



Numerical Python

Programación Unidad 3: Programación orientada a objetos y librerías

Ph.D. Santiago Echeverri-Arteaga

“La diferencia entre arte y ciencia es que la ciencia la entendemos lo suficientemente bien como para explicársela a un ordenador. Arte es todo lo demás” Donald Knuth

Índice

Secuencias

Listas

ndarray

Indexación booleana y reshape

Métodos estadísticos y matemáticos

Álgebra Lineal

Ejercicios

Secuencias

¿Secuencias?

Las secuencias son tipos de datos que contienen listas de datos a los que se accede mediante un índice. En python:

1. No tienen límite de la cantidad de elementos que pueden contener.
2. Comparten operaciones comunes

Operation Name	Operator	Explanation
indexing	[]	Access an element of a sequence
concatenation	+	Combine sequences together
repetition	*	Concatenate a repeated number of times
membership	in	Ask whether an item is in a sequence
length	len	Ask the number of items in the sequence
slicing	[:]	Extract a part of a sequence

3. No hay que decir el tamaño que van a tener
4. Los tipos de secuencias son cadenas, arreglos y conjuntos
5. Pueden ser asignados a variables

Listas

Listas

1. Colecciones ordenadas de cero o más elementos separados por coma, delimitados por corchetes cuadrados
2. Son HETEROGÉNEAS, es decir, los elementos pueden ser cualquier tipo de datos incluido otras listas
3. Tienen las operaciones propias

Method Name	Use	Explanation
append	<code>a_list.append(item)</code>	Adds a new item to the end of a list
insert	<code>a_list.insert(i, item)</code>	Inserts an item at the i^{th} position in a list
pop	<code>a_list.pop()</code>	Removes and returns the last item in a list
pop	<code>a_list.pop(i)</code>	Removes and returns the i^{th} item in a list
sort	<code>a_list.sort()</code>	Modifies a list to be sorted
reverse	<code>a_list.reverse()</code>	Modifies a list to be in reverse order
del	<code>del a_list[i]</code>	Deletes the item in the i^{th} position
index	<code>a_list.index(item)</code>	Returns the index of the first occurrence of item
count	<code>a_list.count(item)</code>	Returns the number of occurrences of item
remove	<code>a_list.remove(item)</code>	Removes the first occurrence of item

4. Algunas operaciones modifican la lista, otras solo alteran el orden de los elementos
5. Pueden ser inicializadas por repetición (`lista = [1, True, 7.8] * 8`)
6. Indexación con signo negativo inicia desde el final

Aún mas operaciones (A listas NUMÉRICAS)

Function	Description
sum	returns the sum of the items in the list
min	returns the minimum of the items in the list
max	returns the maximum of the items in the list

Las operaciones sobre las listas las modifican (L.sort() y no L=L.sort()): MUTABILIDAD

min y max no son exclusivos de listas numéricas

La lista vacía [] es equivalente a " y es interpretada como un 0

Si una lista es larga se puede seguir directamente en la otra línea

Se puede hacer el **for i in list:** y usar el operador **in**

Se pueden modificar los elementos independientes de la lista

ndarray

¿No son suficientes las listas?

- Los array (ndarray) son un tipo de datos distinto a las listas, el cual permite realizar computo matemático mucho más rápido que con las listas por defecto.
- Son más similares a los vectores (matrices) que las listas
- Cada ndarray tiene una *propiedad* llamada **shape** (equivalente al len de las listas).
- Si list es un ndarray, su shape es **list.shape**, una tupla.
- De forma similar se puede encontrar de que tipo de datos son sus items (haciendo **list.dtype**)
- Se puede usar la función **array()** para convertir una secuencia homogénea a un array, con esta una secuencia de secuencias se convierte en un array multidimensional.
- **len** y **shape** difieren para array multidimensionales

ndarray i

- **ones**((n,m)) (**ones**(n)) crea una matriz (vector) $n*m$ ($n*1$) llena de unos
- **zeros**((n,m)) (**zeros**(n)) crea una matriz (vector) $n*m$ ($n*1$) llena de ceros
- **eye**(N) crea una matriz identidad $N*N$
- **arange**(N) es equivalente a **list(range(N))**, pero yo puedo generar una lista de datos entre N y M con una discretización de dx por medio de **arange**(N,M,dx)
- También puedo generar una lista de P datos entre N y M con espaciamiento lineal usando **linspace**(N,M,P)
- De forma similar, puedo generar una lista de P datos entre 10^N y 10^M con espaciamiento logarítmico usando **logspace**(N,M,P)

- Las operaciones de suma y producto entre arrays y producto por escalar funcionan elemento a elemento
- Al elemento n,m de una matriz se puede acceder como $M[n,m]$

- Se puede hacer

```
arr = np.arange(10)
```

```
arr[5:8] = 12
```

Si se tiene `mat = np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9],[10, 11, 12]])`

```
mat[:2, 1:]
```

Indexación booleana y reshape

- ¿Que sucede si tengo en una matriz de N filas las notas de N estudiantes y deseo seleccionar solo las que corresponden a uno de ellos?

```
data = np.random.randint(6,size=(4,10))
names = np.array(['Tatiana','Valeria','Juan','Gabriela'])
names == 'Tatiana'
(names == 'Tatiana') | (names=='Gabriela')
data[names == 'Bob']
data[data < 3] = 0
```

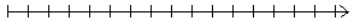
- En algunas ocasiones deseo expresar un vector de tamaño $nm \times 1$ como una matriz de tamaño $n \times m$ o escribir la matriz como el vector. Eso lo consigo con reshape: `np.arange(32).reshape((8, 4))`

¿Cómo trabajo con funciones?

¿Cómo grafico una función 2D?

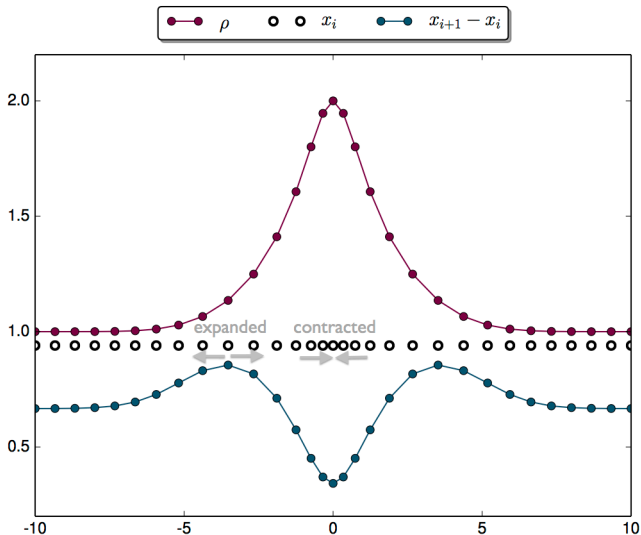
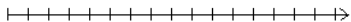
¿Cómo trabajo con funciones?

¿Cómo grafico una función 2D?



¿Cómo trabajo con funciones?

¿Cómo grafico una función 2D?



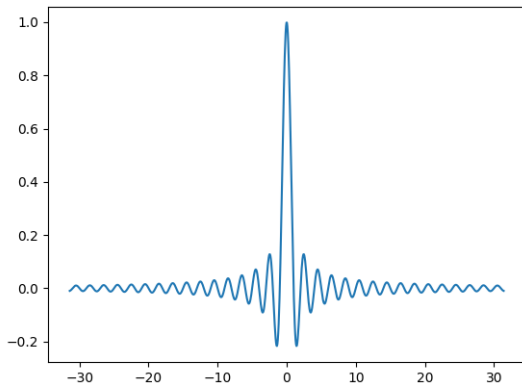
¿Cómo trabajo con funciones?

Representando una función sinc desde $-\pi$ hasta π

```
import numpy as np
```

```
x = np.linspace(-10*np.pi,10*np.pi,1000)
```

```
y = np.sinc(x)
```

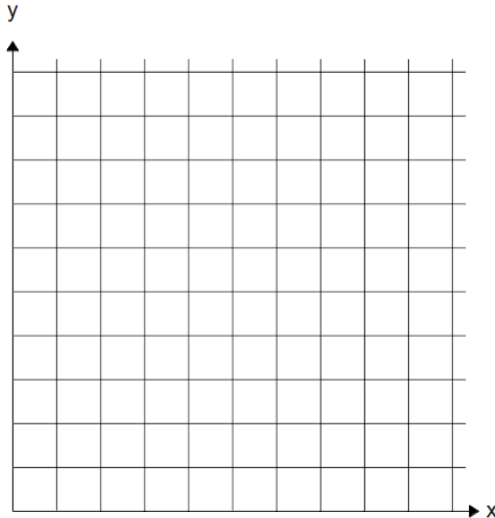


¿Cómo trabajo con funciones?

¿Cómo grafico una función 3D?

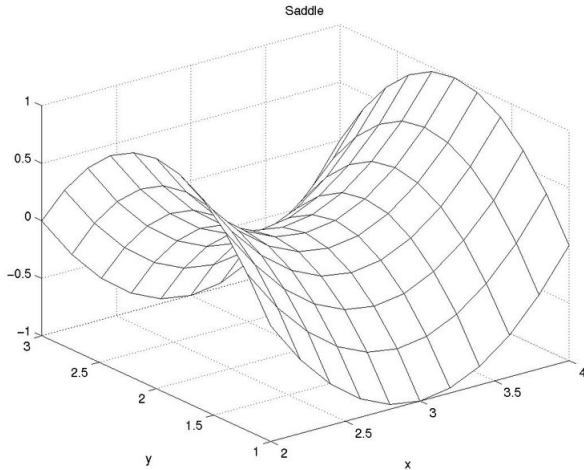
¿Cómo trabajo con funciones?

¿Cómo grafico una función 3D?



¿Cómo trabajo con funciones?

¿Cómo grafico una función 3D?



¿Cómo trabajo con funciones?

Equivale a tener una matriz cuadrada de x , y y z tal que

$$x = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 \end{pmatrix} \quad (1)$$

$$y = \begin{pmatrix} y_1 & y_1 & y_1 \\ y_2 & y_2 & y_2 \\ y_3 & y_3 & y_3 \end{pmatrix} \quad (2)$$

$$z = \begin{pmatrix} f(x_1, y_1) & f(x_2, y_1) & f(x_3, y_1) \\ f(x_1, y_2) & f(x_2, y_2) & f(x_3, y_2) \\ f(x_1, y_3) & f(x_2, y_3) & f(x_3, y_3) \end{pmatrix} \quad (3)$$

¿Cómo trabajo con funciones?

¿Cómo realizo eso con listas?

¿Cómo trabajo con funciones?

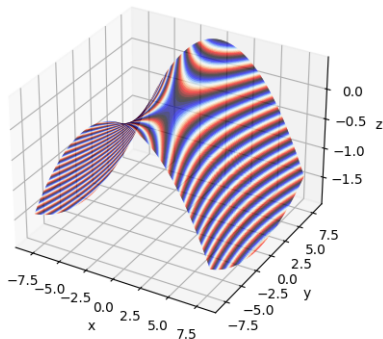
Pero con numpy

```
x_list = np.arange(-5, 5, 0.01)
```

```
y_list = np.arange(-1, 1, 0.01)
```

```
x, y = np.meshgrid(x_list, y_list)
```

```
z = -3*((X/10)**2-(Y/20)**2)
```



¿Cómo trabajo con funciones?

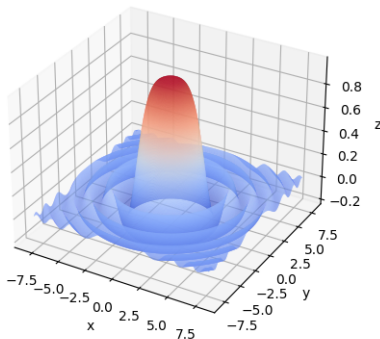
Pero con numpy

```
x_list = np.arange(-5, 5, 0.01)
```

```
y_list = np.arange(-1, 1, 0.01)
```

```
x, y = np.meshgrid(x_list, y_list)
```

```
z_2 = np.sinc((X/np.pi)**2+(Y/np.pi)**2)
```



¿Cómo trabajo con funciones?

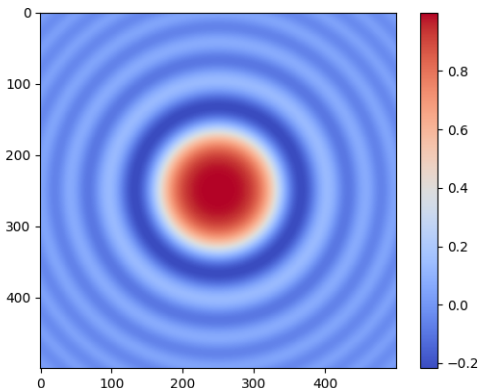
Pero con numpy

```
x_list = np.arange(-5, 5, 0.01)
```

```
y_list = np.arange(-1, 1, 0.01)
```

```
x, y = np.meshgrid(x_list, y_list)
```

```
z_2 = np.sinc((X/np.pi)**2+(Y/np.pi)**2)
```



¿Cómo trabajo con funciones?

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

where

Tengo una lista con el resultado de `data[names='Bob']` que llamo `cond`. Deseo generar una nueva lista, la cual debe tomar un valor de `xarr` si `cond` tiene un valor `True` o de `yarr` si tiene un valor de `False`.

```
xarr = np.array([1.1, 1.2, 1.3, 1.4, 1.5])
```

```
yarr = np.array([2.1, 2.2, 2.3, 2.4, 2.5])
```

```
cond = np.array([True, False, True, True, False])
```

Hacer esto con comprensión de listas sería así:

```
result = [(x if c else y) for x, y, c in zip(xarr, yarr, cond)]
```

Con numpy se hace:

```
result = np.where(cond, xarr, yarr)
```

where

Dada una matriz aleatoria de números positivos y negativos, cambiar todos los números positivos por 2 y negativos por -2. Luego hacer lo mismo pero solo cambiar los negativos por -2

```
arr = np.random.randn(4, 4)
```

```
a = np.where(arr > 0, 2, -2)
```

```
b = np.where(arr > 0, 2, arr)
```

Métodos estadísticos y matemáticos

Métodos estadísticos y matemáticos

- **sum()** Suma de los elementos
- **mean()** Valor medio
- ¿Que sucede si se le da a `arr.sum()` o a `arr.mean()` el argumento de `axis=0` o `axis=1` en una matriz?
- **std()** y **var()** Desviación estandar y varianza
- **min()**, **max()**, **argmin()** y **argmax()** Valor mínimo, máximo, y posición de los mismos
- **sort()** Ordena
- Operaciones que se conseguían trabajando con sets:

<code>unique(x)</code>	Compute the sorted, unique elements in x
<code>intersect1d(x, y)</code>	Compute the sorted, common elements in x and y
<code>union1d(x, y)</code>	Compute the sorted union of elements
<code>in1d(x, y)</code>	Compute a boolean array indicating whether each element of x is contained in y
<code>setdiff1d(x, y)</code>	Set difference, elements in x that are not in y
<code>setxor1d(x, y)</code>	Set symmetric differences; elements that are in either of the arrays, but not both

- **np.random.rand(n,m)** Genera flotantes aleatorios entre 0 y 1
- **np.random.randn(n,m)** Genera números aleatorios de la distribución normal estandar
- **np.random.randint(N,M)** Genera números aleatorios entre N y M

Method	Description
sum	Sum of all the elements in the array or along an axis; zero-length arrays have sum 0
mean	Arithmetic mean; zero-length arrays have NaN mean
std, var	Standard deviation and variance, respectively, with optional degrees of freedom adjustment (default denominator n)
min, max	Minimum and maximum
argmin, argmax	Indices of minimum and maximum elements, respectively
cumsum	Cumulative sum of elements starting from 0
cumprod	Cumulative product of elements starting from 1

Álgebra Lineal

- `dot(x,y)`, `x.dot(y)` o `x@y` calcula el producto matricial de a y b
- `np.linalg.inv(a)` Inversa
- `np.linalg.det(a)` Determinante
- `np.diag(a)` `np.diag(a,k=1)` `np.diag(a,k=-1)` retorna en un array los elementos de la diagonal y los elementos adyacentes a distancia 1 por encima y 1 por debajo
- `np.trace(a)` Traza
- `l,u = np.linalg.eig(a)` Retorna en l los valores propios y en u los vectores propios
- `numpy.linalg.solve(a, b)` Retorna el vector x que soluciona el sistema matricial $ax=b$

¿Qué es descomposición QR?

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

Recordemos álgebra lineal

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

¿Qué es descomposición en valores singulares?

Recordemos álgebra lineal

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

¿Qué es descomposición en valores singulares?

Descomposición de una matriz: $A_{n \times p} = U_{n \times n} S_{n \times p} V_{p \times p}^T$ con S diagonal y U y V ortogonales.

Recordemos álgebra lineal

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

¿Qué es descomposición en valores singulares?

Descomposición de una matriz: $A_{n \times p} = U_{n \times n} S_{n \times p} V_{p \times p}^T$ con S diagonal y U y V ortogonales.

¿Cuáles son los valores singulares de una matriz?

Recordemos álgebra lineal

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

¿Qué es descomposición en valores singulares?

Descomposición de una matriz: $A_{n \times p} = U_{n \times n} S_{n \times p} V_{p \times p}^T$ con S diagonal y U y V ortogonales.

¿Cuáles son los valores singulares de una matriz?

Los elementos de la matriz $S_{n \times p}$, que corresponden a la raíz cuadrada de los autovalores de AA^T

Recordemos álgebra lineal

¿Qué es descomposición QR?

R/ Descomposición de la matriz como producto de una matriz ortogonal por una triangular superior.

¿Qué es descomposición en valores singulares?

Descomposición de una matriz: $A_{n \times p} = U_{n \times n} S_{n \times p} V_{p \times p}^T$ con S diagonal y U y V ortogonales.

¿Cuáles son los valores singulares de una matriz?

Los elementos de la matriz $S_{n \times p}$, que corresponden a la raíz cuadrada de los autovalores de AA^T

En Numpy la función `lstsq` retorna una tupla de cuatro elementos. El primero es la solución del sistema, el segundo es la matriz de residuos, el tercero es el rango de la matriz y el cuarto es los valores singulares de la matriz.

Function	Description
<code>eig</code>	Compute the eigenvalues and eigenvectors of a square matrix
<code>inv</code>	Compute the inverse of a square matrix
<code>pinv</code>	Compute the Moore-Penrose pseudo-inverse of a matrix
<code>qr</code>	Compute the QR decomposition
<code>svd</code>	Compute the singular value decomposition (SVD)
<code>solve</code>	Solve the linear system $Ax = b$ for x , where A is a square matrix
<code>lstsq</code>	Compute the least-squares solution to $Ax = b$

Ejercicios

Ejercicios

Realice los siguientes ejercicios para entregar

1. Realizar una calculadora que le permita al usuario realizar suma, resta, multiplicación de matrices, calcular la adjunta, la inversa, calcular autovalores, autovectores y solucionar un sistema $Ax = B$ con A una matriz y x, b vectores columna
2. Generar archivos para graficar 5 funciones en 3D de su elección y guardarlo en el computador como tablas.
3. Importar una lista de datos como la que se les enviará al correo. Filtrar las notas de cinco estudiantes ingresados por el usuario y hacer su estadística. Tener en cuenta que lo que separa las columnas son tabuladores.