
Tera

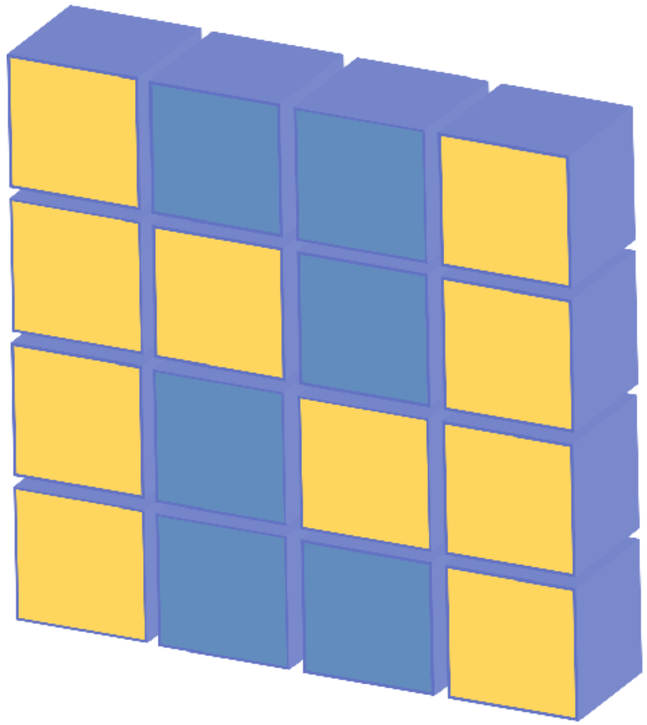
Módulo 2:
Análise e Visualização de Dados

Análise de Dados com Arrays Multidimensionais

Luam Catão Totti

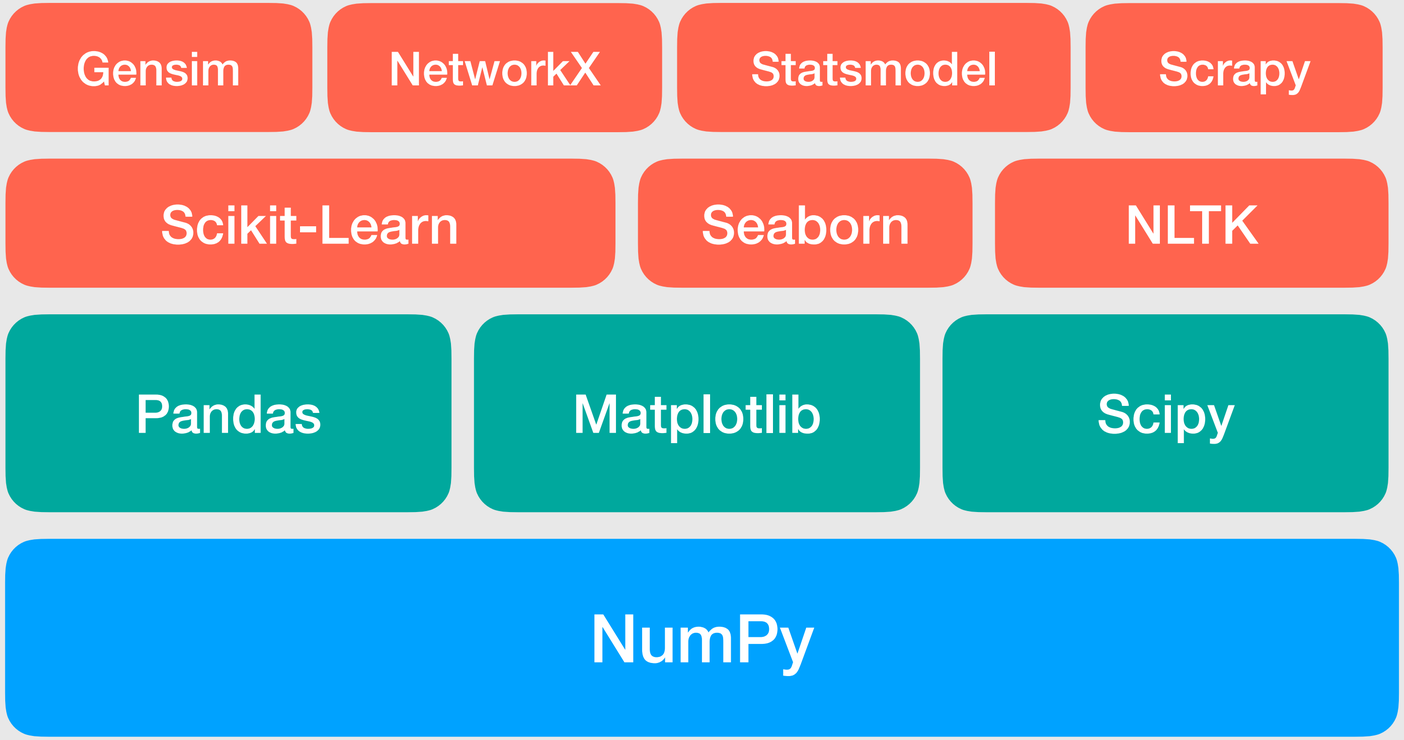
Agenda

1. O que é numpy e por que é importante para mim?
2. O que é vectorização e quais as suas vantagens?
3. O que são arrays e quais as principais operações que eu devo conhecer?
4. Como eu uso arrays para tomar decisões e aprender com meus dados?



NumPy

Eco-sistema

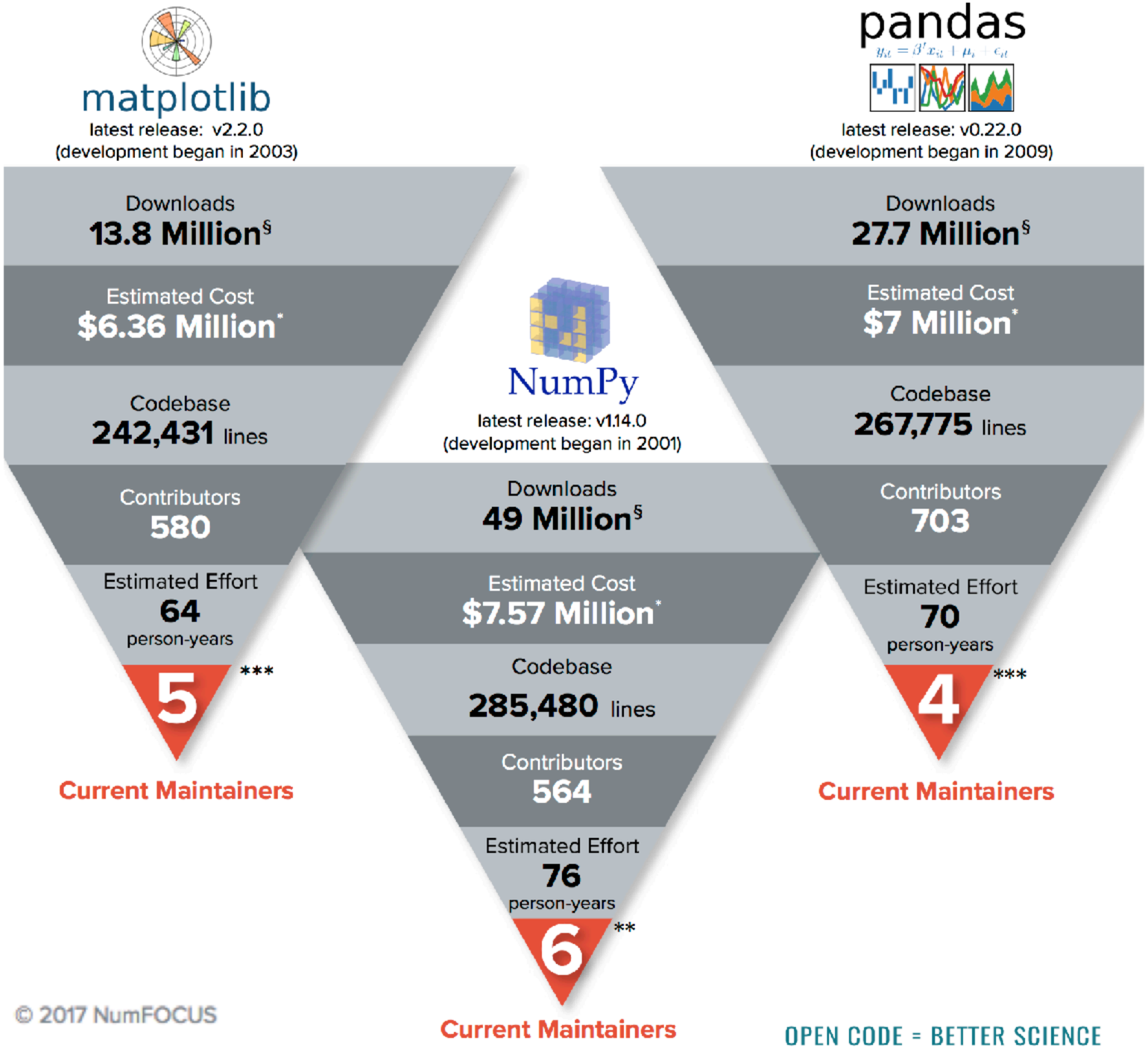

Gensim
NetworkX
Statsmodel
Scrapy
Scikit-Learn
Seaborn
NLTK
Pandas
Matplotlib
Scipy
NumPy

NumPy vs Pandas vs Matplotlib

Numpy: suporte eficiente a arrays multi-dimensionais e múltiplos métodos matemáticas para operar e manipular esses arrays

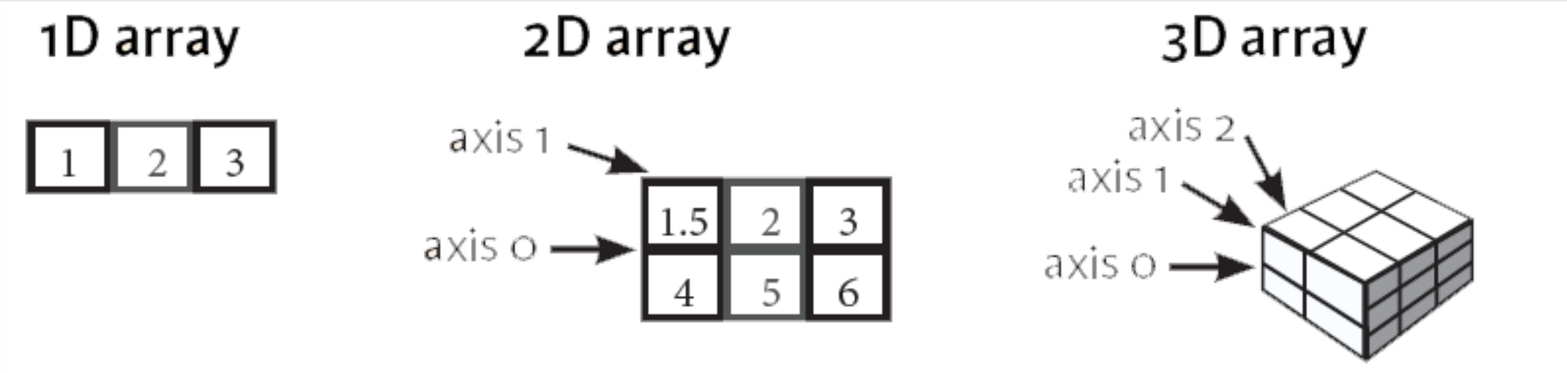
Pandas: manipulação de dados estruturados, com enfoque em dados tabulares e tipos mais complexos (text, datetimes, etc)

Matplotlib: visualização de dados representados como arrays numpy e dataframes pandas


matplotlib
latest release: v2.2.0
(development began in 2003)
Downloads
13.8 Million§
Estimated Cost
$6.36 Million*
Codebase
242,431 lines
Contributors
580
Estimated Effort
64
person-years
5
***
Current Maintainers
pandas
latest release: v0.22.0
(development began in 2009)
Downloads
27.7 Million§
Estimated Cost
$7 Million*
Codebase
267,775 lines
Contributors
703
Estimated Effort
70
person-years
4
***
Current Maintainers
NumPy
latest release: v1.14.0
(development began in 2001)
Downloads
49 Million§
Estimated Cost
$7.57 Million*
Codebase
285,480 lines
Contributors
564
Estimated Effort
76
person-years
6
**
Current Maintainers
© 2017 NumFOCUS
OPEN CODE = BETTER SCIENCE

Numpy Array

- Sequência de valores em N dimensões
- Implementação de vetores e matrizes da álgebra linear
- Extremamente úteis!



<code> … </code>

Recapitulando

- O que são numpy arrays e suas principais propriedades
  - `a.dtype, a.shape, a.ndim, a.itemsize`

- Como criar arrays explicitamente ou utilizando geradores:
  - `np.array([1,2,3])`
  - `np.ones(shape), np.zeros(shape)`
  - `np.random.random(shape)`
  - `np.random.randint(shape)`

Recapitulando

- Como acessar arrays e atribuir valores:
  - `a[0,3], a[-1,-2]`
  - `a[[1,5]], a[True, True, False]`
  - `a[7,7] = 0`

- Utilizando slicing:
  - `a[1:3], a[2:], a[:, -1:-5]`
  - `a[1:5] = [1,2,3,4]`

Recapitulando

- Aprendemos a operar arrays:
  - `a+b, 2*a, a**2, np.add(a, b)`
  - `a>b, a<=b, a>10`
  - `np.sin(a), np.sqrt(b)`
- E fazer agregações:
  - `a.sum(), a.max()`
  - `a.mean(axis=1), np.min(axis=0)`
  - `np.argsort(a), np.argmax(a)`

Conclusões

- Por fim, aprendemos a combinar todas essas ferramentas para manipular dados reais e responder perguntas relevantes
- O poder de generalizar um mesmo código pra múltiplas dimensões é extremamente poderoso
- Aprenderemos novas abstrações e funcionalidades nas próximas aulas, porém todas se baseiam no paradigma de vetorização

DÚVIDAS?!