Exercícios_Estatistica

May 7, 2025

0.1 Exercício 1

Os dados a seguir representam o numero de cliente, por dia, no restaurante Bom Prato desde sua inalguração:

96, 279, 255, 254, 75, 211, 271 e 291. Utilizando esses dados, crie o histograma

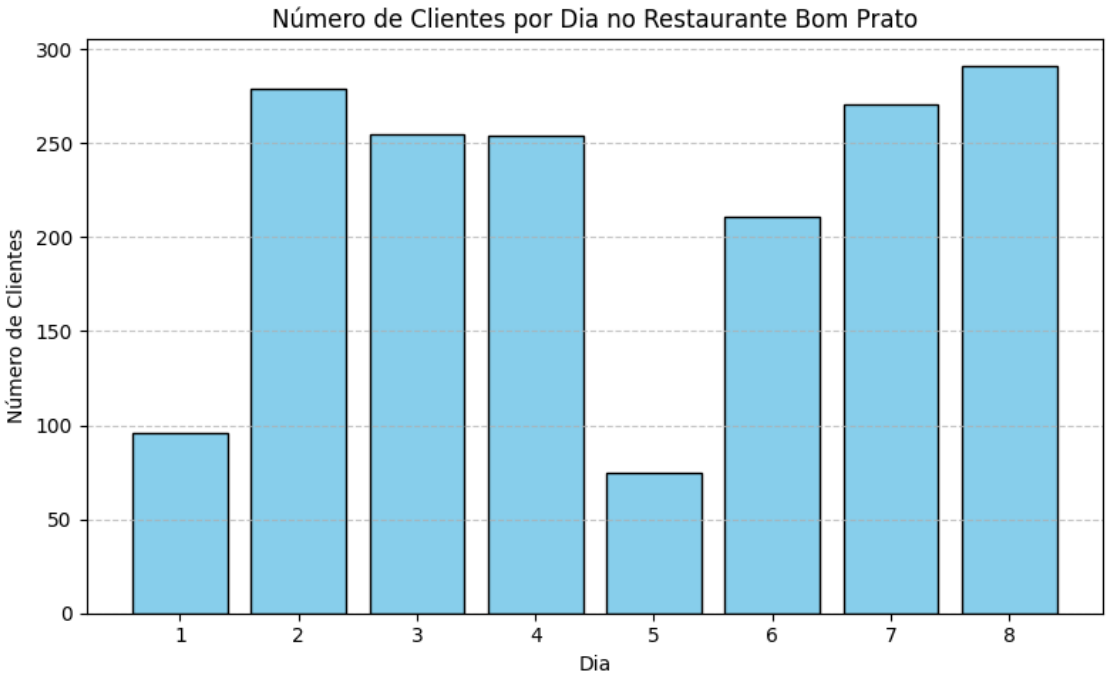
[2]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np


# Gráfico de barras com os dados originais

# Dados de clientes por dia no restaurante
clientes = [96, 279, 255, 254, 75, 211, 271, 291]

# Plotando os valores dos dados originais
dias = list(range(1, len(clientes) + 1))  # Dias desde a inauguração

plt.figure(figsize=(8, 5))
plt.bar(dias, clientes, color='skyblue', edgecolor='black')
plt.title('Número de Clientes por Dia no Restaurante Bom Prato')
plt.xlabel('Dia')
plt.ylabel('Número de Clientes')
plt.xticks(dias)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



[3]:
```
# Criando dados de amplitudo
n = len(clientes)                          # Total de dados
min_clientes = np.min(clientes)              # Mínimo
max_clientes = np.max(clientes)              # Máximo
amplitude_total = max_clientes - min_clientes
```

[4]:
```
# Regra de Sturges

k_sturges = int(np.ceil(1 + np.log2(n)))
h_sturges = int(np.ceil(amplitude_total / k_sturges))
print(f" O número de classes: {k_sturges} e valor da amplitude: {h_sturges}")
```

```
O número de classes: 4 e valor da amplitude: 54
```

[10]:
```
# Regra de Sturges

bins_sturges = np.arange(min_clientes, max_clientes + h_sturges, h_sturges)
labels_sturges = [f'{int(bins_sturges[i])}-{int(bins_sturges[i+1])-1}' for i in␣
 ↪range(len(bins_sturges)-1)]
faixas_sturges = pd.cut(clientes, bins=bins_sturges, labels=labels_sturges,␣
 ↪right=False)
distribuicao_sturges = faixas_sturges.value_counts().sort_index().
 ↪reset_index(name='Frequência')
distribuicao_sturges.rename(columns={'index': 'Faixa Clientes'}, inplace=True)
```

```
# Exibindo os resultados
print("Distribuição - Regra de Sturges:")
print(distribuicao_sturges)
```

```
Distribuição - Regra de Sturges:
  Faixa Clientes  Frequência
0         75-128           2
1        129-182           0
2        183-236           1
3        237-290           4
```

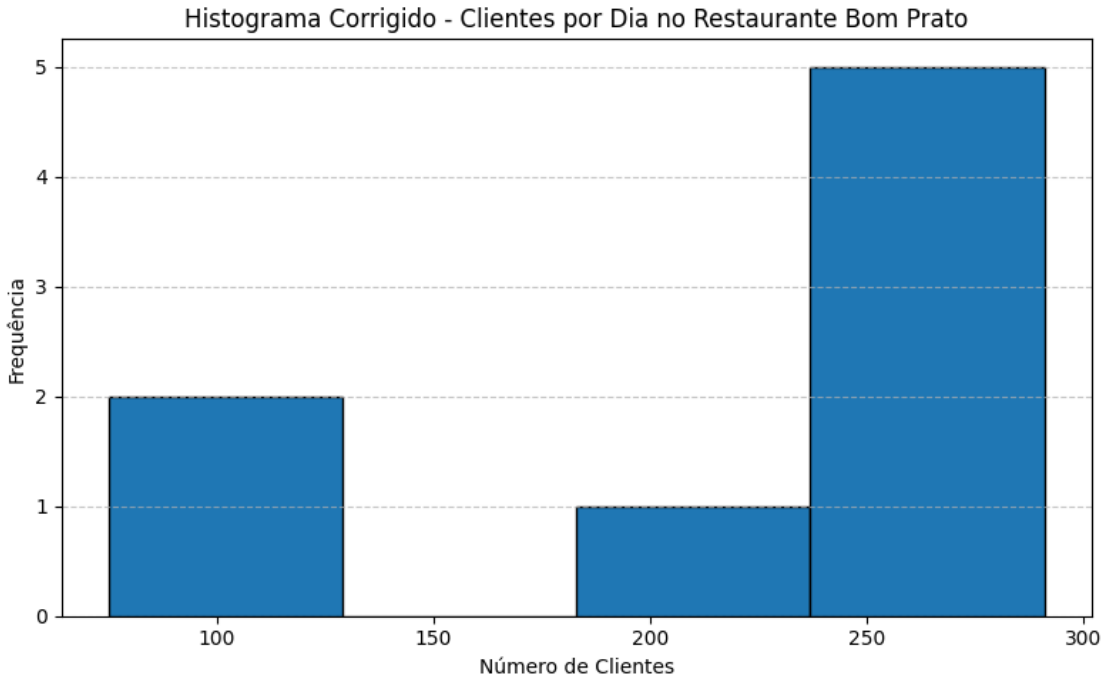
[7]:
```
# Dados de clientes por dia
clientes = np.array([96, 279, 255, 254, 75, 211, 271, 291])

# Determinar os limites dos bins de forma adequada para os dados
min_valor = clientes.min()
max_valor = clientes.max()
bins = np.histogram_bin_edges(clientes, bins='auto')

# Criar tabela de frequência
frequencia, limites = np.histogram(clientes, bins=bins)
tabela_frequencia = pd.DataFrame({
    'Faixa de Clientes': [f'{int(limites[i])} - {int(limites[i+1]-1)}' for i in␣
 ↪range(len(frequencia))],
    'Frequência': frequencia
})
print(f" Tabela de Frequência: {tabela_frequencia}")
```

```
 Tabela de Frequência:   Faixa de Clientes  Frequência
0          75 - 128           2
1         129 - 182           0
2         183 - 236           1
3         237 - 290           5
```

[8]:
```
# Criando histograma ajustado
plt.figure(figsize=(8, 5))
plt.hist(clientes, bins=bins_sturges, edgecolor='black')
plt.title('Histograma Corrigido - Clientes por Dia no Restaurante Bom Prato')
plt.xlabel('Número de Clientes')
plt.ylabel('Frequência')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



0.2 Como interpretar isso?

- A maioria dos dias (5 de 8) teve entre 237 e 290 clientes, indicando um padrão de movimento alto.
- Os valores 75 e 96 indicam dias com baixa frequência, provavelmente os primeiros dias após a inauguração.
- Não há valores médios (na faixa 129–182), o que sugere uma possível curva de crescimento rápida no atendimento.

0.3 Exemplo Boxplot

[11]:
```
# Vamos utilizar o seguinte conjunto de dados para todos os exemplos:

import numpy as np
from scipy import stats

dados = [10, 12, 23, 23, 16, 23, 21, 16]
```

[12]:
```
media = np.mean(dados)
print(f"Média: {media}")
```

```
Média: 18.0
```

[13]:
```
mediana = np.median(dados)
print(f"Mediana: {mediana}")
```

```
Mediana: 18.5
```

[14]:
```python
moda = stats.mode(dados, keepdims=False).mode
print(f"Moda: {moda}")
```

```
Moda: 23
```

[15]:
```python
desvio_padrao = np.std(dados, ddof=1)
print(f"Desvio padrão (amostral): {desvio_padrao}")
```

```
Desvio padrão (amostral): 5.237229365663817
```

[16]:
```python
q1 = np.percentile(dados, 25)
q2 = np.percentile(dados, 50)  # também é a mediana
q3 = np.percentile(dados, 75)

print(f"Q1: {q1}")
print(f"Q2 (Mediana): {q2}")
print(f"Q3: {q3}")
```

```
Q1: 15.0
Q2 (Mediana): 18.5
Q3: 23.0
```

[17]:
```python
iqr = q3 - q1
lim_inf = q1 - 1.5 * iqr
lim_sup = q3 + 1.5 * iqr
outliers = [x for x in dados if x < lim_inf or x > lim_sup]

print(f"IQR: {iqr}")
print(f"Limite inferior: {lim_inf}")
print(f"Limite superior: {lim_sup}")
print(f"Outliers: {outliers}")
```

```
IQR: 8.0
Limite inferior: 3.0
Limite superior: 35.0
Outliers: []
```

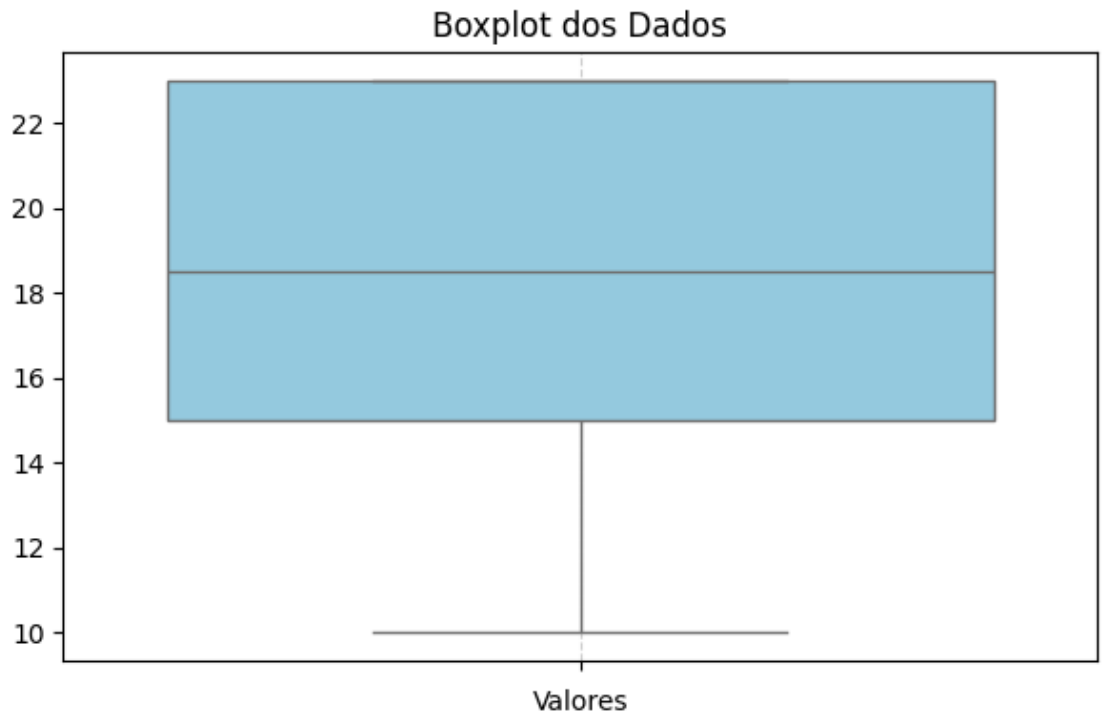
0.4 Boxplot

- O boxplot (ou diagrama de caixa) é uma representação gráfica da distribuição estatística de um conjunto de dados, focando especialmente em medidas de posição e dispersão.

[18]:
```python
# Gerando o boxplot
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Criar o boxplot
```

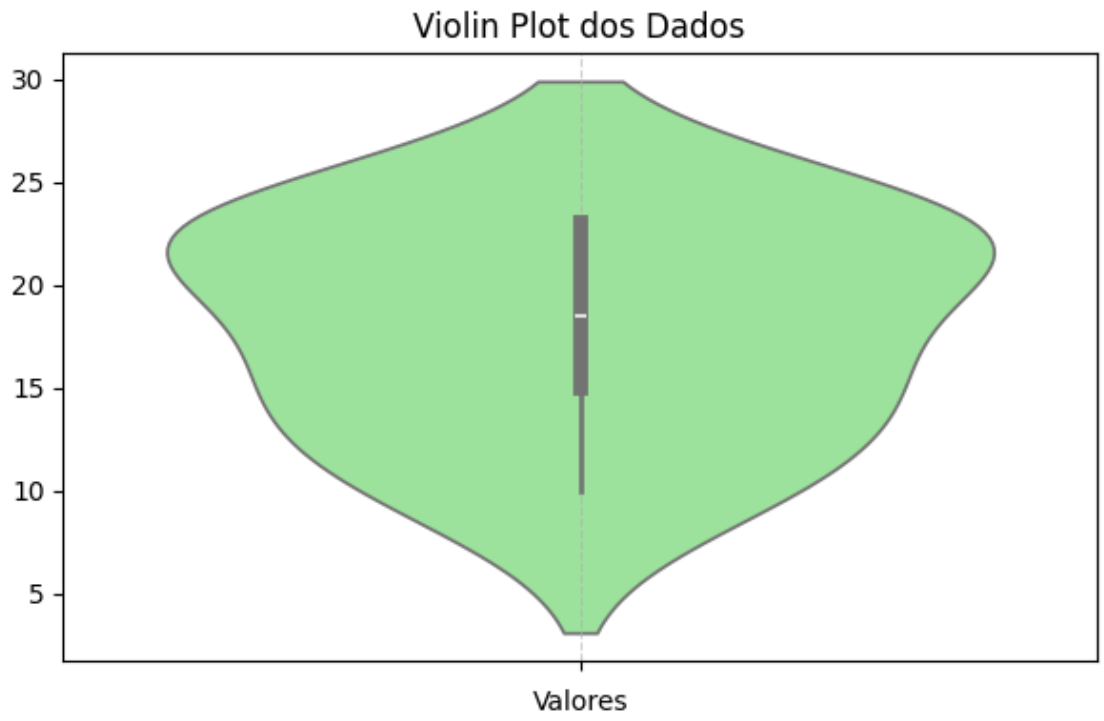
```
plt.figure(figsize=(6, 4))
sns.boxplot(data=dados, orient='v', color='skyblue')
plt.title('Boxplot dos Dados')
plt.xlabel('Valores')
plt.grid(True, axis='x', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



0.5 Boxplot Violino

- É útil quando se deseja visualizar distribuições assimétricas ou multimodais, além da posição central.

[20]:
```
# Gerando o violin plot dos dados
plt.figure(figsize=(6, 4))
sns.violinplot(data=dados, orient='v', color='lightgreen')
plt.title('Violin Plot dos Dados')
plt.xlabel('Valores')
plt.grid(True, axis='x', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```


Violin Plot dos Dados
30
25
20
15
10
5
Valores

[ ]: