

INTRODUCCIÓN A PANDAS y PANDAS SERIES

PYTHON PROGRAMMING



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

1 Introducción a Pandas

1.1 ¿Qué es Pandas?

2 Pandas Series

2.1 Qué es una Serie de Pandas

2.2 Creación de una Serie

2.3 Cómo acceder a sus elementos

2.4 Atributos y métodos



1. INTRODUCCIÓN A PANDAS

¿Qué es Pandas?

Pandas es una de las librerías más populares de Python, está construida a partir de Numpy. Es un recurso esencial en Data Science y Machine Learning, ya que ofrece estructuras y funciones muy útiles para la manipulación y el tratamiento de datos.

¿Qué podemos hacer con Pandas?

- Mediante Pandas podemos definir Series y DataFrames (tablas)
- Permite leer y escribir tablas de datos a partir de distintos tipos de archivos
- Ofrece métodos para conocer nuestros datos, analizarlos, transformarlos y limpiarlos. Además de calcular estadísticas.
- Viene integrado con Matplotlib, una librería que nos permitirá realizar gráficos de manera muy sencilla

Series			Series			DataFrame	
apples			oranges			apples	oranges
0	3	+	0	0	=	0	3
1	2		1	3		1	2
2	0		2	7		2	0
3	1		3	2		3	1
							2



2. PANDAS SERIES

¿Qué es una Serie de Pandas?

Podemos verlo como una especie de lista de Python etiquetada, o parecido a un diccionario en formato tabular, pero sobre todo lo definiríamos como un DataFrame (tabla de datos) de una sola columna

Series 1		Series 2		Series 3		DataFrame			
Mango		Apple		Banana		Mango	Apple	Banana	
0	4	0	5	0	2	0	4	5	2
1	5	1	4	1	3	1	5	4	3
2	6	2	3	2	5	2	6	3	5
3	3	3	0	3	2	3	3	0	2
4	1	4	2	4	7	4	1	2	7



2. PANDAS SERIES

Creación de una Serie

Mediante un diccionario

```
ID_products = {  
    01667: 'Milk',  
    76554: 'Cookies',  
    43382: 'Juice'  
}  
  
pd.Series(ID_products)  
⇒  
    01667  Milk  
    76554  Cookies  
    43382  Juice  
dtype: object
```

Mediante una lista

```
fruits = ['apple', 'orange', 'watermelon']  
pd.Series(fruits)  
⇒  
    0 apple  
    1 orange  
    2 watermelon  
dtype: object
```



La columna adicional que vemos, llamada Índice, no tiene porqué ser numérica. Podemos definirla con otros datos numéricos o strings.

```
pd.Series([9.5, 8.8, 8], index=['Dune', 'Gladiator', 'Mean Girls'])  
⇒  
    Dune      9.5  
    Gladiator  8.8  
    Mean Girls 8  
dtype: float64
```



2. PANDAS SERIES

Cómo acceder a sus elementos

Mediante posición o índice

Ejemplo 1

```
fruits = ['apple', 'orange', 'watermelon']
fruits_serie = pd.Series(fruits)
```

⇒
0 apple
1 orange
2 watermelon
dtype: object

```
fruits_serie[0] ⇒ apple
```

Ejemplo 2

```
s = pd.Series([1,2,3,4,5], index = ['a', 'b', 'c', 'd', 'e'])
```

s[:3] ⇒
a 1
b 2
c 3
dtype: int64

s[-3:] ⇒
c 3
d 4
e 5
dtype: int64

s[2:4] ⇒
c 3
d 4
dtype: int64

s[::2] ⇒
a 1
c 3
e 5
dtype: int64

Mediante index label

```
film_scores = pd.Series( [9.5, 8.8, 8], index=['Dune','Gladiator','Mean Girls'])
```

⇒
Dune 9.5
Gladiator 8.8
Mean Girls 8
dtype: float64

```
film_scores['Dune'] ⇒ 9.5
```

film_scores[['Dune', 'Gladiator']]
⇒
Dune 9.5
Gladiator 8.8
dtype: float64

film_scores['Dune': 'Mean Girls']
⇒
Dune 9.5
Gladiator 8.8
Mean Girls 8
dtype: float64



2. PANDAS SERIES

Atributos y Métodos

Los objetos de Python tienen Atributos y Métodos. Los Atributos son la forma que tenemos para encontrar información sobre un elemento sin manipularlo ni modificarlo. En cambio, los Métodos sí que aplican alguna acción sobre el objeto, sea en una copia de este o en el propio elemento.

Atributos

Sea `s` una serie de Pandas:

- `s.values` devuelve un array con los valores de una serie
- `s.dtype` devuelve el tipo de los datos de la serie
- `s.size` devuelve el número de elementos de una serie

Métodos

Sea `s` una serie de Pandas:

- | | | | |
|---------------------------|----------------------------|--------------------------------|------------------------------|
| • <code>s.head()</code> | • <code>s.sum()</code> | • <code>s.unique()</code> | • <code>s.nlargest()</code> |
| • <code>s.tail()</code> | • <code>s.product()</code> | • <code>s.nunique()</code> | • <code>s.nsmallest()</code> |
| • <code>s.mean()</code> | • <code>s.max()</code> | • <code>s.sort_values()</code> | |
| • <code>s.median()</code> | • <code>s.min()</code> | • <code>s.sort_index()</code> | |

