Persistência de Arquivos: CSV, TSV, Planilhas

QXD0099 - Desenvolvimento de Software para Persistência

Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br

Slides baseados no material do Professor Régis Magalhães

Agenda

- CSV - Comma-separated values
- TSV - Tab-separated values
- Planilhas
- Diferenças entre os formatos (CSV, TSV, Planilhas)
- Manipulação com bibliotecas específicas
- Visualizações e Análise
  - Histogramas e boxplots de notas
  - Gráficos de pizza para status (aprovado/reprovado)
  - Dashboard simples com pandas + matplotlib
- Análise de Frequência e Desempenho dos Alunos
  - Leitura e análise de dados
  - Gráficos simples com matplotlib ou pandas
  - Exportando para Excel
  - Desempenho e uso em grandes arquivos

CSV - Comma-separated values

- O formato CSV é bastante simples e suportado por quase todas as planilhas eletrônicas e SGDB disponíveis no mercado.
- Cada linha do arquivo representa um registro, e os valores dentro desse registro são separados por vírgulas (ou outro delimitador, como ponto e vírgula, em algumas regiões).
- O formato é amplamente utilizado devido à sua simplicidade e compatibilidade com diversos sistemas e programas, como planilhas e bancos de dados.

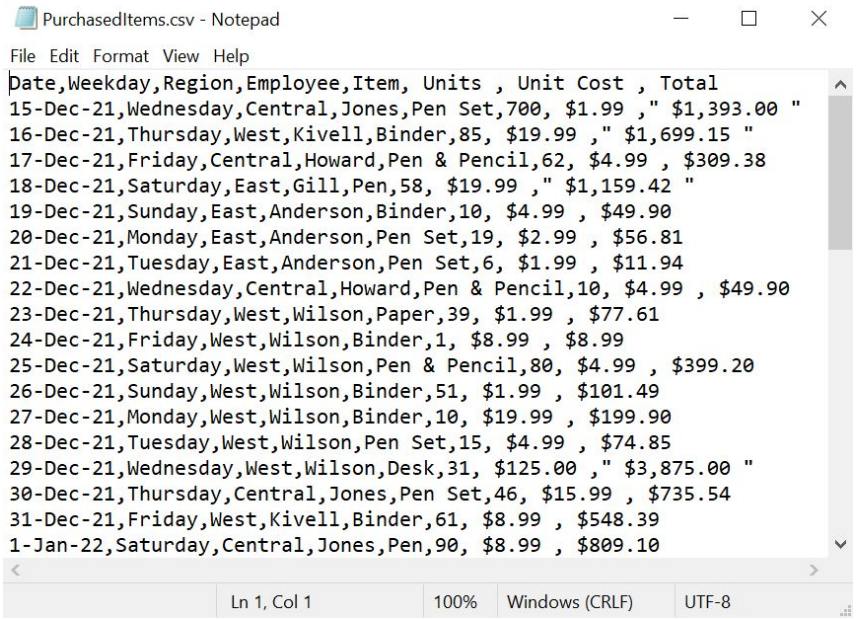
Padrão RFC 4180 para CSV

- RFC 4180 é a especificação mais comum (embora informal) para arquivos CSV. Define como os dados devem ser organizados.

- Principais regras do padrão:
  - Cada linha representa um registro.
  - Os campos são separados por vírgula (,).
  - O primeiro registro (opcional) pode conter cabeçalhos.
  - Campos com vírgulas, aspas ou quebras de linha devem ser envolvidos por aspas duplas (").
  - Aspas duplas dentro de um campo devem ser duplicadas ("valor com ""aspas"" internas").

Exemplo conforme RFC 4180:

- "Nome","Idade","Cidade"
- "João da Silva",30,"Fortaleza"
- "Maria, Lopes",28,"São Paulo"



CSV - Comma-separated values

Year	Make	Model	Description	Price
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

Problemas comuns em arquivos CSV

- **Quebras de linha dentro de campos**
  - Ex: comentário com ENTER no meio
  - Solução: envolver o campo com aspas

```
"Nome","Observações"
"Lucas","Aluno excelente
Participativo em aula"
```

- **Aspas dentro de campos**
  - Devem ser escapadas com aspas duplas

```
"Nome","Comentário"
"João","Disse: ""Gostei muito!"""
```

- **Campos com vírgulas**
  - Devem ser envoltos em aspas

```
"Nome","Cidade"
"Maria","São Paulo, SP"
```

Abrindo arquivos CSV

```
import pandas as pd

df = pd.read_csv('veiculos.csv')
df
```

	Year	Make	Model	Description	Price
0	1997	Ford	E350	ac, abs, moon	3000.0
1	1999	Chevy	Venture "Extended Edition"	NaN	4900.0
2	1999	Chevy	Venture "Extended Edition, Very Large"	NaN	5000.0
3	1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.0

- **import pandas as pd**: Importa a biblioteca pandas e a associa ao alias pd, facilitando seu uso.
- **df = pd.read_csv('veiculos.csv')**: Utiliza a função pd.read_csv() para ler o arquivo veiculos.csv e armazena os dados em um DataFrame chamado df. Esse DataFrame permite manipular, filtrar, agrupar e processar os dados tabulares de forma eficiente.
- **df**: Essa última linha é usada para visualizar o conteúdo do DataFrame df. Em um ambiente de desenvolvimento como Jupyter Notebook, colocar apenas df ao final do código exibirá o conteúdo da tabela carregada.

TSV - Tab-separated values

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa

```
Sepal length Sepal width Petal length Petal width Species
5.1 3.5 1.4 0.2 I. setosa
4.9 3.0 1.4 0.2 I. setosa
4.7 3.2 1.3 0.2 I. setosa
4.6 3.1 1.5 0.2 I. setosa
5.0 3.6 1.4 0.2 I. setosa
```

Planilhas

- Excel
- Google Planilha

- Desenvolvimento de Software Low-Code / No-Code usando Planilhas:
  - AppSheet do Google - https://www.appsheet.com/
  - Bubble - https://bubble.io/
  - Glide - https://www.glideapps.com/

Abrindo arquivos de planilhas

```python
import pandas as pd

# Carrega o arquivo Excel
df = pd.read_excel('nome_do_arquivo.xlsx',
sheet_name='Nome_da_Planilha')

# Exibe os dados carregados
print(df)
```

	Year	Make	Model	Description	Price
3	1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799
4	2024	Toyota	Corolla	Red	100000

- **'nome_do_arquivo.xlsx'**: O nome ou caminho do arquivo Excel que deseja abrir.
- **sheet_name**: O nome ou índice da aba (planilha) que deseja carregar. Pode ser uma string com o nome da aba, um número (começando do 0), ou None (para carregar todas as planilhas como um dicionário de DataFrames).

Carregar todas as planilhas

```
# Carrega todas as abas como um dicionário de DataFrames
dfs = pd.read_excel('nome_do_arquivo.xlsx', sheet_name=None)

# Exibe as chaves do dicionário (nomes das planilhas)
print(dfs.keys())

# Exibe o DataFrame da primeira aba
print(dfs['Nome_da_Planilha'])
```

	Year	Make	Model	Description	Price
3	1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799
4	2024	Toyota	Corolla	Red	100000

- Esse método é especialmente útil para arquivos com múltiplas planilhas que precisam ser manipuladas em conjunto.

Diferenças entre os formatos (CSV, TSV, Planilhas)

Formato	Quando usar
CSV	Dados simples, compatibilidade ampla
TSV	Quando campos podem conter vírgulas
XLSX	Quando precisa de formatação, múltiplas planilhas, ou integração com Excel

- **Problemas comuns:**
  - CSV com vírgulas dentro dos campos (solução: encapsular com aspas).
  - Arquivos com codificação incorreta (UTF-8 preferível).
  - Separadores inconsistentes.

Manipulação com bibliotecas específicas

- csv (módulo nativo do Python)

```python
import csv

with open("dados.csv", mode="w", newline='', encoding="utf-8") as file:
    writer = csv.writer(file)
    writer.writerow(["Nome", "Nota1", "Nota2"])
    writer.writerow(["João", 7.5, 8.0])
```

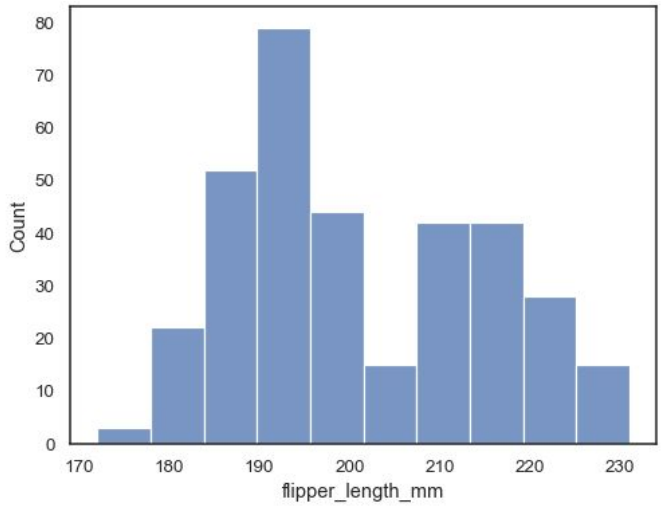
Manipulação com bibliotecas específicas

- pandas (para ler/gravar arquivos como DataFrame)
- openpyxl / xlsxwriter (para manipular planilhas Excel)
- **Vantagens do uso de bibliotecas:**
  - Conversão automática de tipos (número, texto, datas).
  - Leitura e escrita simplificadas.
  - Suporte a arquivos grandes e formatação.

Histogramas

- **O que é:** Representa a distribuição das notas em intervalos (bins).
- **Uso:** Identificar padrões: concentração de notas, presença de extremos, etc.

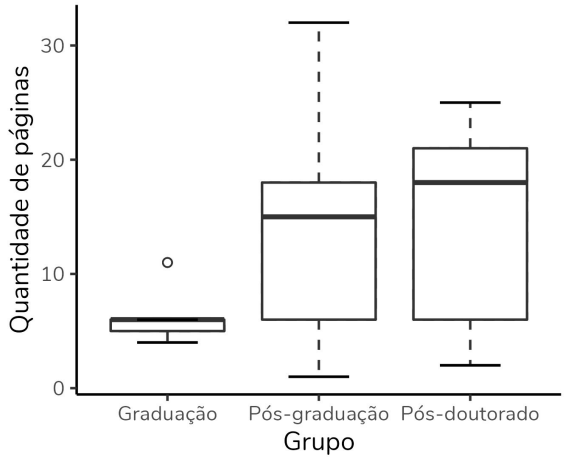
```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("notas.csv")
plt.hist(df["nota"], bins=10)
plt.title("Distribuição das Notas")
plt.xlabel("Nota")
plt.ylabel("Frequência")
plt.grid(True)
plt.show()
```



Boxplot

- **O que é:** Mostra a mediana, quartis, outliers e dispersão das notas.
- **Uso:** Identificar variação e valores discrepantes (outliers).

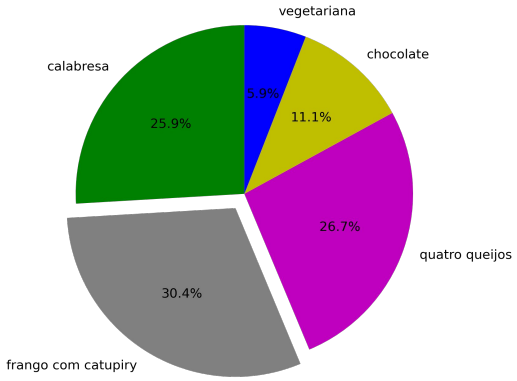
```
plt.boxplot(df["nota"])
plt.title("Boxplot das Notas")
plt.ylabel("Nota")
plt.show()
```



Gráficos de Pizza

- **O que são:** Gráficos de pizza (ou gráficos de setor) são usados para representar proporções de categorias em relação ao todo. Cada fatia do círculo representa uma categoria, com tamanho proporcional à sua frequência ou valor.

```
# Criando gráfico de pizza
plt.pie(contagem, labels=contagem.index, autopct="%1.1f%%",
startangle=90)
plt.title("Distribuição de Aprovados e Reprovados")
plt.axis("equal")  # Garante formato circular
plt.show()
```



Análise de Frequência e Desempenho dos Alunos

- Você recebeu um arquivo chamado frequencia_notas.csv, que contém dados sobre a participação e o desempenho de estudantes ao longo do semestre em uma disciplina da graduação.

- Cada linha do arquivo contém:
  - Aluno: Nome do aluno.
  - Curso: Curso de origem (ex: Computação, Engenharia, Design).
  - Data: Data da aula (formato dd/mm/yyyy).
  - Presenca: Sim ou Não.
  - Nota: Nota atribuída ao aluno naquela aula (pode ser NaN se ele faltou).

Leitura e análise de dados

```python
import pandas as pd

# Carregar o arquivo
df = pd.read_csv("frequencia_notas.csv")

# Filtrar apenas os registros com presença "Sim" e nota não nula
df_presente = df[(df["Presenca"] == "Sim") & (df["Nota"].notna())]

# Calcular média por aluno
medias_alunos = df_presente.groupby("Aluno", as_index=False)["Nota"].mean()
medias_alunos.rename(columns={"Nota": "Média"}, inplace=True)

# Calcular estatísticas
media_geral = medias_alunos["Média"].mean()
aprovados = (medias_alunos["Média"] >= 7).sum()
reprovados = (medias_alunos["Média"] < 5).sum()
```

- **Filtrar apenas os registros com presença “Sim”** (pois só esses têm nota válida).
- **Agrupar por aluno para calcular a média final de cada um.**
- **Contar os alunos aprovados e reprovados com base na média final**:
  - Aprovado: média ≥ 7
  - Reprovado: média < 5

Gráficos Simples com matplotlib ou pandas

```
# --- Histograma de médias ---
plt.figure(figsize=(8, 4))
medias_alunos["Média"].plot.hist(bins=5, edgecolor='black', color='skyblue')
plt.title("Distribuição das Médias dos Alunos")
plt.xlabel("Média")
plt.ylabel("Quantidade de Alunos")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Gráficos Simples com matplotlib ou pandas

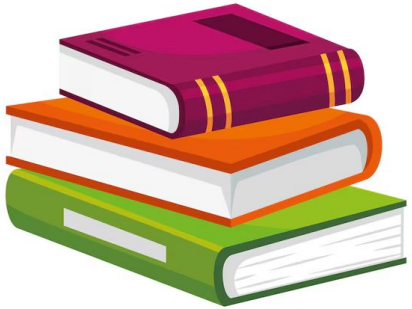
```
# --- Gráfico de Pizza do Desempenho ---
categorias = ["Aprovados (≥ 7)", "Reprovados (< 7)"]
valores = [
    (medias_alunos["Média"] >= 7).sum(),
    (medias_alunos["Média"] < 7).sum()
]

plt.figure(figsize=(6, 6))
plt.pie(valores, labels=categorias, autopct="%1.1f%%", startangle=90, colors=["#4CAF50",
"#F44336"])
plt.title("Desempenho Geral dos Alunos")
plt.axis("equal")  # Deixa a pizza redonda
plt.tight_layout()
plt.show()
```

Bibliografia Básica

- SADALAGE, P. J. E FOWLER, M. NoSQL Essencial. Editora Novatec, São Paulo, 2013.
- REDMOND, E.; WILSON, J. R. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. 1ª edição, 2012. The Pragmatic Programmers.
- ULLMAN, J.D.; WIDOW, J. First Course in Database Systems. 3a edição, 2007. Prentice Hall.
- HAMBRICK, G. et al. Persistence in the Enterprise: A Guide to Persistence Technologies; 1ª edição, 2008. IBM Press.
- ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 4ª edicao, 2009. Pearson/Addison-Wesley.

Bibliografia Complementar

- WHITE, Tom. Hadoop: the definitive guide. California: O´Reilly, 2009. xix, 501 p. ISBN 9780596521974 (broch.).
- AMBLER, S.W., SADALAGE, P.J. Refactoring Databases: Evolutionary Database Design. 1a edição, 2011. Addison Wesley.
- SILBERSCHATZ, A.; SUDARSHAN, S. Sistema de banco de dados. 2006. Campus.
- LYNN, B. Use a cabeça! SQL. 1ª edição, 2008. ALTA BOOKS.
- SMITH, Ben. JSON básico: conheça o formato de dados preferido da web. São Paulo: Novatec, 2015. 400 p. ISBN 9788575224366 (broch.).
- HITZLER, P., KRÖTZSCH, M., and RUDOLPH, S. (2009). Foundations of Semantic Web Technologies. Chapman & Hall/CRC.
- ANTONIOU, G. and HARMELEN, F. (2008). A Semantic Web Primer. Second Edition,Cambridge, MIT Press, Massachusetts.
- HEATH, T. and BIZER, C. (2011). Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, 1st edition.



Obrigado!

Dúvidas?

Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br