

Aula 9 – Manipulação e Visualização de Dados com pandas

EMC410235 - Programação Científica para Engenharia e Ciência Térmicas

Prof. Rafael F. L. de Cerqueira

2025.2

Introdução à Análise de Dados com Pandas

- `conda install pandas`
- Apresentar funcionalidades básicas da biblioteca pandas.
- Explorar leitura, manipulação e exportação de dados com DataFrames.
- Discutir boas práticas para análise de dados experimentais em engenharia.
- Preparar o uso do pandas em dois exemplos aplicados:
 - `.csv`, `.txt` (REDU - UNICAMP)
 - `.json` (FlowDa)
- Para ler/escrever planilhas Excel (`.xls` e `.xlsx`): `conda install xlrd`



Importação da Biblioteca

Para utilizar o pandas, primeiro é necessário importá-lo no seu ambiente Python:

Importação

```
import pandas as pd
```

- A convenção `as pd` facilita o uso de funções da biblioteca.
- É comum também importar o NumPy e matplotlib junto, quando há manipulação numérica e visualização de dados:

```
import matplotlib.pyplot as plt  
import numpy as np
```

Criação de DataFrames

Um DataFrame é a estrutura central do pandas, usada para armazenar e manipular dados tabulares.

Exemplo: a partir de um dicionário

```
import pandas as pd

data = {'Temperatura': [300, 310, 320],
        'Pressao': [1.0, 1.1, 1.2]}

df = pd.DataFrame(data)
print(df)
```

- Cada chave do dicionário vira uma coluna.
- O índice é gerado automaticamente (0, 1, 2...).

Inspeção e Estrutura do DataFrame

Depois de criar ou carregar um DataFrame, é importante visualizar sua estrutura.

Comandos úteis

```
df.head()      # Primeiras linhas
df.tail()      # Últimas linhas
df.shape       # (n_linhas, n_colunas)
df.columns     # Nome das colunas
df.dtypes      # Tipo de dado por coluna
df.info()      # Resumo geral
```

- Esses comandos ajudam a entender a dimensão, nomes e tipos das variáveis.
- Úteis para diagnóstico rápido ao carregar arquivos de dados.

A seleção de dados em um DataFrame pode ser feita por colunas, índices ou condições lógicas.

Exemplos práticos

```
df['Temperatura']      # Seleciona a coluna
df.loc[0]              # Seleciona a linha com índice 0
df.iloc[0, 1]          # Seleciona valor na linha 0, coluna 1
df[df['Temperatura'] > 305] # Filtra linhas com Temperatura > 305
```

- Use `loc[]` para acesso por rótulo e `iloc[]` por posição.
- Expressões booleanas permitem filtros poderosos em seus dados.

O pandas oferece diversas funções para análise estatística e manipulação numérica.

Exemplos úteis

```
df.describe()           # Estatísticas descritivas
df.mean()               # Média por coluna
df['Temperatura'].max()  # Máximo da coluna 'Temperatura'
df.sort_values(by='Pressao') # Ordena pelo valor da coluna
```

- `describe()` fornece um resumo completo: média, desvio padrão, quartis.
- É possível aplicar operações matemáticas diretamente sobre colunas.

O pandas facilita a importação de dados de arquivos em vários formatos.

Funções comuns de leitura

```
df_csv = pd.read_csv('dados.csv')           # Arquivo .csv
df_txt = pd.read_table('dados.txt', sep='\t') # Arquivo .txt
df_json = pd.read_json('dados.json')         # Arquivo .json
```

- Use o parâmetro `sep` para definir o separador correto.
- Verifique a estrutura inicial com `df.head()` e `df.info()`.
- Pode-se combinar com `glob` para ler múltiplos arquivos.

Leitura de Vários Arquivos com glob

Quando há muitos arquivos semelhantes em uma pasta, podemos automatizar a leitura com o módulo glob.

Exemplo: leitura de vários arquivos .csv

```
import pandas as pd
import glob

arquivos = glob.glob("dados_experimento/*.csv")
dfs = [pd.read_csv(f) for f in arquivos]
df_total = pd.concat(dfs, ignore_index=True)
```

- glob.glob() lista os arquivos que correspondem ao padrão.
- pd.concat() junta todos os DataFrames em um só
- Muito útil para ensaios repetidos ou em lotes.

O pandas permite salvar os dados processados em diferentes formatos de forma simples.

Exemplos de exportação

```
df.to_csv('saida.csv', index=False)           # arquivo .csv  
df.to_csv('saida_tab.txt', sep='\t', index=False) # arquivo .txt  
df.to_excel('saida.xlsx', index=False)        # Arquivo .xlsx
```

- Use `sep='\t'` para gerar arquivos legíveis por outros programas (ex: CFD, Tecplot).
- O parâmetro `index=False` evita salvar o índice como coluna.
- Ideal para gerar relatórios ou arquivar resultados de simulação.

Exportação para Excel com Múltiplas Abas

É possível salvar múltiplos DataFrames em diferentes abas de um mesmo arquivo '.xlsx'.

Exemplo com ExcelWriter

```
with pd.ExcelWriter('multi_saida.xlsx') as writer:  
    df1.to_excel(writer, sheet_name='Velocidade X')  
    df2.to_excel(writer, sheet_name='Velocidade Y')  
    df3.to_excel(writer, sheet_name='Velocidade Z')
```

- Cada aba do Excel pode conter um conjunto diferente de dados.
- Útil para organizar componentes vetoriais ou múltiplas variáveis de campo.

O pandas oferece várias ferramentas para tratar dados ausentes, duplicados e criar novas colunas.

Exemplos práticos

```
df.drop_duplicates()    # Remove duplicatas
df.dropna()             # Remove linhas com valores ausentes
df.fillna(0)            # Substitui valores ausentes por zero
df['delta_T'] = df['Temperatura'] - 273.15  # Nova coluna (°C)
```

- A limpeza de dados é uma etapa fundamental na análise experimental.
- Podemos criar novas variáveis com operações diretas entre colunas.

Agregações com groupby

A função `groupby` permite agrupar os dados com base em uma ou mais colunas e aplicar funções de agregação.

Exemplo: média por experimento

```
df.groupby('experimento').mean()
```

- Útil para calcular estatísticas por categoria (ex: experimento, condição, sensor).
- Pode ser combinado com outras funções: `sum()`, `min()`, `std()`, etc.
- Também permite agregar múltiplas colunas ao mesmo tempo.

Plotagem Básica com pandas

O pandas possui integrações com o matplotlib para visualizações rápidas.

Exemplos

```
df.plot()                # Gráfico de todas as colunas numéricas
df['Temperatura'].hist()  # Histograma da coluna 'Temperatura'
df.plot(x='Temperatura', y='Pressao', kind='scatter') # Dispersão
```

- Ideal para análise exploratória rápida.
- Permite criar linhas, histogramas, dispersões e outros gráficos simples.
- Para visualizações mais elaboradas, recomenda-se usar matplotlib ou seaborn.

Agora que vimos as principais funcionalidades do pandas, vamos aplicar esses conceitos em dois exemplos práticos:

- 1 **REDU/UNICAMP – Experimental database of horizontal air-water flow through orifice plate:** leitura e análise de arquivos `.txt` e `.csv`.
- 2 **FlowDa:** leitura e exploração de dados no formato `.json`.

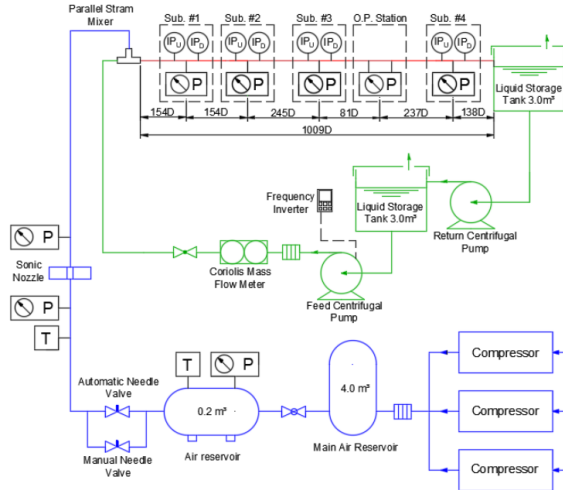
Vamos para a prática!

Dataset: Escoamento Bifásico Ar-Água em Placa de Orifício

- **Origem:** Laboratório Flow&Rs – UNICAMP / CEPETRO
- **Título:** *Experimental Database of Horizontal Air-Water Flow Through Orifice Plate*
- **Objetivo:** Estudar o escoamento bifásico ar-água em placas de orifício horizontais com diferentes razões de contração de área.
- **Dados disponíveis:**
 - Pressões diferenciais medidas com transdutores
 - Sinais de sondas de impedância para fração de vazio
 - Arquivos '.txt' (dados brutos) e '.xls' (dados operacionais)
- **Parâmetros experimentais:**
 - JL: 0.20–0.70 m/s JG: 0–0.80 m/s
 - 2 placas com diâmetros de 7 mm e 9.1 mm
 - 4 estações de medição + estação principal (placa de orifício)

Dataset: Escoamento Bifásico Ar-Água em Placa de Orifício

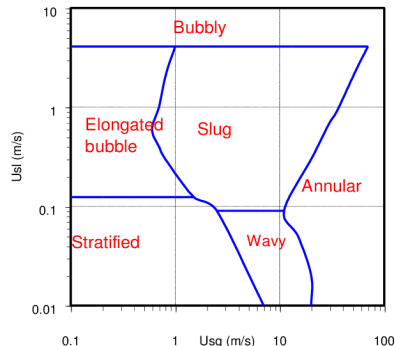
Desenho esquemático da bancada experimental



Dataset: Escoamento Bifásico Ar-Água em Placa de Orifício

Matriz Experimental e Mapa de Padrões de Escoamento

JG \ JL	0.7 m/s	0.6 m/s	0.5 m/s	0.4 m/s	0.3 m/s	0.2 m/s
0.8 m/s	-	-	9.1	9.1	9.1	9.1
0.7 m/s	-	9.1	9.1	9.1	9.1 7.0	9.1 7.0
0.6 m/s	9.1	9.1	9.1	9.1	9.1 7.0	9.1 7.0
0.5 m/s	9.1	9.1	9.1	9.1 7.0	9.1 7.0	9.1 7.0
0.4 m/s	-	-	9.1	9.1 7.0	9.1 7.0	9.1 7.0
0.3 m/s	-	-	-	9.1 7.0	9.1 7.0	9.1 7.0
0.0 m/s	-	7.0	7.0	9.1 7.0	9.1 7.0	9.1 7.0



- **Fonte:** FlowDa, desenvolvido por SINTEF e IFE
- **Histórico:** coleciona dados experimentais de escoamento multifásico em dutos obtidos nos últimos 40 anos
- **Formato:** arquivos `.json`, organizados por projeto/experimento
- **Conteúdo disponível:**
 - Variáveis como velocidade superficial de gás, queda de pressão, fração de líquido, temperatura, regime de escoamento
 - Informações auxiliares sobre o setup experimental: diâmetro do tubo, propriedades dos fluidos, instrumentos usados :
- **Objetivos na aula:**
 - Ler JSONs com `pd.read_json`
 - Plotar gráficos de dispersão e histogramas para análise exploratória

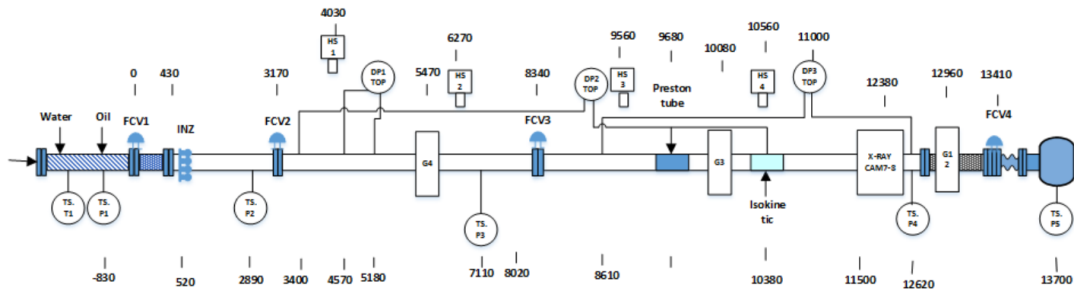


<https://flowda.net>

Experimento FlowDa: MF_C3b_60 (60°)

- **Configuração:** MF_C3b_60 (inclinação de 60°)
- **Fluidos:** mistura de gás SF₆, óleo mineral leve e água
- **Propriedades físicas:**
 - Gás (SF₆): $\mu = 1.5 \times 10^{-5}$ Pa·s, $\rho = 48.4$ kg/m³
 - Óleo: $\mu = 2.9 \times 10^{-3}$ Pa·s, $\rho = 810$ kg/m³
 - Água: $\mu = 1.0 \times 10^{-3}$ Pa·s, $\rho = 998$ kg/m³
- **Condições operacionais:**
 - Vazões: óleo e água até 0.01 m³/s, gás até 0.07 m³/s
 - Pressão: 7.94–8.37 bara
 - Temperatura: 18–27.9 C
 - Diâmetro interno: 98.6 mm (PVC) e 101.5 mm (aço)
- **Instrumentação:** pressão, frações volumétricas, velocidades superficiais, WC, densidades por fase etc.

Experimento FlowDa: MF_C3b_60 (60°) - Tubulação PVC



Steel flow straightner, L = 400 mm



Steel flow mixer

FCV – Fast Closing Valve
 HS – High-Speed video camera
 G – Gamma densitometer
 P – Pressure sensor
 DP – Differential Pressure transducer
 INZ – Liquid feed injector nozzles



Stainless steel pipe, ID = 100 mm



Carbon fiber pipe, ID = 98.7 mm



PVC pipe, ID = 98.6 mm



Stainless steel /glass, ID = 100.0 mm
 Isokinetic/HS4

All dimensions are in mm

Experimento FlowDa: MF_C3b_60 (60°) - Tubulação Aço

