

Manual do Usuário: Integração e Processamento de Dados com Python

Introdução

Este manual orienta sobre como integrar e processar dados de arquivos CSV usando Python. As etapas a seguir descrevem como carregar arquivos, verificar suas informações, integrar dados de múltiplas fontes e exportar os resultados.

1. Importação de Bibliotecas Necessárias

```
import pandas as pd
import os
import chardet
from google.colab import files
```

- ☐ *Pandas: Manipulação de dados.*
- ☐ *os: Operações no sistema de arquivos.*
- ☐ *chardet: Detecção de encoding.*
- ☐ *files: Carregamento e download de arquivos no Google Colab.*

2. Função Para Leitura de Arquivos em CSV

```
def read_csv(file_path):
    try:
        with open(file_path, 'rb') as f:
            result = chardet.detect(f.read())
            encoding = result['encoding']
            return pd.read_csv(file_path, sep=',', decimal='.', encoding=encoding)
    except FileNotFoundError:
        print(f"Arquivo não encontrado: {file_path}")
        return None
```

read_csv: Lê arquivos CSV com detecção automática de encoding e tratamento de erros.

3. Carregar Arquivos do Sistema Local para o Ambiente do Colab

```
uploaded = files.upload()
```

files.upload(): Permite ao usuário carregar arquivos locais para o ambiente do Colab.

4. Verificar os Nomes dos Arquivos Carregados

```
file_rotas = [key for key in uploaded.keys() if 'Rotas.csv' in key][0]
file_clientes = [key for key in uploaded.keys() if 'Clientes.csv' in key][0]
file_fabricas = [key for key in uploaded.keys() if 'Fabricas.csv' in key][0]
```

Identifica os arquivos carregados com base em seus nomes.

5. Abrir os Arquivos Necessários

```
df_rotas = read_csv(file_rotas)
df_clientes = read_csv(file_clientes)
df_fabricas = read_csv(file_fabricas)
```

Utiliza a função read_csv para abrir os arquivos.

6. Verificar se os DataFrames Foram Carregados Corretamente

```
if df_rotas is not None and df_clientes is not None and df_fabricas is not None:
```

Confirma que todos os arquivos foram carregados com sucesso.

7. Exibir Informações sobre o Arquivo de Rotas

```
print("Informações sobre o arquivo de rotas:")
print(df_rotas.info())
```

Mostra informações detalhadas sobre o DataFrame de rotas.

8. Exibir Informações sobre o Arquivo de Clientes

```
print("\nInformações sobre o arquivo de clientes:")
print(df_clientes.info())
```

Mostra informações detalhadas sobre o DataFrame de clientes.

9. Exibir Informações sobre o Arquivo de Fábricas

```
print("\nInformações sobre o arquivo de fábricas:")
print(df_fabricas.info())
```

Mostra informações detalhadas sobre o DataFrame de fábricas.

10. Integrar os DataFrames de Rotas, Clientes e Fábricas

```
rotas_final = df_rotas.merge(df_clientes, left_on='CO.Cliente', right_on='CO.Cliente', how='inner') \
               .merge(df_fabricas, left_on='CO.Fabrica', right_on='CO.Fabrica', how='inner')
```

Realiza a integração dos DataFrames com base em colunas comuns.

11. Exibir Informações sobre o DataFrame Final

```
print("\nInformações sobre o DataFrame final:")
print(rotas_final.info())
```

Mostra informações detalhadas sobre o DataFrame final integrado.

12. Visualizar as Primeiras Linhas do DataFrame Final

```
print("\nCinco primeiras linhas do DataFrame final:")
print(rotas_final.head(5))
```

Exibe as cinco primeiras linhas do DataFrame final para verificação.

13. Criar Nova Coluna com Condição

```
rotas_final['Capacidade'] = None
rotas_final.loc[rotas_final['Veiculo'] == 'P24', 'Capacidade'] = 3600
rotas_final.loc[rotas_final['Veiculo'] == 'P12', 'Capacidade'] = 1800
```

Adiciona uma nova coluna "Capacidade" ao DataFrame final com valores baseados em condições.

14. Criar Nova Coluna com Cálculo

```
rotas_final['Produtividade'] = rotas_final['Qtd.Transp'] / rotas_final['Capacidade'] * 100
```

Adiciona uma coluna "Produtividade" calculada a partir de outras colunas.

15. Exportar DataFrames para Arquivos de Texto

```
rotas_final.to_csv('Rotas_final.txt', sep='\t', index=False)
df_clientes.to_csv('Clientes.txt', sep='\t', index=False)
df_fabricas.to_csv('Fabricas.txt', sep='\t', index=False)
```

Exporta os DataFrames para arquivos de texto com separador de tabulação.

16. Mensagem de Sucesso na Criação dos Arquivos TXT

```
print("\nArquivos TXT criados com sucesso!")
```

Informa ao usuário que os arquivos foram criados com sucesso.

17. Baixar os Arquivos para o Sistema Local

```
files.download('Rotas_final.txt')
files.download('Clientes.txt')
files.download('Fabricas.txt')
```

Permite ao usuário baixar os arquivos gerados para o sistema local

18. Mensagem de Erro ao Carregar Arquivos

```
else:
    print("\nErro ao carregar arquivos. Verifique os caminhos e tente novamente.")
```

Informa ao usuário se houve algum erro ao carregar os arquivos.