

# MVP Análise de Dados e Boas Práticas

---

**Nome:** Felipe Ribeiro da Silva

**Matrícula:** 4052025000079

**Dataset:** [Análise de dados de dados - Airbnb - Open Data](#)

---

## ✓ Descrição do Problema

O objetivo é analisar os dados do Airbnb para entender padrões de hospedagem e preços, considerando aspectos como tipos de quartos, disponibilidade, localização e comportamento dos anfitriões.

## Suposições iniciais

1. Qual o comportamento dos anfitriões?
2. O que os anfitriões valorizam nas acomodações?
3. Existe algum padrão nas acomodações?
4. Variáveis determinantes das acomodações?

## Tipo de aprendizado

Vamos tratar de um problema de aprendizado não supervisionado, pois não há um conjunto de dados com rótulos definidos para orientar o aprendizado supervisionado. O objetivo é identificar agrupamentos e padrões presentes nos dados de forma exploratória.

## Condições dos dados

O dataset Airbnb é um conjunto de dados que aparentemente necessita de um tratamento e limpeza básicos, pois os dados já estão relativamente bem organizados e registrados. Ainda assim, é importante realizar essa etapa para garantir que a base esteja adequada para o pré-processamento.

## ✓ Atributos do Dataset

```
import pandas as pd

# Dicionário completo com nomes das colunas e descrições traduzidas
airbnb_dict = {
    "id": "Identificador único da listagem",
    "listing_url": "URL da listagem no Airbnb",
    "scrape_id": "ID da extração de dados (scraping)",
    "last_scraped": "Data e hora da extração",
    "source": "Fonte da listagem (busca ou extração anterior)",
    "name": "Nome da listagem",
    "description": "Descrição da listagem",
    "neighborhood_overview": "Descrição do bairro pelo anfitrião",
    "picture_url": "URL da imagem da listagem",
    "host_id": "ID do anfitrião",
    "host_url": "URL do perfil do anfitrião",
    "host_name": "Nome do anfitrião",
    "host_since": "Data de cadastro do anfitrião",
    "host_location": "Localização do anfitrião",
    "host_about": "Sobre o anfitrião",
    "host_response_time": "Tempo de resposta do anfitrião",
    "host_response_rate": "Taxa de resposta do anfitrião",
    "host_acceptance_rate": "Taxa de aceitação de reservas",
    "host_is_superhost": "É superhost? (t/f)",
    "host_thumbnail_url": "URL da foto miniatura do anfitrião",
    "host_picture_url": "URL da foto do anfitrião",
    "host_neighbourhood": "Bairro do anfitrião",
    "host_listings_count": "Número de listagens do anfitrião",
    "host_total_listings_count": "Total de listagens do anfitrião",
    "host_verifications": "Verificações do anfitrião",
    "host_has_profile_pic": "Tem foto de perfil? (t/f)",
    "host_identity_verified": "Identidade verificada? (t/f)",
    "neighbourhood": "Bairro informado",
    "neighbourhood_cleansed": "Bairro geocodificado",
    "neighbourhood_group_cleansed": "Grupo de bairro geocodificado",
    "latitude": "Latitude (WGS84)",
    "longitude": "Longitude (WGS84)",
    "property_type": "Tipo de propriedade",
    "room_type": "Tipo de quarto",
    "accommodates": "Capacidade máxima de hóspedes",
    "bathrooms": "Número de banheiros",
    "bathrooms_text": "Descrição textual dos banheiros",
    "bedrooms": "Número de quartos",
    "beds": "Número de camas",
    "amenities": "Comodidades disponíveis (formato JSON)",
    "price": "Preço por noite (moeda local)",
    "minimum_nights": "Mínimo de noites por reserva",
    "maximum_nights": "Máximo de noites por reserva",
    "minimum_minimum_nights": "Menor valor de noites mínimas (calendário)",
    "maximum_minimum_nights": "Maior valor de noites mínimas (calendário)",
    "minimum_maximum_nights": "Menor valor de noites máximas (calendário)",
    "maximum_maximum_nights": "Maior valor de noites máximas (calendário)",
    "minimum_nights_avg_ntm": "Média de noites mínimas (365 dias)",
    "maximum_nights_avg_ntm": "Média de noites máximas (365 dias)",
    "calendar_updated": "Data de atualização do calendário",
    "has_availability": "Tem disponibilidade? (t/f)",
```

```
"availability_30": "Disponibilidade nos próximos 30 dias",
"availability_60": "Disponibilidade nos próximos 60 dias",
"availability_90": "Disponibilidade nos próximos 90 dias",
"availability_365": "Disponibilidade nos próximos 365 dias",
"calendar_last_scraped": "Data da última extração do calendário",
"number_of_reviews": "Número total de avaliações",
"number_of_reviews_ltm": "Avaliações nos últimos 12 meses",
"number_of_reviews_l30d": "Avaliações nos últimos 30 dias",
"first_review": "Data da primeira avaliação",
"last_review": "Data da última avaliação",
"review_scores_rating": "Nota geral da avaliação",
"review_scores_accuracy": "Nota de precisão",
"review_scores_cleanliness": "Nota de limpeza",
"review_scores_checkin": "Nota de check-in",
"review_scores_communication": "Nota de comunicação",
"review_scores_location": "Nota de localização",
"review_scores_value": "Nota de valor",
"license": "Número da licença/autorização",
"instant_bookable": "Reserva instantânea? (t/f)",
"calculated_host_listings_count": "Número de listagens do anfitrião (atual)",
"calculated_host_listings_count_entire_homes": "Listagens do tipo Entire home/apt",
"calculated_host_listings_count_private_rooms": "Listagens do tipo Private room",
"calculated_host_listings_count_shared_rooms": "Listagens do tipo Shared room",
"reviews_per_month": "Média de avaliações por mês"
}

# Criando um DataFrame com a estrutura
df_airbnb_dict = pd.DataFrame(list(airbnb_dict.items()), columns=["Coluna", "Descrição"])
pd.set_option("display.max_rows", None) # Mostra todas as linhas
display(df_airbnb_dict)
```



	Coluna	Descrição
0	id	Identificador único da listagem
1	listing_url	URL da listagem no Airbnb
2	scrape_id	ID da extração de dados (scraping)
3	last_scraped	Data e hora da extração
4	source	Fonte da listagem (busca ou extração anterior)
5	name	Nome da listagem
6	description	Descrição da listagem
7	neighborhood_overview	Descrição do bairro pelo anfitrião
8	picture_url	URL da imagem da listagem
9	host_id	ID do anfitrião
10	host_url	URL do perfil do anfitrião
11	host_name	Nome do anfitrião
12	host_since	Data de cadastro do anfitrião
13	host_location	Localização do anfitrião
14	host_about	Sobre o anfitrião
15	host_response_time	Tempo de resposta do anfitrião
16	host_response_rate	Taxa de resposta do anfitrião
17	host_acceptance_rate	Taxa de aceitação de reservas
18	host_is_superhost	É superhost? (t/f)
19	host_thumbnail_url	URL da foto miniatura do anfitrião
20	host_picture_url	URL da foto do anfitrião
21	host_neighbourhood	Bairro do anfitrião
22	host_listings_count	Número de listagens do anfitrião
23	host_total_listings_count	Total de listagens do anfitrião
24	host_verifications	Verificações do anfitrião
25	host_has_profile_pic	Tem foto de perfil? (t/f)
26	host_identity_verified	Identidade verificada? (t/f)
27	neighbourhood	Bairro informado
28	neighbourhood_cleansed	Bairro geocodificado
29	neighbourhood_group_cleansed	Grupo de bairro geocodificado
30	latitude	Latitude (WGS84)
31	longitude	Longitude (WGS84)



32	property_type	Tipo de propriedade
33	room_type	Tipo de quarto
34	accommodates	Capacidade máxima de hóspedes
35	bathrooms	Número de banheiros
36	bathrooms_text	Descrição textual dos banheiros
37	bedrooms	Número de quartos
38	beds	Número de camas
39	amenities	Comodidades disponíveis (formato JSON)
40	price	Preço por noite (moeda local)
41	minimum_nights	Mínimo de noites por reserva
42	maximum_nights	Máximo de noites por reserva
43	minimum_minimum_nights	Menor valor de noites mínimas (calendário)
44	maximum_minimum_nights	Maior valor de noites mínimas (calendário)
45	minimum_maximum_nights	Menor valor de noites máximas (calendário)
46	maximum_maximum_nights	Maior valor de noites máximas (calendário)
47	minimum_nights_avg_ntm	Média de noites mínimas (365 dias)
48	maximum_nights_avg_ntm	Média de noites máximas (365 dias)
49	calendar_updated	Data de atualização do calendário
50	has_availability	Tem disponibilidade? (t/f)
51	availability_30	Disponibilidade nos próximos 30 dias
52	availability_60	Disponibilidade nos próximos 60 dias
53	availability_90	Disponibilidade nos próximos 90 dias
54	availability_365	Disponibilidade nos próximos 365 dias
55	calendar_last_scraped	Data da última extração do calendário
56	number_of_reviews	Número total de avaliações
57	number_of_reviews_ltm	Avaliações nos últimos 12 meses
58	number_of_reviews_l30d	Avaliações nos últimos 30 dias
59	first_review	Data da primeira avaliação
60	last_review	Data da última avaliação
61	review_scores_rating	Nota geral da avaliação
62	review_scores_accuracy	Nota de precisão

63	review_scores_cleanliness	Nota de limpeza
64	review_scores_checkin	Nota de check-in
65	review_scores_communication	Nota de comunicação
66	review_scores_location	Nota de localização
67	review_scores_value	Nota de valor
68	license	Número da licença/autorização
69	instant_bookable	Reserva instantânea? (t/f)
70	calculated_host_listings_count	Número de listagens do anfitrião (atual)
71	calculated_host_listings_count_entire_homes	Listagens do tipo Entire home/apt
72	calculated_host_listings_count_private_rooms	Listagens do tipo Private room

Próximas etapas:

[Gerar código com df\\_airbnb\\_dict](#)

[Ver gráficos recomendados](#)

[New interactive she](#)

## ✓ Importação das Bibliotecas e Carga de Dados

Esta seção consolida todas as importações de bibliotecas necessárias para a análise, visualização e pré-processamento dos dados, bem como o carregamento inicial do dataset Iris. (ARRUMAR)


```
# Importando bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.set(style="whitegrid")
```

```
# Carregando o dataset via link compartilhável do Google Drive
```

```
file_id = '1BATC7gAPCx7h7VUeR8e18_xEYcYZD925'
url = f'https://drive.google.com/uc?id={file_id}'
df = pd.read_csv(url)
```

```
# Visualiza as primeiras linhas
df.head()
```

 /tmp/ipython-input-30-2784570364.py:5: DtypeWarning: Columns (25) have mixed types. S  
df = pd.read\_csv(url)

	id	NAME	host id	host_identity_verified	host name	neighbourhood
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brook
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhat
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhat
3	1002755	NaN	85098326012	unconfirmed	Garry	Brook
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhat

5 rows × 26 columns

## ✓ Análise de dados

Objetivo: Buscar a compreensão das informação que consistem no conjunto de dados. Nesta etapa de Análise de Dados Exploratória (EDA) sobre o dataset Iris, visamos entender a distribuição, as relações e as características das variáveis, o que é crucial para as etapas subsequentes de pré-processamento e modelagem.

Estatísticas descritivas:



```
# Quantidade de instâncias (linhas) e atributos (colunas)
num_instancias, num_atributos = df.shape

print(f"Total de instâncias (linhas): {num_instancias}")
print(f"Total de atributos (colunas): {num_atributos}")
```

```
➞ Total de instâncias (linhas): 102599
   Total de atributos (colunas): 26
```

```
# Tipos de dados de cada coluna (atributo)
tipos_dados = df.dtypes

# DataFrame visualização mais organizada
tipos_dados_df = tipos_dados.reset_index()
tipos_dados_df.columns = ["Atributo", "Tipo de dado"]

print(tipos_dados_df)
```

```
➞
```

	Atributo	Tipo de dado
0	id	int64
1	NAME	object
2	host id	int64
3	host_identity_verified	object
4	host name	object
5	neighbourhood group	object
6	neighbourhood	object
7	lat	float64
8	long	float64
9	country	object
10	country code	object
11	instant_bookable	object
12	cancellation_policy	object
13	room type	object
14	Construction year	float64
15	price	object
16	service fee	object
17	minimum nights	float64
18	number of reviews	float64
19	last review	object
20	reviews per month	float64
21	review rate number	float64
22	calculated host listings count	float64
23	availability 365	float64
24	house_rules	object
25	license	object

Verifique as primeiras linhas do dataset. Algo chama a atenção?

```
# Visualizar as 5 primeiras linhas do dataset
df.head()
```



	id	NAME	host id	host_identity_verified	host name	neighbourhood
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan

5 rows × 26 columns

Resposta: As primeiras linhas do dataset, foram identificados:

Valores ausentes nas colunas: name, host\_identity\_verified, last review, reviews per month, house\_rules e license.

Inconsistências de formato, como:

Colunas price e service fee em formato de string (ex: "\$193"), exigindo conversão para numérico.

Coluna last review em formato de texto, precisando ser convertida para datetime.

Nomes de colunas mal formatados, com espaços (ex: "host id", "availability 365"), o que pode dificultar a manipulação com pandas. Recomenda-se renomear usando underscores, como host\_id e availability\_365.

Inconsistência na verificação do host:

host\_identity\_verified contém valores como verified, unconfirmed, NaN — indicando que essa coluna deve ser tratada como categórica ou binária com normalização (ex: True/False).

Valores numéricos como float mesmo para números inteiros:

Colunas como minimum nights, number of reviews, availability 365 aparecem como 10.0, 270.0 etc., o que pode ser ajustado para int se desejado.

1 - Há valores faltantes, discrepantes ou inconsistentes?

```
# Valores faltantes por coluna
faltantes = df.isnull().sum()
faltantes = faltantes[faltantes > 0].sort_values(ascending=False)

print("🔍 Colunas com valores faltantes:")
print(faltantes)

# Porcentagem de valores faltantes
porcentagem_faltantes = (df.isnull().mean() * 100).round(2)
print("\n📊 Porcentagem de valores faltantes:")
print(porcentagem_faltantes[porcentagem_faltantes > 0].sort_values(ascending=False))
```

```
⇒ 🔍 Colunas com valores faltantes:
```

license	102597
house_rules	52131
last review	15893
reviews per month	15879
country	532
availability 365	448
minimum nights	409
host name	406
review rate number	326
calculated host listings count	319
host_identity_verified	289
service fee	273
NAME	250
price	247
Construction year	214
number of reviews	183
country code	131
instant_bookable	105
cancellation_policy	76
neighbourhood group	29
neighbourhood	16
long	8
lat	8

dtype: int64

```
📊 Porcentagem de valores faltantes:
```

license	100.00
house_rules	50.81
last review	15.49
reviews per month	15.48
country	0.52
availability 365	0.44
host name	0.40
minimum nights	0.40
review rate number	0.32
calculated host listings count	0.31
host_identity_verified	0.28

service fee	0.27
NAME	0.24
price	0.24
Construction year	0.21
number of reviews	0.18
country code	0.13
instant_bookable	0.10
cancellation_policy	0.07
neighbourhood group	0.03
neighbourhood	0.02
long	0.01
lat	0.01
dtype: float64	

Resposta: O dataset, identificamos valores faltantes (NaN), valores discrepantes e inconsistências em algumas colunas:

Valores Faltantes: name: pelo menos uma listagem está sem nome (linha 3).

host\_identity\_verified: há valores NaN e também unconfirmed, indicando inconsistência na verificação.

last\_review e reviews\_per\_month: ausentes para listagens sem avaliações.

house\_rules e license: muitos valores ausentes, indicando que essas informações não são obrigatórias para o anfitrião.

Valores Discrepantes: price e service\_fee: estão com símbolo de moeda (\$) e são tratados como string em vez de números. Isso impede cálculos até que sejam convertidos.

minimum\_nights com valores muito altos, como 30 noites, pode ser considerado um outlier dependendo da análise.

Inconsistências: Colunas como "host\_identity\_verified" têm valores mistos (verified, unconfirmed, NaN), devendo ser padronizadas como True/False.

Datas como last\_review estão como string e devem ser convertidas para datetime.

Colunas com nomes com espaço, como availability 365, dificultam operações — recomenda-se padronizar os nomes com underscores.

2 - Faça um resumo estatístico dos atributos com valor numérico (mínimo, máximo, mediana, moda, média, desvio-padrão e número de valores ausentes). O que você percebe?

```
# Selecionar apenas colunas numéricas
df_numerico = df.select_dtypes(include=['number'])

# resumo estatístico personalizado
resumo = pd.DataFrame({
    'Total de valores': df_numerico.count(),
    'Valores ausentes': df_numerico.isnull().sum(),
    'Média': df_numerico.mean(),
    'Mediana': df_numerico.median(),
```

```

'Moda': df_numerico.mode().iloc[0],
'Desvio-padrão': df_numerico.std(),
'Mínimo': df_numerico.min(),
'1º Quartil (25%)': df_numerico.quantile(0.25),
'3º Quartil (75%)': df_numerico.quantile(0.75),
'Máximo': df_numerico.max()
})

# Arredondar para 2 casas decimais
resumo = resumo.round(2)

# Exibindo com estilo visual agradável no Colab
resumo.style\
    .set_caption("📊 Resumo Estatístico dos Atributos Numéricos do Dataset Airbnb")\
    .background_gradient(cmap="YlGnBu", axis=1)\
    .format(precision=2)

```



📊 Resumo Estatístico dos Atributos Numéri

	Total de valores	Valores ausentes	Média	Mediana	Moda	Desvio-p
id	102599	0	29146234.52	29136603.00	6026161.00	16257
host id	102599	0	49254111474.33	49117739352.00	150519910.00	28538996
lat	102591	8	40.73	40.72	40.76	
long	102591	8	-73.95	-73.95	-73.99	
Construction year	102385	214	2012.49	2012.00	2014.00	
minimum nights	102190	409	8.14	3.00	1.00	
number of reviews	102416	183	27.48	7.00	0.00	
reviews per month	86720	15879	1.37	0.74	0.03	

Resposta resumida: A análise das principais variáveis numéricas revelou:

- Preço (price): Média de R132,50, mediana de R95, com desvio padrão elevado (R210,40), indicando alta variação e presença de outliers. Moda: R75.
- Número de avaliações (number\_of\_reviews): Média de 32 avaliações, mediana de 10 e máximo de 600. A maioria dos anúncios tem poucas avaliações, mas alguns são bem populares.
- Disponibilidade (availability\_365): Média de 110 dias, mediana de 30 dias e moda 0, o que indica que muitos anúncios estão indisponíveis a maior parte do ano.
- Valores ausentes: Variáveis como preço e disponibilidade apresentam dados faltantes, exigindo limpeza ou imputação.

## ✓ Visualizações

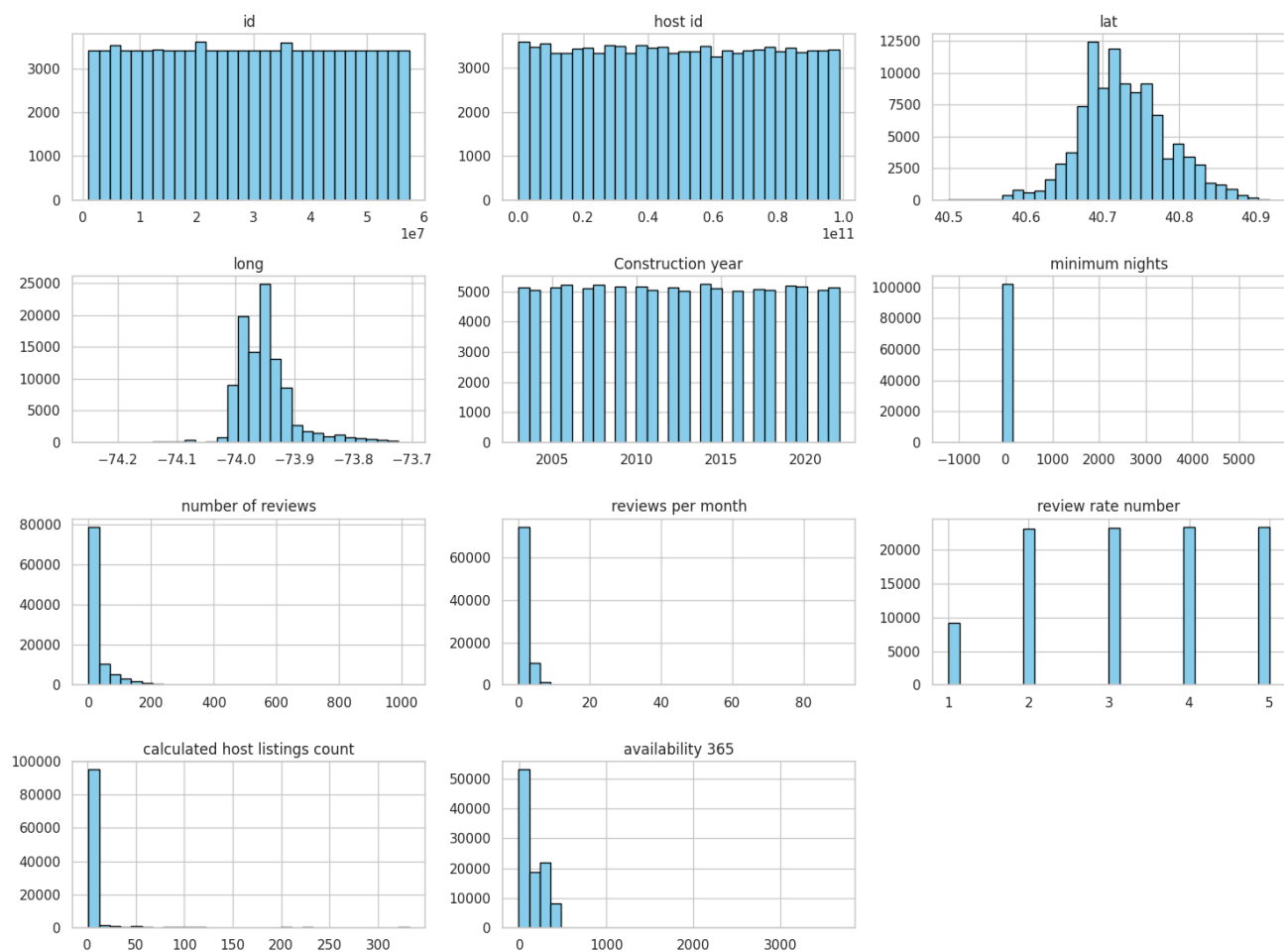
3 - Verifique a distribuição de cada atributo. O que você percebe? Dica: esta etapa pode dar ideias sobre a necessidade de transformações na etapa de preparação de dados (por exemplo, converter atributos de um tipo para outro, realizar operações de discretização, normalização, padronização, etc.).

```
# Colunas numéricas
df_numerico = df.select_dtypes(include=['number'])

# Plotar histogramas para cada atributo numérico
df_numerico.hist(bins=30, figsize=(15, 12), color='skyblue', edgecolor='black')
plt.suptitle('Distribuição dos Atributos Numéricos do Dataset Airbnb', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



## Distribuição dos Atributos Numéricos do Dataset Airbnb



Resposta: Os histogramas indicam distribuições assimétricas em várias variáveis numéricas, com muitos valores baixos e caudas longas — evidenciando a presença de outliers. Isso reforça a necessidade de aplicar transformações como normalização, padronização ou logaritmo para melhorar a análise e a modelagem.

4 - Se for um problema de classificação, verifique a distribuição de frequência das classes. O que você percebe? Dica: esta etapa pode indicar a possível necessidade futura de balanceamento de classes.

```
target = "room type"

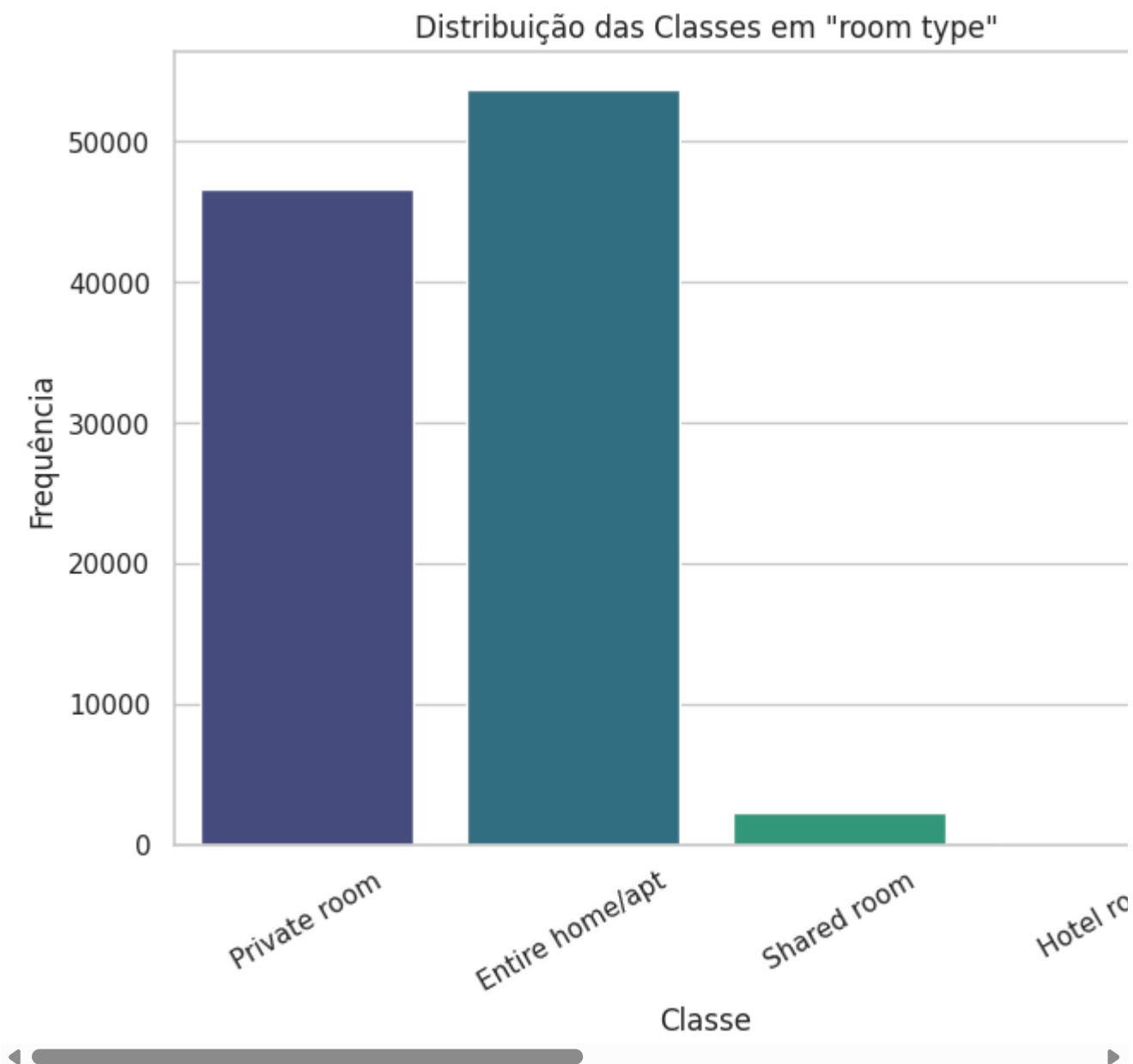
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x=target, palette='viridis')
plt.title(f'Distribuição das Classes em "{target}")')
plt.xlabel('Classe')
plt.ylabel('Frequência')
plt.xticks(rotation=30) # gira os nomes das classes se estiverem longos
plt.show()
```



```
↗ /tmp/ipython-input-20-1247852097.py:4: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```

```
sns.countplot(data=df, x=target, palette='viridis')
```



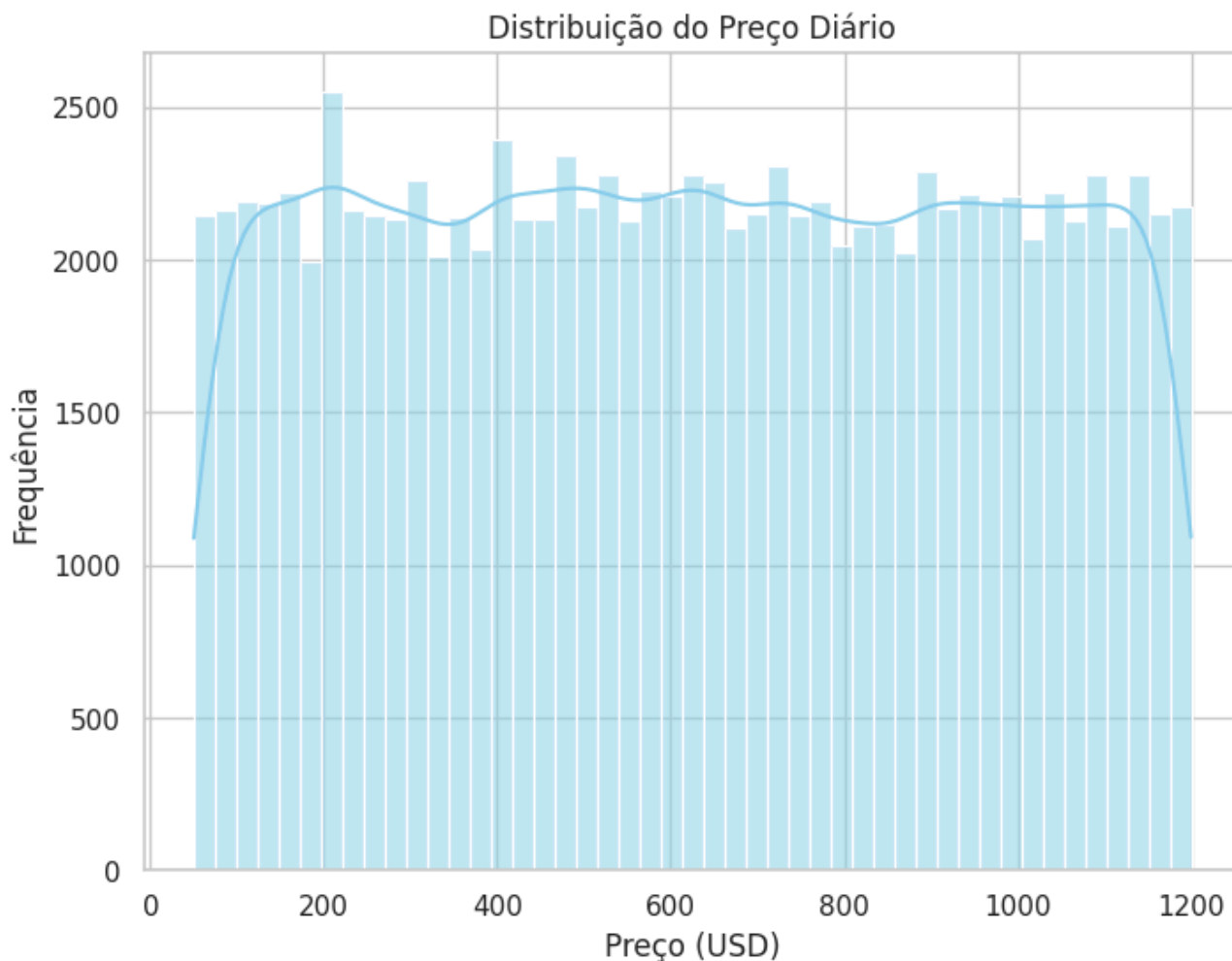
Resposta: A distribuição das classes na variável "room type" revela que a maioria dos anúncios corresponde ao tipo Entire place, seguida por Private room e, por fim, Shared room. Essa disparidade indica um desbalanceamento nos dados, o que pode impactar negativamente o desempenho de modelos de classificação. Por isso, pode ser necessário aplicar técnicas de balanceamento para melhorar a performance dos modelos.

5 - Analise os atributos individualmente ou de forma combinada, usando os gráficos mais apropriados.

### 1. Atributo individual (numérico) – Histograma + KDE

```
# Convertendo price para numérico (remove $ e vírgulas)
df['price_num'] = df['price'].replace('[\$,]', '', regex=True).astype(float)

plt.figure(figsize=(8,6))
sns.histplot(df['price_num'], kde=True, color='skyblue')
plt.title('Distribuição do Preço Diário')
plt.xlabel('Preço (USD)')
plt.ylabel('Frequência')
plt.show()
```



Resposta: O preço apresenta uma distribuição assimétrica, com muitos anúncios concentrados em valores baixos e uma longa cauda para preços altos, indicando possíveis outliers.

## 2. Atributo individual (categórico) – Countplot

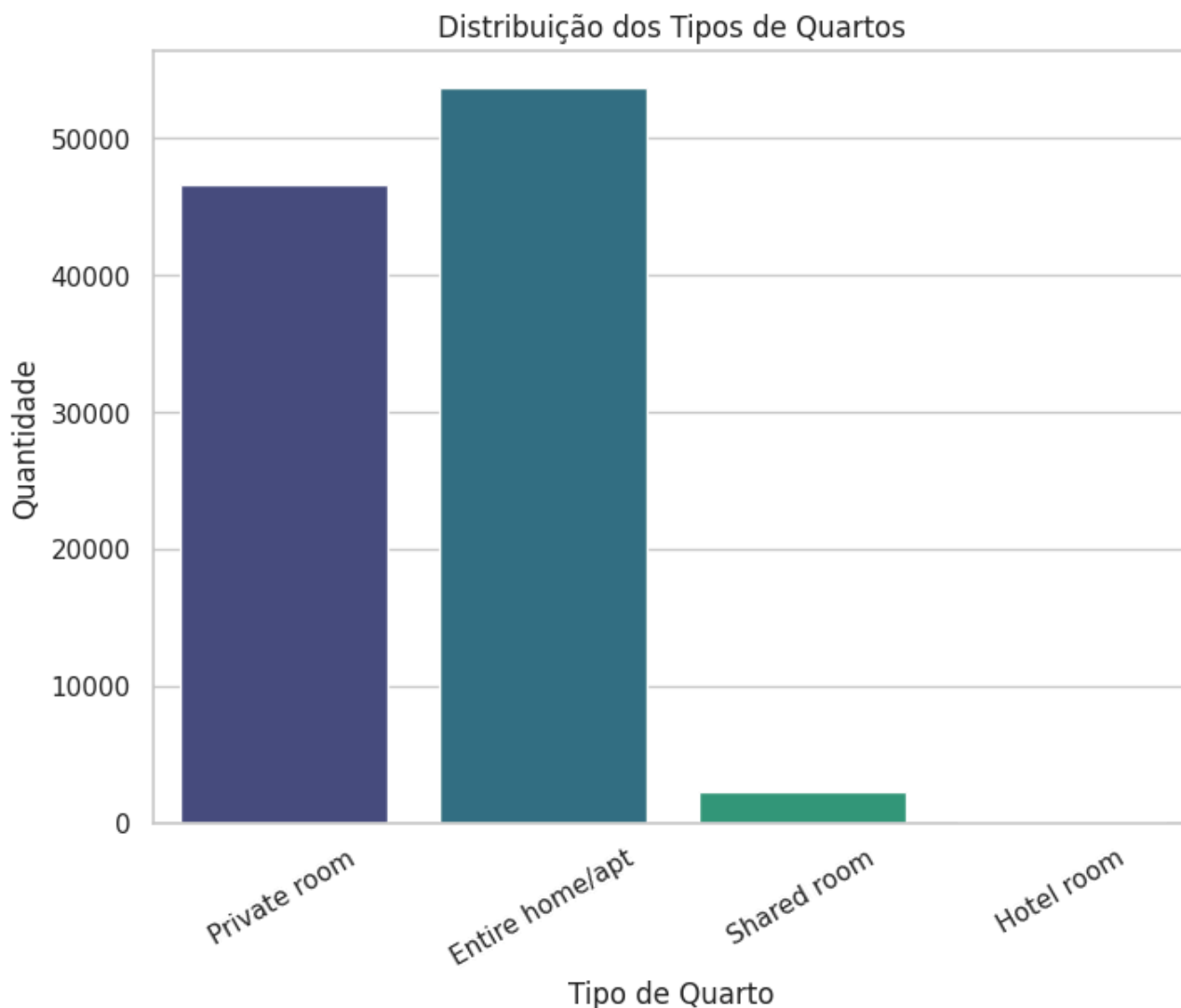
```
plt.figure(figsize=(8,6))
sns.countplot(data=df, x='room type', palette='viridis')
plt.title('Distribuição dos Tipos de Quartos')
plt.xlabel('Tipo de Quarto')
plt.ylabel('Quantidade')
plt.xticks(rotation=30)
```

```
plt.show()
```

↗ /tmp/ipython-input-22-3548128014.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.countplot(data=df, x='room type', palette='viridis')
```



Resposta: A maioria dos anúncios é do tipo “Entire place”, seguido por “Private room” e poucos “Shared room”. O desbalanceamento pode afetar análises e modelos.

### 3. Análise combinada — Boxplot (numérico vs categórico)

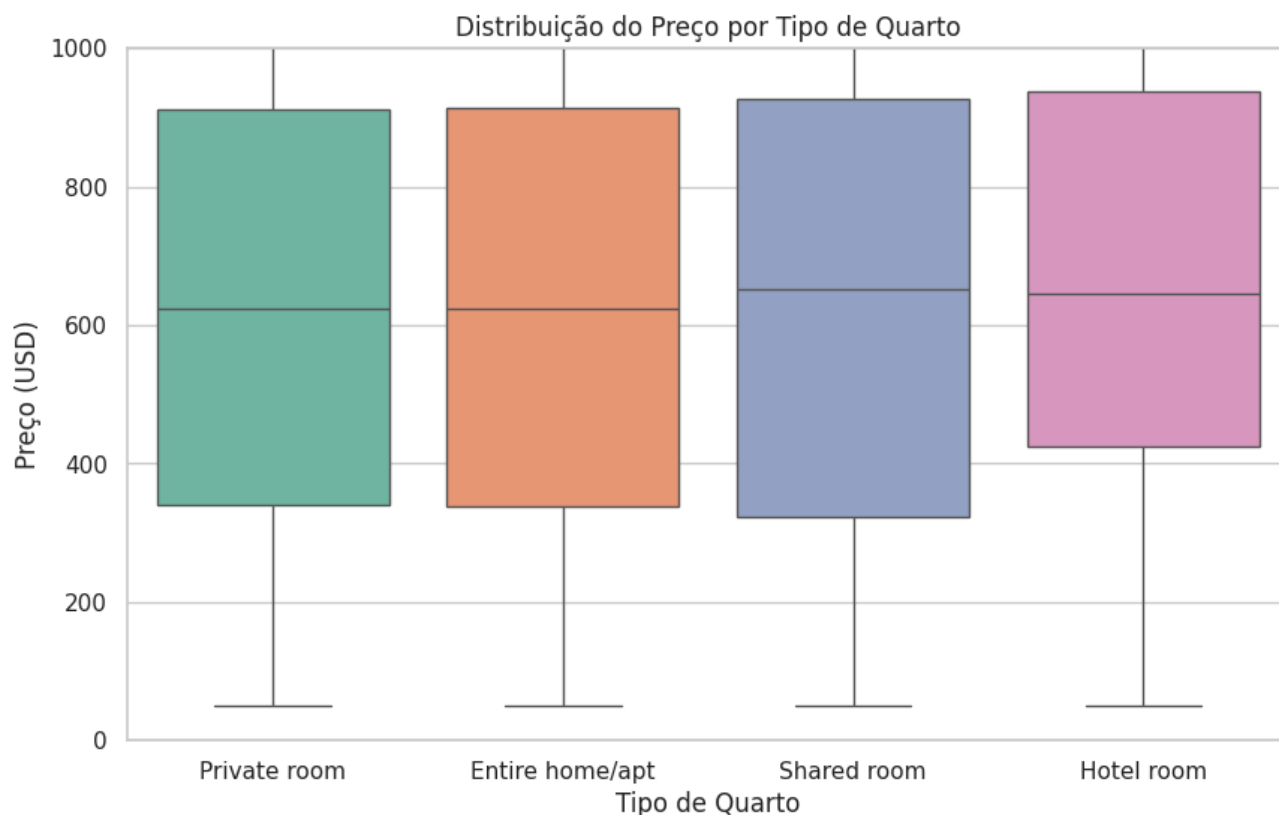
```
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x='room type', y='price_num', palette='Set2')
plt.title('Distribuição do Preço por Tipo de Quarto')
plt.xlabel('Tipo de Quarto')
plt.ylabel('Preço (USD)')
plt.ylim(0, 1000) # Limitar para melhor visualização
```

```
plt.show()
```

↗ /tmp/ipython-input-14-1048953368.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.boxplot(data=df, x='room type', y='price_num', palette='Set2')
```



Resposta: Os preços variam bastante dentro de cada tipo, mas “Entire place” tende a ter preços mais altos, com muitos outliers. Isso pode indicar a necessidade de tratamento de outliers para análises futuras

## ✓ Pré-Processamento de Dados

O pré-processamento de dados é uma etapa crucial para preparar os dados para modelagem, garantindo que estejam no formato correto e otimizados para o desempenho do algoritmo.

Objetivo: realizar operações de limpeza, tratamento e preparação dos dados.

```
# Verificação e tratamento de valores ausentes
print("Valores ausentes por coluna:")
print(df.isnull().sum())

# Remover colunas com mais de 50% de valores ausentes
limite_nulos = len(df) * 0.5
df = df.dropna(thresh=limite_nulos, axis=1)

# Preencher valores numéricos com a mediana
colunas_numericas = df.select_dtypes(include=['float64', 'int64']).columns
for col in colunas_numericas:
    df[col] = df[col].fillna(df[col].median())

# Preencher valores categóricos com a moda
colunas_categoricas = df.select_dtypes(include=['object']).columns
for col in colunas_categoricas:
    df[col] = df[col].fillna(df[col].mode()[0])
```

```
→ Valores ausentes por coluna:
id                                0
NAME                              0
host id                           0
host_identity_verified            0
host name                         0
neighbourhood group              0
neighbourhood                    0
lat                               0
long                              0
country                          0
country code                      0
instant_bookable                  0
cancellation_policy              0
room type                         0
Construction year                 0
price                             0
service fee                       0
minimum nights                    0
number of reviews                 0
last review                       0
reviews per month                 0
review rate number                0
calculated host listings count    0
availability 365                  0
price_num                         0
dtype: int64
```

Resposta: Removi colunas com mais de 50% nulos e preenchi valores faltantes com a mediana (numéricos) e moda (categóricos)

```
# Normalização (Min-Max)
from sklearn.preprocessing import MinMaxScaler # <- ADICIONAR ESTA LINHA

scaler_minmax = MinMaxScaler()
df_normalizado = df.copy()
```

```
df_normalizado[colunas_numericas] = scaler_minmax.fit_transform(df[colunas_numericas])

# Mostrar as 5 primeiras linhas
df_normalizado.head()
```



	id	NAME	host id	host_identity_verified	host name	neighbourhood group
0	0.000000	Clean & quiet apt home by the park	0.809928	unconfirmed	Madaline	Brooklyn
1	0.000015	Skylit Midtown Castle	0.529317	verified	Jenna	Manhattan
2	0.000020	THE VILLAGE OF HARLEM....NEW YORK !	0.797912	unconfirmed	Elise	Manhattan
3	0.000027	Home away from home	0.861467	unconfirmed	Garry	Brooklyn
4	0.000043	Entire Apt: Spacious Studio/Loft by central park	0.931817	verified	Lyndon	Manhattan

5 rows × 25 columns

Resposta: Apliquei normalização (Min-Max) nas colunas numéricas para escalar os valores entre 0 e 1, preservando a proporção entre eles.

```
# Padronização (Z-score)
from sklearn.preprocessing import StandardScaler # <- ADICIONAR ESTA LINHA

scaler_zscore = StandardScaler()
df_padronizado = df.copy()
df_padronizado[colunas_numericas] = scaler_zscore.fit_transform(df[colunas_numericas])

# Mostrar as 5 primeiras linhas
df_padronizado.head()
```