

Actividad 2.2 – Cuaderno demo UT2 - Ejercicios de ampliación

Link al repositorio: https://github.com/Javiert54/bigData/tree/main/CEIABD_SNS/Actividad%202.2%20%E2%80%93%20Cuaderno%20demo%20UT2%20-%20Ejercicios%20de%20ampliación%C3%B3n

- Ejer 1: Mostrar la columna 4

```
▶ # Ejer 1: Mostrar la columna 4
  x[:,3]

[9] ✓ 0.0s
... array([ 4.,  8., 12.]
```

- Ejer 2: Mostrar la fila 1

```
#Ejer 2: mostrar la fila 1
x[0,:]

[11] ✓ 0.0s
... array([1., 2., 3., 4.]
```

- Ejer 3: Extraer la submatriz de las filas 1 y 3

```
▶ #Ejer 3: Extraer la submatriz de las filas 1 y 3
  x_new = x[1:3,:]
  x_new

[14] ✓ 0.0s
... array([[ 5.,  6.,  7.,  8.],
          [ 9., 10., 11., 12.]])
```

- Ejer 4: Crear un dataframe con una lista (ficticia) de municipios de Gran canaria y el número de habitantes separados por géneros (Masculino, Femenino, etc...)

```
data = pd.DataFrame({
    'Municipio': ['Las Palmas de Gran Canaria', 'Telde', 'Santa Brígida', 'Arucas', 'Gáldar'],
    'Masculino': [200000, 50000, 10000, 15000, 12000],
    'Femenino': [210000, 52000, 11000, 16000, 13000],
    'Otros': [500, 100, 50, 75, 60]
})
data
```

✓ 0.0s

	Municipio	Masculino	Femenino	Otros
0	Las Palmas de Gran Canaria	200000	210000	500
1	Telde	50000	52000	100
2	Santa Brígida	10000	11000	50
3	Arucas	15000	16000	75
4	Gáldar	12000	13000	60

- Ejer 5: Mostrar los tipos de datos asociados a cada columna

```
#Ejer 5: Mostrar los tipos de datos asociados a cada columna
data.dtypes

[19] ✓ 0.0s

... Municipio      object
    Masculino      int64
    Femenino        int64
    Otros           int64
dtype: object
```

- Ejer 6: modificar uno o más valores de una de las columnas

```
#Ejer 6: modificar uno o más valores de una de las columnas
data.at[2, 'Masculino'], data.at[3, 'Femenino'] = 2, 5
data

[27] ✓ 0.0s

...

```

	Municipio	Masculino	Femenino	Otros
0	Las Palmas de Gran Canaria	200000	210000	500
1	Telde	50000	52000	100
2	Santa Brígida	2	11000	50
3	Aruca	15000	5	75
4	Gáldar	12000	13000	60

- Ejer 7: Cambiar alguno de los valores de forma directa

```
#Ejer 7: Cambiar alguno de los valores de forma directa
data.at[2, 'Masculino'], data.at[3, 'Femenino'] = 2, 5
data

1 ✓ 0.0s


```

	Municipio	Masculino	Femenino	Otros
0	Las Palmas de Gran Canaria	200000	210000	500
1	Telde	50000	52000	100
2	Santa Brígida	2	11000	50
3	Aruca	15000	5	75
4	Gáldar	12000	13000	60

- Ejer 8: Obtener estadística básica descriptiva

```
#Ejer 8: Obtener estadística básica descriptiva
data.describe()

[31] ✓ 0.0s

...

```

	Masculino	Femenino	Otros
count	5.000000	5.000000	5.000000
mean	55400.400000	57201.000000	157.000000
std	82950.253772	87655.901142	192.665513
min	2.000000	5.000000	50.000000
25%	12000.000000	11000.000000	60.000000
50%	15000.000000	13000.000000	75.000000
75%	50000.000000	52000.000000	100.000000
max	200000.000000	210000.000000	500.000000

- Ejer 9: Realizar un ejemplo de cada uno de los siguientes comandos:

df_fechas.head() primeros registros
df_fechas.tail() ultimos registros
df_fechas.index nombre de las filas
df_fechas.columns nombre de las columnas
df_fechas.describe() estadística basica descriptiva
df_fechas.T transpuesta

```
print('primeros registros:\n')
df_fechas.head()
```

[27] ✓ 0.0s

... primeros registros:

...

	col1	col2
2022-09-10	-1.489056	-1.104105
2022-09-11	0.476611	-1.461715
2022-09-12	0.871972	1.727704
2022-09-13	0.249003	-0.352231
2022-09-14	-1.257495	-0.202970

```
print('ultimos registros:\n')
df_fechas.tail()
```

✓ 0.0s

ultimos registros:

	col1	col2
2022-11-14	1.318505	-1.023711
2022-11-15	0.011550	-0.443787
2022-11-16	-1.364608	1.252477
2022-11-17	-0.682532	-0.482006
2022-11-18	1.053966	-1.249484

```
print('nombre de las filas:\n')
df_fechas.index
```

✓ 0.0s

nombre de las filas:

DatetimeIndex(['2022-09-10', '2022-09-11', '2022-09-12', '2022-09-13',
'2022-09-14', '2022-09-15', '2022-09-16', '2022-09-17',
'2022-09-18', '2022-09-19', '2022-09-20', '2022-09-21',
'2022-09-22', '2022-09-23', '2022-09-24', '2022-09-25',
'2022-09-26', '2022-09-27', '2022-09-28', '2022-09-29',
'2022-09-30', '2022-10-01', '2022-10-02', '2022-10-03',
'2022-10-04', '2022-10-05', '2022-10-06', '2022-10-07',
'2022-10-08', '2022-10-09', '2022-10-10', '2022-10-11',
'2022-10-12', '2022-10-13', '2022-10-14', '2022-10-15',
'2022-10-16', '2022-10-17', '2022-10-18', '2022-10-19',
'2022-10-20', '2022-10-21', '2022-10-22', '2022-10-23',
'2022-10-24', '2022-10-25', '2022-10-26', '2022-10-27',
'2022-10-28', '2022-10-29', '2022-10-30', '2022-10-31',
'2022-11-01', '2022-11-02', '2022-11-03', '2022-11-04',
'2022-11-05', '2022-11-06', '2022-11-07', '2022-11-08',
'2022-11-09', '2022-11-10', '2022-11-11', '2022-11-12',
'2022-11-13', '2022-11-14', '2022-11-15', '2022-11-16',
'2022-11-17', '2022-11-18'],
dtype='datetime64[ns]', freq='D')

```
print('nombre de las columnas:\n')
df_fechas.columns
```

✓ 0.0s

nombre de las columnas:

Index(['col1', 'col2'], dtype='object')

```
print('estadística basica descriptiva:\n')
df_fechas.describe()
```

✓ 0.0s

estadística basica descriptiva:

	col1	col2
count	70.000000	70.000000
mean	-0.059278	-0.069673
std	1.051065	0.966128
min	-2.859789	-2.927366
25%	-0.815393	-0.737844
50%	0.056524	-0.220849
75%	0.697519	0.545106
max	2.235068	2.046250

```
print('transpuesta:\n')
df_fechas.T
```

✓ 0.0s

transpuesta:

	2022-09-10	2022-09-11	2022-09-12	2022-09-13	2022-09-14
col1	-1.489056	0.476611	0.871972	0.249003	-1.257495
col2	-1.104105	-1.461715	1.727704	-0.352231	-0.202970

2 rows × 70 columns

Ejer 10: Crear una función anónima denominada `mi_funcion_rango`, que pasándole por parámetro la columna "C" de `df` obtenga el rango de los valores que forman parte de dicha columna

```
mi_funcion_rango = lambda a : f"min: {a.min()}, máx: {a.max()}"  
print(mi_funcion_rango(df['C']))  
✓ 0.0s  
min: 1.0, máx: 5.0
```

Ejer 11: Volver a cambiar a CANARIAS

```
#Ejer 11: Volver a cambiar a CANARIAS  
Datos.iloc[4,0]= 'CANARIAS'  
Datos  
✓ 0.0s
```

	Comunidad	Renta	DiferenciaRenta
0	ANDALUCIA	17747	-9.1
1	ARAGON	26512	-7.8
2	ASTURIAS	21149	-9.0
3	BALEARS	22048	-22.7
4	CANARIAS	17448	-18.4

Ejer 12: Filtrando los datos en el dataframe `Datos`, identificar las comunidades autónomas cuya renta

Apartado 1: Renta entre 26000€ y 28000€

Apartado 2: Renta superior a 30000€

```
# Apartado 1: Renta entre 26000€ y 28000€  
apartado_1 = Datos[(Datos['Renta'] >= 26000) & (Datos['Renta'] <= 28000)]  
# Apartado 2: Renta superior a 30000€  
apartado_2 = Datos[Datos['Renta'] > 30000]  
  
print("apartado 1:\n",apartado_1)  
print("apartado 2:\n",apartado_2)  
  
✓ 0.0s
```

apartado 1:

	Comunidad	Renta	DiferenciaRenta
1	ARAGON	26512	-7.8
8	CATALUNYA	27812	-10.9

apartado 2:

	Comunidad	Renta	DiferenciaRenta
12	MADRID	32048	-11.1
15	PAIS VASCO	30401	-10.4