



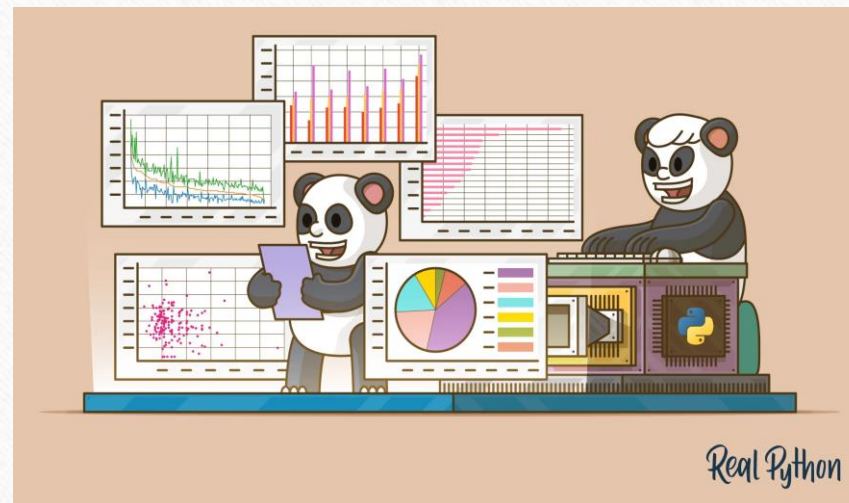
Biblioteca Pandas - Python

Senai – Almirante Tamandaré 24/09/2022

Pablo Ricardo de abreu

• O que é a biblioteca Pandas ?

- Pandas é uma ferramenta de análise e manipulação de dados de código aberto.
- Rápida, poderosa, flexível e fácil de usar.
- Construído sobre a linguagem Python.





Instalando o Pandas no VSCODE

The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, and Termina. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Remote Explorer. The main editor area is dark and contains a large, faint watermark of a stylized 'X' shape. Overlaid on this watermark is the text: "No terminal digite o seguinte comando: **pip install pandas**". A red arrow points from this text to the terminal window. The terminal window is located at the bottom of the interface and has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing a PowerShell prompt. The text in the terminal reads: "Copyright (C) Microsoft Corporation. Todos os direitos reservados. Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! <https://aka.ms/PSWindows> PS Z:\FBDDAP-SMB-01_PABLO_ABREU\AULA 8 PY> **pip install pandas**". The command "pip install pandas" is highlighted with a red rectangular box.

```
File Edit Selection View Go Run Termina AULA 8 PY - Visual Stu... | 0%
```

No terminal digite o seguinte comando:
pip install pandas

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... powershell + -

Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! <https://aka.ms/PSWindows>

PS Z:\FBDDAP-SMB-01_PABLO_ABREU\AULA 8 PY> **pip install pandas**



O que é Dataframe e Series ?

- Um Dataframe é semelhante a uma matriz mas as suas colunas têm nomes e podem conter dados de tipo diferente.
- Um Dataframe pode ser visto como uma tabela de uma base de dados, em que cada linha corresponde a um registo (linha) da tabela.

| Jupyter format | | | | | Standard Python format | | | | |
|----------------|----------|------|------|------|------------------------|----------|------|------|------|
| YEARMODA | TEMP | MAX | MIN | | YEARMODA | TEMP | MAX | MIN | |
| 0 | 20160601 | 65.5 | 73.6 | 54.7 | 0 | 20160601 | 65.5 | 73.6 | 54.7 |
| 1 | 20160602 | 65.8 | 80.8 | 55.0 | 1 | 20160602 | 65.8 | 80.8 | 55.0 |
| 2 | 20160603 | 68.4 | 77.9 | 55.6 | 2 | 20160603 | 68.4 | 77.9 | 55.6 |
| 3 | 20160604 | 57.5 | 70.9 | 47.3 | 3 | 20160604 | 57.5 | 70.9 | 47.3 |
| 4 | 20160605 | 51.4 | 58.3 | 43.2 | 4 | 20160605 | 51.4 | 58.3 | 43.2 |
| 5 | 20160606 | 52.2 | 59.7 | 42.8 | 5 | 20160606 | 52.2 | 59.7 | 42.8 |
| 6 | 20160607 | 56.9 | 65.1 | 45.9 | 6 | 20160607 | 56.9 | 65.1 | 45.9 |
| 7 | 20160608 | 54.2 | 60.4 | 47.5 | 7 | 20160608 | 54.2 | 60.4 | 47.5 |
| 8 | 20160609 | 49.4 | 54.1 | 45.7 | 8 | 20160609 | 49.4 | 54.1 | 45.7 |
| 9 | 20160610 | 49.5 | 55.9 | 43.0 | 9 | 20160610 | 49.5 | 55.9 | 43.0 |

Pandas DataFrame
pandas.core.frame.DataFrame

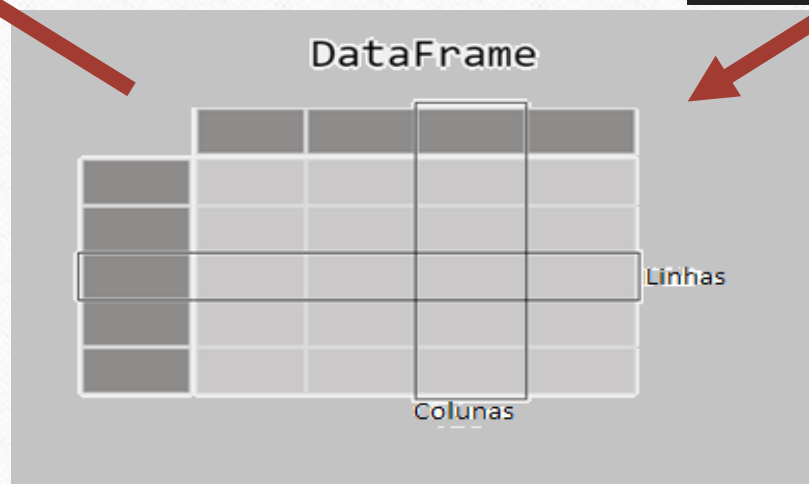
Standard Python format

0 65.5
1 65.8
2 68.4
3 57.5
4 51.4
5 52.2
6 56.9
7 54.2
8 49.4
9 49.5
Name: TEMP, dtype: float64

Pandas Series

pandas.core.series.Series

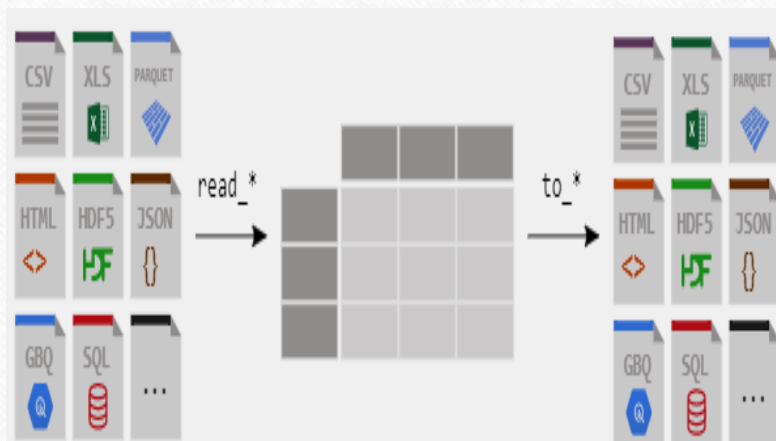
```
In [1]: import pandas as pd  
  
In [2]: pd.DataFrame({  
Out[2]:
```





Quais dados são possíveis de Importar e Exportar no Pandas e seus métodos ?

- A biblioteca pandas tem um arsenal de possibilidades onde posso extrair de um banco de dados ou de uma estrutura não estruturados para realizar análises e tratamentos dos dados, exemplos: CSV, Excel, HTML, SQL, Json e etc.
- Podemos formatar esses dados e exportar com outros formatos diferentes do original importado.



```
In [143]: df.to_csv("foo.csv")
```

```
In [144]: pd.read_csv("foo.csv")
```

```
Out[144]:
```

| | Unnamed: 0 | A | B | C | D |
|-----|------------|------------|-----------|-----------|------------|
| 0 | 2000-01-01 | 0.350262 | 0.843315 | 1.798556 | 0.782234 |
| 1 | 2000-01-02 | -0.586873 | 0.034907 | 1.923792 | -0.562651 |
| 2 | 2000-01-03 | -1.245477 | -0.963406 | 2.269575 | -1.612566 |
| 3 | 2000-01-04 | -0.252830 | -0.498066 | 3.176886 | -1.275581 |
| 4 | 2000-01-05 | -1.044057 | 0.118042 | 2.768571 | 0.386039 |
| ... | ... | ... | ... | ... | ... |
| 995 | 2002-09-22 | -48.017654 | 31.474551 | 69.146374 | -47.541670 |
| 996 | 2002-09-23 | -47.207912 | 32.627390 | 68.505254 | -48.828331 |
| 997 | 2002-09-24 | -48.907133 | 31.990402 | 67.310924 | -49.391051 |
| 998 | 2002-09-25 | -50.146062 | 33.716770 | 67.717434 | -49.037577 |
| 999 | 2002-09-26 | -49.724318 | 33.479952 | 68.108014 | -48.822030 |

```
[1000 rows x 5 columns]
```




Real Python



Quais os principais comandos e para que servem ?

- Comando Dataframe:

```
1 import pandas as pd
2 df = pd.DataFrame(
3     {
4         "Name": [
5             "Braund, Mr. Owen Harris",
6             "Allen, Mr. William Henry",
7             "Bonnell, Miss. Elizabeth",
8         ],
9         "Age": [22, 35, 58],
10        "Sex": ["male", "male", "female"],
11    }
12 )
13
14 print(df)
15
```



| | Name | Age | Sex |
|---|--------------------------|-----|--------|
| 0 | Braund, Mr. Owen Harris | 22 | male |
| 1 | Allen, Mr. William Henry | 35 | male |
| 2 | Bonnell, Miss. Elizabeth | 58 | female |

PS Z:\FBDDAP-SMB-01_PABLO_ABREU\AULA 8 PY> █

Esse comando transforma uma entrada de dados bem parecido com uma biblioteca ou matrizes e vetores em uma tabela com títulos e referencias das linhas, similar ao que popularmente conhecemos como é a estrutura de tabelas do Excel.



Quais os principais comandos e para que servem ?

- Após definir os dados usaremos algumas ferramentas de análise:

```
In [2]: titanic = pd.read_csv("data/titanic.csv")
```

Read: função para ler dados armazenados como um arquivo

```
In [4]: titanic.head(8)
```

Out[4]:

| | PassengerId | Survived | Pclass | ... | Fare | Cabin | Embarked |
|---|-------------|----------|--------|-----|---------|-------|----------|
| 0 | 1 | 0 | 3 | ... | 7.2500 | NaN | S |
| 1 | 2 | 1 | 1 | ... | 71.2833 | C85 | C |
| 2 | 3 | 1 | 3 | ... | 7.9250 | NaN | S |
| 3 | 4 | 1 | 1 | ... | 53.1000 | C123 | S |
| 4 | 5 | 0 | 3 | ... | 8.0500 | NaN | S |
| 5 | 6 | 0 | 3 | ... | 8.4583 | NaN | Q |
| 6 | 7 | 0 | 1 | ... | 51.8625 | E46 | S |
| 7 | 8 | 0 | 3 | ... | 21.0750 | NaN | S |

[8 rows x 12 columns]

Head: função para priorizar a visualização da quantidade de linhas determinadas.

```
In [23]: adult_names = titanic.loc[titanic["Age"] > 35, "Name"]
```

```
In [24]: adult_names.head()
```

Out[24]:

| | |
|----|---|
| 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... |
| 6 | McCarthy, Mr. Timothy J |
| 11 | Bonnell, Miss. Elizabeth |
| 13 | Andersson, Mr. Anders Johan |
| 15 | Hewlett, Mrs. (Mary D Kingcome) |

Name: Name, dtype: object

Dtypes: função que identifica o tipo dos dados usados no Dataframe.

Loc: Localiza as informações conforme solicitado e realiza um filtro e mostra apenas o que foi selecionado.

```
In [5]: titanic.dtypes
```

Out[5]:

| | |
|-------------|---------|
| PassengerId | int64 |
| Survived | int64 |
| Pclass | int64 |
| Name | object |
| Sex | object |
| Age | float64 |
| SibSp | int64 |
| Parch | int64 |
| Ticket | object |
| Fare | float64 |
| Cabin | object |
| Embarked | object |
| dtype: | object |



Quais os principais comandos e para que servem ?

- Agora conseguimos realizar algumas manipulações dos dados:

```
In [4]: titanic["Age"].mean()  
Out[4]: 29.69911764705882
```

Mean: Calcula a media dos dados selecionados.

```
In [5]: titanic[["Age", "Fare"]].median()  
Out[5]:  
Age      28.0000  
Fare     14.4542  
dtype: float64
```

Median: calcula a media de mais de um agrupamento de dados.

```
In [8]: air_quality = pd.concat([air_quality_pm25, air_quality_no2], axis=0)
```

```
In [9]: air_quality.head()
```

Out[9]:

| | date.utc | location | parameter | value |
|---|---------------------------|----------|-----------|-------|
| 0 | 2019-06-18 06:00:00+00:00 | BETR801 | pm25 | 18.0 |
| 1 | 2019-06-17 08:00:00+00:00 | BETR801 | pm25 | 6.5 |
| 2 | 2019-06-17 07:00:00+00:00 | BETR801 | pm25 | 18.5 |
| 3 | 2019-06-17 06:00:00+00:00 | BETR801 | pm25 | 16.0 |
| 4 | 2019-06-17 05:00:00+00:00 | BETR801 | pm25 | 7.5 |

Concat: Concatena dois dataframes diferentes em um, combinando seus cabeçalhos.



Quais os principais comandos e para que servem ?

- Agora conseguimos realizar algumas manipulações dos dados:

```
In [20]: air_quality = pd.merge(air_quality, stations_coord, how="left", on="location")
```

```
In [21]: air_quality.head()
```

```
Out[21]:
```

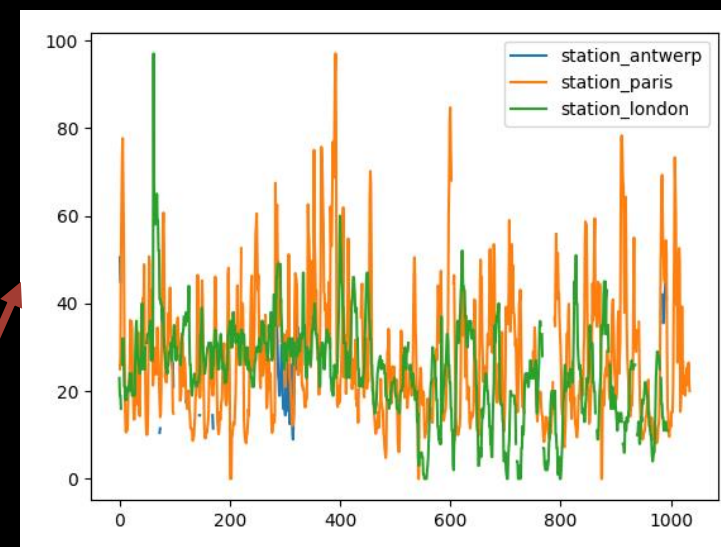
| | date.utc | ... coordinates.longitude |
|---|---------------------------|---------------------------|
| 0 | 2019-05-07 01:00:00+00:00 | ... -0.13193 |
| 1 | 2019-05-07 01:00:00+00:00 | ... 2.39390 |
| 2 | 2019-05-07 01:00:00+00:00 | ... 2.39390 |
| 3 | 2019-05-07 01:00:00+00:00 | ... 4.43182 |
| 4 | 2019-05-07 01:00:00+00:00 | ... 4.43182 |

```
[5 rows x 6 columns]
```

Merge: Combina as linhas de Dataframe diferentes, usando a referencia da coluna chave para concatenar em um único Dataframe.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv")
5
6 air_quality.head()
7
8 print(air_quality)
9
10 air_quality.plot()
11
12 plt.show()
13
```

Plot: Unindo o pandas com outra biblioteca conseguimos gerar gráficos de nossos dataframes e manipulações de dados.





Quais os comandos de filtragem dentro do Pandas ?

```
>>> df = pd.DataFrame({'A': range(1, 6),  
...                    'B': range(10, 0, -2),  
...                    'C': range(10, 5, -1)})  
>>> df  
   A  B  C  
0  1 10 10  
1  2  8  9  
2  3  6  8  
3  4  4  7  
4  5  2  6  
>>> df.query('A > B')  
   A  B  C  
4  5  2  6
```

Query: Realiza um filtro conforme comparação ou valor de índice determinado.

```
In [23]: adult_names = titanic.loc[titanic["Age"] > 35, "Name"]  
  
In [24]: adult_names.head()  
Out[24]:  
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  
6                McCarthy, Mr. Timothy J  
11                Bonnell, Miss. Elizabeth  
13                Andersson, Mr. Anders Johan  
15                Hewlett, Mrs. (Mary D Kingcome)  
Name: Name, dtype: object
```

Loc: Localiza as informações conforme solicitado e realiza um filtro e mostra apenas o que foi selecionado.



Quais os comandos de filtragem dentro do Pandas ?

```
>>> # select columns by name
>>> df.filter(items=['one', 'three'])
```

| | one | three |
|--------|-----|-------|
| mouse | 1 | 3 |
| rabbit | 4 | 6 |

Filter: Realiza um filtro conforme comparação ou valor de índice determinado, nesse caso filtramos pelas colunas.

```
>>> index = pd.DatetimeIndex(["2017-07-05", "2017-07-06", None,
...                           "2017-07-08"])
>>> index
DatetimeIndex(['2017-07-05', '2017-07-06', 'NaT', '2017-07-08'],
              dtype='datetime64[ns]', freq=None)
>>> pd.isna(index)
array([False, False,  True, False])
```

Isna: Realiza um filtro conforme índice e retorna pra nós com dados Booleanos onde que estão os dados vazios de nosso Dataframe.

```
import pandas as pd

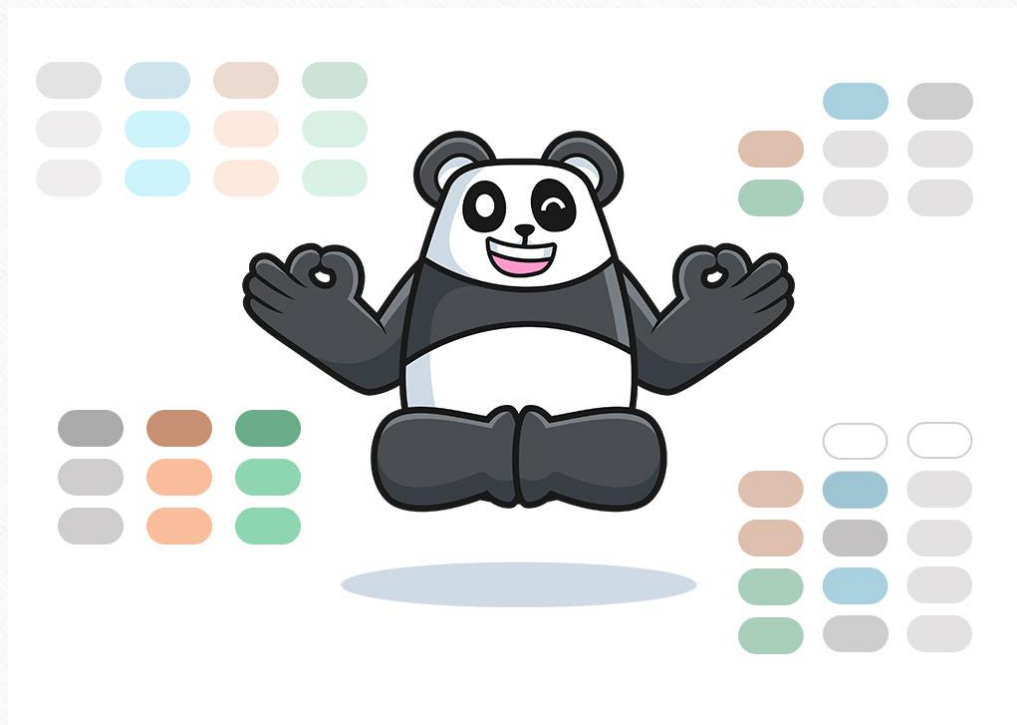
dates=['April-10', 'April-11', 'April-12', 'April-13', 'April-14', 'April-16']
sales=[200,300,400,200,300,300]
prices=[3, 1, 2, 4,3,2]

df = pd.DataFrame({'Date':dates ,
                   'Sales':sales ,
                   'Price': prices})

df_mask=df['Sales']>=300
filtered_df = df[df_mask]
print(filtered_df)
```

Logica: Realiza um filtro conforme a logica de programação acessando diretamente o Dataframe.

Agradecimientos



- Pablo Ricardo de Abreu
- Aula 8
- Biblioteca Pandas
- Github:<https://github.com/Pabloabreu1277?tab=repositories>