



UNIVERSIDAD NACIONAL AGRARIA

LA MOLINA

Estadística descriptiva con Python

Librerías Python

Ana Vargas
anavargas@lamolina.edu.pe

25 y 26 de agosto de 2018

Contenido

1. Reglas de alcance LEGB y Clases

2. Vectorización

3. Indexación en NumPy Arrays

4. Introducción a Matplotlib

Ejemplo

- L: local
- E: enclosing function
- G: global
- B: built-in

```
def actualiza(n,x):  
    n=2  
    x.append(4)  
    print('actualiza:',n,x)  
  
def main():  
    n=1  
    x=[0,1,2,3]  
    print('principal:',n,x)  
    actualiza(n,x)  
    print('principal:',n,x)  
  
main()
```

Clases - POO

Un objeto consiste de datos internos y métodos que ejecuta operaciones sobre los datos.

Una clase a menudo se utiliza cuando se necesita crear un nuevo tipo de objeto que no se ajusta a los existentes, pero que puede heredar características de los tipos existente a esto se llama Herencia (una característica de la POO).

```
class MiLista(list):  
    def remover_min(self):  
        self.remove(min(self))  
    def remover_max(self):  
        self.remove(max(self))  
  
class Persona:  
    def __init__(self, nombre, dni, edad):  
        self.nombre = nombre  
        self.dni = dni  
        self.edad = edad  
  
    def iniciales(self):  
        cadena = ''  
        for caracter in self.nombre:  
            if caracter >= 'A' and caracter <= 'Z':  
                cadena = cadena + caracter + '. '  
        return cadena
```


Vectorización usando NumPy

NumPy es un módulo de Python para realizar cálculos computacionales.

NumPy tiene diferentes útiles aplicaciones como: NumPy arrays objetos de matriz n-dimensional que son el componente central de la computación científica y numérica en Python.

NumPy proporciona elementos para integrar código con C, C++ y Fortran.

NumPy proporciona elementos para ejecutar álgebra lineal, generar números aleatorios, etc.

NumPy Array

NumPy Array son un tipo de dato proporcionado por NumPy. Un objeto de este tipo tiene elementos del mismo tipo y su tamaño es fijado cuando es construido.

Por defecto los elementos son de puntos flotantes.

```
import numpy as np

x = np.array([1,2,3])
y = np.array([2,4,6])

A = [[1,3],[5,9]]

B = np.array([[1,3],[5,9]])

B.transpose()

B.trace()
```

Indexación en NumPy Arrays

`NumPy Array` puede ser indexado con otro arreglo u objeto de otro tipo como lista de secuencias.

Se obtiene diferentes resultados al dividir un arreglo usando el operador coma (lo que se obtiene es una vista) que cuando se indexa (ya que lo que se obtiene es una copia)

```
import numpy as np

z1 = np.array([1,3,5,7,9])
w = z1[0:3]
w[0] = 3
z1 = np.array([1,3,5,7,9])
ind = np.array([0,1,2])
w = z1[ind]
w[0] = 3
```

Introducción a Matplotlib

`Matplotlib` es una librería de Python para graficar.

`Pyplot` es una colección de funciones que hacen que `Matplotlib` funcione como Matlab y es útil en trabajo interactivo como explorar datos.

```
import matplotlib.pyplot as plt
```

```
x = np.linspace(0,10,20)
```

```
y1 = x**2
```

```
plt.plot(x,y1, "bo-", linewidth=2, markersize=4)
```

```
x = np.logspace(-1,1,40)
```

```
y1 = x**2
```

```
y2 = x**1.5
```

```
plt.loglog(x,y1, "bo-", linewidth=2, markersize=4, label="y1")
```

```
plt.loglog(x,y2, "gs-", linewidth=2, markersize=4, label="y2")
```

```
plt.xlabel("X")
```

```
plt.ylabel("Y")
```

```
plt.axis([-0.5,10.5,-5,105])
```

```
plt.legend(loc="upper left")
```

Histograma con Pyplot

```
matplotlib.pyplot.hist(x, bins=None, range=None,  
density=None, weights=None, cumulative=False, bottom=None,  
histtype='bar', align='mid', orientation='vertical',  
rwidth=None, log=False, color=None, label=None,  
stacked=False, normed=None, hold=None, data=None,  
**kwargs) [source]
```



```
x = np.random.normal(size=1000) #genera 1000 números aleat  
plt.hist(x,density=True, bins=np.linspace(-5,5,21))  
plt.show()
```


Recursos de consulta

- <https://matplotlib.org/api>
- <http://www.numpy.org/>
- <https://jakevdp.github.io/PythonDataScienceHandbook/>