

Numpy

Documentación en <https://docs.scipy.org/doc/numpy/>
Librería para hacer arreglos matriciales

Importar Numpy

```
import numpy as np
```

Creación de arrays

Para llamar arreglos de Numpy es necesario pasarlos en forma de listas.

```
#Dimensión 1 (Unidimensional)
arr1 = np.array([22,34,15])
#Dimensión 2 (Matriz)
arr2 = np.array([(22,32,15),(12,7,8)])
#Dimensión 3 (Cubo)
arr3 = np.array([[(22,32,15),(12,7,8)],
                 [(31,50,27),(11,3,2)]]])
```

Atributos y arrays especiales

Algunos atributos de Numpy

```
#Número de dimensiones del array
print(arr3.ndim)
#Tamaño de los arrays de cada
#dimensión
print(arr3.shape)
#Tipo de elementos del array
print(arr3.dtype)
```

Arrays especiales

```
#Arreglos rellenos de ceros
np.zeros((5,2))
#Arreglos rellenos de unos
np.ones((3,2,3))
#Arreglos crecientes en un
#rango y con saltos definidos
np.arange(10,50,2)
#Arreglos crecientes y
#divididos en tamaños iguales
np.linspace(0,2,9)
#Matriz identidad
np.eye(3)
#Creación de arreglos con
#valores aleatorios
np.random.random((3,2))
```

Funciones matemáticas

```
#Sumar todo el array
arr1.sum()
#Mínimo del array
arr1.min()
#Máximo del array
arr1.max()
#Promedio del array
arr1.mean()
#Suma acumulada por renglón
arr1.cumsum()
```

Funciones probabilísticas

```
#Valores uniformes entre 0 y 1
np.random.rand()
#Valores normal estándar
np.random.randn()
#Valores discretos en rango definido
np.random.randint(3,10)
#Muestro aleatorio de un array
np.random.choice(7,3,replace = False)
#Barajar valores de un array
np.random.shuffle(arr1)
```

Pandas

Documentación en <https://pandas.pydata.org/>
Librería para abordar estructuras de datos

Importar Pandas

```
import pandas as pd
```

Series y DataFrames

```
Crear Serie sencilla

s1 = pd.Series([23,45,6,2,4,6])
dtype: int64
```

Crear DataFrame con arrays de numpy

```
d1 = pd.DataFrame(
    np.random.rand(3,4),
    index = np.arange(3),
    columns = ['X1','X2','X3','X4'])
```

Leer y exportar DataFrames a CSV

Leer CSV y guardar en DataFrames

```
read = pd.read_csv('AAPL.csv')
read.head() #Explorar sólo los primeros 5 datos
read.tail() #Explorar sólo los últimos 5 datos
```

Exportar DataFrames a CSV

```
export = d1.to_csv('d1.csv')
```

Selección de datos

Diferentes formas de seleccionar elementos del DataFrame

```
d1.iloc[[2],[3]] #Por posición (1)
d1.iloc[2,3] #Por posición (2)
d1.loc[[2],['X4']] #Por etiqueta
```

	X4
2	0.853266

Selección por índices lógicos

```
d1[(d1.X2 > 0.3)]
```

Combinar columnas y renglones

	X1	X2	X3	X4
d1	0 0.649542	0.419919	0.519321	0.915293
	1 0.431524	0.311905	0.471838	0.651635
	2 0.417658	0.633393	0.204923	0.631622

	X1	X2	X3	X4
d2	0 0.931822	0.797931	0.779658	0.160618
	1 0.245780	0.339333	0.793106	0.553437
	2 0.468972	0.895550	0.384651	0.133785

```
#Unir DataFrames por Renglón
pd.concat([d1,d2],axis = 0)
```

Si tenemos un nuevo DataFrame

	X5	X6	X7	X8
d3	0 0.365896	0.038291	0.825559	0.628553
	1 0.299406	0.967713	0.364213	0.374344
	2 0.115664	0.158988	0.948092	0.797649

```
#Unir DataFrames por Columna
pd.concat([d1,d3],axis = 1)
```

	X1	X2	X3	X4	X5	X6	X7	X8
0	0.649542	0.419919	0.519321	0.915293	0.365896	0.038291	0.825559	0.628553
1	0.431524	0.311905	0.471838	0.651635	0.299406	0.967713	0.364213	0.374344
2	0.417658	0.633393	0.204923	0.631622	0.115664	0.158988	0.948092	0.797649

```
#Genera muestras
d1.sample(n = 1)
#Elimina duplicados
#por columna
d1.drop_duplicates()
```

Créditos

Luis A. Alcántara (Autor)
Stefani J. Ruiz (Manager)
Ana L. Aldeco (Senior Manager)