos basicos estadisticos
    resum = {
        'màximo': np.max(array),
        'mínimo': np.min(array),
        'promedio': np.mean(array),
        'mediana': np.median(array),
        'desviacion estandar': np.std(array)
    }
    return resum

array = np.array([1, 2, 3, 4, 5])
resum = calcular(array)
print(resum)

{'màximo': 5, 'mínimo': 1, 'promedio': 3.0, 'mediana': 3.0, 'desviacion estandar': 1.414
2135623730951}
```

Crea una funció que et generi un quadrat NxN de nombres aleatoris entre el 0 i el 100.

```
In [5]: import numpy as np

def generar(N):
    cuadrado = np.random.randint(0, 100, size=(N, N))
    return cuadrado

cuadrado_aleatorio = generar(5)
print(cuadrado_aleatorio)

[[11 57 42 62 97]
 [17 53 43 75 58]
 [91 50 32 82  1]
 [ 4 66 51 98 15]
 [63 46 46 86 63]]
```

Crea una funció que dada una taula de dos dimensions (NxM), et calcule els totals per fila i els totals per columna.

```
In [7]: import numpy as np

def calcular(matriz):
    total_fila = np.sum(matriz, axis=1)
    total_columna = np.sum(matriz, axis=0)

    return {'total_fila': total_fila, 'total_columna': total_columna}

matriz = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
totales = calcular(matriz)
print(totales)

{'total_fila': array([ 6, 15, 24]), 'total_columna': array([12, 15, 18])}
```

Implementa manualment una funció que calcule el coeficient de correlació. Informa sobre els seus usos e

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def correlation_and_plot(x, y):
    # formula para calcular la correlacion
    corr = np.corrcoef(x, y)[0,1]

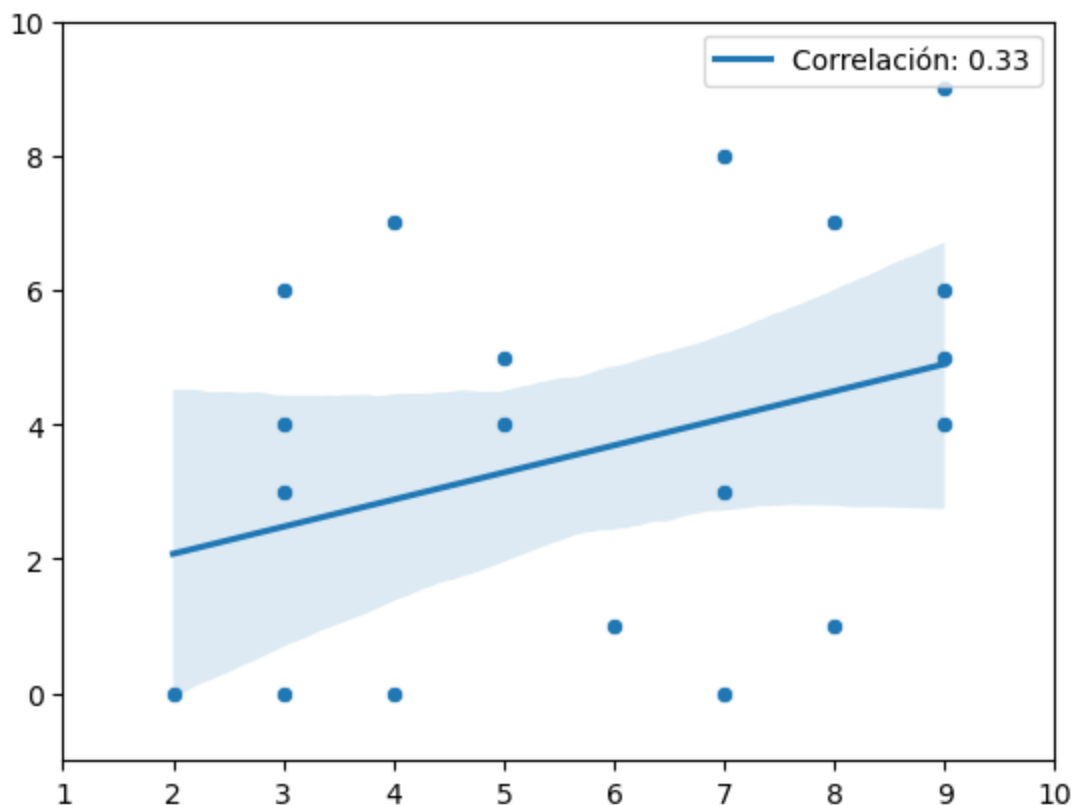
    # linea para grafico de dispersion
    sns.scatterplot(x=x, y=y)

    # Agregar la línea de regresión y el coeficiente de correlación al gráfico
    sns.regplot(x=x, y=y, scatter=False, label=f'Correlación: {corr:.2f}')

    # Configurar los límites de los ejes y mostrar el gráfico
    plt.xlim(np.min(x)-1, np.max(x)+1)
    plt.ylim(np.min(y)-1, np.max(y)+1)
    plt.legend()
    plt.show()

    # Generar datos aleatorios
    x = np.random.randint(0, 10, size=20)
    y = np.random.randint(0, 10, size=20)

    # Calcular el coeficiente de correlación y graficar los datos
    correlation_and_plot(x, y)
```



Para entender mejor el coeficiente de correlación se hace dentro de un gráfico. Por ejemplo creo que es importante según lo investigado cuando estamos analizando los datos de un estudio de mercado, muchas veces nos encontramos con la necesidad de saber si entre dos variables de tipo cuantitativo existe algún tipo de relación. Por ejemplo, a la hora de evaluar un producto o un servicio de una compañía, podemos querer saber si existe alguna relación entre la puntuación que se le ha dado a ese producto y el nivel de

ingresos. Hay diferentes formas de analizar estos datos, pero una de ellas es comprobar si existe correlación entre esas dos variables. Veo que el concepto es muy amplio para seguir aprendiendo

In []: