



INSIGHT

# BIBLIOTECAS PARA MANIPULAÇÃO DE DADOS



# AGENDA

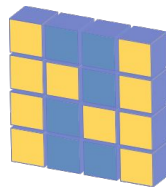
1. NumPy
2. Pandas
3. Matplotlib

# 1. NUMPY

Pacote fundamental para computação científica em Python

# NUMPY

- Operações com **vetores** e **matrizes** de N dimensões
- É essencial para a **computação científica** com Python
- É baseado na linguagem **C**
- Ferramentas para **álgebra linear**
- Diversas operações algébricas básicas
  - Métricas estatísticas
  - Auto-decomposição



NumPy

## EXEMPLOS DE APLICAÇÃO

Uma **matriz** contendo:

- valores de um experimento/simulação em tempos discretos;
- sinal gravado por um equipamento de medição, por exemplo, ondas sonoras;
- pixels de uma imagem, escala de cinza ou coloridos;
- dados tridimensionais;

**Por que é muito útil:** é eficiente na questão da memória e provê operações numéricas rápidas.

## NUMPY OBJECT

**ndarray** (array) - matriz n-dimensional **homogênea**

- É uma tabela de elementos do **mesmo tipo**
- **Indexados** por uma tupla de números inteiros positivos
- As dimensões são chamadas de **axes**
- O tamanho dos **axes** é o **shape**
- O número de **axes** é **rank**



8

VAMOS PARA  
O JUPYTER  
NOTEBOOK!!





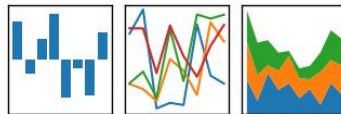
## 2. PANDAS

Easy-to-use data structures and data analysis tools

# PANDAS

- Estruturas de dados **rápidas, flexíveis e expressivas**
- Trabalhar com dados **relacionais** ou **labeled** de forma fácil e intuitiva
- Manipular e analisar dados de forma eficiente
- É bem adequado para diferentes **tipos de dados**:
  - dados tabulares com colunas heterogêneas (SQL table ou .xlsx)
  - séries temporais ordenadas e não ordenadas
  - matrizes com rótulos nas linhas e colunas

**pandas**  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



# DATA STRUCTURES

- Estruturas primárias: **Series** (1-d) e **DataFrame** (2-d)
- É construído sob o **NumPy**
- **Funcionalidades** para manipular e analisar dados de forma eficiente
- Coisas que o pandas **faz bem**:
  - manipulação fácil de *missing data*
  - *size mutability* - colunas podem ser inseridas e excluídas
  - *group* - funcionalidade de agrupamento poderosa e flexível
  - *converter* - conversão de dados irregulares
  - **label-based slicing**
  - **merging / joining**
  - **ferramentas robusta de IO**

12

DE VOLTA AO  
JUPYTER!

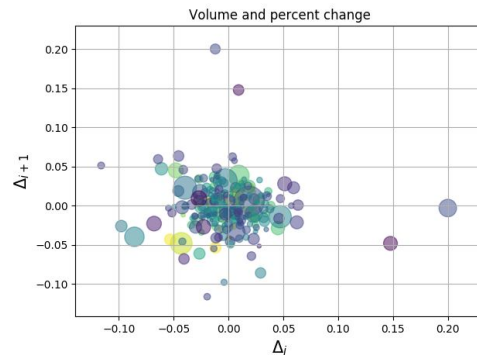
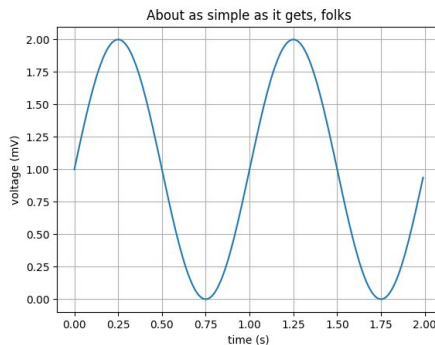
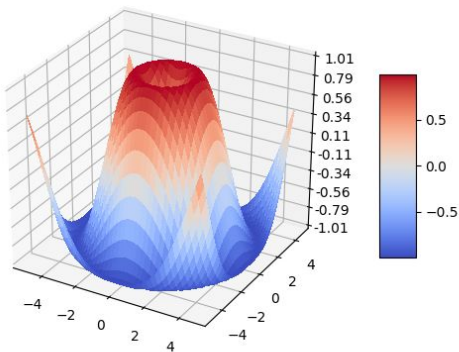


# 3. MATPLOTLIB

2D plotting library

# MATPLOTLIB

- Biblioteca para geração de figuras 2D;
- Integração com Jupyter Notebook
- API *Matlab-like* chamada **pyplot**





# MATPLOTLIB

- Diversos gráficos já implementados:
  - **plot(X,Y)** - gráfico de linha
  - **scatter(X,Y,[c],[s])** - gráfico de dispersão
  - **bar(names, values)** - gráfico de barras
  - **hist(values, bins)**- histograma

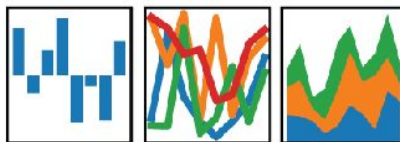


# INTEGRAÇÃO COM PANDAS

- O pandas já traz métodos para gerar gráficos dos dataframes:
  - Gera para todas as colunas (ou subconjunto)
  - Decide o gráfico adequado para o tipo do dado
  - Gráfico pode ser personalizado pela API do Matplotlib

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

17

PARA O  
JUPYTER E  
AVANTE!



ALLONS-Y!