

Trabajo Práctico sobre Pandas

Ejercicio 1

Dada lista con los siguientes valores de ventas diarias de una tienda en una semana:

`ventas = [120, 150, 90, 200, 210, 130, 160]`, realiza las siguientes tareas:

1. Crea una serie de Pandas con los días de la semana como índice (asumiendo que el primer valor corresponde al Lunes).
2. Calcula y muestra la suma total de las ventas de la semana. Almacena este valor en una variable llamada: `suma_total_ventas`
3. Encuentra y muestra el día de mayores ventas. Almacena este valor en una variable llamada: `dia_mayores_ventas`
4. Calcula y muestra el promedio de ventas de la semana. Almacena este valor en una variable llamada: `promedio_ventas`

Ejercicio 2

Crea una serie de Pandas llamada: `serie_numeros` que contenga los números del 1 al 20. Luego, escribe un código que filtre y muestre solo aquellos números que sean mayores que 10. Utiliza una variable llamada `filtro` para almacenar la serie filtrada.

Ejercicio 3

Dada una serie de Pandas que contiene los siguientes valores: `[18, 22, 7, 9, 15, 8]`, filtra y muestra solo aquellos valores que sean pares.

1. Primero, crea una serie de valores booleanos que represente la condición. Nombra a esta variable como: `condicion_valores_pares`
2. Luego aplica esta serie para filtrar los valores originales.

Ejercicio 4

Crea una serie de Pandas llamada `frutas` y que contenga los siguientes elementos: `["manzana", "banana", "cereza", "durazno", "frambuesa"]`.

1. Escribe un código que filtre y muestre solo aquellos elementos que contengan la letra "e" en su nombre. Almacena los elementos filtrados en una variable llamada `frutas_con_e`
2. Utiliza una condición que aplique un método de strings para lograr este filtrado.

Ejercicio 5

Utilizando la biblioteca Pandas, crea un DataFrame llamado `df_empleados` que contenga dos columnas: `nombre` y `edad`.

La columna `nombre` debe contener los nombres de tres empleados: 'Ana', 'Luis' y 'Carlos'

La columna `edad` debe contener las edades correspondientes: 30, 25 y 40

Finalmente, muestra el DataFrame `df_empleados` utilizando la función `print()`.

Ejercicio 6

Dado el DataFrame `df_empleados` creado en el ejercicio anterior:

- La columna `nombre` debe contener los nombres de tres empleados: 'Ana', 'Luis' y 'Carlos'
 - La columna `edad` debe contener las edades correspondientes: 30, 25 y 40
1. Utiliza el método adecuado para seleccionar solo la columna `edad` y almacénala en una variable llamada `edades`.
 2. Luego, muestra el contenido de `edades` utilizando la función `print()`.
 3. Asegúrate de demostrar que el tipo de dato de `edades` es una Serie de Pandas. Utilizando la función `type()`.
 4. Explorar sus atributos principales: `shape`, `columns`, y `index` utilizando las siguientes variables respectivamente: `shape_df`, `columns_df`, `index_df` Imprime los resultados de cada variable..

El resultado esperado es el siguiente:

```
(3, 2)
Index(['nombre', 'edad'], dtype='object')
RangeIndex(start=0, stop=3, step=1)
```

Ejercicio 7

Crea un DataFrame llamado `datos_clima` usando Pandas para cargar un archivo CSV llamado "`clima.csv`" que contiene datos sobre temperaturas y precipitaciones en diferentes ciudades. Asegúrate de importar Pandas antes de intentar cargar el archivo.

Considera que tu archivo "`clima.csv`" se encuentra en la misma ubicación que el archivo de tu código, por lo cual puedes importarlo usando su nombre incluyendo la extensión `.csv`

1. Aplica el método `describe()` para explorar la estructura del DataFrame y obtener un resumen estadístico de las columnas numéricas.
2. Asegurate de almacenar el resultado en una variable llamada `describe_df`

3. Extraer exactamente la primera fila de `'datos_clima'`. Almacena el contenido en una variable llamada `head_df`
4. Extraer exactamente la última fila de `'datos_clima'`. Almacena el contenido en una variable llamada `tail_df`

Ejercicio 8

Dada una tabla (Diccionario) de ventas que contiene información sobre productos vendidos, incluyendo ID, Producto, Cantidad y Precio,

```
1. data = {  
2.     'ID': [1, 2, 3, 4, 5],  
3.     'Producto': ['Producto A', 'Producto B', None, 'Producto D', 'Producto E'],  
4.     'Cantidad': [10, 20, 30, None, 50],  
5.     'Precio': [100, 200, 300, 400, None]  
6. }
```

Escribe un código en Python usando Pandas para contar los valores nulos por columnas.

Ejercicio 9

Tu tabla de ventas, la columna "Precio" tiene algunos valores nulos.

```
1. data = {  
2.     'ID': [1, 2, 3, 4],  
3.     'Producto': ['Producto A', 'Producto B', 'Producto C', 'Producto D'],  
4.     'Cantidad': [10, 20, 30, 40],  
5.     'Precio': [100, None, 300, None]  
6. }
```

Escribe un código en Python usando Pandas para reemplazar los valores nulos en la columna Precio por el promedio de los precios no nulos de esa columna.

Ejercicio 10

El conjunto de datos proporcionado tiene algunas entradas duplicadas.

```
1. data = {  
2.     'ID': [1, 2, 3, 4, 1],  
3.     'Producto': ['Producto A', 'Producto B', 'Producto C', 'Producto D', 'Producto A'],  
4.     'Cantidad': [10, 20, 30, 40, 50],  
5.     'Precio': [100, 200, 300, 400, 100]  
6. }
```

Escribe un script en Python usando Pandas para eliminar las filas duplicadas en el conjunto de datos. Almacena el resultado en una variable llamada: `df_sin_duplicados`

Ejercicio 11

Partiendo del DataFrame `df` mencionado anteriormente:

1. `# Datos proporcionados`
2. `data = {`
3. `'Nombre': ['Ana', 'Luis', 'Carlos', 'Sara'],`
4. `'Edad': [25, 30, 22, 27],`
5. `'Ciudad': ['Madrid', 'Barcelona', 'Valencia', 'Bilbao']`
6. `}`

Realiza las siguientes operaciones:

1. Crea un dataframe llamado `df`
2. Agrega una nueva columna llamada `Edad_en_10_años` que contenga la edad de las personas dentro de 10 años.
3. Modifica la columna `Ciudad` para que todas las ciudades estén en mayúsculas.
4. Crea una nueva columna `Es_Mayor_de_25` que contenga valores booleanos: `True` si la persona tiene 25 años o más, y `False` en caso contrario.