



Introducción a Python

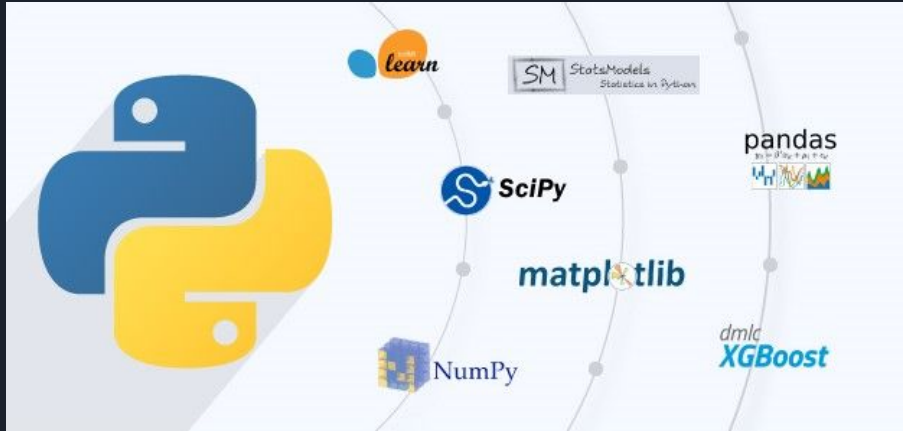
Pandas y Numpy

Prof. Jhoan Steven Delgado V.

Prof. Juan Manuel Reyes

¿Qué es una librería?

- Una librería de Python es una colección de módulos relacionados. Contiene paquetes de código que pueden ser utilizados repetidamente en diferentes programas
- Hace que la programación en Python sea más simple y conveniente para el programador
- Las librerías de Python juegan un papel muy importante en los campos de Machine Learning, Data Science, Data Visualization, etc.



The NumPy logo consists of a large red rectangle with a smaller red rectangle on top of it. The word "NumPy" is written in white, sans-serif font in the center of the larger rectangle. The background features a light gray pattern of concentric circles and dashed lines.

NumPy

Ecosistema SciPy

 StatsModels
Statistics in Python


 xarray

 scikit-image
image processing in python

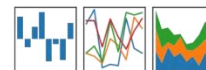
 scikit-learn
machine learning in Python

 Cartopy

And many,
many more...

 SciPy

pandas
 $y_{it} = \beta'x_{it} + \mu_i + \epsilon_{it}$



matplotlib

 bokeh

 NumPy

 jupyter

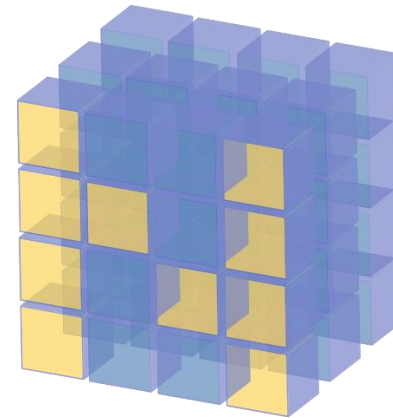
IP[y]:
IPython

 python™

 spyder

NumPy

- Es el paquete fundamental para computación científica de datos con Python



NumPy

NumPy

- El principal objeto de NumPy es el **arreglo multidimensional homogéneo**.
- Este arreglo es una tabla de elementos (usualmente números), todos del mismo tipo, indexados por una tupla de enteros no negativos.
- En NumPy, las dimensiones son llamadas ejes.
- Por ejemplo, el arreglo [8, 2, 5] tiene un eje. Ese eje tiene 3 elementos en él, por tanto decimos que tiene un largo de 3.
- Ahora otro ejemplo de dos dimensiones, o mejor 2 ejes. El primer eje tiene largo 2 y el segundo largo 3.

```
[[ 1., 0., 0.],  
 [ 0., 1., 2.]]
```



ndarray

- Lo primero que debemos hacer para utilizar NumPy es importar la librería que ya viene instalada como parte del ScyPy stack en Anaconda.

```
import numpy as np
a = np.array([2,3,4])
print(a)
print(a.dtype)
print(a.dtype.name)
print(type(a))
print(a.itemsize)
print(a.size)
print(a.ndim)
print(a.shape)
```

Secuencias de Secuencias

- La función **array** de la librería NumPy transforma secuencias de secuencias en arreglos de dos dimensiones. Secuencias de secuencias de secuencias en arreglos de tres dimensiones y así.

```
b = np.array([(3.2, 8, 2), (5, 9.4, 6.1)])  
print(b)
```

```
c = np.array([[3,2,1],[9,8,7]], dtype='float64')  
print(c)
```

```
d = np.array([(2,4,6),(1,3,5)], dtype=complex)  
print(d)
```


Arreglos con Valores por Defecto

- Podemos crear un arreglo de las dimensiones que indiquemos, inicializado en zeros, unos, vacío o valores aleatorios.

```
e = np.zeros((3,4))  
print(e)  
f = np.ones((3,2,4), dtype=np.int16)  
print(f)  
g = np.empty((5,4))  
print(g)  
h = np.random.random((2,4))  
print(h)
```

Modificar dimensiones de un arreglo

```
l = np.array([3,9,1,4,10,6,12,8,2,5,11,7])  
m = l.reshape((3,4))  
print("l=%s"%l)  
print("m=%s"%m)
```

```
o = m.reshape((2,6))  
print(o)
```

```
p = m.reshape((2,3,2))  
print(p)
```



arange y
linspace

```
w = np.arange(10,30,5)
```

```
print(w)
```

```
x = np.arange(0,2,0.3)
```

```
print(x)
```

```
y = np.linspace(0,2,9)
```

```
print(y)
```

```
z = np.linspace(0,2*np.pi,100)
```

```
print(z)
```

```
ss = np.sin(z)
```

```
print(ss)
```

Operaciones Básicas

```
a = np.array([100,200,300,400,500])
b = np.array([500,400,300,200,100])
c = a-b
print(c)

d = np.arange(10)**2
print(d)

e = 10*np.cos(a)
print(e)

f = a<35
print(f)
```

Mas Operaciones

```
A = np.array([[1,1],[0,1]])
```

```
B = np.array([[2,0],[3,4]])
```

```
C = A * B
```

```
print(C)
```

```
D = A @ B
```

```
print(D)
```

```
E = A.dot(B)
```

```
print(E)
```

Operaciones Unarias Frecuentes

```
a = np.random.random((3,4))  
print(a)  
  
print(a.sum())  
  
print(a.min())  
  
print(a.max())
```

```
z = np.arange(4)  
print(np.exp(z))  
print(np.sqrt(z))
```

Indexado y Slicing

```
a = np.arange(10)**3
print(a)

print(a[2:5])
print(a[3:7:2])

a[1:4] = -1
print(a)

print(a[::-1])

a[2:9:3] = 1010
print(a)
```