** Introducción a la librería científica con Python: Pandas (parte I)**

¿Qué es la librería Pandas?

👉 Pandas facilita la manipulación de grandes volúmenes de datos a través de un conjunto de métodos y estructuras diseñadas para tal fin.

👉 Extiende las funcionalidades de Numpy, por lo que sus estructuras de datos son totalmente compatibles.

👉 Es de uso libre, su primera versión se creó en 2008.

👉 Está enmarcado en un proyecto llamado NUMFOCUS (2015)

Tres estructuras fundamentales

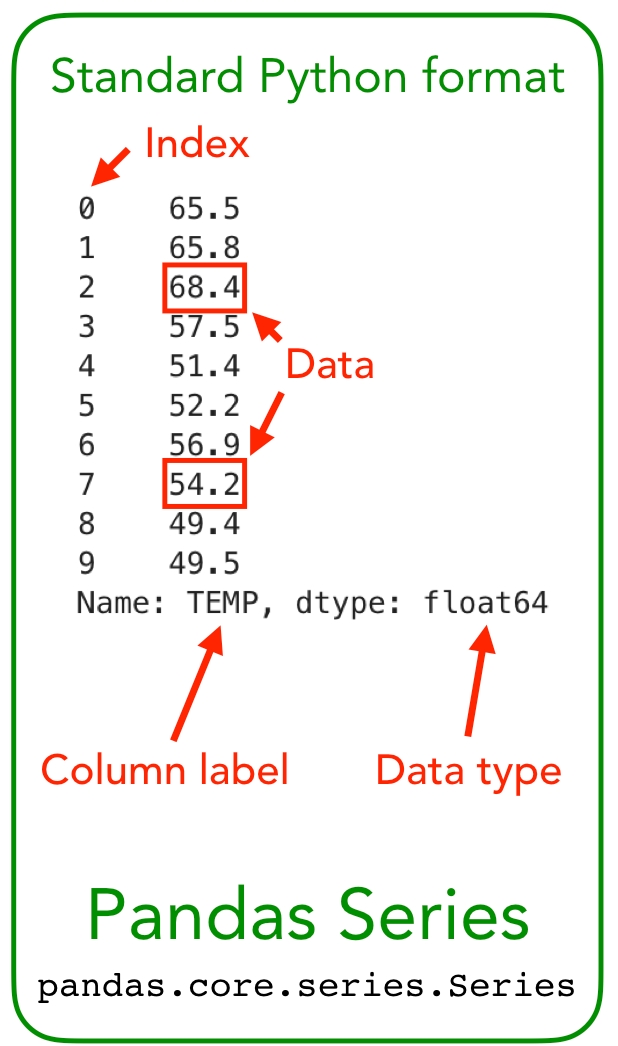
* Series (1D)
* Dataframes (2D)
* Paneles (3D)

**En este curso trabajaremos principalmente con Series y DataFrames.**

Pandas Series

**¿Qué son y cómo están compuestas?**

Las series son objetos de una dimensión, con tres componentes principales:

1. Índices: Pueden ser cualquier tipo de variable
2. **Valores:** Asociados a cada índice
3. **Nombre:** Que caracteriza a la serie

* Se construyen a partir de otros objetos particulares, como las listas o los Numpy arrays.
* Tienen índice propio.
* Están formados por dos objetos vinculados: el arreglo de índices y el arreglo de valores
* Tanto el arreglo de índices como el de valores pueden modificarse

Pandas Data Frames

* Son una **extensión** de los objetos **Series**.
* Pueden pensarse como una **tabla**,formada por varias Series de igual longitud.
* Como toda tabla, consta de filas y columnas (2D).
* Cada fila tiene un elemento índice asociado.

En resumen las series son objetos 2D (filas x columnas) con las siguientes propiedades:

* **Índices:** Pueden ser cualquier tipo de variable
* **Valores:** Asociados a cada índice y a cada columna
* **Nombre:** Que caracteriza al Dataframe

Son, entonces, una generalización de series

Manipulación de datos en Pandas

Selección de datos

**Indexamiento Series**

Con Pandas, existen tres formas de seleccionar elementos:

* Mediante el índice
* Mediante el **método loc()**
* Mediante el **método iloc()** que utiliza sólo índices numéricos

**Indexamiento Data Frame**

También podemos seleccionar partes específicas del Data Frame, como índices, columnas y valores.

* Selección de **columna**
* Selección de **fila**
* Selección **con condición**

Operaciones básicas de

datos en Pandas

**Transposición**

* Consiste en **intercambiar** filas de un Data Frame por sus columnas.
* Puede resultar más cómodo trabajar con el Data Frame transpuesto que con el original.

Funciones Vectorizadas

**Desde Numpy, en Pandas**

* Las **funciones vectorizadas** o **ufuncs** de Numpy pueden realizarse también sobre Data Frames y Series.
* Tras ejecutar la operación **se conservan los índices**.

Ufuncs sobre Data Frames  
Los Data Frames también admiten operaciones vectorizadas

DATOS AUSENTES

Houston… Tenemos un problema   
**¿Qué hacemos con los valores faltantes? Lidiando con valores faltantes**

* La mayoría de las operaciones de Pandas admiten un parámetro **fill\_value**, que indica el valor a insertar en caso de resultar un NaN.

Numeros\_serie\_2\_porcion.add(Numeros\_serie\_3\_porcion, fill\_value=0)

En este caso, especificamos que en caso de encontrar un valor faltante lo reemplace por 0.

¿Datos ausentes, por qué?

Como futuros Data Scientists, comúnmente nos toparemos con valores faltantes o **ausentes** que podrían provenir de las siguientes situaciones:

**👉 Fallas** en algún paso de la carga de datos.

**👉 Omisión** directa de la carga de datos.

**👉 Reticencia** de parte de un encuestado a dar una respuesta determinada.

¡Los valores faltantes son más comunes de lo que se piensa!

Not a Number

El representante del valor faltante

* NaN significa **Not a Number** y es el **valor faltante** **por defecto**.
* Es un tipo de dato especial de punto flotante.
* Tiene **propiedades especiales**: cualquier operación que involucre NaN da como resultado NaN.

**Trabajando con datos ausentes**

* Estos valores **podrían no ser adecuados** para algunos algoritmos de Data Science. Por ello, deben ser manejados correctamente.
* Pandas nos provee de herramientas para trabajar con ellos.

**Metódo pct\_change()**

El método pct\_change() de Pandas se aplica en series con datos numéricos para calcular el cambio de porcentual de un valor respecto al anterior.

Resulta útil al momento de monitorear variables numéricas.

Si tenemos n datos numéricos entonces tendremos n-1 cambios porcentuales

Glosario de términos

**Pandas**: librería fundamental que nos permite trabajar con archivos planos (.csv, .txt, xlsx) en Python

**Series**: estructuras 1D en Pandas que almacenan vectores con índice, nombre y valores

**DataFrames**: estructuras 2D (filas x columnas) que son la generalización de Series en Pandas

**Panel**: estructuras 3D (filas x columnas x profundidad) que permiten almacenar varios Dataframes

**Indexación de datos en Pandas**: métodos loc y iloc que permiten extraer filas con base en condiciones

**Operaciones vectorizadas**: toda operación matemática consistente que se puede ejecutar en vectores y matrices

**Datos ausentes**: Conocidos como NaN en Python representan valores sin registro que pueden llegar a ser problemáticos en los análisis.

**Metódo pct\_change**: Implícito en pandas que permite calcular variaciones porcentuales de observaciones consecutivas para monitorear cambios

**Lectura de datos en pandas**: se pueden leer archivos planos con las funciones pd\_read\_csv o pd\_read \_excel pero también se pueden leer archivos en formatos tipo .JSON y .SQL.