Sprint 3 - Tarea M3 T01 - Programación numérica, dataframes y análisis estadístico

August 5, 2022

1 Programación numérica, dataframes y análisis estadístico

1.1 Ejercicio 1

Crea una función que dado un Array de una dimensión, te haga un resumen estadístico básico de los datos. Si detecta que el array tiene más de una dimensión, debe mostrar un mensaje de error.

[86]:
```
import numpy as np
import pandas as pd
import scipy
import scipy.stats as st

arr = np.array([1, 2, 3, 4, 5, 5, 8, 9])

print ("Tipo de dato:", arr.dtype)
print ("Forma:", arr.shape)
print ("Valor Mínimo", np.min(arr))
print ("Valor Máximo", np.max(arr))
print ("Promedio:", round(np.mean(arr),2))
print ("Mediana:", np.median(arr))
print ("Moda:", st.mode(arr))
print ("Desviación Estandar:", round(np.std(arr),2))
cuartiles = [0, 25, 50, 75, 100]
print("Cuartiles:", np.percentile(arr, q = cuartiles))


if (arr.ndim) > 1:
    print ('ERROR ---> ARRAY CON MÁS DE UNA DIMENSIÓN')
```

```
Tipo de dato: int32
Forma: (8,)
Valor Mínimo 1
Valor Máximo 9
Promedio: 4.62
Mediana: 4.5
Moda: ModeResult(mode=array([5]), count=array([2]))
Desviación Estandar: 2.6
```

```
Cuartiles: [1.   2.75 4.5  5.75 9.  ]
```

1.2 Ejercicio 2

Crea una función que genere un cuadrado NxN de números aleatorios entre el 0 y el 100.

[64]:
```
import numpy as np
from numpy import random

n = int(input("Introduzca el tamaño del array NxN:"))

cuadrado = random.randint(100, size=(n,n))

print(cuadrado)
```

```
Introduzca el tamaño del array NxN:5
[[ 4 12 59 22 61]
 [21 60 83  6 66]
 [88 53 53 33 11]
 [72 75 52  3  2]
 [99 48 43 32 22]]
```

1.3 Ejercicio 3

Crea una función que dada una tabla de dos dimensiones (NxM), te calcule los totales por fila y los totales por columna.

[87]:
```
import numpy as np
from numpy import random

x = int(input("Introduzca el tamaño del array (# Filas):"))
y = int(input("Introduzca el tamaño del array (# Columnas):"))

matriz = random.randint(100, size=(x,y))
print(matriz)

row = np.sum(matriz, axis = 1)
col = np.sum(matriz, axis = 0)

print("Total por fila:", row)
print("Total por Columna:", col)
```

```
Introduzca el tamaño del array (# Filas):3
Introduzca el tamaño del array (# Columnas):4
[[49 86 59 34]
 [27 36 87 65]
 [48 19 27 15]]
Total por fila: [228 215 109]
Total por Columna: [124 141 173 114]
```

1.4 Ejercicio 4

Implementa manualmente una función que calcule el coeficiente de correlación. Infórmate sobre sus usos e interpretación.

[88]:
```
np.corrcoef(matriz)
```

[88]:
```
array([[ 1.        , -0.27330601, -0.12838183],
       [-0.27330601,  1.        , -0.45786805],
       [-0.12838183, -0.45786805,  1.        ]])
```

El coeficiente de correlación de Pearson mide la asociación lineal entre variables. Su valor varóa entre -1 y 1.

Si este coeficiente es igual a 1 o -1 (o cercano a estos valores) significa que una variable es fruto de una transformación lineal de la otra. Teniendo una relación directa al tratarse de 1 (cuando una variable aumenta, la otra también), mientras que existirá una relación inversa al tratarse de -1 (cuando una variable aumenta la otra disminuye).

Si el valor es 0 (o cercano) no existe relación lineal.

Para el ejemplo existe una pequeña relación inversa para los valores fuera de la diagonal (valores negativos y cercanos a cero)