Numerical Python

Volodymyr Byelobrov

Kariera IT, Wroclaw

September 21, 2019

Table of Contents

Guido van Rossum's first intentions were to create a script language suitable for teaching purposes. Still because of his interest in Maths, a lot of mathematical features were directly implemented in the core of Python.

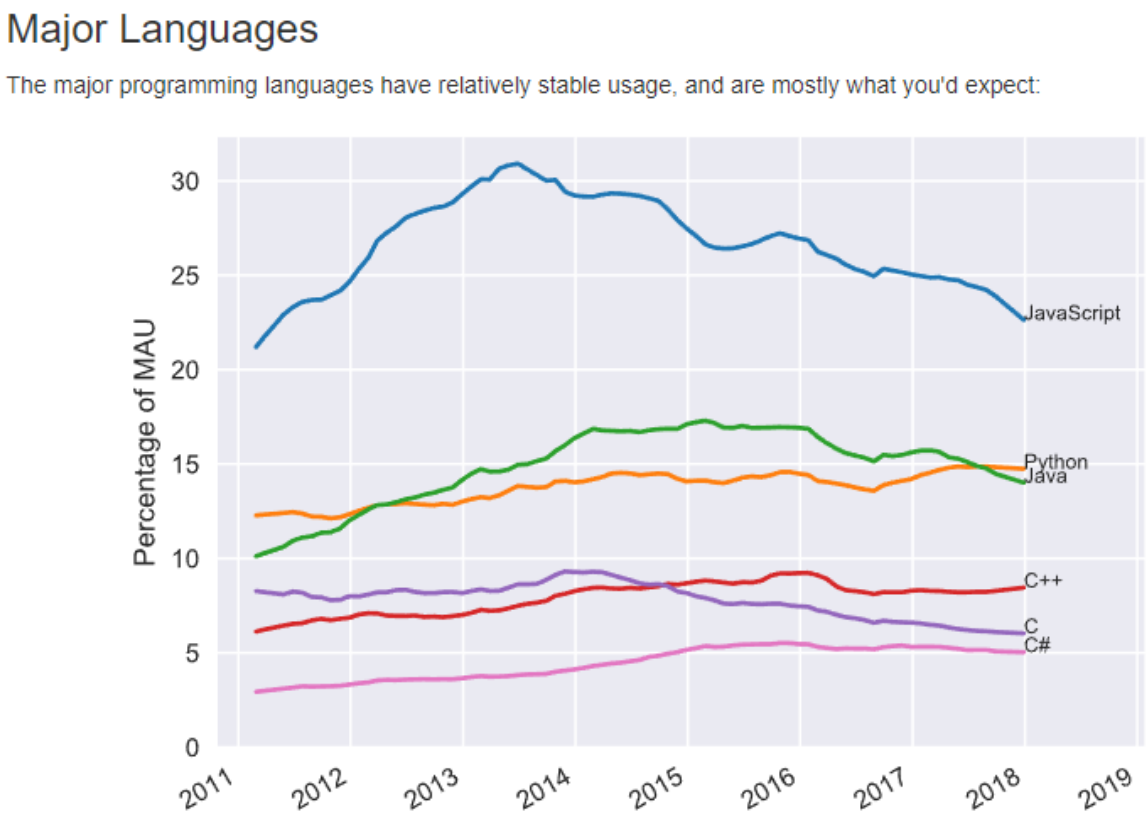
1. Complex numbers
2. Logic operations
3. Aggregation functions
   min() ; *max()* ; *sum()* ; *round()*

Complex numbers in different languages

Build-in type	Standard library	No Standard library
Fortran	C	Java
Python	C++	
Go	Perl	
Common Lisp	Ruby	
	OCaml	
	Haskell	

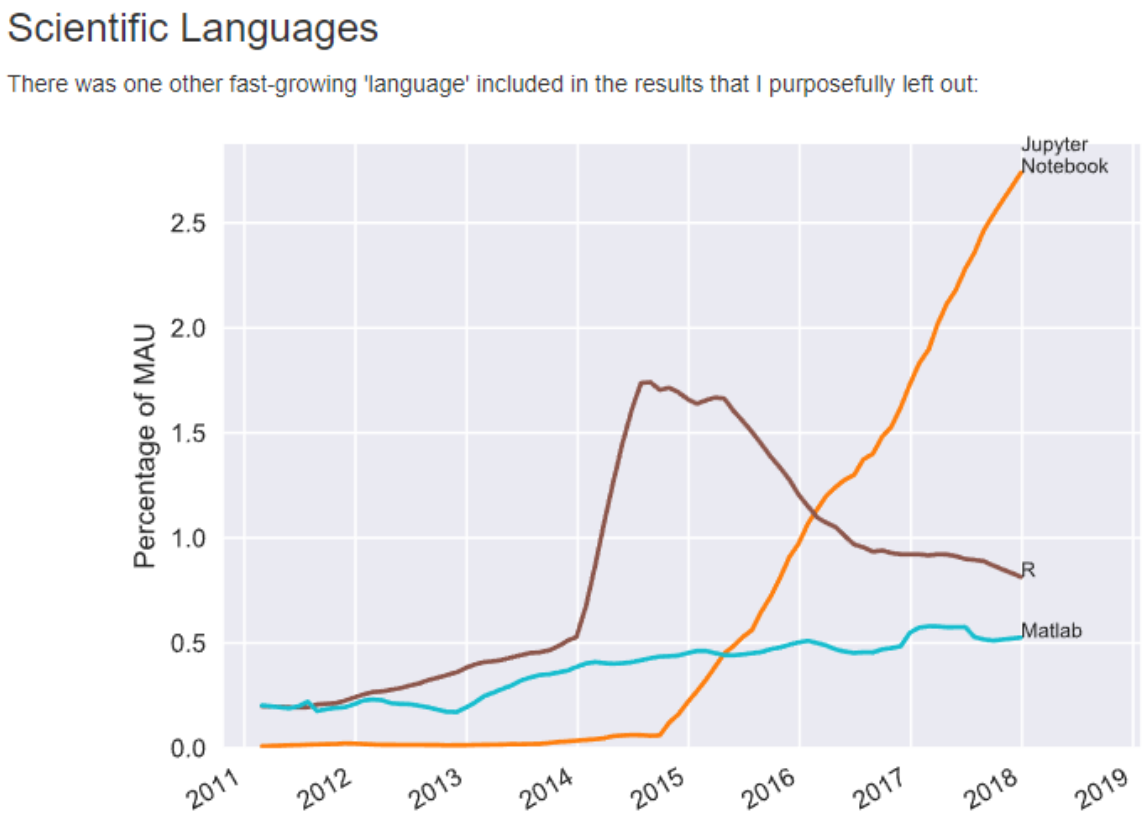
https://en.wikipedia.org/wiki/Complex_data_type

Python is super



https://www.benfrederickson.com

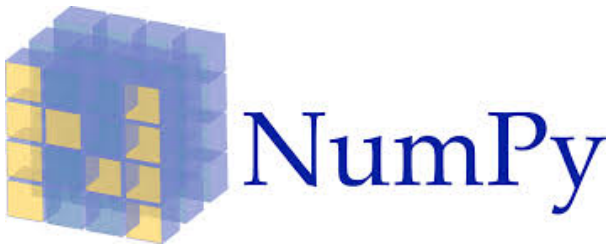
Numerical Python is superb

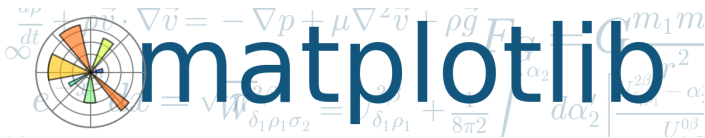


www.benfrederickson.com

Python vs Matlab

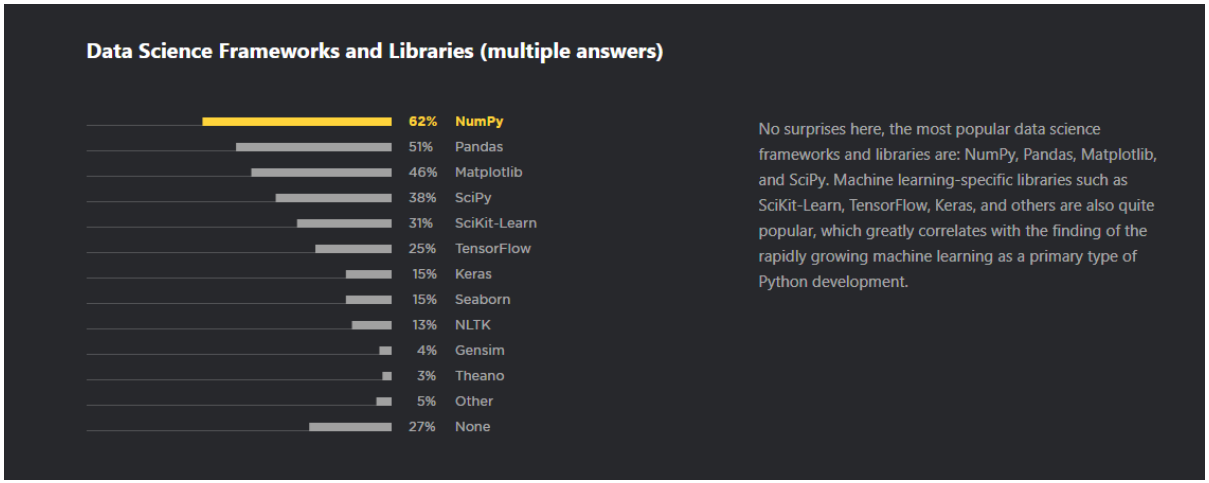
Even most NumPy, SciPy, and Matplotlib function replicates Matlab ones, currently the former libraries create a more popular platform for numerical calculations.





NumPy in Data Science

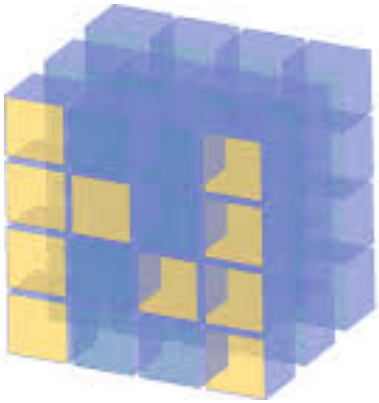
Python Developers Survey 2018 showed NumPy was the most popular Data Science framework.



www.jetbrains.com/research/python-developers-survey-2018

Table of Contents

NumPy is Everything



NumPy

2447 packages depend on NumPy:

scipy, Keras, pandas, spacy, bokeh, dask, gym, netCDF4, numexpr, moviepy, textblob, nibabel, PyWavelets, autograd, tensorboardX, mxnet, folium, datasketch, rasterio, pymc3, ChatterBot, Bottleneck, thinc, OpenFisca-Core, holoviews, h5py, etc.

libraries.io/pypi/numpy/dependents

NumPy documentation

NumPy has a very rich official documentation, provided for both NumPy and SciPy

docs.scipy.org

Still, there is a huge number of books, courses, videos, and ect. Among numerous other authors I dare to sugeest Jacob VanderPlas. Here is a link to his page vanderplas.com.

Numerical packages

NumPy

1. np.ndarray
2. np.random
3. basic Maths functions

SciPy

1. Special functions (scipy.special)
2. Integration (scipy.integrate)
3. Optimization (scipy.optimize)
4. Interpolation (scipy.interpolate)
5. Fourier Transforms (scipy.fftpack)
6. Signal Processing (scipy.signal)
7. Linear Algebra (scipy.linalg)
8. Sparse Eigenvalue Problems with ARPACK
9. Spatial data structures and algorithms (scipy.spatial)
10. Statistics (scipy.stats)

NumPy ndarray

Numpy ndarray is the Python object that represents a table of elements, all of the same type, indexed by a tuple of positive integers (starting from zero).
NumPy has many methods for ndarrays‘ declaration

1. **empty array**
   np.ndarray()
2. **filled array**
   np.zeros() , ***np.zeros_like()*** , ***np.ones()*** , ***np.eye()*** , ***np.full()*** , etc
3. **shape and dimention manipulation**
   np.reshape() , ***np.flatten()*** , etc

NumPy supports a much greater variety of numerical types than Python does. They could be platform-dependent and independent. Here some former ones.

Numpy type	C type	Description
np.int8	int8_t	Byte(-128 to 127)
np.int16	int16_t	Integer(-32768 to 32767)
np.int32	int32_t	Integer(-2147483648 to 2147483647)
np.int64	int64_t	Integer (-9223372036854775808 to 9223372036854775807)
np.float32	float	
np.float64 / np.float_	double	precision of the builtin python float
np.complex64	float complex	Complex number, represented by two 32-bit floats (real and imaginary)

NumPy ndarray indexing support

1. **Slicing** The rules of lists‘ slicing are also applied to ndarrays.
2. **Integer array indexing** Lists or ndarrays of integers are passed for indexing for each dimension.
3. **Boolean array indexing** List or ndarray of boolean type of the same shape are passed to obtain a slice for True values.

A universal function (or ufunc for short) is a function that operates on ndarrays in an element-by-element fashion, supporting array broadcasting, type casting, and several other standard features. That is, a ufunc is a "vectorized" wrapper for a function that takes a fixed number of specific inputs and produces a fixed number of specific outputs.

1. Arithmetic Operations: $+$ $-$ $*$ $@$ $/$ $//$ $\%$ $**$
2. Bitwise Operations: $\&$ $|$ $\backsim$ $\wedge$ $>>$ $<<$
3. Comparison Operations: $<$ $>$ $<=$ $=>$ $==$ $!=$
4. Mathematical Functions: np.sin, np.cos np.sqrt np.exp, np.log, np.log10, etc.
5. and many, many more

Thank you for your attention :)

github.com/sh-am-si/numpy-presentation