

```
# === IMPORTS ===
import pandas as pd          # manipulação de dados em tabelas
(DataFrame)
import numpy as np           # operações numéricas
import seaborn as sns        # visualizações estatísticas (construído
sobre matplotlib)
import matplotlib.pyplot as plt # controle fino dos gráficos

# === PARTE A - CARREGAMENTO E INSPEÇÃO ===
# Lê o CSV e converte a coluna "Data" para datetime automaticamente
df = pd.read_csv("dados1.csv", parse_dates=["Data"])

# df.info() imprime no console informação sobre o DataFrame (colunas,
tipos, non-null)
print("Informações do DataFrame (df.info()):")
df.info()                    # retorna None, mas imprime o resumo no
stdout

Informações do DataFrame (df.info()):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Data                   120 non-null   datetime64[ns]
1   Companhia              120 non-null   object
2   Aeroporto Origem       120 non-null   object
3   Aeroporto Destino      120 non-null   object
4   Passageiros            120 non-null   int64
5   Distância (km)         120 non-null   float64
6   Ocupação (%)           120 non-null   float64
7   Receita (R$)           120 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(3)
memory usage: 7.6+ KB

# Mostra dimensão (linhas, colunas)
print("\nDimensão:", df.shape)

Dimensão: (120, 8)
```

```
# Tipos de dados por coluna
print("\nTipos de dados:\n", df.dtypes)
```

```
Tipos de dados:
  Data          datetime64[ns]
Companhia          object
Aeroporto Origem  object
Aeroporto Destino object
Passageiros       int64
Distância (km)    float64
Ocupação (%)      float64
Receita (R$)      float64
dtype: object
```

```
# Primeiras 5 linhas para inspeção rápida
print("\nPrimeiras linhas:\n", df.head())
```

```
Primeiras linhas:
      Data Companhia Aeroporto Origem Aeroporto Destino Passageiros
Distância (km) Ocupação (%)  Receita (R$)
0 2023-01-01      Azul          GIG          GIG          273
1443.2          87.8      149848.36
1 2023-01-02       Gol          BSB          BSB          77
2444.2          57.0      323685.94
2 2023-01-03      Azul          CGH          BSB          115
1281.0          68.6      145811.26
3 2023-01-04      Azul          CGH          BSB          66
290.2          99.4      227845.78
4 2023-01-05     Latam          CGH          GRU          268
1381.0          95.1      353254.97
```

```
# Verifica valores nulos por coluna (útil para limpeza de dados)
print("\nValores nulos por coluna:\n", df.isnull().sum())
```

```
Valores nulos por coluna:
  Data      0
Companhia  0
Aeroporto Origem  0
Aeroporto Destino  0
Passageiros      0
Distância (km)    0
Ocupação (%)      0
Receita (R$)      0
dtype: int64
```

```
# === PARTE B - ESTATÍSTICAS DESCRITIVAS ===
# Para cada coluna numérica relevante, calculamos média, mediana, desvio
padrão e variância
for col in ["Passageiros", "Distância (km)", "Ocupação (%)", "Receita
(R$)"]:
    print(f"\n--- Estatísticas para '{col}' ---")
    print("Média:", df[col].mean())          # média aritmética
    print("Mediana:", df[col].median())      # percentil 50
    print("Desvio padrão:", df[col].std())    # dispersão (raiz da
variância)
    print("Variância:", df[col].var())        # variância

--- Estatísticas para 'Passageiros' ---
Média: 170.56666666666666
Mediana: 172.5
Desvio padrão: 75.90537379071473
Variância: 5761.625770308124

--- Estatísticas para 'Distância (km)' ---
Média: 1318.6625
Mediana: 1334.0500000000002
Desvio padrão: 613.1847268598605
Variância: 375995.50925420166

--- Estatísticas para 'Ocupação (%)' ---
Média: 76.66749999999999
Mediana: 77.75
Desvio padrão: 14.886247460168669
Variância: 221.60036344537815

--- Estatísticas para 'Receita (R$)' ---
Média: 236158.11333333334
Mediana: 244611.075
Desvio padrão: 97011.39964733057
Variância: 9411211661.53409
```

```

# Percentis (quantis) para a receita: 25%, 50% (mediana), 75%
print("\nPercentis da Receita (25%, 50%, 75%):")
print(df["Receita (R$)"].quantile([0.25, 0.5, 0.75]))

Percentis da Receita (25%, 50%, 75%):
0.25    155383.9225
0.50    244611.0750
0.75    319910.9725
Name: Receita (R$), dtype: float64

# Companhia com maior receita total:
# groupby soma a 'Receita (R$)' por companhia; idxmax retorna o rótulo com
o maior valor
comp_maior_receita = df.groupby("Companhia")["Receita
(R$)"].sum().idxmax()
print("\nCompanhia com maior receita total:", comp_maior_receita)

Companhia com maior receita total: Avianca

# Companhia com maior número total de passageiros
comp_mais_passageiros =
df.groupby("Companhia")["Passageiros"].sum().idxmax()
print("Companhia com mais passageiros:", comp_mais_passageiros)

Companhia com mais passageiros: Avianca

# Contagem de voos por companhia (quantos registros/linhas por companhia)
print("\nVoos por companhia:")
print(df["Companhia"].value_counts())

Voos por companhia:
Companhia
Azul      32
Avianca   30
Gol       29
Latam     29
Name: count, dtype: int64

```

```

# Receita média por companhia
print("\nReceita média por companhia:")
print(df.groupby("Companhia")["Receita (R$)"].mean())

Receita média por companhia:
Companhia
Avianca      242510.513667
Azul         217029.835313
Gol          248945.684138
Latam        237906.159310
Name: Receita (R$), dtype: float64

# Receita média por aeroporto de origem
print("\nReceita média por aeroporto de origem:")
print(df.groupby("Aeroporto Origem")["Receita (R$)"].mean())

Receita média por aeroporto de origem:
Aeroporto Origem
BSB      248305.172222
CGH      228657.067812
GIG      222235.633214
GRU      249153.187000
SDU      243036.039091
Name: Receita (R$), dtype: float64

# === PARTE C - VISUALIZAÇÕES ===
# Histórico: distribuição de passageiros (histograma + KDE)
plt.figure()
sns.histplot(df["Passageiros"], kde=True) # kde=True desenha uma
estimativa de densidade
plt.title("Distribuição de Passageiros")
plt.xlabel("Passageiros")
plt.ylabel("Frequência")
plt.show(block=False) # block=False mantém o script rodando (útil em
notebooks/terminais interativos)

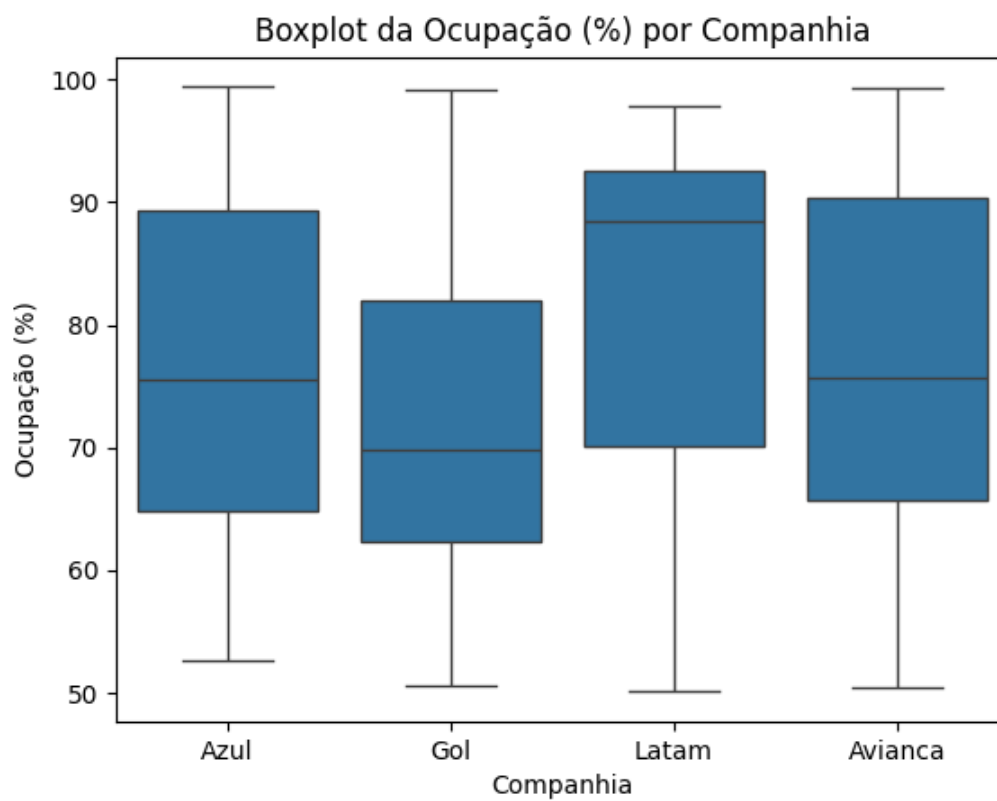
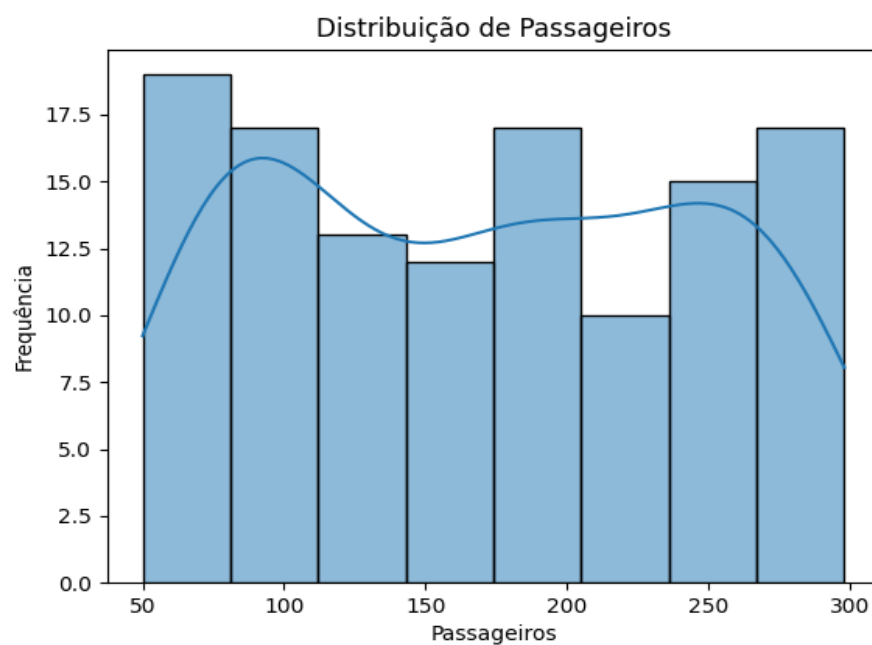
# Boxplot da ocupação por companhia (mostra mediana, quartis e possíveis
outliers)
plt.figure()
sns.boxplot(x="Companhia", y="Ocupação (%)", data=df)
plt.title("Boxplot da Ocupação (%) por Companhia")
plt.xlabel("Companhia")
plt.ylabel("Ocupação (%)")
plt.show(block=False)

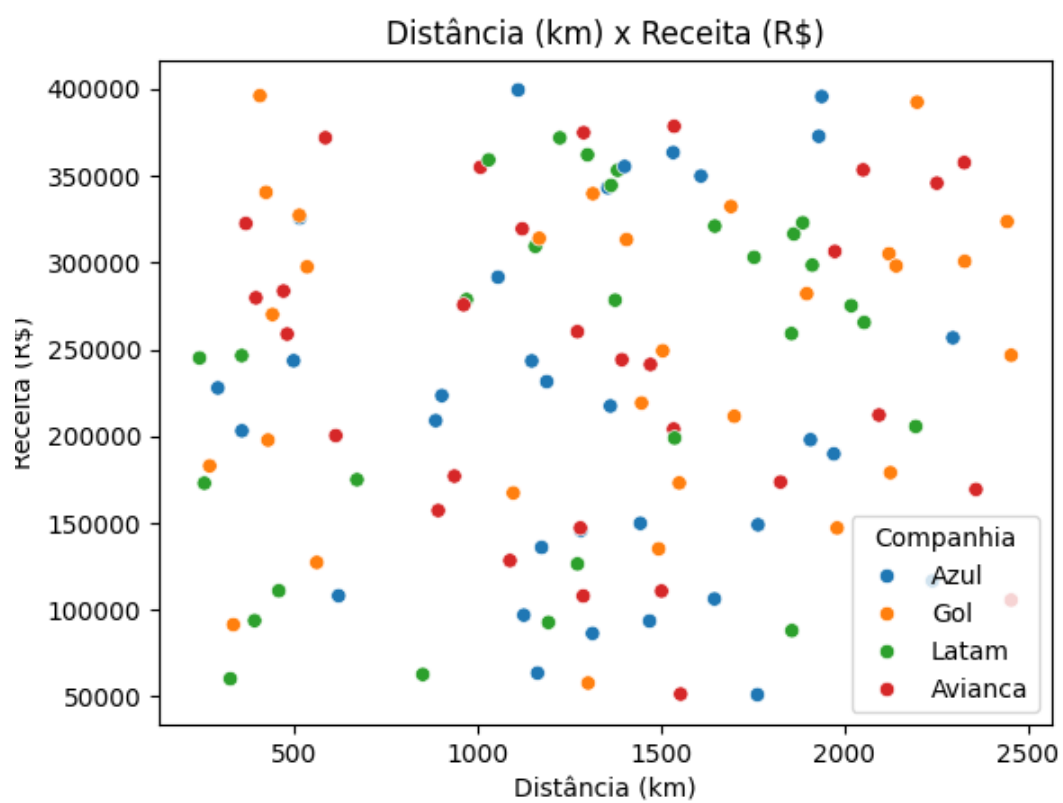
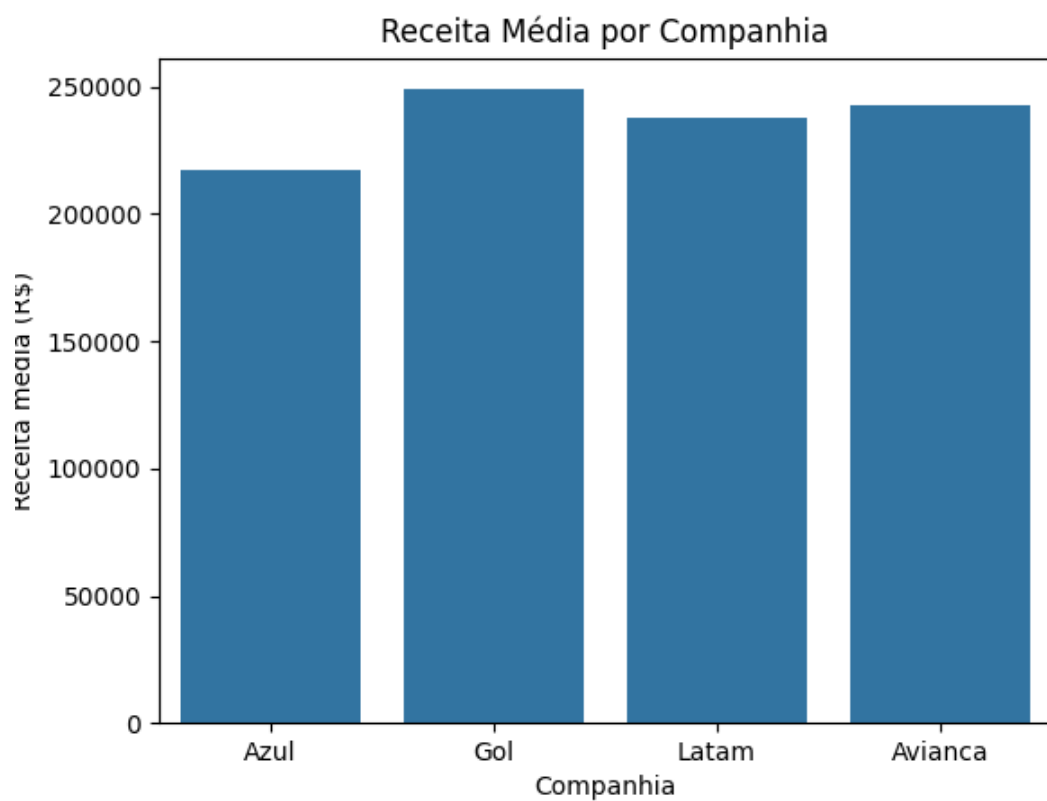
```

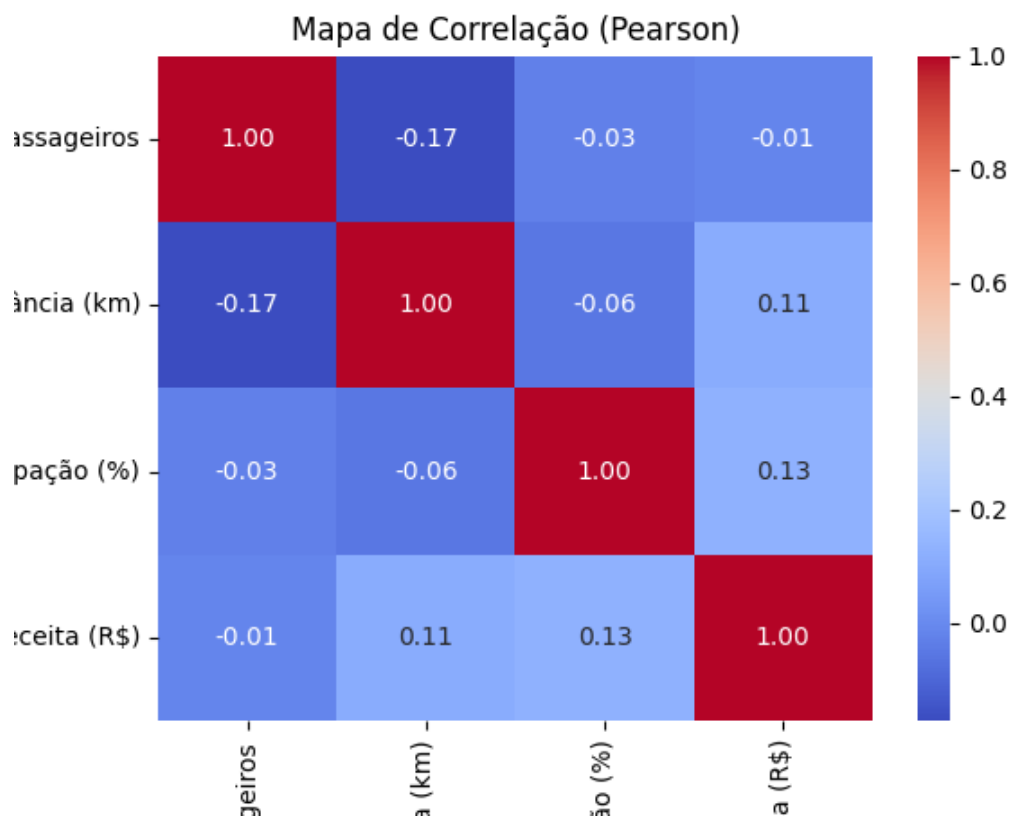
```
# Barra: receita média por companhia (estimator=np.mean calcula a média
para cada grupo)
plt.figure()
sns.barplot(x="Companhia", y="Receita (R$)", data=df, estimator=np.mean,
ci=None)
plt.title("Receita Média por Companhia")
plt.xlabel("Companhia")
plt.ylabel("Receita média (R$)")
plt.show(block=False)

# Scatter: relação entre distância e receita, colorido por companhia para
ver padrões por operadora
plt.figure()
sns.scatterplot(x="Distância (km)", y="Receita (R$)", hue="Companhia",
data=df)
plt.title("Distância (km) x Receita (R$)")
plt.xlabel("Distância (km)")
plt.ylabel("Receita (R$)")
plt.legend(title="Companhia")
plt.show(block=False)

# Heatmap de correlação entre variáveis numéricas – ajuda a ver forças de
relação linear
plt.figure()
corr = df[["Passageiros", "Distância (km)", "Ocupação (%)", "Receita
(R$)"]].corr()
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Mapa de Correlação (Pearson)")
plt.show(block=False)
```







```
# === PARTE D - PERGUNTAS ANALÍTICAS (COM GRÁFICOS) ===
# 1) Companhia com mais voos: usar value_counts para ordenar e idxmax para
#    pegar a maior
comp_mais_voos = df["Companhia"].value_counts().idxmax()
print("\nCompanhia com mais voos:", comp_mais_voos)

Companhia com mais voos: Azul

# Gráfico: número de voos por companhia (countplot)
plt.figure()
sns.countplot(x="Companhia", data=df,
order=df["Companhia"].value_counts().index)
plt.title("Número de Voos por Companhia")
plt.xlabel("Companhia")
plt.ylabel("Número de voos (contagem de registros)")
plt.show(block=False)

# 2) Verificar se distância influencia receita – usamos correlação de
#    Pearson
corr_dist_receita = df["Distância (km)"].corr(df["Receita (R$)"])
print("\nCorrelação distância x receita:", corr_dist_receita)
```

```
# Comentário: coeficiente próximo de 0 indica pouca relação linear;  
próximo de 1/-1 indica relação forte.
```

```
Correlação distância x receita: 0.10829976175936555
```

```
# 3) Verificar se ocupação está relacionada à receita  
corr_ocup_receita = df["Ocupação (%)"].corr(df["Receita (R$)"])  
print("\nCorrelação ocupação x receita:", corr_ocup_receita)
```

```
Correlação ocupação x receita: 0.1256031396843081
```

```
# 4) Aeroportos de origem com mais voos (contagem)  
print("\nAeroportos de origem com mais voos:")  
print(df["Aeroporto Origem"].value_counts())
```

```
Aeroportos de origem com mais voos:
```

```
Aeroporto Origem
```

```
CGH      32
```

```
GIG      28
```

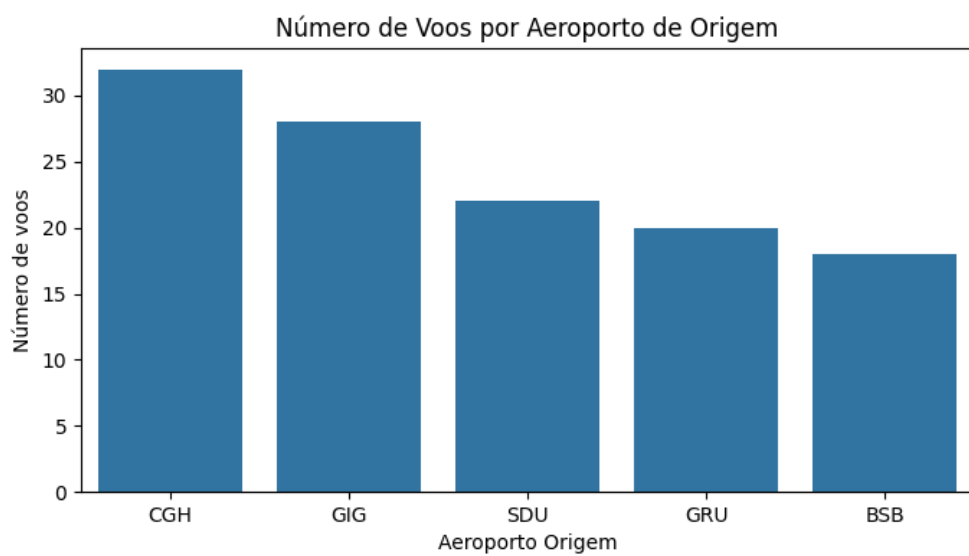
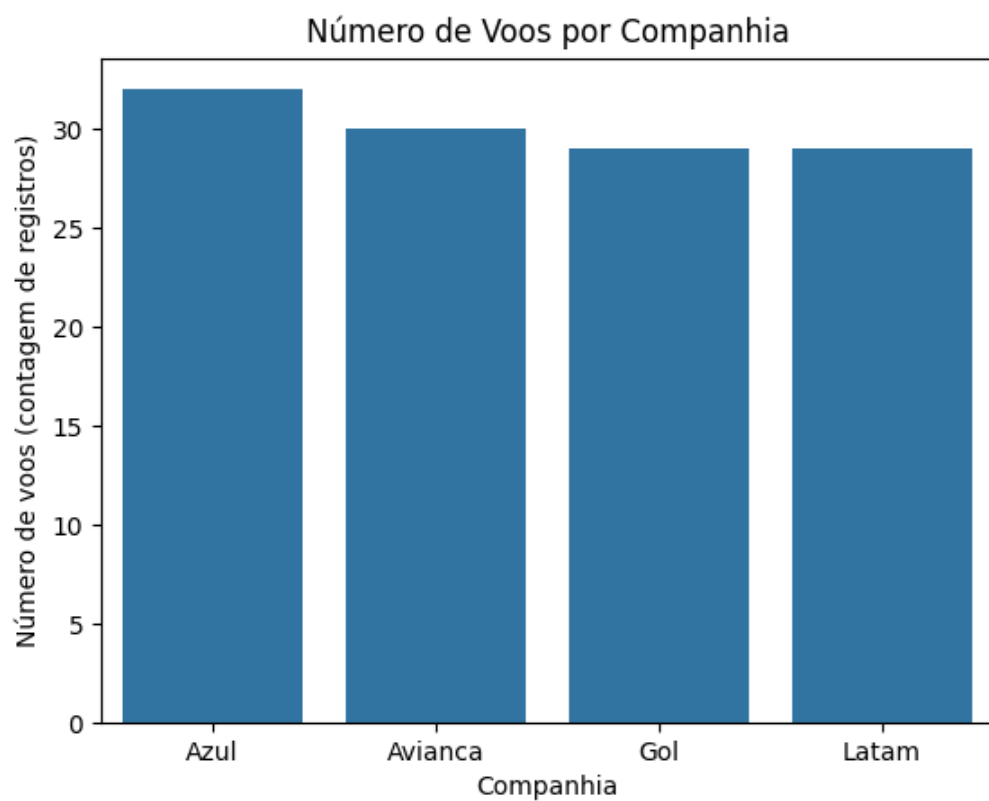
```
SDU      22
```

```
GRU      20
```

```
BSB      18
```

```
Name: count, dtype: int64
```

```
# Gráfico: número de voos por aeroporto de origem  
plt.figure(figsize=(8,4))  
sns.countplot(x="Aeroporto Origem", data=df, order=df["Aeroporto  
Origem"].value_counts().index)  
plt.title("Número de Voos por Aeroporto de Origem")  
plt.xlabel("Aeroporto Origem")  
plt.ylabel("Número de voos")  
plt.show(block=False)
```



```
# === EXTENSÃO ===
# Cria coluna 'Mes' (período mensal) para análises agregadas por mês
df["Mes"] = df["Data"].dt.to_period("M")

# Receita média por mês (agregação por período)
print("\nReceita média por mês:")
print(df.groupby("Mes")["Receita (R$)"].mean())

Receita média por mês:
Mes
2023-01    251552.492258
2023-02    237749.441429
2023-03    224936.445161
2023-04    230361.072667
Freq: M, Name: Receita (R$), dtype: float64

# Gráfico: receita média por mês (linha)
plt.figure(figsize=(8,4))
df.groupby("Mes")["Receita (R$)"].mean().plot(marker="o")
plt.title("Receita Média por Mês")
plt.xlabel("Mês")
plt.ylabel("Receita média (R$)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show(block=False)

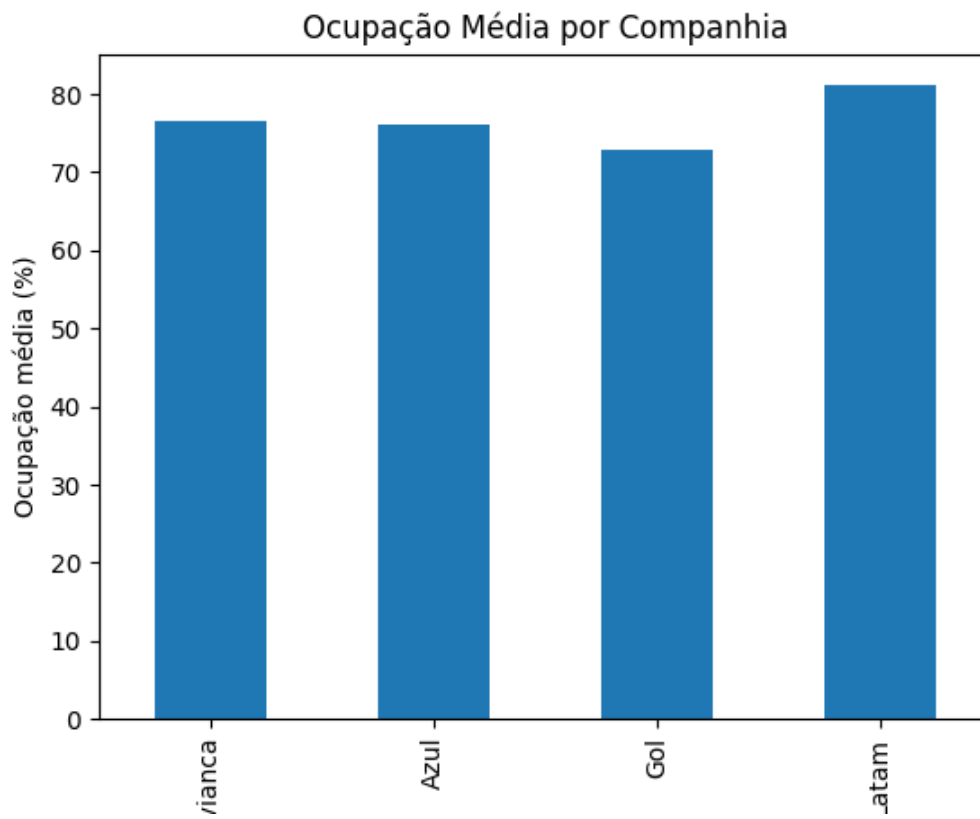
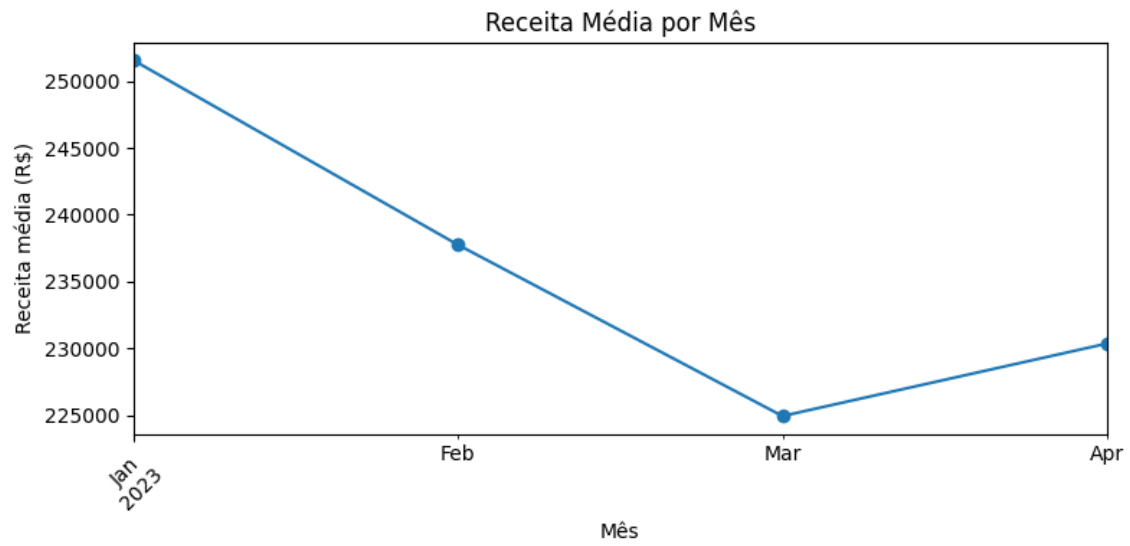
# Ocupação média por companhia
print("\nOcupação média por companhia:")
print(df.groupby("Companhia")["Ocupação (%)"].mean())

Ocupação média por companhia:
Companhia
Avianca    76.583333
Azul       76.159375
Gol        72.872414
Latam      81.110345
Name: Ocupação (%), dtype: float64
```

```
# Gráfico: ocupação média por companhia (barra)
plt.figure()
df.groupby("Companhia")["Ocupação (%)"].mean().plot(kind="bar")
plt.title("Ocupação Média por Companhia")
plt.xlabel("Companhia")
plt.ylabel("Ocupação média (%)")
plt.show(block=False)

# === FINALIZAÇÃO: mantém os gráficos abertos até o usuário pressionar
Enter ===
# Em ambientes interativos, plt.show() já pode ser suficiente. Aqui usamos
input para permitir inspeção.
input("Pressione Enter para fechar os gráficos...")
plt.close('all')

Pressione Enter para fechar os gráficos...
```



1. Companhia com maior participação em número de voos

A companhia com maior participação em número de voos é a Azul, com 32 voos registrados. Na sequência, aparecem Avianca (30), Gol (29) e Latam (29).

Voos por companhia:

Azul 32 | Avianca 30 | Gol 29 | Latam 29

2. A distância influencia a receita?

A correlação entre distância e receita foi de aproximadamente 0.1083, indicando uma correlação fraca e positiva. Isso significa que a distância exerce pouca influência direta sobre a receita, sendo outros fatores, como ocupação e preço do bilhete, mais relevantes.

Correlação distância x receita: 0.10829976175936555

3. Os voos com maior ocupação são necessariamente os de maior receita?

A correlação entre ocupação e receita foi de aproximadamente 0.1256, também fraca e positiva. Assim, voos mais cheios não são necessariamente os que geram mais receita, pois o resultado depende também da distância percorrida e do preço médio das passagens.

Correlação ocupação x receita: 0.1256031396843081

4. Quais aeroportos de origem concentram mais voos?

Os aeroportos com maior concentração de voos foram:

- Congonhas (CGH): 32 voos
- Galeão (GIG): 28 voos
- Santos Dumont (SDU): 22 voos
- Guarulhos (GRU): 20 voos
- Brasília (BSB): 18 voos

O aeroporto que mais concentrou voos foi o de Congonhas (CGH).

Aeroporto Origem

CGH 32 | GIG 28 | SDU 22 | GRU 20 | BSB 18

Conclusão

A análise mostra que a Azul lidera em número de voos, enquanto a Avianca lidera em receita total e número de passageiros. A distância e a ocupação influenciam a receita de forma fraca, e o aeroporto de Congonhas se destaca como o mais movimentado.

```

df.groupby(['Companhia', 'Aeroporto Origem', 'Aeroporto Destino']).agg({
    'Passageiros': 'sum', 'Receita (R$)': 'sum', 'Ocupação (%)': 'mean'
}).reset_index()

# === EVOLUÇÃO MENSAL DO TOTAL DE PASSAGEIROS POR COMPANHIA ===
# Agrupa os dados por mês e companhia
passageiros_mes = (
    df.groupby(["Mes", "Companhia"])["Passageiros"]
      .sum()
      .reset_index()
)

# Converter Mes de Period para string (ex: "2023-01")
passageiros_mes["Mes"] = passageiros_mes["Mes"].astype(str)

print("\n=== Total de passageiros por mês e companhia ===")
print(passageiros_mes.head())

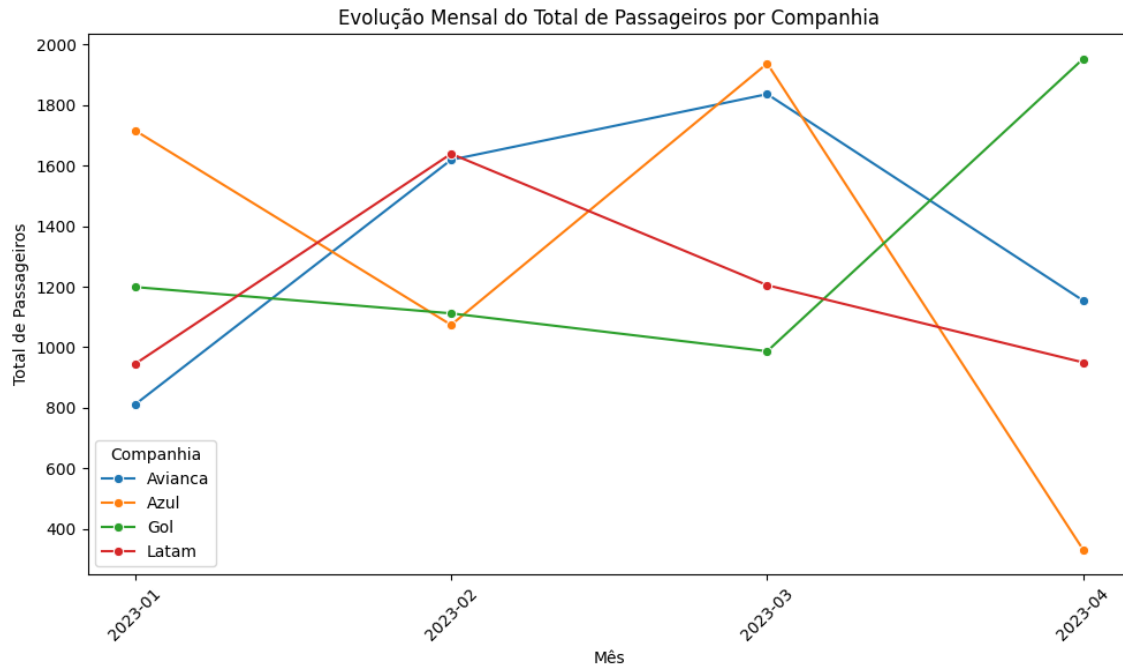
=== Total de passageiros por mês e companhia ===
   Mes Companhia  Passageiros
0  2023-01   Avianca         811
1  2023-01    Azul        1716
2  2023-01     Gol        1199
3  2023-01    Latam         945
4  2023-02   Avianca        1619

```

```

# Gráfico
plt.figure(figsize=(10, 6))
sns.lineplot(data=passageiros_mes, x="Mes", y="Passageiros",
             hue="Companhia", marker="o")
plt.title("Evolução Mensal do Total de Passageiros por Companhia")
plt.xlabel("Mês")
plt.ylabel("Total de Passageiros")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show(block=False)

```

```
# Criar coluna "Rota"
df["Rota"] = df["Aeroporto Origem"] + " → " + df["Aeroporto Destino"]
# Receita por passageiro
df["Receita_por_Passageiro"] = df["Receita (R$)"] / df["Passageiros"]
# Agrupamento por Companhia e Rota
rotas = df.groupby(["Companhia", "Rota"]).agg({
    "Ocupação (%)": "mean",
    "Receita_por_Passageiro": "mean"
}).reset_index()

# Ordenar pelas rotas mais eficientes (receita por passageiro,
# decrescente)
top5_rotas = rotas.sort_values("Receita_por_Passageiro",
    ascending=False).head(5)

print("\n=== Top 5 Rotas mais eficientes por Receita por Passageiro ===")
print(top5_rotas)

=== Top 5 Rotas mais eficientes por Receita por Passageiro ===
   Companhia  Rota  Ocupação (%)  Receita_por_Passageiro
26      Azul  CGH → SDU          69.8          4833.086338
72     Latam  SDU → GIG          95.4          4800.519310
51       Gol  SDU → GIG          80.5          4154.726500
65     Latam  GIG → GRU          77.6          3981.587714
67     Latam  GRU → GIG          84.6          3973.611467
```

```
# Garantir que Data é datetime
df["Data"] = pd.to_datetime(df["Data"])

# Criar coluna de mês (no formato AAAA-MM)
df["Mes"] = df["Data"].dt.to_period("M")

# Agrupar por mês e companhia
passageiros_mes = df.groupby(["Mes",
                              "Companhia"])["Passageiros"].sum().reset_index()

print("\n=== Total de passageiros por mês e companhia ===")
print(passageiros_mes.head())
```

```
=== Total de passageiros por mês e companhia ===
```

	Mes	Companhia	Passageiros
0	2023-01	Avianca	811
1	2023-01	Azul	1716
2	2023-01	Gol	1199
3	2023-01	Latam	945
4	2023-02	Avianca	1619