Ferramentas de Softwares para uso em ciência de Dados II

OBJETIVO

Nasser Santiago Boan

> arquivologia - UNB

> tecnologia do Petróleo e Gás - UFRJ

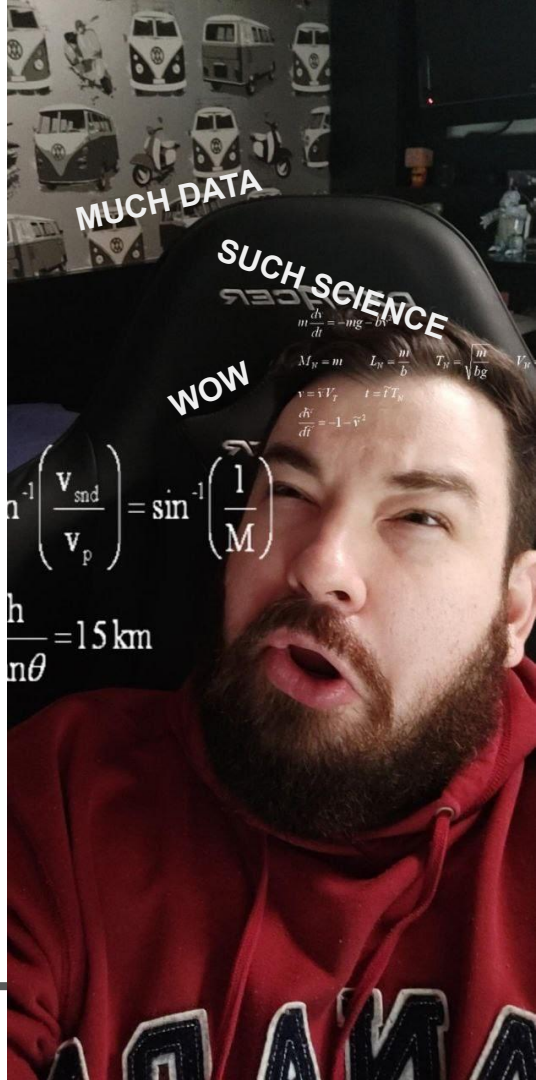
> MBA Processos de Negócio - IBMEC

> IBM Data Science Professional Certified

> líder de Ciência de Dados - Certsys

> Cientista de Dados - Stefanini

> docente Pós-Graduação Ciência de Dados



Ementa

- Básico
  - pip
  - conda
  - miniconda
  - jupyter notebook
- Nivelamento 2
  - Variáveis e operações matemáticas
  - Operações com strings
  - Listas
  - Tuplas
  - Dicionários
  - Condições lógicas
  - Loops (for + compreensão de lista + while)
  - Funções (def + lambda)
  - Funções built-in (zip, enumerate,map e filter)

Ementa

- Files
  - Lendo arquivos com OPEN
  - Escrevendo arquivos com OPEN
- Numpy
  - Numpy array
  - Operações com arrays
  - Indexing e slicing
  - Stacking
  - Funções estatísticas (mean, var, std, median, quantiles)

IESB

Ementa

- Pandas
  - Dataframes
  - read_ / to_
  - Indexing e slicing (.loc , .iloc , filter , head, sample)
  - Tipos de dados e datas (to_datetime)
  - Funções estatísticas (describe, corr, skew, mode, unique, nunique)
  - Transformações (apply + operações com séries)
  - Reshaping e sorting (pivot + transpose + nlargest + nsmallest + value_counts + sort_values)
  - Concat e merge
  - Valores nulos (isna, isnull, fillna-- imputação --, drop, dropna, replace)
  - Method Chaining

IESB

Ementa

- matplotlib
  - Arquitetura (pyplot e axes)
  - Anatomia dos gráficos em matplotlib
  - Plotando (figure, plt.bar, plt.scatter, plt.plot, title, labels, legend, colors, axvline, axhline)
  - Subplots
  - Abrindo imagens
  - Exportando gráficos (plt.savefig)

IESB

Ementa

- Scikit-Learn
  - Pre-processamento (standartscaler, minmaxscaler, onehotencoding)
  - Conceitos de Regressão / Classificação / Clusterização
    - Dataset de treino e teste (train_test_split)
    - Underfitting / overfitting
  - Regressão linear simples (na mão + coeficientes explícitos)
    - Métricas de regressão e função de custo
  - Regressão linear múltipla e polynomial features
  - DecisionTree / RandomForest
    - Métricas de classificação
  - KMeans
    - métricas de clusterização
  - Kfold e GridsearchCV

Avaliação

- Lista 01 (python)
- Lista 02 (pandas + matplotlib)
- Projeto 01 (análise de dados)
- Lista 03 (conceitos de machine learning 1)
- Lista 04 (conceitos de machine learning 2)
- Projeto 02 (projeto ML)

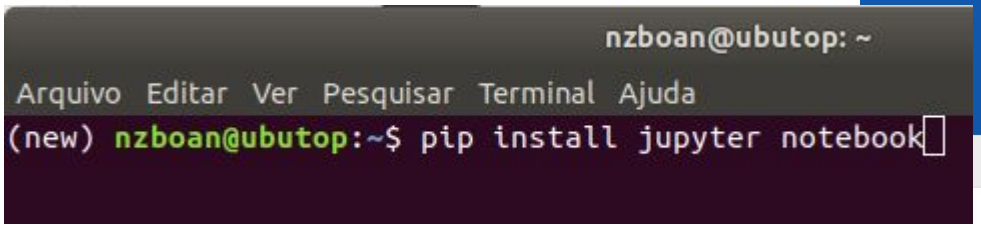
IESB

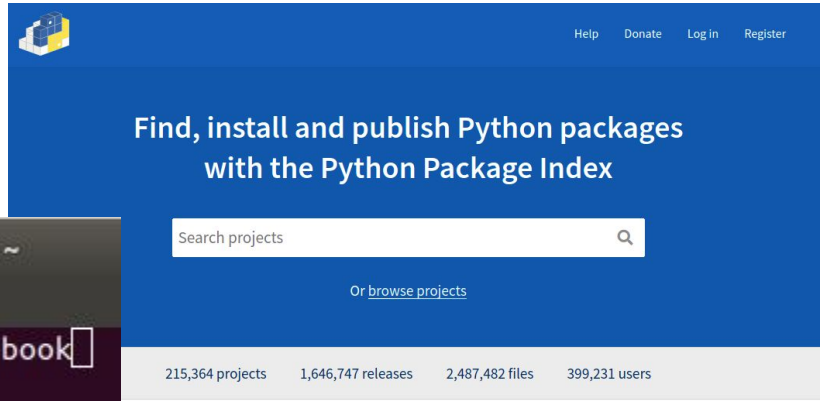
Módulo Básico

pip

O pip é o gerenciador de pacotes do python.

```
$ pip install <nome do pacote>

$ pip --version

$ pip list

$ pip install -r requirements.txt
```





conda e miniconda

O conda vai além e gerência ambientes e dependências

```
$ conda create env -n <nome do ambiente>

$ conda create env -n <nome do ambiente> python=3.6

$ conda env create -f environment.yml

$ conda list

$ conda install <nome do pacote>

$ conda activate <nome do ambiente>

$ conda deactivate <nome do ambiente>
```

IESB

conda e miniconda

O conda vai além e gerência ambientes e dependências

```
(base) nzboan@ubutop:~$ conda list
# packages in environment at /home/nzboan/miniconda3:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                         main
_tflow_select             2.1.0                       gpu
absl-py                   0.8.1                    py37_0
altair                    3.2.0                    py37_0
asn1crypto                1.2.0                    py37_0
astor                     0.8.0                    py37_0
astroid                   2.3.3                    py37_0
attrs                     19.3.0                     py_0
backcall                  0.1.0                    py37_0
beautifulsoup4            4.8.1                    py37_0
biopython                 1.74                     pypi_0    pypi
blas                      1.0                         mkl
bleach                    3.1.0                    py37_0
blosc                     1.16.3               hd408876_0
bokeh                     1.4.0                    py37_0
boto3                     1.9.232                  pypi_0    pypi
botocore                  1.12.232                 pypi_0    pypi
branca                    0.3.1                      py_0    conda-forge
bzip2                     1.0.8                h7b6447c_0
c-ares                    1.15.0               h7b6447c_1001
ca-certificates           2019.11.27                  0
cachetools                3.1.1                    pypi_0    pypi
cairo                     1.14.12              h8948797_3
certifi                   2019.11.28               py37_0
cffi                      1.13.2           py37h2e261b9_0
chardet                   3.0.4                 py37_1003
click                     7.0                      py37_0
click-plugins             1.1.1                    pypi_0    pypi
cligj                     0.5.0                    pypi_0    pypi
cloudpickle               1.2.2                      py_0
cogroo-interface          0.3                      pypi_0    pypi
colorcet                  2.0.2                      py_0
conda                     4.8.1                    py37_0
```

IESB

conda e miniconda

O conda vai além e gerência ambientes e dependências

```
(base) nzboan@ubutop:~$ conda activate new
(new) nzboan@ubutop:~$ conda list
# packages in environment at /home/nzboan/miniconda3/envs/new:
#
# Name                    Version                   Build  Channel
(new) nzboan@ubutop:~$
```

```
In [5]: from sklearn.datasets import load_iris

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-5-56d11ecab3a7> in <module>
----> 1 from sklearn.datasets import load_iris

ModuleNotFoundError: No module named 'sklearn'
```


IESB

conda e miniconda

```
(new) nzboan@ubutop:~$ conda install scikit-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/nzboan/miniconda3/envs/new

  added / updated specs:
    - scikit-learn


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2020.1.1   |                0         125 KB
    certifi-2019.11.28         |           py38_0         153 KB
    ld_impl_linux-64-2.33.1    |       h53a641e_7         568 KB
    mkl-service-2.3.0          |   py38he904b0f_0          62 KB
    mkl_fft-1.0.15             |   py38ha843d7b_0         159 KB
    mkl_random-1.1.0           |   py38h962f231_0         340 KB
    numpy-1.18.1               |   py38h4f9e942_0           5 KB
    numpy-base-1.18.1          |   py38hde5b4d6_1         4.2 MB
    openssl-1.1.1d             |       h7b6447c_3         2.5 MB
    pip-20.0.2                 |           py38_1         1.7 MB
    python-3.8.1               |       h0371630_1        49.5 MB
    scikit-learn-0.22.1        |   py38hd81dba3_0         5.4 MB
    scipy-1.3.2                |   py38h7c811a0_0        14.0 MB
    setuptools-45.1.0          |           py38_0         522 KB
    six-1.14.0                 |           py38_0          27 KB
    sqlite-3.30.1              |       h7b6447c_0         1.1 MB
    wheel-0.34.1               |           py38_0          50 KB
    ------------------------------------------------------------
                                           Total:        80.3 MB
```

```
In [2]: from sklearn.datasets import load_iris

In [9]: data = load_iris()

In [10]: data.target

Out[10]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```


IESB

conda e miniconda

environment.yml

```yaml
name: stats
dependencies:
  - jupyterlab
  - pandas
  - numpy
  - scikit-learn
```

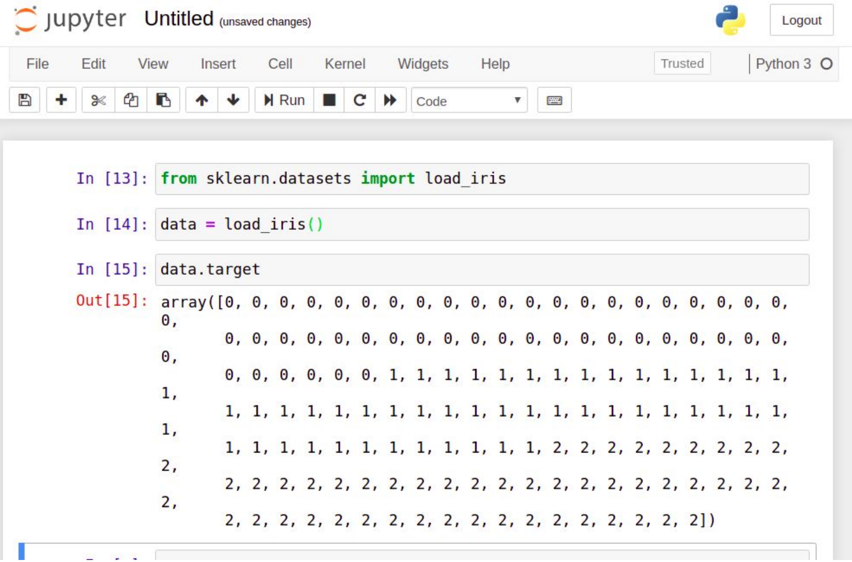
```
(base) nzboan@ubutop:~$ conda env create -f environment.yml
```

```
(base) nzboan@ubutop:~$ conda env create -f environment.yml
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages
pandas-1.0.0         | 8.8 MB    | ##################################### | 100%
jupyterlab-1.2.5     | 2.8 MB    | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate stats
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) nzboan@ubutop:~$
```

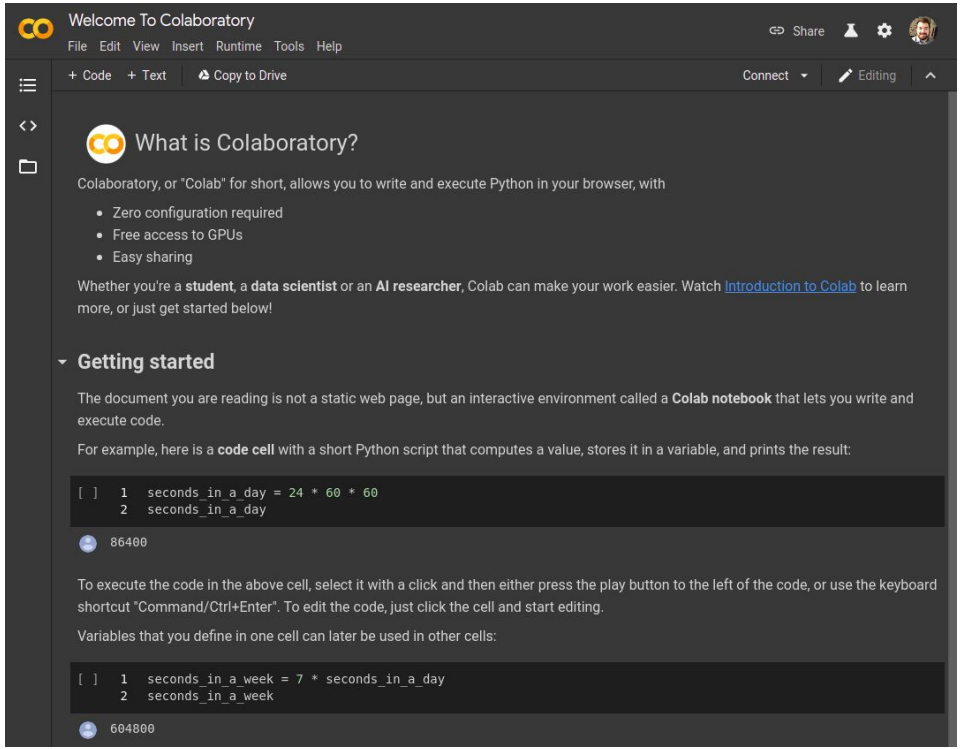

IESB

jupyter notebook

O jupyter notebook é documento que contém tanto código quanto elementos de texto. Ele executa código célula a célula, pode ser lido por humanos (human-readable) ou pela máquina (machine-readable).



google colab



referências importantes

instalação do conda

https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html#regular-installation

instalação do pip

https://pip.pypa.io/en/stable/installing/

instalação do jupyter notebook

https://jupyter.org/install.html

google colab

https://colab.research.google.com/

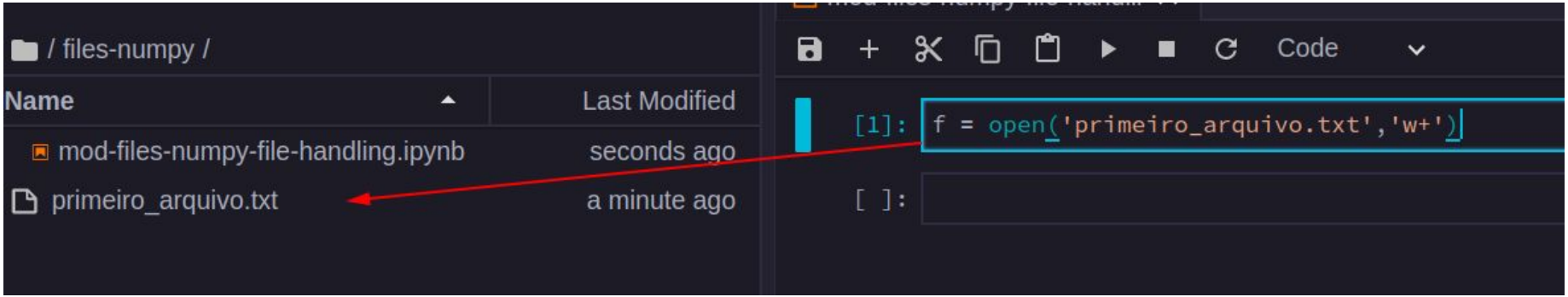
Nivelamento (again...)

https://github.com/aulas-iesb/CastoresIndomaveis

Files e Numpy

File Handling

Em python não há necessidade de importar bibliotecas para ler e escrever arquivos. As funções built-in open e write resolvem bem as grande maioria das necessidades.



open()

A função open() abre um arquivo e retorna um objeto. A função possui vários e o objeto pode ou não ser alterado de acordo com o modo passado. Os modos que a função pode receber:

Modo	Descrição
‘r’	Modo padrão da função. Abre um arquivo para leitura.
‘w’	Esse modo abre um arquivo para escrita. Se o arquivo não existir esse modo cria o arquivo. Se o arquivo já existe ele substitui o arquivo.
‘x’	Cria um arquivo novo. Se o arquivo já existir a função retorna um erro
‘a’	Abre um arquivo no modo ‘append’. Se o arquivo não existe ele cria um novo.
‘+’	Esse modo abre um arquivo para leitura e escrita.

read()

```
[33]: f = open('lorem.txt','r')
      conteudo = f.read()
      f.close()

[35]: print(conteudo)
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sagittis quis massa sed euismod. Aenean at mattis nisl. Vestibulum a imperdiet lacus.
Duis hendrerit, justo maximus convallis placerat, purus nibh vulputate nibh, ac varius libero est eu orci. Fusce ac dolor id ex dictum mattis. Donec f
inibus pellentesque ligula, et semper tellus varius maximus. In rutrum vitae quam nec suscipit. Orci varius natoque penatibus et magnis dis parturient
montes, nascetur ridiculus mus. Nullam laoreet dolor non felis tincidunt dignissim. Suspendisse lacinia accumsan risus vitae rhoncus. Nulla luctus con
vallis enim, ut blandit arcu dignissim at. Pellentesque pharetra ornare efficitur. In molestie, quam in tristique ullamcorper, purus nulla accumsan m
i, et lobortis purus magna quis risus. Cras magna ipsum, accumsan in purus eget, tincidunt feugiat est. Proin laoreet velit a nunc ullamcorper varius.

Cras vel pharetra lacus. In imperdiet at erat ac suscipit. Donec eget urna vel lectus fermentum maximus. Quisque eu tempus nisl. Sed ultricies lacus q
uis condimentum efficitur. Quisque non metus at eros eleifend eleifend. Fusce et semper nisi, lobortis consequat erat. Praesent condimentum pulvinar n
unc in scelerisque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Duis in maximus lacus, vel eleifend ligula. Integer in elementum leo. Nam interdum pharetra neque, a lacinia arcu. Donec scelerisque velit sit amet ju
sto sagittis, eu commodo quam viverra. Vivamus non est viverra mi vehicula scelerisque. Donec id suscipit ligula. Nullam condimentum semper elit, non
accumsan justo consectetur a. Quisque non nisl elit. Ut ultrices risus ut tortor tincidunt accumsan. Curabitur aliquam mi nec mi aliquam lacinia. Maec
enas eget felis non mi ultrices suscipit.
```


IESB

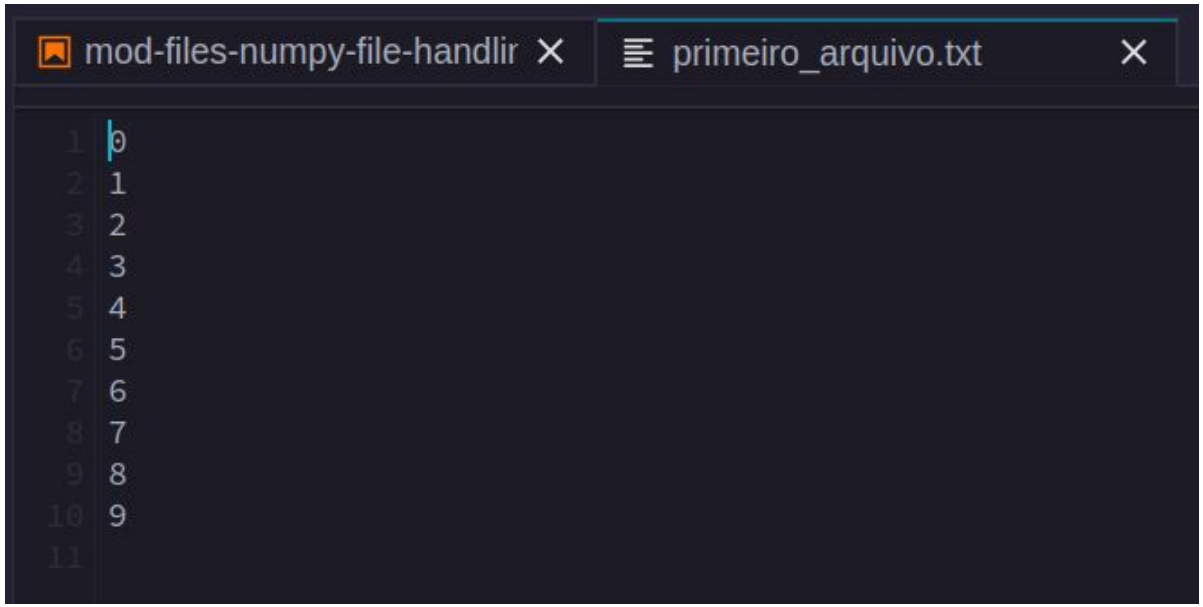
readlines()

```
[37]: f = open('lines.txt','r')
      conteudo = f.readlines()
      for i in conteudo:
          print(i)

      linha 1

      linha 2

      linha 3

      linha 4

      linha 5

      linha 6
```


IESB

write()

```
[51]: with open('primeiro_arquivo.txt','w') as f:
          for i in range(10):
              f.write(f'{i}\n')
```



csv.reader()

```
[38]: import csv

[40]: with open('avocado.csv') as csv_file:
          csv_reader = csv.reader(csv_file,delimiter=',')

          for row in csv_reader:
              print(row)

['', 'Date', 'AveragePrice', 'Total Volume', '4046', '4225', '4770', 'Total Bags', 'Small Bags', 'Large Bags', 'XLarge Bags', 'type', 'year', 'regio
n']
['0', '2015-12-27', '1.33', '64236.62', '1036.74', '54454.85', '48.16', '8696.87', '8603.62', '93.25', '0.0', 'conventional', '2015', 'Albany']
['1', '2015-12-20', '1.35', '54876.98', '674.28', '44638.81', '58.33', '9505.56', '9408.07', '97.49', '0.0', 'conventional', '2015', 'Albany']
['2', '2015-12-13', '0.93', '118220.22', '794.7', '109149.67', '130.5', '8145.35', '8042.21', '103.14', '0.0', 'conventional', '2015', 'Albany']
['3', '2015-12-06', '1.08', '78992.15', '1132.0', '71976.41', '72.58', '5811.16', '5677.4', '133.76', '0.0', 'conventional', '2015', 'Albany']
['4', '2015-11-29', '1.28', '51039.6', '941.48', '43838.39', '75.78', '6183.95', '5986.26', '197.69', '0.0', 'conventional', '2015', 'Albany']
```


IESB

csv.reader()

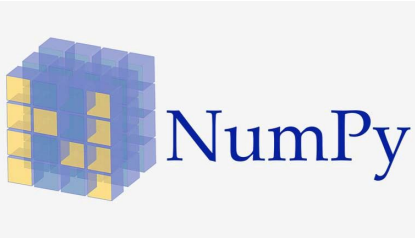
```
[44]: avg_price = []


      with open('avocado.csv') as csv_file:
          csv_reader = csv.reader(csv_file,delimiter=',')
          next(csv_reader)
          for row in csv_reader:
              avg_price.append(row[2])

[47]: avg_price[:10]

[47]: ['1.33',
       '1.35',
       '0.93',
       '1.08',
       '1.28',
       '1.26',
       '0.99',
       '0.98',
       '1.02',
       '1.07']
```

csv.writer()

```
[76]: import csv

      with open('empregados.csv','w') as f:
          empregados_writer = csv.writer(f,delimiter=',',quotechar='"')

          empregados_writer.writerow(['nome','idade','salario'])
          empregados_writer.writerow(['nasser',29,1000])
```

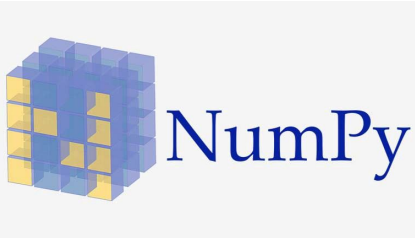
Delimiter: ,

	nome	idade	salario
1	nasser	29	1000
2	joao	20	1500



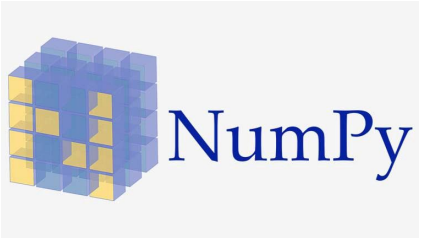

```
[2]: import numpy as np

     a = np.array([[1,2,2],[2,2,2]])
     a

[2]: array([[1, 2, 2],
            [2, 2, 2]])
```


IESB



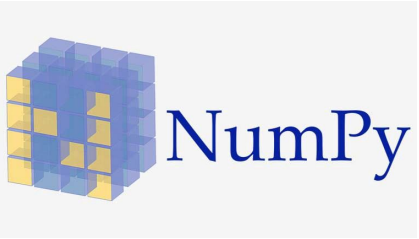

```
[3]: a.ndim

[3]: 2
```

```
[31]: np.zeros((40,50,50,50)).ndim

[31]: 4
```




```
[38]: a.shape

[38]: (2, 3)
```

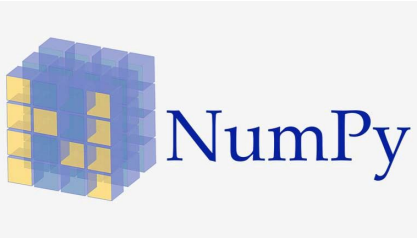
```
[13]: b = np.array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]])
      b

[13]: array([[ 1,  2,  3,  4,  5],
             [ 6,  7,  8,  9, 10],
             [11, 12, 13, 14, 15]])

[39]: b.shape

[39]: (3, 5)
```




```
[43]: a.size
[43]: 6
```

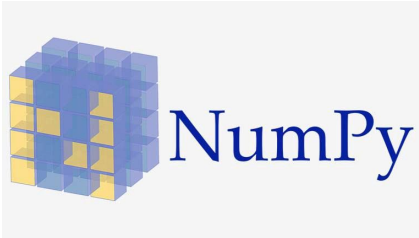
```
[44]: b.size
[44]: 15
```

```
[42]: np.zeros((40,50,50,50)).size
[42]: 5000000
```




```
[61]: a + 1

[61]: array([[2, 3, 3],
             [3, 3, 3]])

[62]: a ** 2

[62]: array([[1, 4, 4],
             [4, 4, 4]])

[63]: a - 10

[63]: array([[-9, -8, -8],
             [-8, -8, -8]])

[66]: (a*3) > 5

[66]: array([[False,  True,  True],
             [ True,  True,  True]])
```

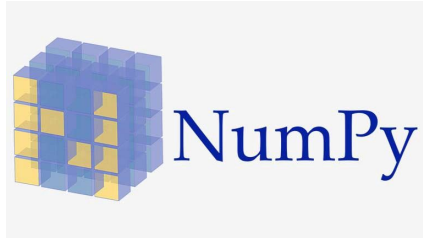
operações




```
[52]: a + b

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-52-bd58363a63fc> in <module>
----> 1 a + b

ValueError: operands could not be broadcast together with shapes (2,3) (3,5)
```

```
[57]: c = np.array([[2,1,1],[1,1,1]])
      c

[57]: array([[2, 1, 1],
             [1, 1, 1]])
```

operações



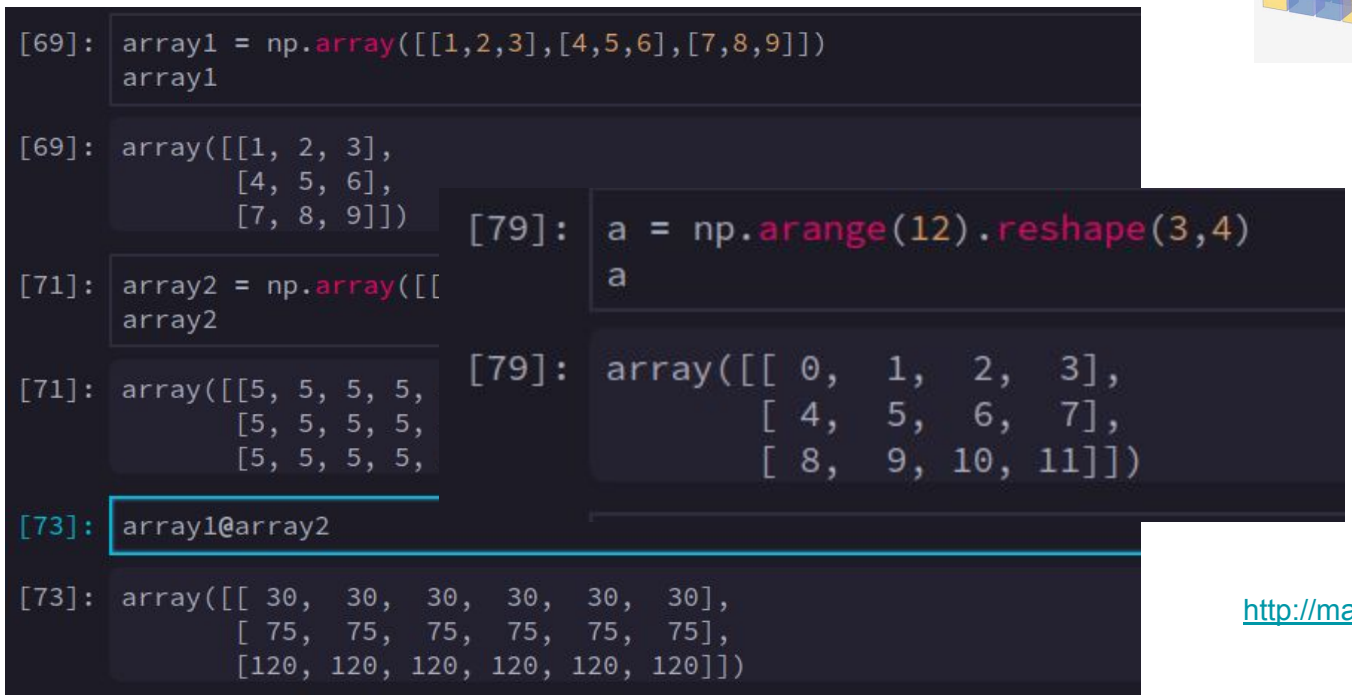

```
[57]: c = np.array([[2,1,1],[1,1,1]])
      c

[57]: array([[2, 1, 1],
             [1, 1, 1]])
```

```
[56]: a + c

[56]: array([[3, 3, 3],
             [3, 3, 3]])
```

```
[67]: a - c

[67]: array([[-1,  1,  1],
             [ 1,  1,  1]])
```

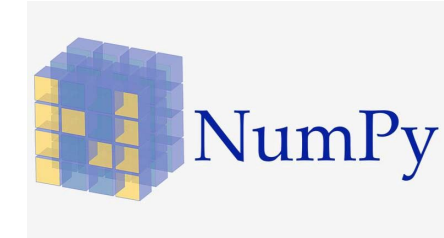

IESB

operações



http://matrixmultiplication.xyz/

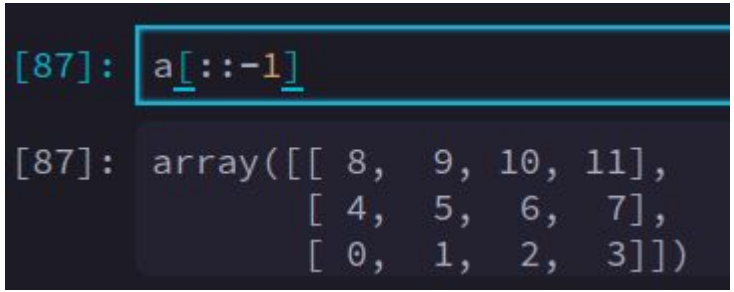
indexing & slicing

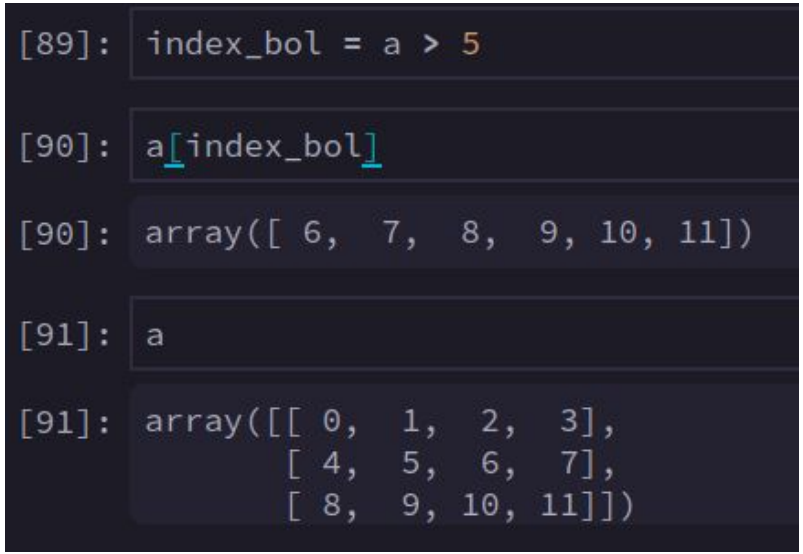



```
[79]: a = np.arange(12).reshape(3,4)
      a

[79]: array([[ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11]])
```

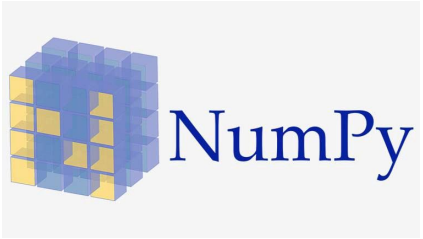
```
[82]: a[1][2]

[82]: 6

[83]: a[:2]

[83]: array([[0, 1, 2, 3],
             [4, 5, 6, 7]])

[86]: a[::2]

[86]: array([[ 0,  1,  2,  3],
             [ 8,  9, 10, 11]])
```

indexing & slicing





stacking

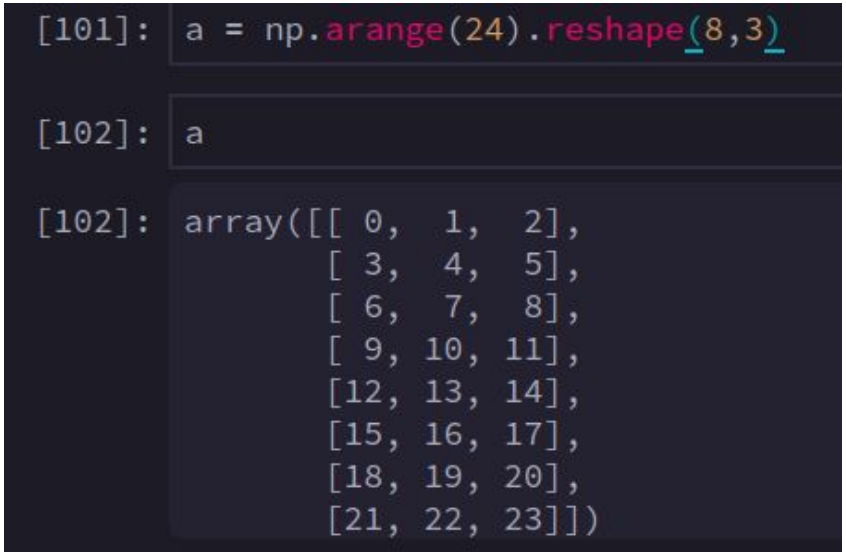



```
a

[79]: array([[ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11]])
```

```
[96]: b

[96]: array([[ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11],
             [12, 13, 14, 15]])
```

```
[99]: np.vstack((a,b))

[99]: array([[ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11],
             [ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11],
             [12, 13, 14, 15]])
```

funções estatísticas



funções estatísticas