

# Data Analysis con Python

Juan Felipe Acevedo Pérez  
Monitor Junior

Curso libre:  
Data Analysis con  
Python

Monitor encargado:  
Juan Felipe Acevedo  
Pérez

**Correo:** [uniic\\_bog@unal.edu.co](mailto:uniic_bog@unal.edu.co)

**Tel:** 3165000 **Ext:** 12301

# 2

## Sesión 2

*Uso de la vectorización e introducción a Pandas*

Correo: [uniic\\_bog@unal.edu.co](mailto:uniic_bog@unal.edu.co)

Teléfono: 3165000 ext 12301

# Contenido



**Funciones  
Universales**



**Vectorización**



**Pandas**

# Funciones Universales

Una función universal, o ufunc, es una función que realiza operaciones elementales sobre datos en ndarrays. Puede pensar en ellos como envoltorios vectorizados rápidos para funciones simples que toman uno o más valores escalares y producen uno o más resultados escalares.

Pueden ser categorizadas en Unary ufuncs o Binary ufuncs dependiendo de la cantidad de argumentos que requieren.

Table 4-3. Unary ufuncs

Function	Description
abs, fabs	Compute the absolute value element-wise for integer, floating-point, or complex values
sqrt	Compute the square root of each element (equivalent to <code>arr ** 0.5</code> )
square	Compute the square of each element (equivalent to <code>arr ** 2</code> )
exp	Compute the exponent $e^x$ of each element
log, log10, log2, log1p	Natural logarithm (base $e$ ), log base 10, log base 2, and $\log(1 + x)$ , respectively
sign	Compute the sign of each element: 1 (positive), 0 (zero), or -1 (negative)
ceil	Compute the ceiling of each element (i.e., the smallest integer greater than or equal to that number)
floor	Compute the floor of each element (i.e., the largest integer less than or equal to each element)
rint	Round elements to the nearest integer, preserving the dtype
modf	Return fractional and integral parts of array as a separate array
isnan	Return boolean array indicating whether each value is NaN (Not a Number)
isfinite, isinf	Return boolean array indicating whether each element is finite (non- <code>inf</code> , non-NaN) or infinite, respectively
cos, cosh, sin, sinh, tan, tanh	Regular and hyperbolic trigonometric functions
arccos, arccosh, arcsin, arcsinh, arctan, arctanh	Inverse trigonometric functions
logical_not	Compute truth value of not <code>x</code> element-wise (equivalent to <code>~arr</code> ).



Table 4-4. Binary universal functions

Function	Description
add	Add corresponding elements in arrays
subtract	Subtract elements in second array from first array
multiply	Multiply array elements
divide, floor_divide	Divide or floor divide (truncating the remainder)
power	Raise elements in first array to powers indicated in second array
maximum, fmax	Element-wise maximum; fmax ignores NaN
minimum, fmin	Element-wise minimum; fmin ignores NaN
mod	Element-wise modulus (remainder of division)
copysign	Copy sign of values in second argument to values in first argument

Function	Description
greater, greater_equal, less, less_equal, equal, not_equal	Perform element-wise comparison, yielding boolean array (equivalent to infix operators >, >=, <, <=, ==, !=)
logical_and, logical_or, logical_xor	Compute element-wise truth value of logical operation (equivalent to infix operators &  , ^)

# Vectorización y Funciones Universales



- NumPy es tan importante en el mundo de la ciencia de datos de Python, por su capacidad de proporcionar una interfaz fácil y flexible para el cálculo *optimizado* con matrices de datos.

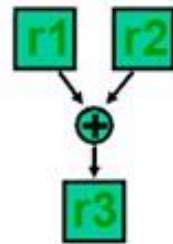
¿De qué depende la velocidad de ejecución?

- El cálculo en NumPy arrays puede ser muy rápido o muy lento. La clave para hacerlo rápido es usar operaciones **vectorizadas**, generalmente implementadas a través de las funciones universales de NumPy (ufuncs). La necesidad de los ufuncs de NumPy, se enfoca en la posibilidad de hacer cálculos repetidos en los elementos de la matriz de forma mucho más **eficientes**.

# ¿Qué es la vectorización?

- Las computadoras modernas cuentan con procesos vectoriales que permiten resultados más eficientes. Existen dos tipos de procesadores:
  1. *Procesador escalar*: Trabaja sobre un valor a la vez.
  2. *Procesador vectorial*: Puede operar sobre varios valores a la vez (un arreglo 1D o vector).
- Los procesadores vectoriales (presentes en los computadores del nuevo siglo), permiten ejecutar operaciones vectoriales que realizan operaciones simultáneamente y no de forma iterativa. En lugar de recorrer dato por dato (ciclo for), aplicamos la función a todos los datos de forma simultánea.

### SCALAR (1 operation)



ADD r3, r1, r2

### VECTOR (N operations)



ADDV v3, v1, v2

Suma  
vectorial vs  
suma escalar



# Pandas

- Pandas serán una herramienta fundamental para cualquier analista. Contiene estructuras de datos y herramientas de manipulación de datos diseñadas para hacer que *la limpieza y el análisis de datos sean rápidos y fáciles en Python*.
- Pandas adopta partes significativas del estilo idiomático de computación basada en arreglos de NumPy, especialmente funciones basadas en arreglos y una preferencia por el procesamiento de datos **sin bucles for**.



# ¿Pandas o NumPy?

- Si bien pandas adopta muchos lenguajes de codificación de NumPy, la mayor diferencia es que **pandas** está diseñado para trabajar con *datos tabulares o heterogéneos*. **NumPy**, por el contrario, es más adecuado para trabajar con *datos de matriz numérica homogénea*.



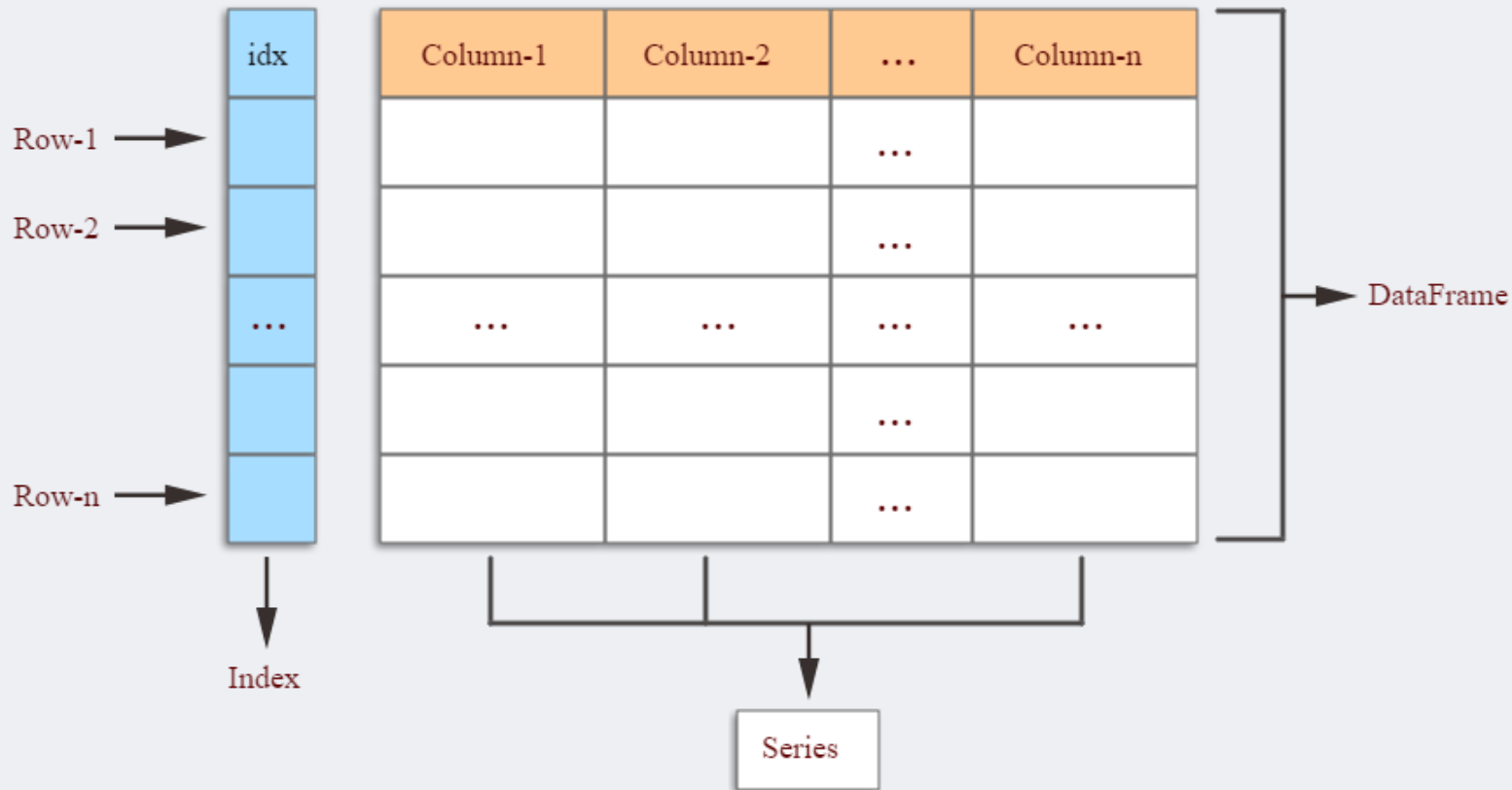
- Pandas presenta una gran ventaja: Desde que se convirtió en un proyecto de código abierto en 2010, pandas ha sido una librería bastante grande que se puede aplicar en un amplio conjunto de **casos de uso del mundo real**. La comunidad de desarrolladores ha crecido a más de 800 colaboradores distintos, que han ayudado a construir el proyecto a medida que lo utilizan para resolver sus *problemas de datos cotidianos*.

# Tipos de datos en Pandas

Para comenzar con pandas, deberá familiarizarse con sus dos principales caballos de batalla: **Series y DataFrame**. Si bien no son estructuras que brinden una solución universal para todos los problemas, permiten una base sólida y fácil de usar para la mayoría de las aplicaciones.

Series 1		Series 2		Series 3		DataFrame
<b>Mango</b>		<b>Apple</b>		<b>Banana</b>		<b>Mango Apple Banana</b>
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

## Pandas Data structure



## Estructuras de Pandas

- **Series:** Una serie es un objeto similar a una matriz unidimensional que contiene una *secuencia de valores* (de tipos similares a los tipos NumPy) y una matriz asociada de etiquetas de datos, denominada *índice*.
- **DataFrame:** representa una tabla rectangular de datos y contiene una *colección ordenada de columnas*. Tiene un índice de fila y columna; se puede considerar como un dict de Series que comparten el mismo índice.

© w3resource.com