PANDAS 2: LIMPIEZA Y TRANSFORMACIÓN DE COLUMNAS

PYTHON PROGRAMMING

nuclio digital school

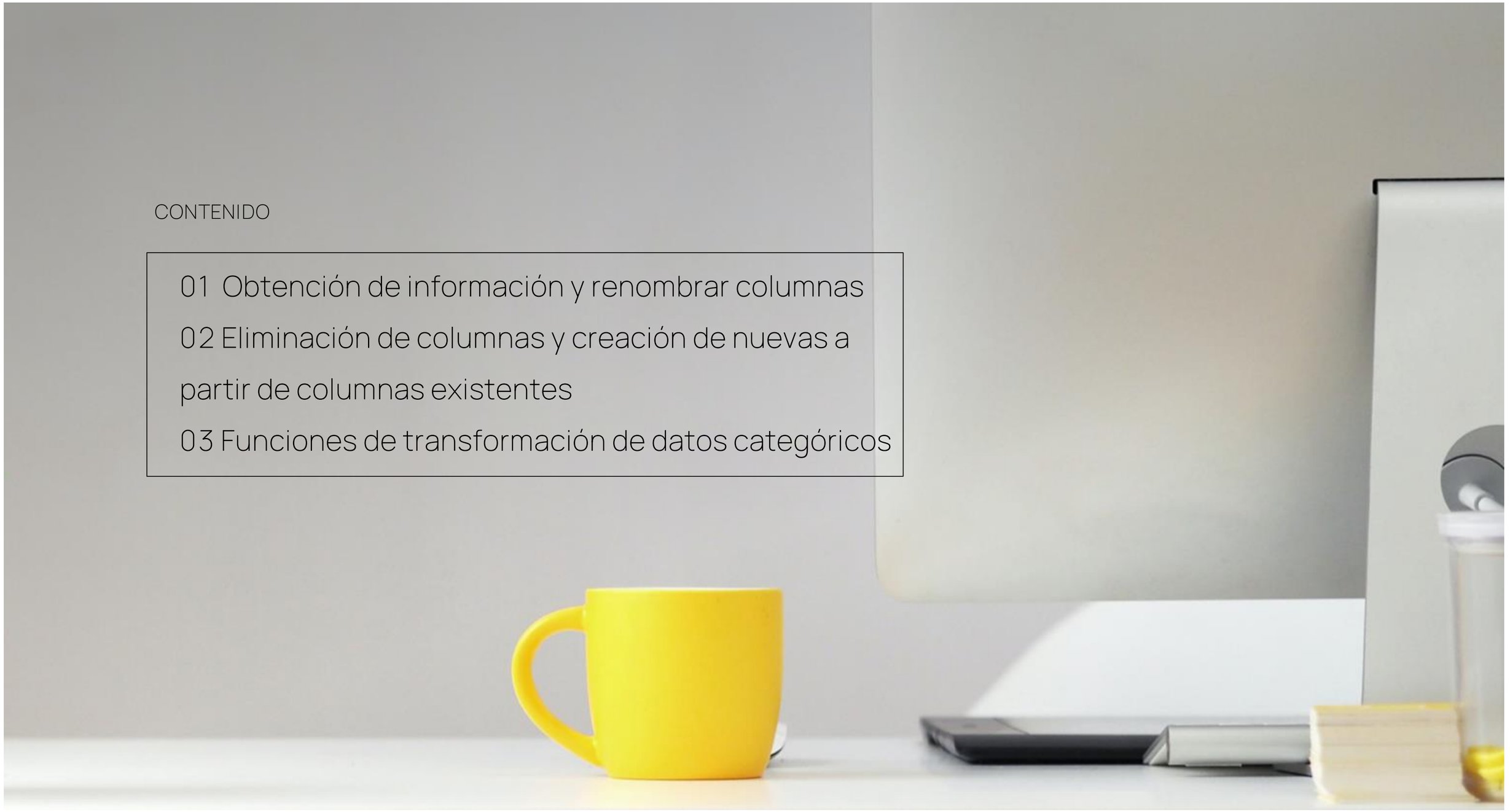
Learn [to be] the future

Anna Perea Navarro

Senior Data Scientist

Graduada en Matemáticas por la Universidad de Barcelona y máster en Ingeniería Computacional y Matemática en la Universidad Rovira y Virgili.

CONTENIDO



01 Obtención de información y renombrar columnas

Obtención de información sobre el DataFrame

- Además de hacer una previsualización de los datos, podemos obtener más información sobre lo que contiene nuestro DataFrame mediante el método `.info()`
- Al aplicarlo obtendremos el número de filas que tiene df, el número de columnas, de que tipo es cada columna y el número de registros no nulos en cada una de estas.

Renombrar columnas

Si observamos que las columnas de nuestro DataFrame tienen nombres imprecisos, que pueden llevar a confusión, hay algún typo en ellos, o bien son demasiado largos e incómodos para su recurrente escritura a la hora de analizar los datos; podemos renombrar las columnas con el método `.rename()`

Sea df un DataFrame, para renombrar las columnas pasaremos un diccionario al parámetro columns con los nombres actuales de
las columnas que queremos editar y los nuevos nombres que tendrán:
`df.rename(columns={'name1' : 'new_name1', 'name2' : 'new_name2'})`

Recuerda aplicar el parámetro `inplace=True` si quieres que se modifique el propio df.

02 Eliminación de columnas y creación de nuevas a partir de columnas existentes

Crear una nueva columna

A partir de columnas ya existentes en nuestro DataFrame, podemos crear nuevas columnas que nos proporcionen nuevas métricas para avanzar en nuestro análisis de los datos.

Para ello, podemos usar las operaciones presentadas en módulos anteriores (+, -, /, *, %) e igualar el resultado a la selección de la nueva columna que queramos generar.

Por ejemplo, si quisiéramos obtener una nueva columna que fuera la suma de los ingresos de 2019 y 2020 a partir de las columnas de ingresos 'ing_2019' e 'ing_2020' podríamos hacerlo de la siguiente manera:

```
df['ing_2019_2020'] = df['ing_2019'] + df['ing_2020']
```

Eliminar columnas

Sea df un DataFrame si queremos eliminar una columna de nuestro DataFrame llamada 'col', podemos usar el método .drop() con la siguiente sintaxi:

```
df.drop(columns=['col'])
```

Recuerda añadir el parámetro inplace=True para modificar el propio df.

Es posible eliminar una sola columna o múltiples columnas a la vez:

```
df.drop(columns=['col1', 'col2', 'col3'])
```

03 Funciones de transformación de datos categóricos

Existen numerosas funciones para llevar a cabo procesos de transformación en los datos que contienen las columnas de los DataFrames. Veamos a continuación algunas de las más útiles:

División de una columna en múltiples

`df['col'].str.split('separator', expand=True)`

Donde 'separator' indica el carácter por el que queremos separar la información (p.ej. un espacio o un guión)

Transformación de texto

`df['col'].str.upper()` para convertir el texto en mayúsculas

`df['col'].str.lower()` para convertir el texto en minúsculas

Combinación de columnas en una

`df['col'] = df['col1'].str.cat(df['col2'], sep = 'separator')`

Donde 'col' será la columna que contendrá la combinación de 'col1' y 'col2' y 'separator' será el carácter que usemos para combinar los datos de 'col1' y de 'col2', p. ej. un espacio

Substitución de caracteres

`df['col'].str.replace('value', 'new_value')`

Donde 'value' indica el carácter que queremos substituir y 'new_value' indica el valor por el que lo queremos reemplazar