Práctica II

2025

Enunciado

- La práctica debe entregarse antes del **28 de noviembre de 2025**.
- El resultado final deberá incluir un fichero *Jupyter Notebook* (`.ipynb`) y los ficheros `.py` correspondientes.
- Cada ejercicio debe resolverse en una celda independiente. Puede utilizarse un único bloque de código o una única función por ejercicio, aunque también es posible definir funciones adicionales si se considera oportuno.
- Se debe realizar un `git commit` por cada apartado de cada ejercicio. Una vez completado el trabajo, desde la terminal se deberá ejecutar el siguiente comando:

  ```
  git log > log.txt
  ```

  Este comando generará un fichero `log.txt`, que también deberá subirse junto con el resto de archivos de la práctica.

Ejercicios

1. **Creación de *DataFrames***
   (a) Crea un `DataFrame` `df1` con una columna llamada `numeros` que contenga los valores del 0 al 19.
   (b) Crea un `DataFrame` `df2` con una columna `aleatorios` de tamaño 10 con números aleatorios entre 0 y 1.
   (c) Combina los `DataFrames` anteriores `df1` y `df2` en un nuevo `DataFrame` llamado `df3`.
   (d) Guarda el `DataFrame` `df3` en un fichero `.csv`.

2. **Indexación y selección (10 pts)**
   Dado el `DataFrame`:

```python
import pandas as pd
df = pd.DataFrame({
    "A": range(1, 11),
    "B": [5,3,6,8,2,9,1,7,4,10]
})
```

   (a) Selecciona las filas donde el valor de la columna `B` sea mayor que 5.

   (b) Cambia los valores menores que 4 en la columna `B` a `NaN`. Para usar el valor `NaN` hay que utilizar el código `pd.NA`.

3. **Análisis de Datos**

   Descargad el conjunto de datos disponible en la siguiente página web y utilizad el fichero `winequality-red.csv`. Este dataset contiene información físico-química sobre diferentes vinos tintos de Portugal.

   Se pide:

   (a) Obtén los estadísticos descriptivos más relevantes (*describe*) de cada columna. Explica brevemente qué mide cada una de ellas.

   (b) Convierte la columna `quality` en una columna categórica con tres categorías:
   - **bueno**
   - **regular**
   - **malo**

   (c) Selecciona los vinos clasificados como *malo* y elimínelos del `DataFrame`.

   (d) Convierte los valores de la columna `citric acid` a porcentajes.

   (e) Obtén los siete vinos con mayor valor de `pH`.

   Ahora crea un fichero `.py` que defina las siguientes funciones:

   (a) `eliminaDecimales(df, col)`: recibe un `DataFrame` y el nombre de una columna, y crea una nueva columna llamada `<col>_sin_decimales` con el mismo valor pero truncado (sin decimales).

   (b) `calculaRatio(df, col1, col2)`: calcula el ratio entre dos columnas y genera una excepción si algún valor del ratio es mayor que 1.

Aplica la primera función a la columna `alcohol` y la segunda a las columnas `total sulfur dioxide` y `free sulfur dioxide`.

4. **Combinando información en pandas**

   Dispones de los siguientes `DataFrames`:

```python
import pandas as pd

estudiantes = pd.DataFrame({
    "id": [1, 2, 3, 4],
    "nombre": ["Ana", "Luis", "Marta", "Jorge"]
})

notas = pd.DataFrame({
    "id": [1, 2, 2, 5],
    "asignatura": ["Matemáticas", "Historia", "Química", "Biología"],
    "nota": [8.5, 7.0, 6.5, 9.0]
})
```

   Realiza las siguientes tareas:

   (a) Realiza un `merge` interno (*inner*) entre ambos `DataFrames`. ¿Cuántas filas se obtienen? ¿Por qué?

   (b) Realiza un `merge` externo (*outer*). Explica las filas que aparecen con valores nulos.

   (c) Realiza un `left merge` usando `estudiantes` como tabla izquierda. ¿Qué información se pierde respecto al `outer`?

   (d) Realiza un `right merge` usando `notas` como tabla derecha. ¿Qué estudiantes desaparecen en el resultado?

   (e) Añade una columna llamada `curso` al `DataFrame estudiantes`. Después, combina los datos utilizando un `merge` múltiple por `id` y explica el resultado.

5. **Agregaciones y operaciones con `groupby`**

   En este ejercicio se trabajará con un conjunto de datos que contiene información sobre ventas realizadas por una empresa en distintas regiones.

   Considera el siguiente `DataFrame`:

```python
import pandas as pd

ventas = pd.DataFrame({
    "region": ["Norte","Norte","Sur","Este","Sur","Este","Oeste",
               "Oeste","Norte","Sur"],
    "vendedor": ["Ana","Luis","Ana","Jorge","Luis","Ana","Marta",
                 "Luis","Ana","Jorge"],
    "ventas": [1200, 800, 950, 1100, 400, 1600, 700, 450, 895, 900],
    "producto": ["A","A","B","A","B","A","B","A","B","B"]
})
```

   Realiza las siguientes tareas:

   (a) Obtén el total de ventas por región.

   (b) Calcula la media, el valor máximo y el mínimo de las ventas agrupadas por vendedor.

   (c) Calcula el número de ventas realizadas por cada combinación de `region` y `producto`.

   (d) Crea una función llamada `rango_ventas` en un fichero `.py` que calcule la diferencia entre el valor máximo y mínimo de un grupo. Aplícala a las ventas agrupando por región.

   (e) Identifica la región con el mayor número total de ventas y muestra solo esa región.

6. **Gestión de datos ausentes**. Utilizando los mismos datos que en el ejercicio cuatro realiza un `merge` por la izquierda, siendo el primer elemento el DataFrame con la información de estudiantes y el segundo elemento el DataFrame de notas.

   (a) Cuenta cuantos NaNs aparecen en cada columna.

   (b) Propón dos estrategias diferentes para tratar los NaNs del DataFrame resultantes y aplica estás estrategias. Justifica las estrategias.