



Numerical Development Academy

An open source initiative to support numerical developers



Leitura de arquivo de dados

Uma das funcionalidades mais comuns dentro de uma pesquisa é a leitura de dados coletados para geração de gráficos e para futuras análises. Visando ajudar nesse processo, esse material vai exemplificar como carregar arquivos nos formatos .txt, .csv e .xlsx, que são comumente utilizados.



Pandas é uma das bibliotecas disponíveis em Python mais utilizadas para manipulação e análise de dados. Nela é possível fazer o carregamento de dados de diferentes formatos e posteriormente fazer as manipulações desejadas.

Para instalar essa biblioteca basta executar esse código no terminal:

```
pip install pandas
```

Dados no formato .txt

Pode-se carregar dados no formato .txt da seguinte maneira:

```
import pandas as pd
```

```
# Pasando o caminho do arquivo .txt e o separador utilizado no arquivo:
```

```
dado = pd.read_csv(file_path, sep="\t")
```

```
# Carregando as primeiras 5 linhas do arquivo
```

```
dado.head(5)
```

Dados no formato .txt

Problemas que a falta do separador pode gerar no arquivo carregado:

Sem separador

time\tf(x)\ty(x)\tg(x)				
0	0.1	4	3.6	10.51271096
1	0.2	5	4.4	11.05170918
2	0.3	6	5.4	11.61834243
3	0.4	7	6.6	12.21402758
4	0.5	8	8.0	12.84025417

Informações misturadas, sem identificação.

Com separador

time	f(x)	y(x)	g(x)	
0	0.1	4	3.6	10.512711
1	0.2	5	4.4	11.051709
2	0.3	6	5.4	11.618342
3	0.4	7	6.6	12.214028
4	0.5	8	8.0	12.840254

Informações separadas por colunas.

OBS: O erro na carregamento dos dados não afeta o dado original, apenas gera um erro no objeto pandas gerado.

Dados no formato .csv

Pode-se carregar dados no formato .csv da mesma maneira:

```
import pandas as pd
```

```
# Pasando o caminho do arquivo .csv e o separador utilizado no arquivo:
```

```
dado = pd.read_csv(file_path, sep=";")
```

```
# Carregando as primeiras 5 linhas do arquivo
```

```
dado.head(5)
```

Dados no formato .csv

Problemas que a falta do separador pode gerar no arquivo carregado:

Sem separador

	time;f(x);y(x);g(x)
0	0.1;4;3.6;10.51271096
1	0.2;5;4.4;11.05170918
2	0.3;6;5.4;11.61834243
3	0.4;7;6.6;12.21402758
4	0.5;8;8;12.84025417

Informações misturadas, sem identificação.

Com separador

	time	f(x)	y(x)	g(x)
0	0.1	4	3.6	10.512711
1	0.2	5	4.4	11.051709
2	0.3	6	5.4	11.618342
3	0.4	7	6.6	12.214028
4	0.5	8	8.0	12.840254

Informações separadas por colunas.

OBS: O erro na carregamento dos dados não afeta o dado original, apenas gera um erro no objeto pandas gerado.

Dados no formato .xlsx

Pode-se carregar dados no formato .xlsx da mesma maneira:

```
import pandas as pd
```

```
# Pasando o caminho do arquivo .xlsx e o separador utilizado no arquivo:
```

```
dado = pd.read_csv(file_path)
```

```
# Carregando as primeiras 5 linhas do arquivo
```

```
dado.head(5)
```


Bônus: Meu primeiro gráfico

Para fazer nosso primeiro gráfico vamos utilizar o **matplotlib** que é uma biblioteca muito conhecida.

Para instalar essa biblioteca basta executar esse código no terminal:

```
pip install matplotlib
```

```
import matplotlib.pyplot as plt

# preparando os dados para serem plotados
time=excel['time']
f_x=excel['f(x)']
y_x=excel['y(x)']
g_x=excel['g(x)']
```

Bônus: Meu primeiro gráfico

```
# instanciando uma figura para fazer o gráfico de tamanho 6 por 6
fig, ax = plt.subplots(figsize=(6, 6))
# atribuindo valores para os eixos x e y do gráfico
ax.plot(time, f_x, label="f(t)")
ax.plot(time, y_x, label="y(t)")
ax.plot(time, g_x, label="g(t)")
# atribuindo título aos eixos e ao gráfico
ax.set(xlabel='time (s)',
       ylabel='functions values',
       title='My first python plot')
# exibindo as linhas de guia do gráfico
ax.grid()
# adicionado legendas ao gráfico
ax.legend()
# exibindo o gráfico
plt.show()
```

Bônus: Meu primeiro gráfico

