*import* numpy *as* np

*import* pandas *as* pd

*import* matplotlib.pyplot *as* plt

*import* seaborn *as* sns

*# 1.*

*# Arquivo CSV com dados de voos*

df *=* pd.read\_csv("dados1.csv", parse\_dates*=*["Data"])

*# 2.*

*# A: Carregar e explorar os dados*

*print*(df.head())

*print*(df.info())

*# Verificar valores nulos*

*print*(df.isnull().sum())

*# B: Estatísticas Descritivas*

*# Média, mediana, desvio-padrão e variância de:*

*# Passageiros*

*# Distância (km)*

*# Ocupação (%)*

*# Receita (R$)*

stats *=* df[['Passageiros', 'Distância (km)', 'Ocupação (%)', 'Receita (R$)']].describe()

*print*(stats)

*# Calcular o percentil 25%, 50%, 75% da receita.*

percentiles *=* np.percentile(df['Receita (R$)'], [25, 50, 75])

*print*(f'Percentis da Receita (R$): {percentiles}')

*# Encontrar a companhia com maior receita total e com maior número de passageiros.*

max\_revenue\_company *=* df.groupby('Companhia')['Receita (R$)'].sum().idxmax()

max\_passengers\_company *=* df.groupby('Companhia')['Passageiros'].sum().idxmax()

*print*(f'Companhia com maior receita total: {max\_revenue\_company}')

*# Contagem de voos por companhia.*

flight\_counts *=* df['Companhia'].value\_counts()

*print*(flight\_counts)

*# Receita média por companhia e por aeroporto de origem.*

mean\_revenue\_by\_company *=* df.groupby('Companhia')['Receita (R$)'].mean()

mean\_revenue\_by\_origin *=* df.groupby('Aeroporto Destino')['Receita (R$)'].mean()

*# C. Visualizações com Seaborn*

*# Histograma da distribuição de passageiros.*

plt.figure(figsize*=*(10, 6))

sns.histplot(df['Passageiros'], bins*=*30, kde*=*True)

plt.title('Distribuição de Passageiros')

plt.xlabel('Número de Passageiros')

plt.ylabel('Frequência')

plt.show(block*=*False)

*# Boxplot da ocupação (%) separada por companhia aérea.*

plt.figure(figsize*=*(12, 6))

sns.boxplot(x*=*'Companhia', y*=*'Ocupação (%)', data*=*df)

plt.title('Ocupação (%) por Companhia Aérea')

plt.xlabel('Companhia Aérea')

plt.ylabel('Ocupação (%)')

plt.xticks(rotation*=*45)

plt.show(block*=*False)

*# Gráfico de barras da receita média por companhia.*

plt.figure(figsize*=*(12, 6))

sns.barplot(x*=*mean\_revenue\_by\_company.index, y*=*mean\_revenue\_by\_company.values)

plt.title('Receita Média por Companhia Aérea')

plt.xlabel('Companhia Aérea')

plt.ylabel('Receita Média (R$)')

plt.xticks(rotation*=*45)

plt.show(block*=*False)

*# Scatterplot de distância x receita para verificar relação.*

plt.figure(figsize*=*(10, 6))

sns.scatterplot(x*=*'Distância (km)', y*=*'Receita (R$)', data*=*df)

plt.title('Distância vs Receita')

plt.xlabel('Distância (km)')

plt.ylabel('Receita (R$)')

plt.show(block*=*False)

*# (Desafio) Heatmap de correlação entre variáveis numéricas (Passageiros, Distância (km), Ocupação (%), Receita (R$)).*

plt.figure(figsize*=*(8, 6))

correlation\_matrix *=* df[['Passageiros', 'Distância (km)', 'Ocupação (%)', 'Receita (R$)']].corr()

sns.heatmap(correlation\_matrix, annot*=*True, cmap*=*'coolwarm', fmt*=*".2f")

plt.title('Matriz de Correlação')

plt.show(block*=*False)

*# D. Perguntas Analíticas*

*# Qual companhia tem maior participação em número de voos?*

most\_flights\_company *=* flight\_counts.idxmax()

*print*(f'Companhia com maior número de voos: {most\_flights\_company}')

*# A distância influencia a receita?*

correlation\_distance\_revenue *=* df['Distância (km)'].corr(df['Receita (R$)'])

*print*(f'Correlação entre Distância e Receita: {correlation\_distance\_revenue}')

*# Os voos com maior ocupação são necessariamente os de maior receita?*

correlation\_occupancy\_revenue *=* df['Ocupação (%)'].corr(df['Receita (R$)'])

*print*(f'Correlação entre Ocupação e Receita: {correlation\_occupancy\_revenue}')

*# Quais aeroportos de origem concentram mais voos?*

most\_flights\_origin *=* df['Aeroporto Destino'].value\_counts().idxmax()

*print*(f'Aeroporto de origem com mais voos: {most\_flights\_origin}')

*# 5️.*

*# Usar groupby para analisar:*

*# Receita média por mês.*

df['Data do Voo'] *=* pd.to\_datetime(df['Data'])

df['Mês'] *=* df['Data'].dt.month

mean\_revenue\_by\_month *=* df.groupby('Mês')['Receita (R$)'].mean()

*print*(mean\_revenue\_by\_month)

*# Ocupação média por companhia.*

mean\_occupancy\_by\_company *=* df.groupby('Companhia')['Ocupação (%)'].mean()

*print*(mean\_occupancy\_by\_company)

*# Análises extras:*

*# 1. Qual a Rota mais eficiente por companhia (baseado em Ocupação média ou Receita por passageiro)?*

*# Mostrar as 5 melhores em ordem decrescente*

*# Criar coluna de Rota*

df['Rota'] *=* df['Aeroporto Origem'] *+* ' - ' *+* df['Aeroporto Destino']

*# Calcular Receita por passageiro*

df['ReceitaPorPassageiro'] *=* df['Receita (R$)'] */* df['Passageiros']

*# Agrupar por Companhia e Rota e calcular a média de Receita por Passageiro*

eficiencia *=* df.groupby(['Companhia', 'Rota'])['ReceitaPorPassageiro'].mean().reset\_index()

*# Obter top 5 rotas mais eficientes por companhia*

top5\_por\_companhia *=* (

eficiencia.sort\_values('ReceitaPorPassageiro', ascending*=*False)

.groupby('Companhia')

.head(5)

)

*# Exibir resultado final*

*print*(top5\_por\_companhia.sort\_values(['Companhia', 'ReceitaPorPassageiro'], ascending*=*[True, False]))

*# 2. Calcular e mostrar graficamente a Evolução mensal do total de passageiros por companhia*

*import* matplotlib.pyplot *as* plt

*import* seaborn *as* sns

*# Converter coluna Data*

df['Data'] *=* pd.to\_datetime(df['Data'])

*# Criar coluna de Mês (Ano-Mês)*

df['AnoMes'] *=* df['Data'].dt.to\_period('M').astype(str)

*# Agrupar total de passageiros por mês e companhia*

evolucao *=* df.groupby(['AnoMes', 'Companhia'])['Passageiros'].sum().reset\_index()

*# Plotar gráfico*

plt.figure(figsize*=*(14, 7))

sns.lineplot(data*=*evolucao, x*=*'AnoMes', y*=*'Passageiros', hue*=*'Companhia', marker*=*'o')

plt.title('Evolução Mensal do Total de Passageiros por Companhia')

plt.xlabel('Ano-Mês')

plt.ylabel('Total de Passageiros')

plt.xticks(rotation*=*45)

plt.grid(True)

plt.tight\_layout()

plt.show(block*=*False)

*input*("Pressione Enter para fechar tudo...")

plt.close('all')

