

```

import pandas as pd

# Caminho do arquivo de entrada
file_path =
'/content/drive/MyDrive/Python/TrensRJ/Dados_Supervia.xlsx'

# Carregar a planilha
df = pd.read_excel(file_path, sheet_name='Supervia2019_2024')

# Converter cabeçalhos para datetime quando possível
colunas = []
for col in df.columns:
    try:
        colunas.append(pd.to_datetime(col, format='%d/%m/%Y'))
    except (ValueError, TypeError):
        colunas.append(col)

df.columns = colunas

# Definir intervalo de datas
start_date = pd.to_datetime('2019-01-01')
end_date = pd.to_datetime('2024-03-31')

# Selecionar colunas de data dentro do intervalo
colunas_data = [col for col in df.columns if isinstance(col,
pd.Timestamp) and start_date <= col <= end_date]

# Criar DataFrame para médias trimestrais
media_trimestral = pd.DataFrame()

# Calcular média trimestral por estação
for station in df['Estação'].unique():
    dados_estacao = df[df['Estação'] == station][colunas_data]
    dados_estacao = dados_estacao.apply(pd.to_numeric,
errors='coerce')

    # Transpor, agrupar por trimestre, e calcular média
    dados_transp = dados_estacao.T
    dados_transp.index = pd.to_datetime(dados_transp.index)
    media_trimestre = dados_transp.resample('Q').mean().T
    media_trimestre['Estação'] = station

    # Adicionar ao DataFrame principal
    media_trimestral = pd.concat([media_trimestral,
media_trimestre.set_index('Estação')])

# Ordenar colunas por data
media_trimestral = media_trimestral.sort_index(axis=1)

```

```
# Exibir tabela final
print(media_trimestral)

# Salvar como arquivo Parquet
media_trimestral.to_parquet('/content/drive/MyDrive/Python/TrensRJ/
media_trimestral_supervia.parquet')
print("Arquivo 'media_trimestral_supervia.parquet' salvo com
sucesso.")
```