

Analise simples de dois arquivos csv gerados do MySQL. Um desses arquivos é uma consulta

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib as plt
4 import seaborn as sns
```

Arquivo aluguel_veiculos

```
1 df = pd.read_csv('/content/aluguel_veiculos.csv', sep=';', encoding='latin1')
2
```

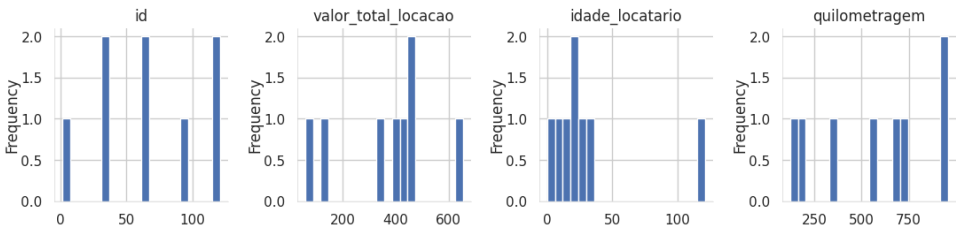
```
1 df.head()
```

	id	valor_total_locacao	qtde_portas	ar_condicionado	data_inicio_locacao	idade_locatario	genero	quilometragem	cotacao_dolar	Esta
0	1	368.38	2 portas	sem ar condicionado	2021-06-01 00:00:00	23	Masculino	957.44	4.41	Mir Ger
1	2	446.85	4 portas	com ar condicionado	2021-05-12 00:00:00	18	Feminino	829.53	5.63	Ba
2	3	414.73	5 portas	sem ar condicionado	2021-04-14 00:00:00	28	Feminino	923.30	8.81	Rio Jane

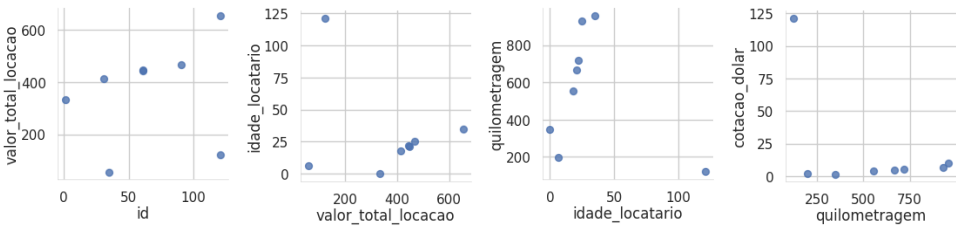
```
1 df.describe()
```

	id	valor_total_locacao	idade_locatario	quilometragem	cotacao_dolar	qtde_dias
count	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000
mean	61.000000	442.444380	21.801653	720.032314	5.266694	3.066116
std	35.073732	56.488951	6.445437	195.643711	2.192854	0.823969
min	1.000000	333.740000	0.000000	345.790000	1.610000	2.000000
25%	31.000000	414.730000	18.000000	554.700000	4.260000	2.000000
50%	61.000000	446.850000	21.000000	665.700000	4.720000	3.000000
75%	91.000000	467.600000	25.000000	930.700000	6.930000	4.000000
max	121.000000	654.000000	35.000000	957.440000	9.910000	5.000000

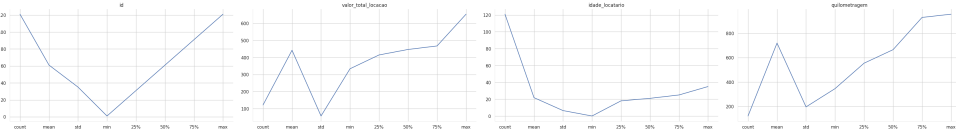
Distributions



2-d distributions



Values



```
1 df.shape
```

(121, 12)

✓ Obs: .shape[0] --> gera a quantidade da consulta. Sem esse comando é gerada uma consulta com todos os dados solicitados

```
1 locacoes_2 = df[df['qtde_diarias'] == 2].shape[0]
2 locacoes_2
```

34

```
1 locatario_menor25 = df[df['idade_locatario'] < 25].shape[0]
2 locatario_menor25
```

81

```
1 locatario_maior25 = df[df['idade_locatario'] > 25].shape[0]
2 locatario_maior25
```

27

```
1 valor_locacao = df['valor_total_locacao'].sum().round()
2 valor_locacao
```

53536.0

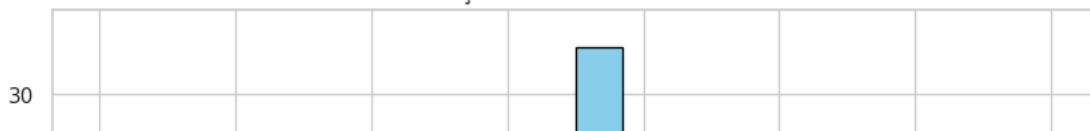
```
1 locacoes_totais = df['id'].count()
2 locacoes_totais
```

121

Distribuição de Idades dos Locatários

```
1 import matplotlib.pyplot as plt
2
3 # Histograma da idade dos locatários
4 plt.figure(figsize=(8, 6))
5 plt.hist(df['idade_locatario'], bins=20, color='skyblue', edgecolor='black')
6 plt.title('Distribuição de Idades dos Locatários')
7 plt.xlabel('Idade do Locatário')
8 plt.ylabel('Frequência')
9 plt.show()
10
```

Distribuição de Idades dos Locatários



Relação entre Valor Total de Locação e Quilometragem

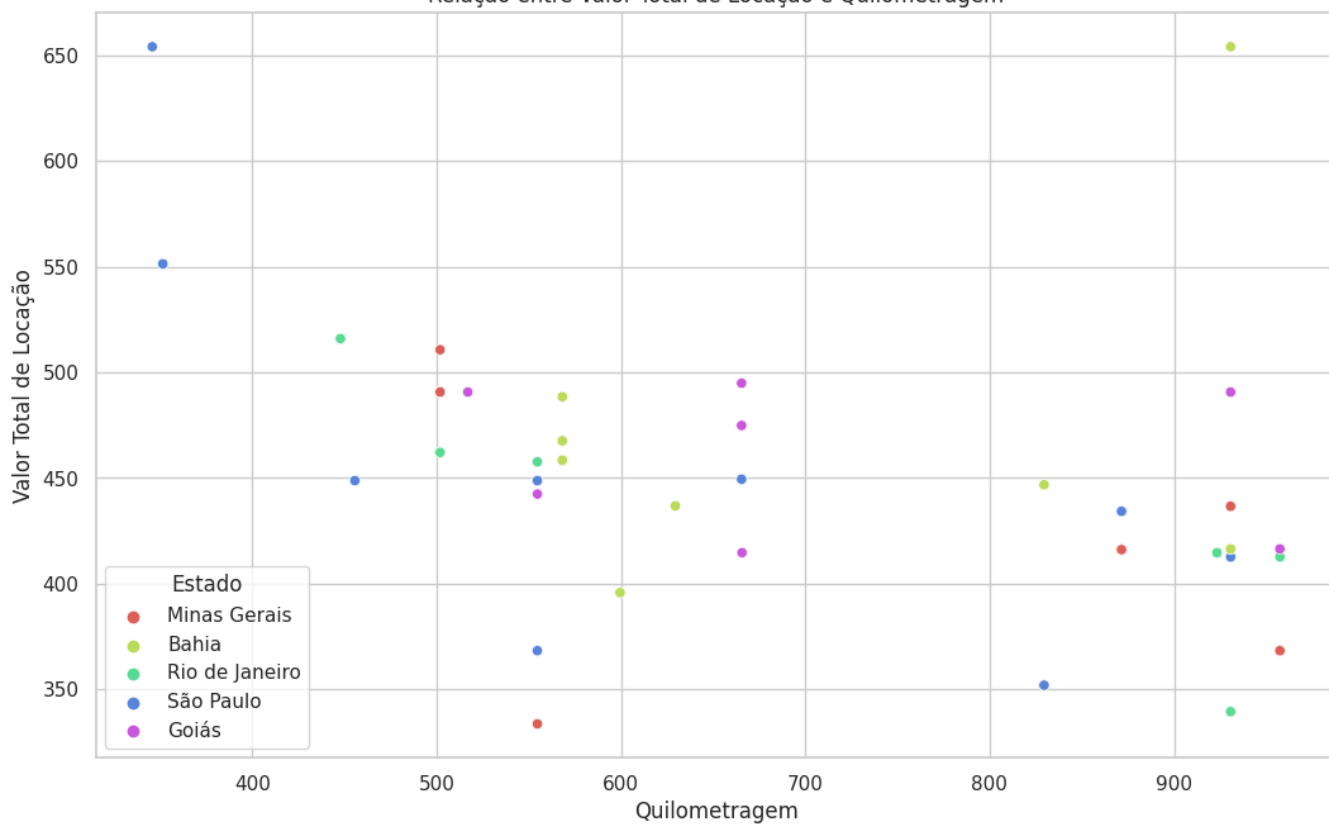


```

1 import seaborn as sns
2
3 # Gráfico de dispersão
4 plt.figure(figsize=(10, 6))
5 sns.scatterplot(data=df, x='quilometragem', y='valor_total_locacao', hue='Estado', palette='hls')
6 plt.title('Relação entre Valor Total de Locação e Quilometragem')
7 plt.xlabel('Quilometragem')
8 plt.ylabel('Valor Total de Locação')
9 plt.legend(title='Estado')
10 plt.show()
11

```

Relação entre Valor Total de Locação e Quilometragem

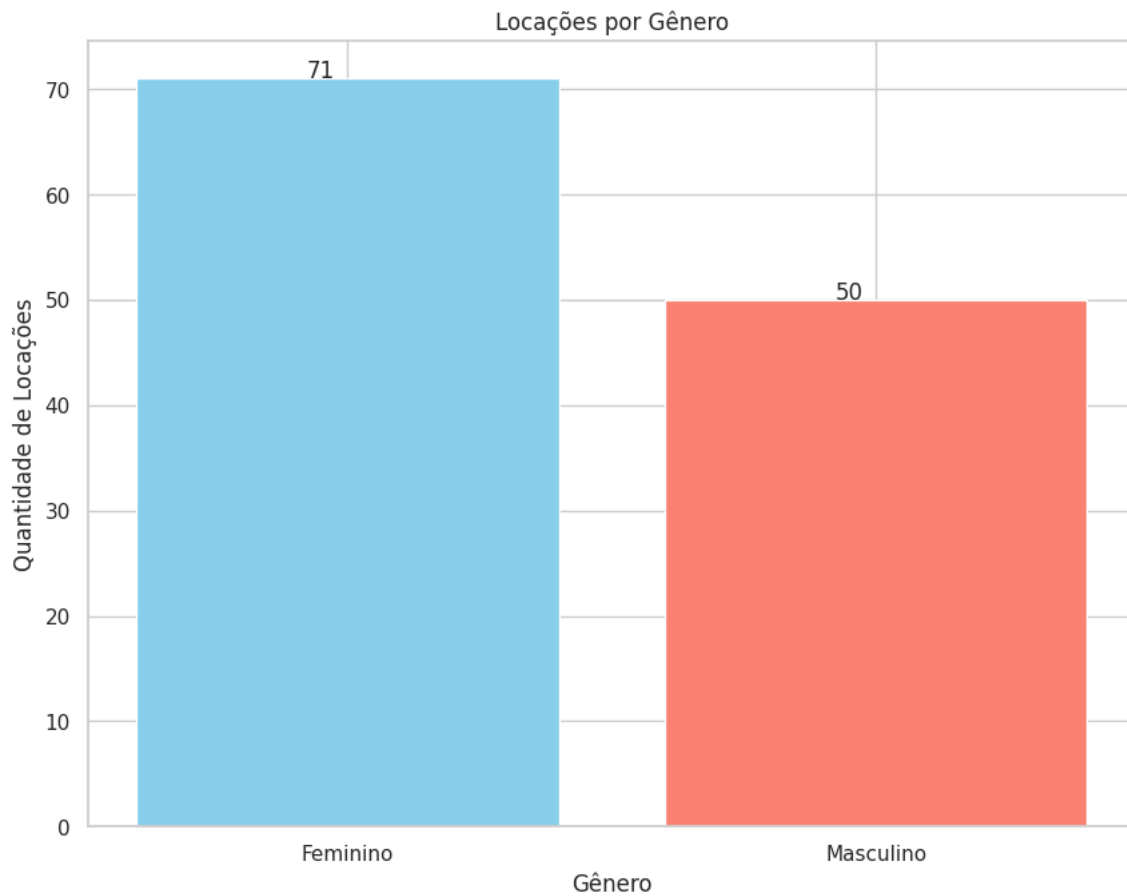


Locações por Gênero

```

1 # Contagem de locações por gênero
2 locacoes_por_genero = df['genero'].value_counts()
3
4 # Gráfico de barras com números nas barras
5 plt.figure(figsize=(8, 6))
6 bars = plt.bar(locacoes_por_genero.index, locacoes_por_genero, color=['skyblue', 'salmon'])
7
8 # Adiciona números nas barras
9 for bar in bars:
10     plt.text(bar.get_x() + bar.get_width() / 2 - 0.05, bar.get_height() + 0.1, str(bar.get_height()), ha='center')
11
12 plt.title('Locações por Gênero')
13 plt.xlabel('Gênero')
14 plt.ylabel('Quantidade de Locações')
15 plt.show()
16

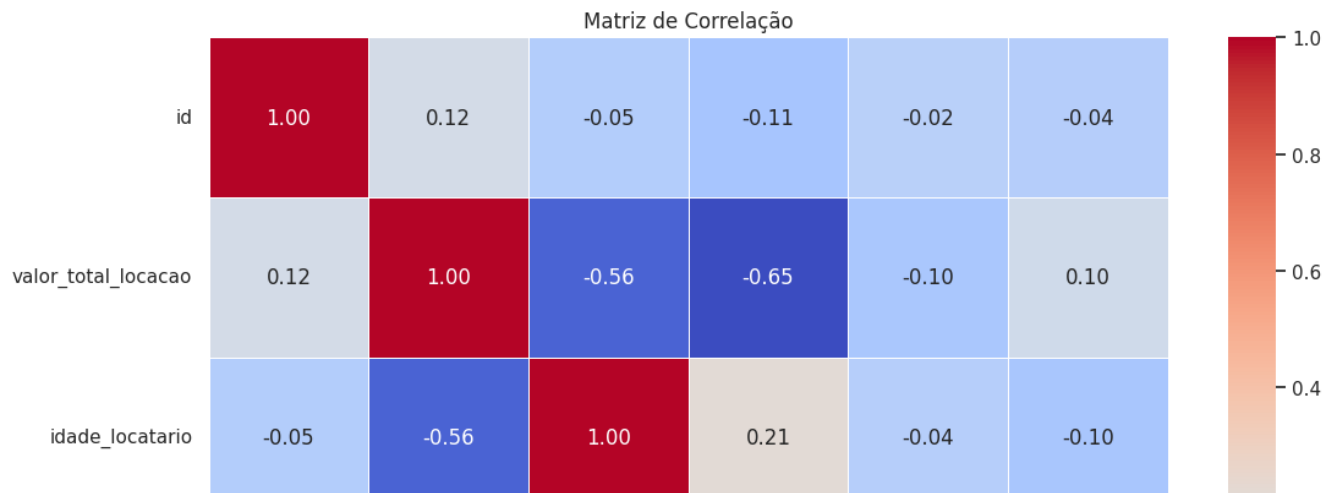
```



Correlação entre Variáveis

```
1 # Matriz de correlação
2 correlation_matrix = df.corr()
3
4 # Mapa de calor
5 plt.figure(figsize=(10, 8))
6 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
7 plt.title('Matriz de Correlação')
8 plt.show()
9
```

<ipython-input-76-382aea06b271>:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_



Comparação da idade dos Locatários



```
1 # Contagem de locatários com idade maior que 25
2 locatario_maior25 = df[df['idade_locatario'] > 25].shape[0]
3
4 # Contagem de locatários com idade menor que 25
5 locatario_menor25 = df[df['idade_locatario'] < 25].shape[0]
6
7 # Criando o gráfico de barras
8 plt.figure(figsize=(8, 6))
9 plt.bar(['Maior que 25', 'Menor que 25'], [locatario_maior25, locatario_menor25], color=['skyblue', 'salmon'])
10
11 # Adicionando números nas barras
12 for i, value in enumerate([locatario_maior25, locatario_menor25]):
13     plt.text(i, value + 0.1, str(value), ha='center')
14
15 plt.title('Comparação de Locatários por Faixa Etária')
16 plt.xlabel('Faixa Etária')
17 plt.ylabel('Quantidade de Locatários')
18 plt.show()
19
```

Comparação de Locatários por Faixa Etária

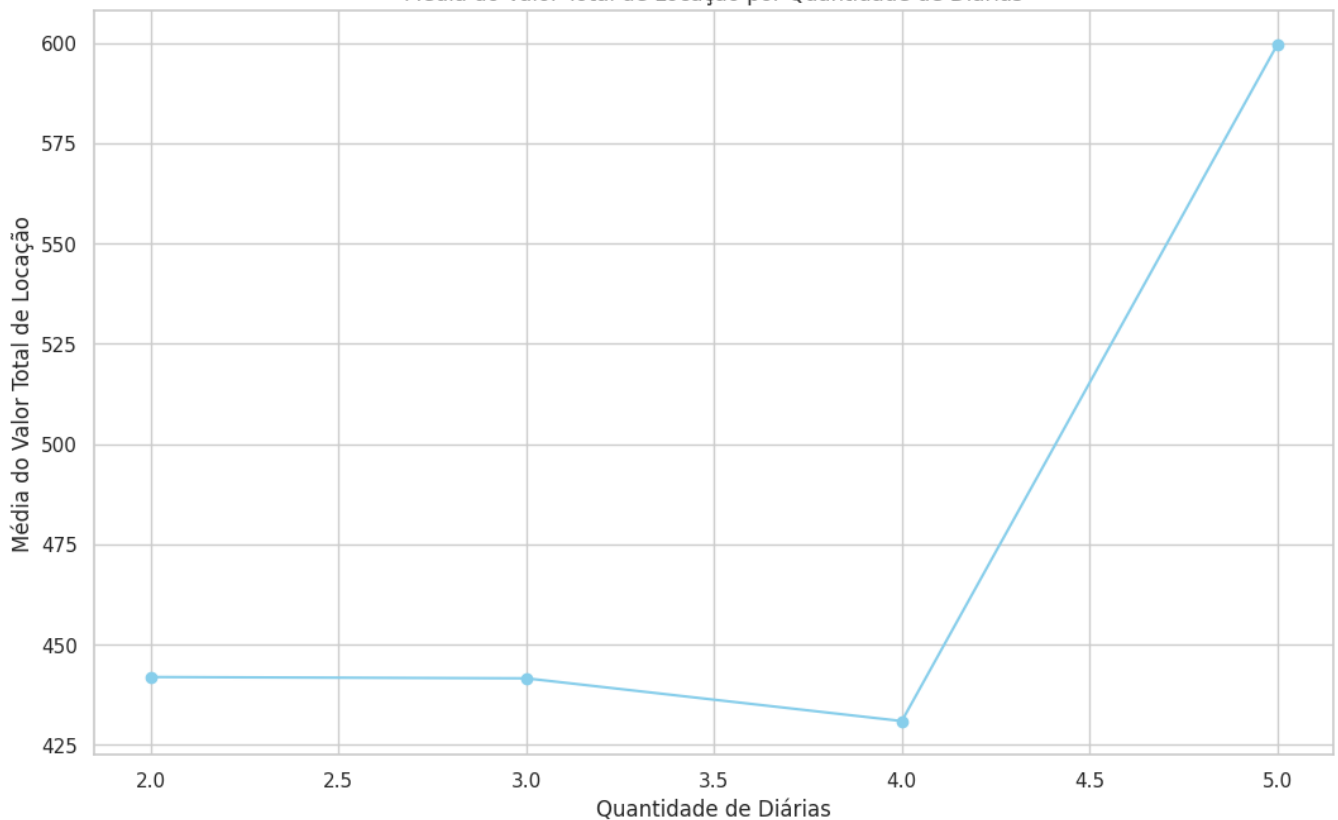
Valor da Locação

```
1 valor_locacao = df['valor_total_locacao']
2 valor_locacao
```

53536.0

```
1 # Agrupando os dados pela quantidade de diárias e calculando a média do valor total de locação
2 media_valor_diaria = df.groupby('qtde_dias')['valor_total_locacao'].mean().reset_index()
3
4 # Criando o gráfico de linha
5 plt.figure(figsize=(10, 6))
6 plt.plot(media_valor_diaria['qtde_dias'], media_valor_diaria['valor_total_locacao'], marker='o', color='skyblue')
7
8 # Adicionando rótulos e título
9 plt.title('Média do Valor Total de Locação por Quantidade de Diárias')
10 plt.xlabel('Quantidade de Diárias')
11 plt.ylabel('Média do Valor Total de Locação')
12
13 # Exibindo o gráfico
14 plt.show()
15
```

Média do Valor Total de Locação por Quantidade de Diárias



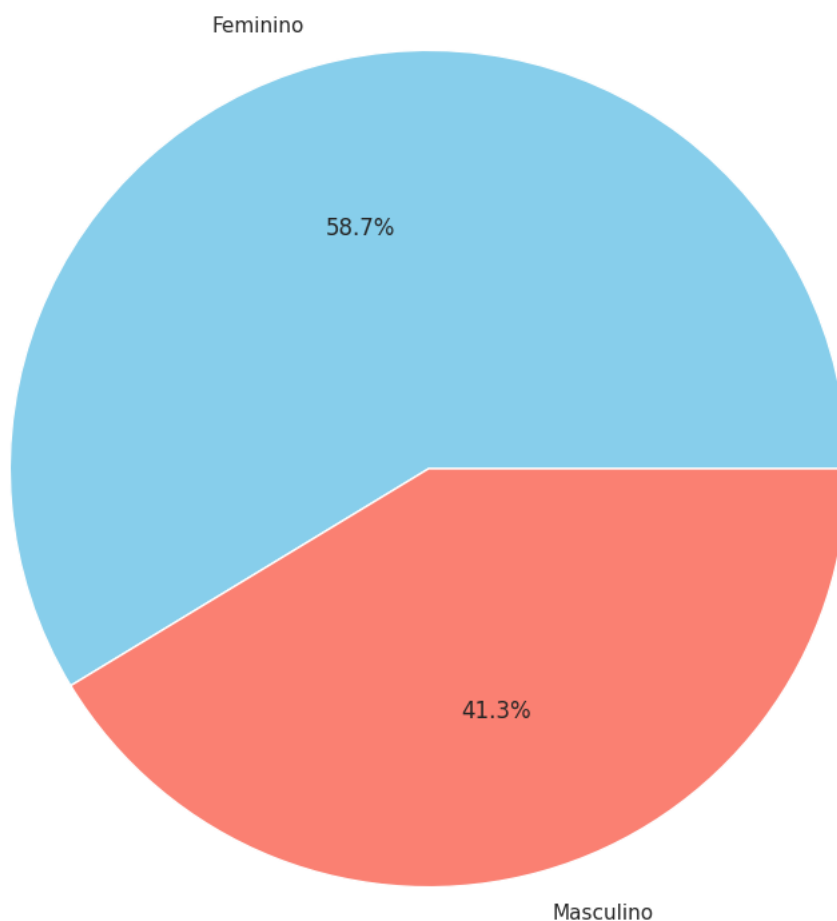
Locações por Gênero

```
1 locacoes_por_genero = df['genero'].value_counts()
2 locacoes_por_genero
```

```
Feminino    71
Masculino    50
Name: genero, dtype: int64
```

```
1 import matplotlib.pyplot as plt
2
3 locacoes_por_genero = df['genero'].value_counts()
4
5 plt.figure(figsize=(8, 8))
6 plt.pie(locacoes_por_genero, labels=locacoes_por_genero.index, autopct='%1.1f%%', colors=['skyblue', 'salmon'])
7 plt.title('Distribuição de Locações por Gênero')
8 plt.show()
9
```

Distribuição de Locações por Gênero



Média de Idade dos Locatários

```
1 media_idade_locatarios = df['idade_locatario'].mean()
2 media_idade_locatarios.round()

22.0
```

Total de Locações por Estado

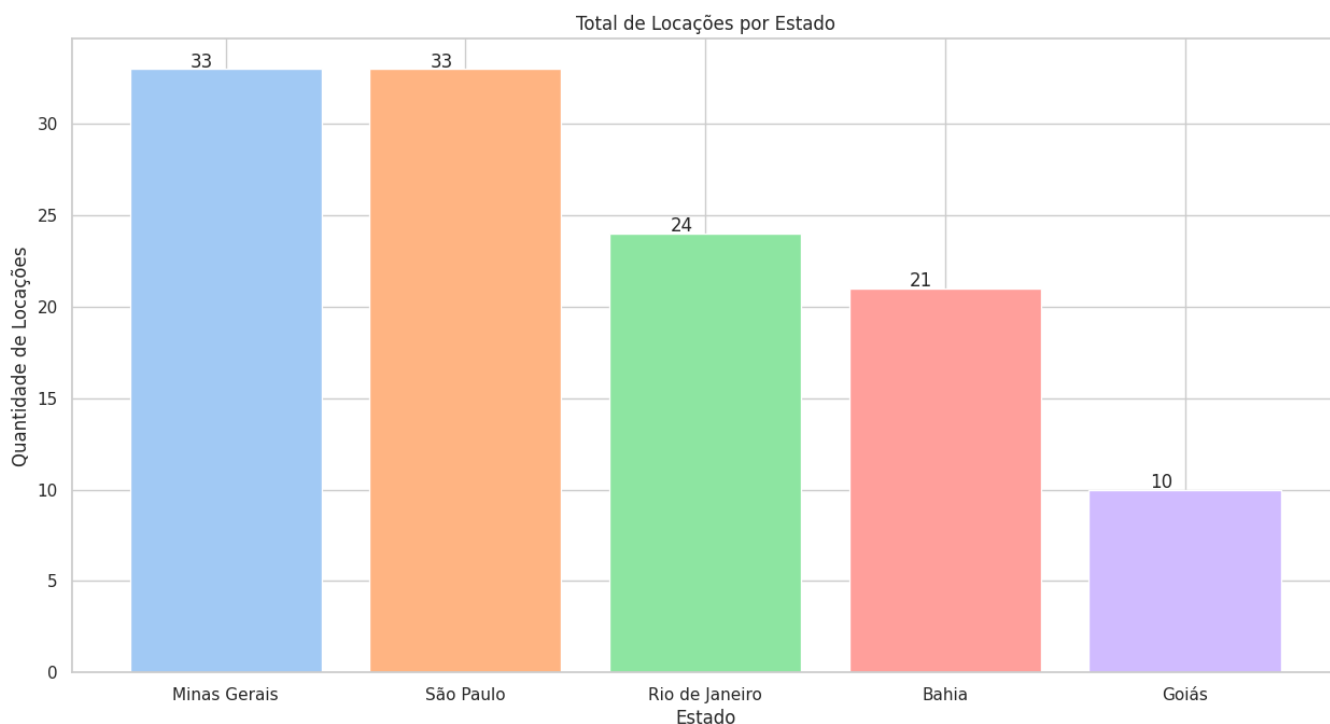
```
1 locacoes_por_estado = df['Estado'].value_counts()
2 locacoes_por_estado
```

```
Minas Gerais    33
São Paulo       33
Rio de Janeiro  24
Bahia           21
Goiás           10
Name: Estado, dtype: int64
```

```

1 locacoes_por_estado = df['Estado'].value_counts()
2
3 # Definindo uma paleta de cores única para cada estado
4 cores_por_estado = sns.color_palette('pastel', n_colors=len(locacoes_por_estado))
5
6 plt.figure(figsize=(12, 6))
7 bars = plt.bar(locacoes_por_estado.index, locacoes_por_estado, color=cores_por_estado)
8
9 # Adicionando números nas barras
10 for bar in bars:
11     plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, bar.get_height() + 0.1, str(int(bar.get_height())), ha='center')
12
13 plt.title("Total de Locações por Estado")
14 plt.xlabel('Estado')
15 plt.ylabel('Quantidade de Locações')
16 plt.show()
17

```



Média de Quilometragem percorrida

```

1 media_quilometragem = df['quilometragem'].mean()
2 media_quilometragem.round(2)

```

720.03

Locações com Ar-condicionado

```

1 locacoes_com_ar_condicionado = df[df['ar_condicionado'] == 'com ar condicionado'].shape[0]
2 locacoes_com_ar_condicionado

```

70

Valor Médio de Locação por Diária

```

1 valor_medio_diaria = df['valor_total_locacao'].mean() / df['qtde_diarias'].mean()
2 valor_medio_diaria.round(2)

```

144.3

Locações com Quilometragem Alta (> 300km)

```
1 locacoes_quilometragem_alta = df[df['quilometragem'] > 300].shape[0]
2 locacoes_quilometragem_alta

121
```

Locações mais Caras

```
1 locacoes_mais_caras = df.sort_values(by='valor_total_locacao', ascending=False).head()
2 locacoes_mais_caras
```

	id	valor_total_locacao	qtde_portas	ar_condicionado	data_inicio_locacao	idade_locatario	genero	quilometragem	cotacao_dolar	Est
86	87	654.00	89 portas	sem ar condicionado	2021-03-27 00:00:00	0	Masculino	345.79	3.98	F
94	95	654.00	97 portas	com ar condicionado	2021-06-29 00:00:00	0	Masculino	930.70	4.52	E
20	21	551.32	23 portas	com ar condicionado	2021-06-06 00:00:00	21	Feminino	351.55	5.56	F

Média do Valor de Locação por Faixa Etária

```
1 df['faixa_etaria'] = pd.cut(df['idade_locatario'], bins=[18, 25, 35, 50, 100], labels=['18-25', '26-35', '36-50', '51+'])
2 media_valor_por_faixa_etaria = df.groupby('faixa_etaria')['valor_total_locacao'].mean().round()
3 media_valor_por_faixa_etaria

faixa_etaria
18-25    456.0
26-35    390.0
36-50      NaN
51+      NaN
Name: valor_total_locacao, dtype: float64
```

Locações por Mês

```
1 # Convertendo 'data_inicio_locacao' para formato de data
2 df['data_inicio_locacao'] = pd.to_datetime(df['data_inicio_locacao'])
3
4 # Criando a coluna 'mes'
5 df['mes'] = df['data_inicio_locacao'].dt.month
6
7 # Contagem de locações por mês
8 locacoes_por_mes = df['mes'].value_counts()
9
10 # Mostrar resultados
11 locacoes_por_mes
12

6    41
4    37
5    36
3     3
2     2
1     2
Name: mes, dtype: int64
```

Distribuição de Valores de Locação: Histograma

```
1 plt.figure(figsize=(10, 6))
2
3 # Histograma
4 n, bins, patches = plt.hist(df['valor_total_locacao'], bins=20, color='skyblue', edgecolor='black')
5
6 # Adiciona os números nas barras
7 for value, count, patch in zip(bins[:-1], n, patches):
8     plt.text(value + (bins[1] - bins[0]) / 2, count, str(int(count)), ha='center', va='bottom')
9
10 plt.title('Distribuição de Valores de Locação')
11 plt.xlabel('Valor Total de Locação')
12 plt.ylabel('Frequência')
13 plt.show()
14
```

