

"""

Este script classifica preços em 'Baixo', 'Médio' e 'Alto' utilizando quartis e visualiza os resultados em gráficos estilizados para Dark Mode.

Explicação sobre Quartis e Quantis:

Quartis e quantis são medidas estatísticas que ajudam a dividir um conjunto de dados em partes iguais, mas têm significados ligeiramente diferentes.

Quartis:

- Dividem um conjunto de dados em quatro partes iguais.
- Existem três quartis principais:
  - Primeiro Quartil (Q1): Separa os 25% menores valores dos 75% maiores.
  - Segundo Quartil (Q2): É a mediana, dividindo o conjunto de dados ao meio.
  - Terceiro Quartil (Q3): Separa os 75% menores valores dos 25% maiores.

Quantis:

- São valores que dividem um conjunto de dados em partes iguais de forma mais geral.
- Incluem percentis (100 partes), decis (10 partes), e quartis (4 partes).

Diferença:

- Quartis são um tipo específico de quantil que divide os dados em quatro partes iguais.
- Quantis são mais gerais e podem dividir os dados em qualquer número de partes iguais.

Neste script, usamos quartis para classificar os preços:

- Baixo: Preço abaixo do Primeiro Quartil ( $P < Q1$ )
- Médio: Preço entre o Primeiro Quartil e o Terceiro Quartil ( $Q1 \leq P \leq Q3$ )
- Alto: Preço acima do Terceiro Quartil ( $P > Q3$ )

Passos do script:

1. Carregar os dados e verificar valores ausentes.
2. Calcular quartis e classificar preços.
3. Visualizar resultados com histogramas e boxplots, incluindo melhorias como normalização e análise estatística.
4. Salvar o DataFrame modificado em um novo arquivo CSV.

"""

# Importação das bibliotecas necessárias

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

# Carregando o DataFrame

```
df = pd.read_csv('dados_limpos.csv') # Substitua 'dados_limpos.csv' pelo nome correto do arquivo
```

# Verificação de valores ausentes na coluna 'price'

```
if df['price'].isnull().sum() > 0:
    print(f"Atenção: Existem {df['price'].isnull().sum()} valores ausentes na coluna 'price'.")
    df['price'] = df['price'].fillna(df['price'].median()) # Preenchendo com a mediana
```

# Normalizando a coluna 'price'

```
df['price_normalized'] = (df['price'] - df['price'].min()) / (df['price'].max() - df['price'].min())
```

# Calculando os quartis da coluna 'price'

```
Q1 = df['price'].quantile(0.25)
```

```
Q3 = df['price'].quantile(0.75)
```

```

# Criando a coluna 'Categoria_Preco' e classificando os preços
df['Categoria_Preco'] = 'Médio' # Inicializando com 'Médio'
df.loc[df['price'] < Q1, 'Categoria_Preco'] = 'Baixo' # Preços abaixo do Primeiro Quartil
df.loc[df['price'] > Q3, 'Categoria_Preco'] = 'Alto' # Preços acima do Terceiro Quartil

# Exibindo as primeiras linhas do DataFrame
print(df.head())

# Análise estatística básica da coluna 'price'
mean_price = df['price'].mean()
median_price = df['price'].median()
std_price = df['price'].std()
print(f"Média do Preço: {mean_price:.2f}")
print(f"Mediana do Preço: {median_price:.2f}")
print(f"Desvio Padrão do Preço: {std_price:.2f}")

# Configuração do estilo para Dark Mode
plt.style.use('dark_background')

# Visualização: Histograma de Preços com linha de densidade
plt.figure(figsize=(10, 6))
sns.histplot(df['price'], kde=True, color='cyan', bins=30, edgecolor='white')
plt.title('Histograma de Preços com Densidade')
plt.xlabel('Preço')
plt.ylabel('Frequência')
plt.axvline(Q1, color='red', linestyle='dashed', linewidth=2, label='Q1')
plt.axvline(Q3, color='green', linestyle='dashed', linewidth=2, label='Q3')
plt.legend()
plt.show()

# Visualização: Boxplot de Preço por Categoria
plt.figure(figsize=(10, 6))
sns.boxplot(x='Categoria_Preco', y='price', data=df, palette='coolwarm')
plt.title('Boxplot de Preço por Categoria')
plt.xlabel('Categoria de Preço')
plt.ylabel('Preço')
plt.show()

# Visualização: Gráfico de violino para mostrar a distribuição dos preços em cada categoria
plt.figure(figsize=(10, 6))
sns.violinplot(x='Categoria_Preco', y='price', data=df, palette='coolwarm')
plt.title('Distribuição de Preços por Categoria')
plt.xlabel('Categoria de Preço')
plt.ylabel('Preço')
plt.show()

# Visualização interativa com Plotly
fig = px.histogram(df, x='price', color='Categoria_Preco', nbins=30,
                  title='Histograma Interativo de Preços por Categoria')
fig.show()

# Salvando o DataFrame modificado em um novo arquivo CSV
df.to_csv('dados_classificados.csv', index=False)
print("DataFrame classificado salvo em 'dados_classificados.csv'.")

```