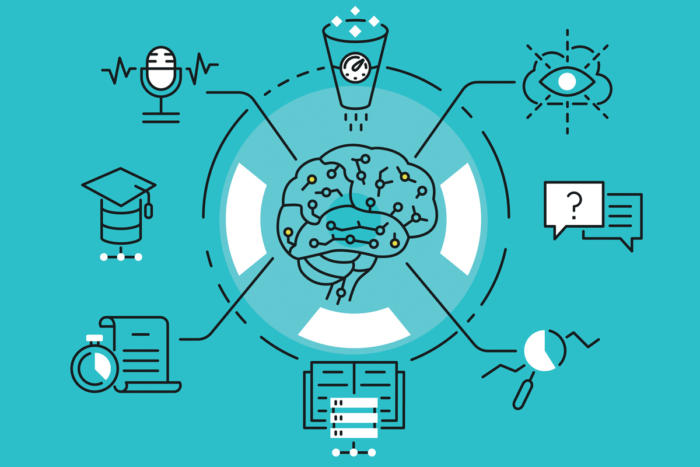
Cuaderno demo UT2 - Ejercicios de ampliación



Adrián Yared Armas de la Nuez

**Contenido**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

[**1. Enunciado 3**](#_2c35568x6wg8)

[**2. Ejercicios 3**](#_3ccb7ndqh1ta)

[**2.1 Ejer 1 3**](#_15b5q6sweoq5)

[**2.1.1 Código 3**](#_watipp8gg8b8)

[**2.1.2 Ejecución 3**](#_toxfinpcm0za)

[**2.2 Ejer 2 3**](#_d2dhmohg7wb)

[**2.2.1 Código 3**](#_hx15ao42o4ak)

[**2.2.2 Ejecución 3**](#_65mcfwkhwrb)

[**2.3 Ejer 3 3**](#_94h5lrbdxcxb)

[**2.3.1 Código 3**](#_mplksc4vdf73)

[**2.3.2 Ejecución 4**](#_ah50xzl7m7f1)

[**2.4 Ejer 4 4**](#_kokoh3qu5skf)

[**2.4.1 Código 4**](#_4jijlkz673q9)

[**2.4.2 Ejecución 4**](#_w628zafzdakz)

[**2.5 Ejer 5 5**](#_35ea4k5f6eeg)

[**2.5.1 Código 5**](#_o5yvdf9lxv55)

[**2.5.2 Ejecución 5**](#_3kck72h1o37g)

[**2.6 Ejer 6 5**](#_x1ekytnblgi9)

[**2.6.1 Código 5**](#_x1t5d0gjhmhw)

[**2.6.2 Ejecución 5**](#_uff7gu5xlehk)

[**2.7 Ejer 7 5**](#_jhztszptsm4z)

[**2.7.1 Código 5**](#_gpujk2jh1qbf)

[**2.7.2 Ejecución 6**](#_vs0yql5c9vnh)

[**2.8 Ejer 8 6**](#_xaz5xunlpjq0)

[**2.8.1 Código 6**](#_kagu10gi8p1w)

[**2.8.2 Ejecución 6**](#_gq2s7k6mo2im)

[**2.9 Ejer 9 6**](#_v8u2hfq9bu1e)

[**2.9.1 Mostrar los primeros 5 registros 6**](#_qd6p0u6y7pmc)

[**2.9.2 Ejecución 7**](#_n9wzqf1m2y8w)

[**2.9.3 Mostrar los últimos 5 registros 7**](#_lkrmygdi83dk)

[**2.9.4 Ejecución 7**](#_bkq5wliurfsg)

[**2.9.5 Mostrar el nombre de las filas (índices) 7**](#_qg2m1qoo5ska)

[**2.9.6 Ejecución 7**](#_abig09x7e9do)

[**2.9.7 Mostrar el nombre de las columnas 8**](#_kq21m7hjzndx)

[**2.9.8 Ejecución 8**](#_lexh313qv0el)

[**2.9.9 Obtener estadísticas básicas descriptivas 8**](#_u3k06bm9zng3)

[**2.9.10 Ejecución 8**](#_uj3jhkm09f88)

[**2.9.11 Obtener la transpuesta del DataFrame 8**](#_vbocf5ahvr3d)

[**2.9.12 Ejecución 8**](#_82mp98gxft5g)

[**2.10 Ejer 10 9**](#_y68wu67gqgrv)

[**2.10.1 Código 9**](#_p5jo3egsaqko)

[**2.10.2 Ejecución 9**](#_76oaiaqhqhvw)

[**2.11 Ejer 11 9**](#_zc62bt8uwrtd)

[**2.11.1 Código 9**](#_jmab2861r03o)

[**2.11.2 Ejecución 10**](#_oxm21l1ulweo)

[**2.12 Ejer 12 10**](#_7gqsege3vdyz)

[**2.12.1 Código 10**](#_zd4v36r29sse)

[**2.12.2 Ejecución 11**](#_8fj83xxcxni)

[**3. Github y Colab 11**](#_lmdngf8dlin3)

## 

## 

## 

## 

## 

## 

## 

## 

## **1. Enunciado**

Utilizando como fuente de partida el cuaderno suministrado por el profesor y titulado

Cuaderno demo UT2 - Ejercicios de ampliación

Url: <https://colab.research.google.com/drive/1fJiRWF1sEbMa1ykSpnocz8Dw4_LB-kw9?usp=sharing>

Completar los ejercicios propuestos

## **2. Ejercicios**

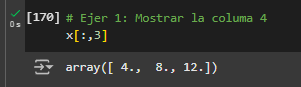
### **2.1 Ejer 1**

#### **2.1.1 Código**

# Ejer 1: Mostrar la columa 4

x[:,3]

#### **2.1.2 Ejecución**



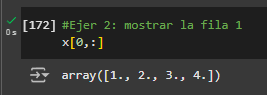
### **2.2 Ejer 2**

#### **2.2.1 Código**

#Ejer 2: mostrar la fila 1

x[0,:]

#### **2.2.2 Ejecución**



### **2.3 Ejer 3**

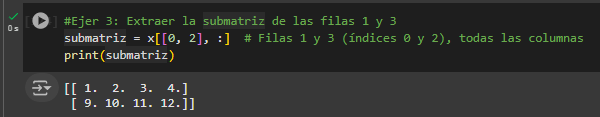
#### **2.3.1 Código**

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

submatriz = x[[0, 2], :] # Filas 1 y 3 (índices 0 y 2), todas las columnas

print(submatriz)

#### **2.3.2 Ejecución**



### **2.4 Ejer 4**

#### **2.4.1 Código**

#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...)

import pandas as pd

# Crear los datos ficticios

datos = {

'Municipio': ['Las Palmas de Gran Canaria', 'Telde', 'San Bartolomé de Tirajana',

'Santa Lucía de Tirajana', 'Arucas', 'Gáldar', 'Ingenio'],

'Total\_Habitantes': [383343, 102647, 53225, 68455, 37927, 24939, 30889],

'Masculino': [189000, 50500, 26500, 33700, 18300, 12500, 15600],

'Femenino': [194343, 52147, 26725, 34755, 19627, 12439, 15289],

'Otros': [343, 147, 75, 55, 27, 39, 89]

}

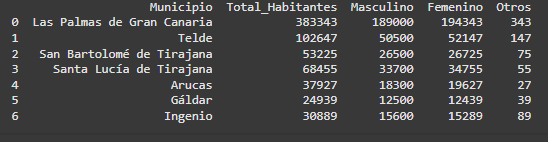
# Crear el DataFrame

df = pd.DataFrame(datos)

# Mostrar el DataFrame

print(df)

#### **2.4.2 Ejecución**



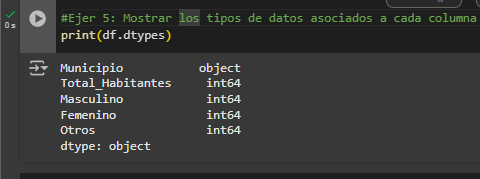
### **2.5 Ejer 5**

#### **2.5.1 Código**

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

print(df.dtypes)

#### **2.5.2 Ejecución**



### **2.6 Ejer 6**

#### **2.6.1 Código**

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

# Modificar valores de la columna 'Masculino' para dos municipios específicos

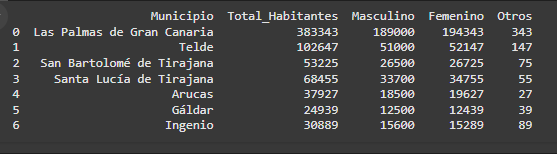
df.loc[df['Municipio'] == 'Telde', 'Masculino'] = 51000

df.loc[df['Municipio'] == 'Arucas', 'Masculino'] = 18500

# Mostrar el DataFrame modificado

print(df)

#### **2.6.2 Ejecución**



### **2.7 Ejer 7**

#### **2.7.1 Código**

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

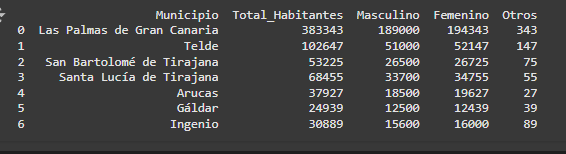
# Modificar directamente un valor en la columna 'Femenino'

df.at[6, 'Femenino'] = 16000 # Cambia el valor en la fila 6, columna 'Femenino'

# Mostrar el DataFrame modificado

print(df)

#### **2.7.2 Ejecución**



### **2.8 Ejer 8**

#### **2.8.1 Código**

#Ejer 8: Obtener estadística basica descriptiva

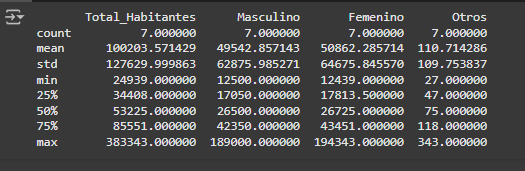
# Obtener estadística descriptiva del DataFrame

estadisticas = df.describe()

# Mostrar las estadísticas

print(estadisticas)

#### **2.8.2 Ejecución**



### **2.9 Ejer 9**

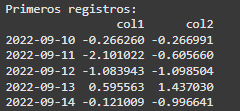
#### **2.9.1 Mostrar los primeros 5 registros**

# Mostrar los primeros 5 registros

print("Primeros registros:")

print(df\_fechas.head())

#### **2.9.2 Ejecución**



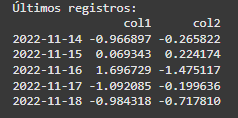
#### **2.9.3 Mostrar los últimos 5 registros**

# Mostrar los últimos 5 registros

print("\nÚltimos registros:")

print(df\_fechas.tail())

#### **2.9.4 Ejecución**



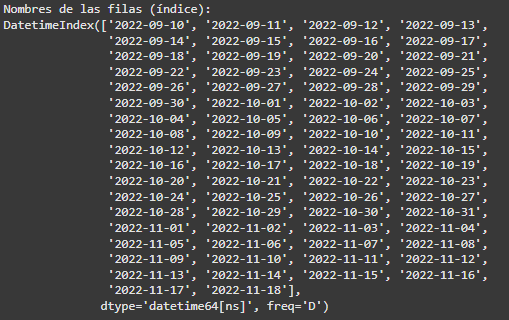
#### **2.9.5 Mostrar el nombre de las filas (índices)**

# Mostrar el nombre de las filas (índices)

print("\nNombres de las filas (índice):")

print(df\_fechas.index)

#### **2.9.6 Ejecución**



#### **2.9.7 Mostrar el nombre de las columnas**

# Mostrar el nombre de las columnas

print("\nNombres de las columnas:")

print(df\_fechas.columns)

#### **2.9.8 Ejecución**



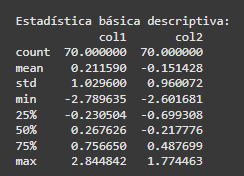
#### **2.9.9 Obtener estadísticas básicas descriptivas**

# Obtener estadísticas básicas descriptivas

print("\nEstadística básica descriptiva:")

print(df\_fechas.describe())

#### **2.9.10 Ejecución**



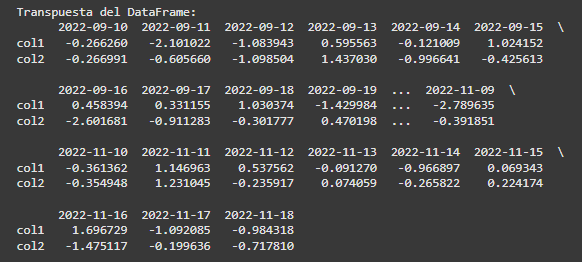
#### **2.9.11 Obtener la transpuesta del DataFrame**

# Obtener la transpuesta del DataFrame

print("\nTranspuesta del DataFrame:")

print(df\_fechas.T)

#### **2.9.12 Ejecución**



### **2.10 Ejer 10**

#### **2.10.1 Código**

#Ejer 10: Crear una función anónima denomminada mi\_funcion\_rango, que pasándole por parámetro la columna "C" de df obtenga el rango

# de los valores que forman parte de dicha columna

# Crear un DataFrame de ejemplo con una columna "C"

df = pd.DataFrame({

"C": [10, 20, 15, 25, 30]

})

# Crear la función anónima para calcular el rango

mi\_funcion\_rango = lambda columna: columna.max() - columna.min()

# Aplicar la función a la columna "C"

rango\_columna\_c = mi\_funcion\_rango(df["C"])

# Mostrar el rango calculado

print("El rango de la columna 'C' es:", rango\_columna\_c)

#### **2.10.2 Ejecución**



### **2.11 Ejer 11**

#### **2.11.1 Código**

#Ejer 11: Volver a cambiar a CANARIAS

Datos.iloc[4,0]= 'CANARIAS'

Datos

#### **2.11.2 Ejecución**



### **2.12 Ejer 12**

#### **2.12.1 Código**

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

# Apartado 1: Filtramos las comunidades con renta entre 26000€ y 28000€

comunidades\_26000\_28000 = Datos[(Datos["Renta"] >= 26000) & (Datos["Renta"] <= 28000)]

print("Comunidades autónomas con renta entre 26000€ y 28000€:")

print(comunidades\_26000\_28000[["Comunidad", "Renta"]])

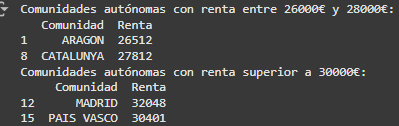
# Apartado 2: Filtramos las comunidades con renta superior a 30000€

comunidades\_superior\_30000 = Datos[Datos["Renta"] > 30000]

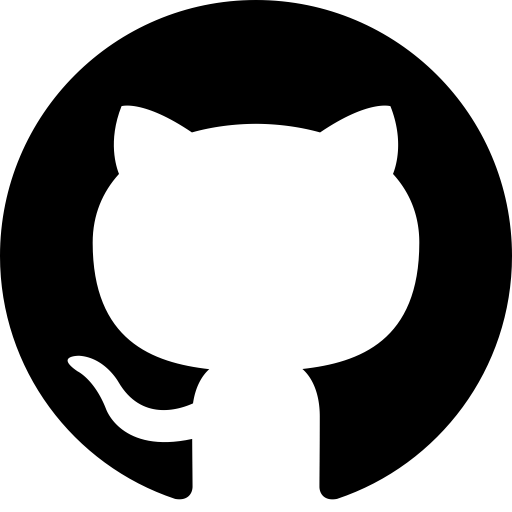
print("Comunidades autónomas con renta superior a 30000€:")

print(comunidades\_superior\_30000[["Comunidad", "Renta"]])

#### **2.12.2 Ejecución**



## **3. Github y Colab**

[](https://github.com/AdrianYArmas/IaBigData/tree/main/SNS/2%20T%C3%A9cnicas%20y%20herramientas%20de%20sistemas%20de%20aprendizaje%20autom%C3%A1tico%20(machine%20learning)/2.2%20Ejercicios%20de%20ampliaci%C3%B3n) [](https://colab.research.google.com/drive/1j5pSEPWMQPZLoAiujquTUi-K8fdDDKLt?usp=sharing)