

Algoritmos e estrutura de dados I



Pandas - Parte 1

Introdução

Pandas é uma biblioteca open-source, amplamente utilizada no python

Fácil manipulação de estrutura de dados

Alta performance

Bastante utilizado para realizar a análise de dados

Para utilizar a biblioteca pandas é necessário importá-la antes de utilizar

```
>>> import pandas as pd
```

Criando um dataframe

```
>>>d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4],  
index=['a', 'b', 'c', 'd'])}  
>>>dados = pd.DataFrame(d)  
>>>dados
```

	one	two
a	1	1
b	2	2
c	3	3
d	Nan	4

Salvando um dataframe

Podemos salvar o conteúdo de um dataframe em um arquivo, por exemplo, um CSV

```
>>>dados.to_csv('dados.csv', index=False, header=False)
```

Importando um CSV

A maneira mais comum de utilizar o pandas é importando um arquivo CSV contendo dados.

O arquivo CSV também pode ser lido em um programa de planilhas (Excel, Calc, etc)

```
>>> data = pd.read_csv('arquivo.csv')
```

Comandos básicos

Obtendo informações do dataframe

```
>>>dados.info()
```

Obtendo uma amostra dos primeiros valores do dataframe

```
>>>dados.head(3)
```

Número de linhas e colunas

```
>>>dados.shape
```

Nomes das colunas

```
>>>dados.columns
```

Comandos básicos

Número de valores únicos

```
>>>dados["coluna"].nunique()
```

Amostra dos valores únicos

```
>>>dados["coluna"].unique()
```

Contagem dos valores

```
>>>dados["coluna"].value_counts()
```

Comandos básicos

Contagem de valores nulos por colunas

```
>>>dados.isnull().sum()
```

Apagar as linhas que contém valor nulo

```
>>>dados.dropna()
```

Substituir valor nulo por um valor padrão

```
>>>dados.fillna(-9999)
```

Calcular a média de um valor

```
>>>dados.mean()
```


Comandos básicos

Selecionando apenas uma coluna

```
>>>dados["sepal_length"]
```

Selecionando várias colunas e inserindo em outro dataframe

```
>>>X = dados[["a", "b", "c"]]
```

Seleciona a coluna por número

```
>>>dados.iloc[:, [1,3,4]]
```

Apagando uma coluna do dataframe

```
>>>X = dados.drop("b", axis=1)
```

Renomeando colunas

```
>>>dados.rename(columns={"a": "novo_a", "b": "novo_b"})
```

Comandos básicos

Ordenando os valores

```
>>>dados.sort_values(by = "a", ascending = True)
```

Criando uma nova coluna através de outra

```
>>>dados['nova_col'] = dados["a"]*2
```

Criando uma nova coluna a partir de uma condição

```
>>>dados['nova_col'] = ["valor 1" if i < 3 else "valor 2" for i in dados["a"]]
```

Comandos básicos

Selecionando da linha 3 até a 10

```
>>>dados.iloc[3:10,]
```

Seleciona 10 linhas aleatórias

```
>>>dados.sample(10)
```

Localizando as linhas com um valor específico

```
>>>dados[dados["a"].isin(["valor 1"])]
```

Comandos básicos

Filtrando através de uma condição

```
>>>dados[dados.a >= 5]
```

```
>>>dados[(dados.a > 1) & (dados.b=="abc") | (dados.c < 3)]
```

Apagando linhas

```
>>>dados.drop(dados.index[1])
```