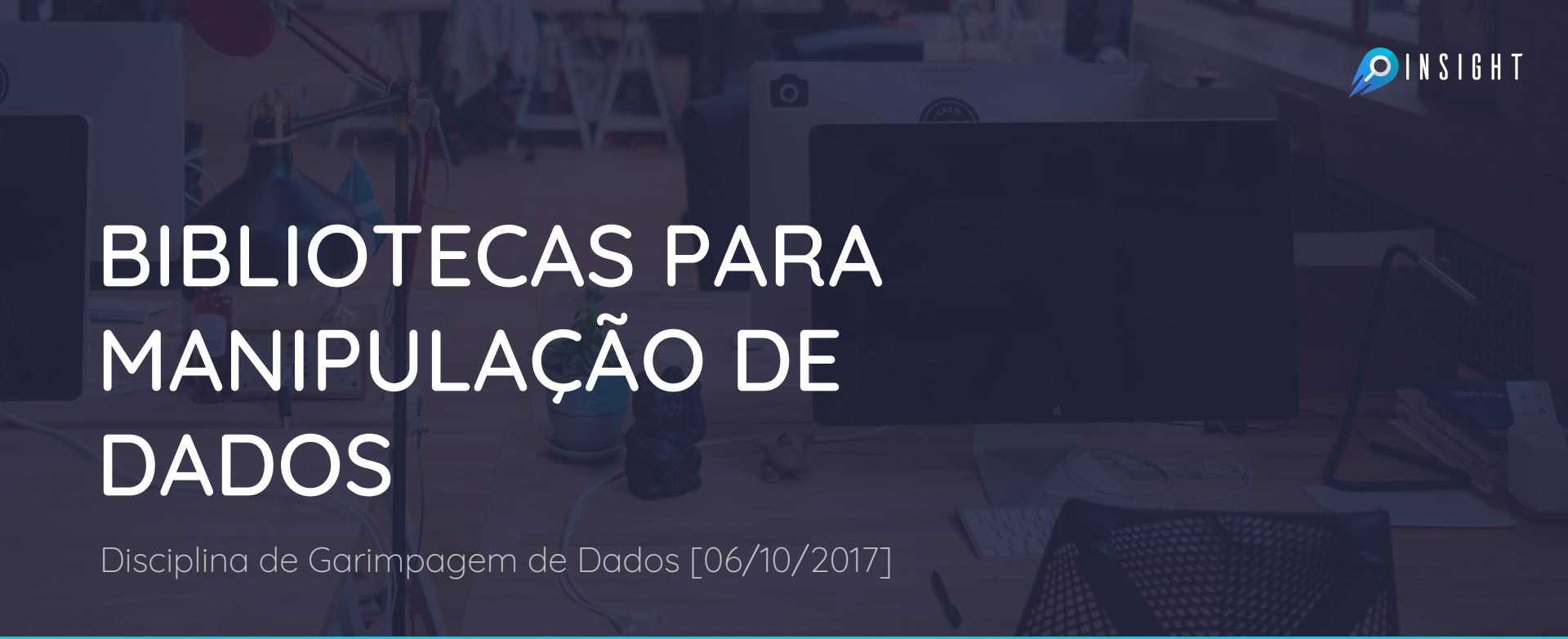

INSIGHT



INSIGHT

BIBLIOTECAS PARA MANIPULAÇÃO DE DADOS

Disciplina de Garimpagem de Dados [06/10/2017]

DEPARTAMENTO DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DO CEARÁ

AGENDA

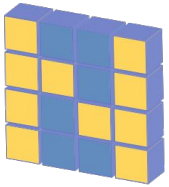
1. NumPy
2. Pandas
3. Matplotlib

1.
NUMPY

Pacote fundamental para computação científica em Python

NUMPY

- Pacote que suporta operações com **vetores** e **matrizes** de N dimensões
- É essencial para a **computação científica** com Python
- É baseado na linguagem **C**
- Objeto **array** para a implementação de arranjos multidimensionais
- Objeto **matrix** para o cálculo com matrizes
- Ferramentas para **álgebra linear**
- Transformadas de **Fourier** básicas
- Ferramentas sofisticadas para geração de **números aleatórios**



NumPy

NumPy, 2017. [http://www.numpy.org]

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 medidos em posições X, Y, Z diferentes, por exemplo, MRI scan;
- entre outros...

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

NumPy and Scipy Documentation, 2017. [https://docs.scipy.org/doc]

INSTALAÇÃO

Acessar o ***environment* conda** criado na aula passada:

- $ conda install numpy

Ajuda interativa:

- $ python
- $ import numpy as np
- $ help(np.array)

Busca específica:

- $ np.lookfor('create array')

NUMPY OBJECT

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

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

Por exemplo: coordenadas de um ponto em um espaço 3D - **[1, 2, 1]**

Qual o rank do array? **1**

Qual o tamanho do axis? **3**

numpy.array != array.array

Quickstart tutorial, 2017. [https://docs.scipy.org/doc/numpy-dev/user/quickstart.html#prerequisites]

VAMOS PRATICAR EM UM JUPYTER NOTEBOOK!!

2.
PANDAS

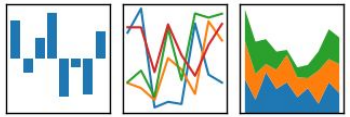
Easy-to-use data structures and data analysis tools

PANDAS

- Estruturas de dados **rápidas**, **flexíveis** e **expressivas**
- Projetadas para tornar o trabalho com dados **relacionais** ou **labeled** de forma fácil e intuitiva
- **Funcionalidades** para 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 (homogêneas ou heterogêneas) com rótulos nas linhas e colunas
  - qualquer outra forma de conjuntos de dados observacionais/estatísticos

pandas

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



Pandas, 2017. [http://pandas.pydata.org]

DATA STRUCTURES

- Estruturas primárias: **Series** (1-d) e **DataFrame** (2-d)
  - Maioria dos casos de **usos típicos**: finanças, estatísticas, ciências sociais e muitas áreas da engenharia
- 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 em um DataFrame
  - ***data alignment*** - automática e explícita
  - ***group*** - funcionalidade de agrupamento poderosa e flexível (split-apply-combine op / agg / trans)
  - ***converter*** - conversão de dados irregulares e indexados do Python, NumPy em DataFrames
  - **label-based slicing**
  - **merging / joining**
  - **reshaping and pivoting of data sets**
  - **rotulagem hierárquica de eixos**
  - **ferramentas robusta de IO**
  - **funcionalidades específicas para séries temporais**

INSTALAÇÃO

Acessar o ***environment* conda** criado na aula passada:

- $ conda install pandas
- $ conda install matplotlib

E o que é o **Matplotlib**?

- Python 2D plotting
- Diversos formatos de gráfico
- Alta qualidade

VAMOS PRATICAR EM UM JUPYTER NOTEBOOK!!