
Estatística para Ciência de Dados



Profa. Rebeca Valgueiro

Quem sou eu?

- Graduada em Engenharia Civil
- MBA em Gestão Empresarial
- Trabalho a mais de 4 anos no mercado de tecnologia atuando em projetos de:
 - Desenvolvimento web
 - Desenvolvimento desktop- windows
 - Data Science
 - IA e visão computacional



<https://www.linkedin.com/in/rebecavalgueiro/>

Estatística Descritiva

Imagine que você tem um conjunto de dados

- As alturas de todos os alunos da sua turma
- O número de vendas de um produto ao longo do último ano.



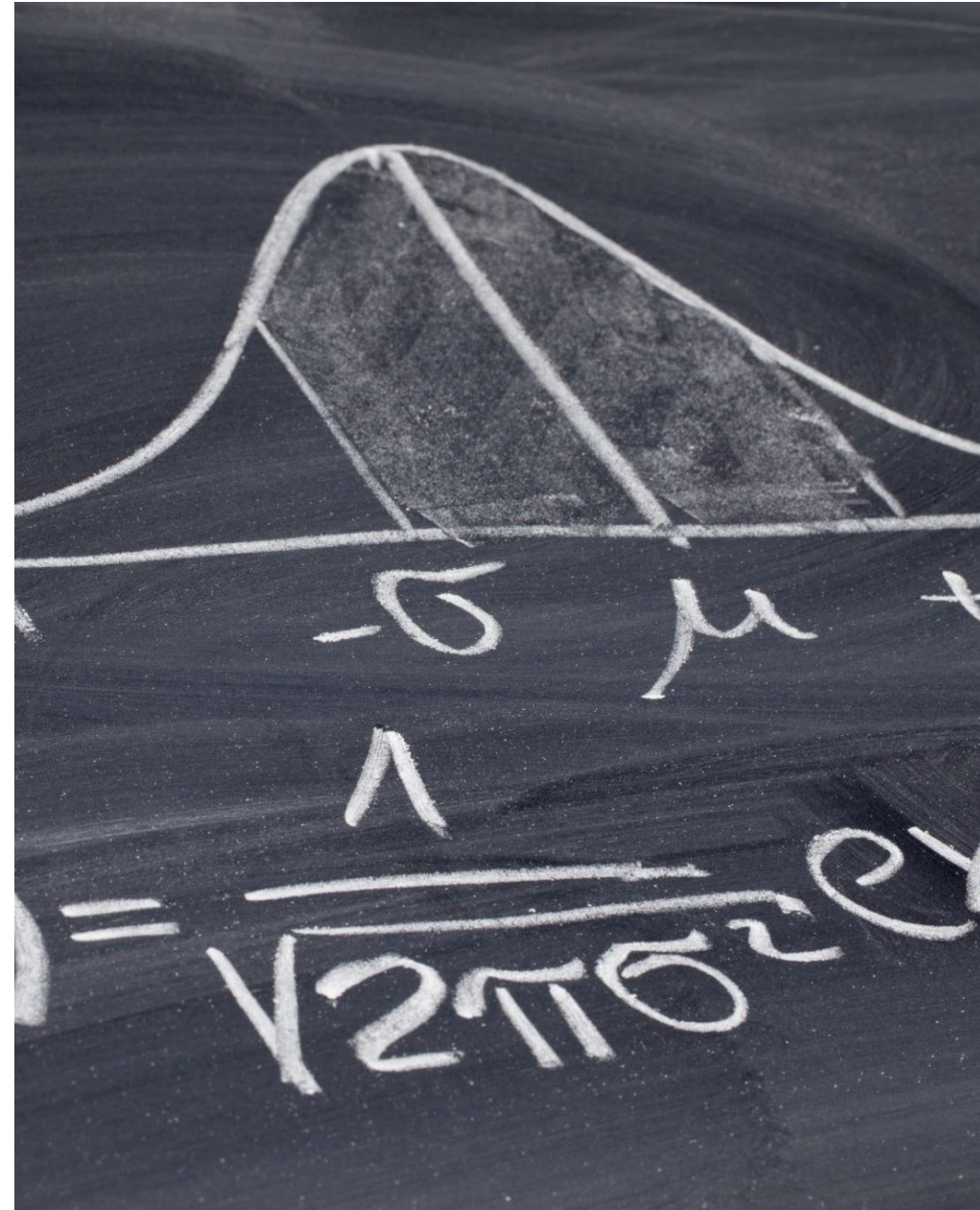
Estatística Descritiva

Conjunto de métodos que visam tornar os dados coletados mais fáceis de entender por meio de:

- Organização;
- Simplificação;
- Descrição e
- Apresentação de dados.

Usa tabelas, gráficos e medidas que resumem os dados brutos

- **Medidas de Tendência Central** ✓
 - Média
 - Mediana
 - Moda
- **Medidas de Dispersão** ✓
 - Amplitude
 - Variância
 - Desvio Padrão
 - Percentis e Quartis
- **Visualizações de Dados**
 - Histogramas
 - Gráficos de Barras
 - Gráficos de Pizza
 - Box Plots (Diagramas de Caixa)
 - Tabelas de Frequência



Visualizações de Dados

Representações gráficas de informações e dados. Ao transformar dados brutos em imagens compreensíveis, podemos identificar **padrões, tendências e insights** que seriam difíceis de perceber em tabelas ou listagens.

— Por que as Visualizações de Dados são Importantes?

- **Facilitam a Compreensão:** Nosso cérebro processa informações visuais muito mais rapidamente do que textuais ou numéricas.
- **Identificam Padrões e Tendências:** Visualizações podem revelar correlações, outliers, tendências de alta ou baixa, e outros padrões que podem passar despercebidos em dados brutos.
- **Suportam a Tomada de Decisão:** Ao apresentar insights de forma clara, as visualizações ajudam os tomadores de decisão a entender o cenário, identificar oportunidades e riscos, e fazer escolhas mais informadas.
- **Exploração de Dados:** As visualizações interativas permitem aos usuários explorar os dados por conta própria, fazer perguntas e descobrir novos insights.

Gráfico de Barras

Diagrama que usa barras retangulares para comparar valores. É uma forma visual de representar dados, sendo ideal **comparar valores entre diferentes categorias**.

Um gráfico de barras utiliza barras retangulares, onde:

- Um **eixo** (geralmente o horizontal) representa as categorias que estão sendo comparadas (ex: nomes de produtos, bairros, anos).
- O **outro eixo** (geralmente o vertical) representa a escala de valores da variável que está sendo medida (ex: número de vendas, população, pontuação).
- A **altura** (ou comprimento, se as barras forem horizontais) de cada barra é proporcional ao valor da categoria que ela representa.

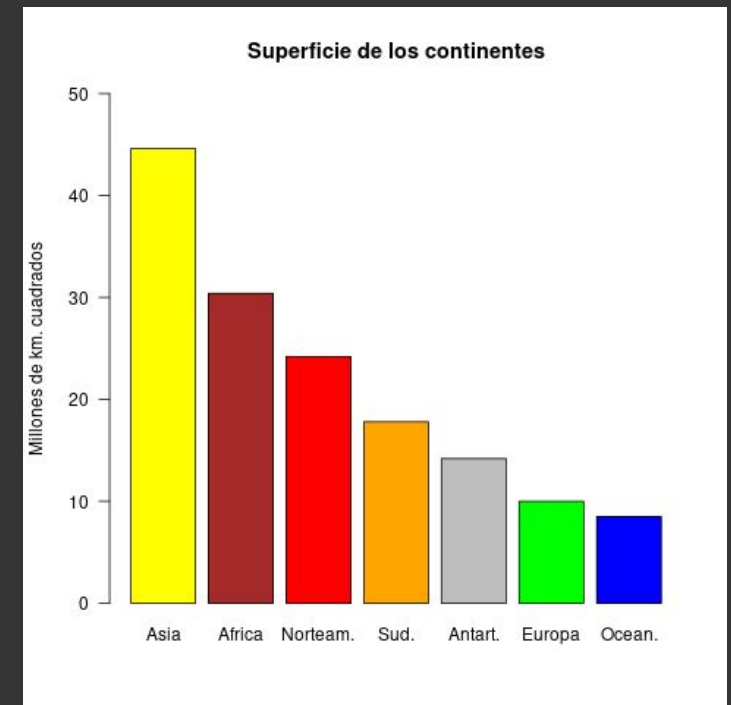


Gráfico de Barras

Vantagens:

- Facilidade de Compreensão
- Comparação Direta: É excelente para comparar valores exatos entre diferentes grupos ou itens.
- Leitura Precisa: Permite uma leitura relativamente precisa dos valores individuais de cada categoria
- Versatilidade: Pode ser usado para diversos tipos de dados categóricos (nominais ou ordinais) e para comparar contagens, médias, totais, ou qualquer outra métrica agregada.
- Impacto Visual: Gráficos de barras bem desenhados podem ser visualmente atraentes e eficazes para comunicar informações em apresentações, relatórios e infográficos.

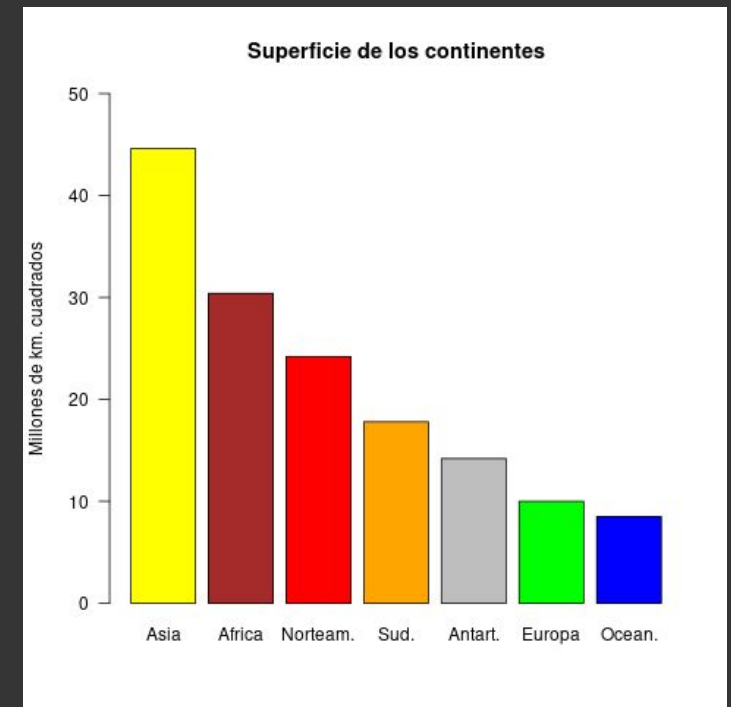
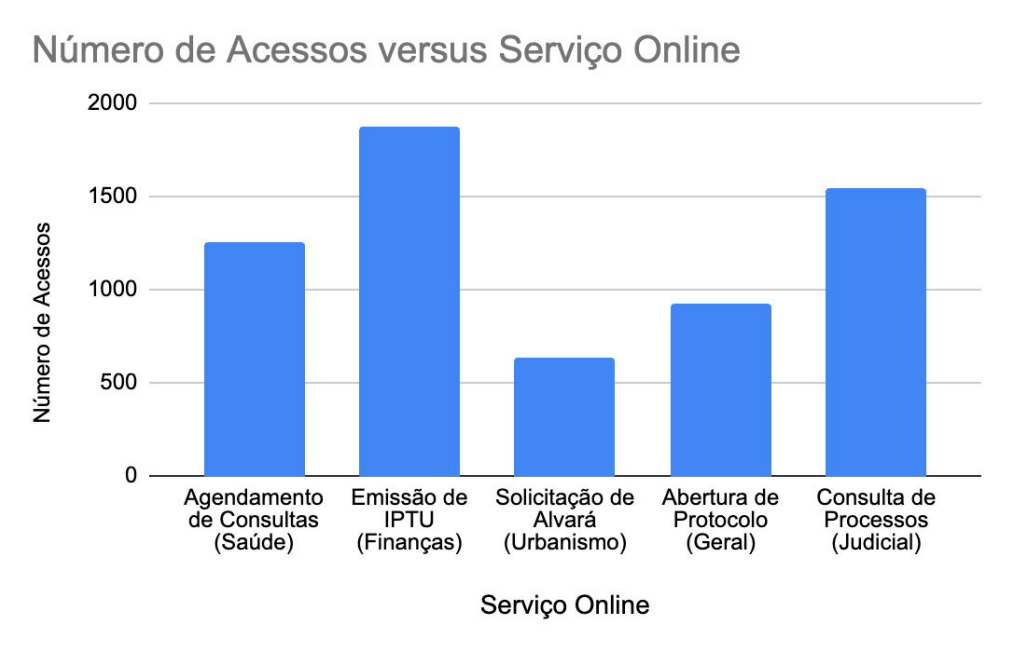


Gráfico de Barras - Exemplo

Imagine que a Prefeitura do Recife, através da iniciativa "Recife Digital", coletou dados sobre o número de acessos a diferentes serviços online oferecidos aos cidadãos durante o mês de Abril de 2025. Os dados coletados foram os seguintes:

Serviço Online	Número de Acessos
Agendamento de Consultas (Saúde)	1250
Emissão de IPTU (Finanças)	1875
Solicitação de Alvará (Urbanismo)	630
Abertura de Protocolo (Geral)	920
Consulta de Processos (Judicial)	1540

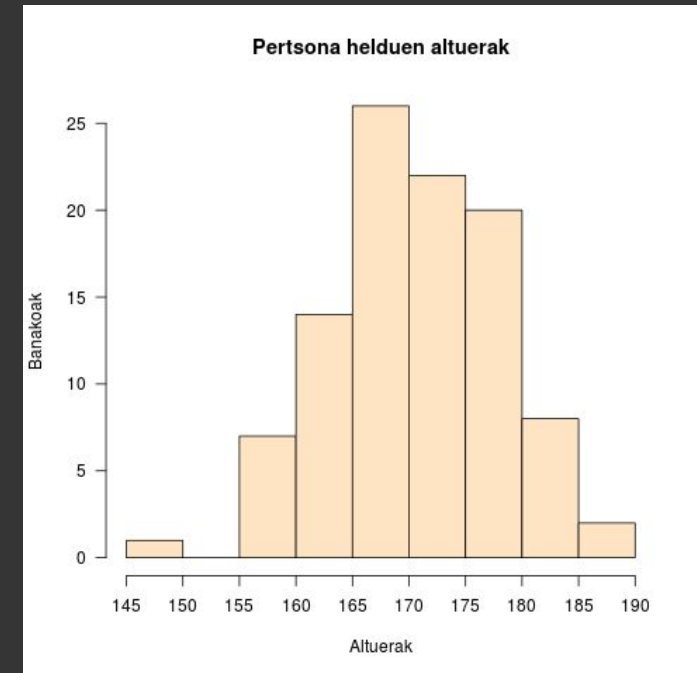


Histograma

É um tipo de gráfico de barras que nos permite visualizar a distribuição de uma única variável numérica contínua. Em vez de comparar categorias distintas, o histograma agrupa os dados em intervalos e exibe a frequência de observações que caem em cada intervalo

Um histograma utiliza:

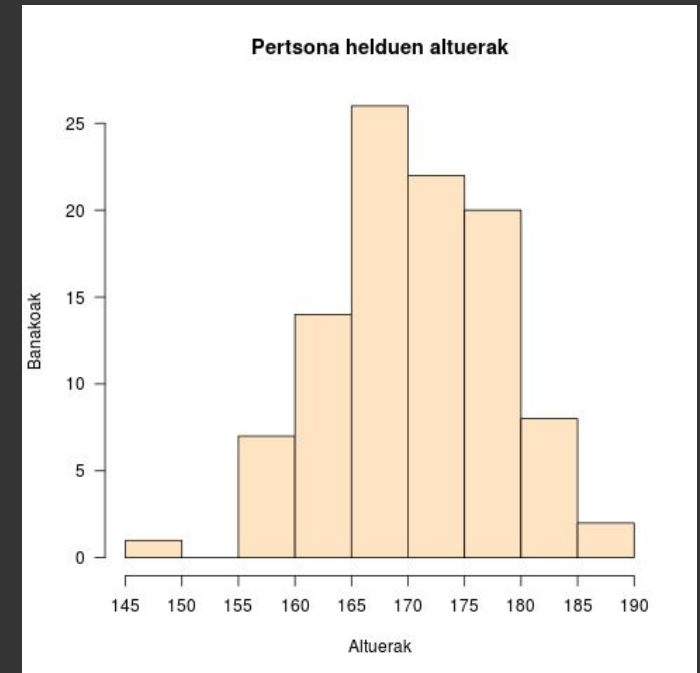
- **Eixo Horizontal (Eixo X):** Representa a variável numérica contínua que está sendo analisada. Este eixo é dividido em intervalos de igual largura.
- **Eixo Vertical (Eixo Y):** Representa a frequência (o número de observações) que caem em cada intervalo.
- **Barras:** Para cada intervalo no eixo horizontal, há uma barra vertical cuja altura é proporcional à frequência das observações dentro desse intervalo.



Histograma

Vantagens:

- **Visualização da Distribuição:** fornece uma representação visual da forma da distribuição dos dados.
- **Identificação de Tendências Centrais e Dispersão:** Embora não mostre diretamente a média ou mediana, podemos inferir sua localização aproximada observando o pico da distribuição.
- **Detecção de Outliers:** Valores extremos que caem em intervalos isolados nas extremidades do histograma podem ser identificados como potenciais outliers.



Histograma - Exemplo 1

Imagine que a Secretaria de Turismo e Lazer do Recife realizou uma pesquisa amostral durante o Carnaval de 2025 para entender a distribuição etária dos participantes. Os dados coletados sobre a idade de 100 participantes foram os seguintes (em anos):

[18, 22, 25, 30, 19, 28, 35, 21, 27, 40, 23, 32, 26, 29, 45, 20, 31, 24, 33, 38, 18, 26, 29, 36, 22, 30, 37, 25, 34, 42, 21, 28, 39, 23, 31, 27, 33, 41, 20, 30, 24, 35, 28, 32, 43, 19, 26, 34, 29, 37, 22, 30, 38, 25, 33, 40, 21, 27, 36, 23, 31, 29, 35, 44, 18, 26, 33, 28, 39, 24, 32, 30, 37, 42, 20, 25, 34, 27, 38, 21, 29, 36, 31, 40, 23, 33, 26, 35, 28, 39, 22, 30, 37, 41, 19, 27, 34, 32, 43]

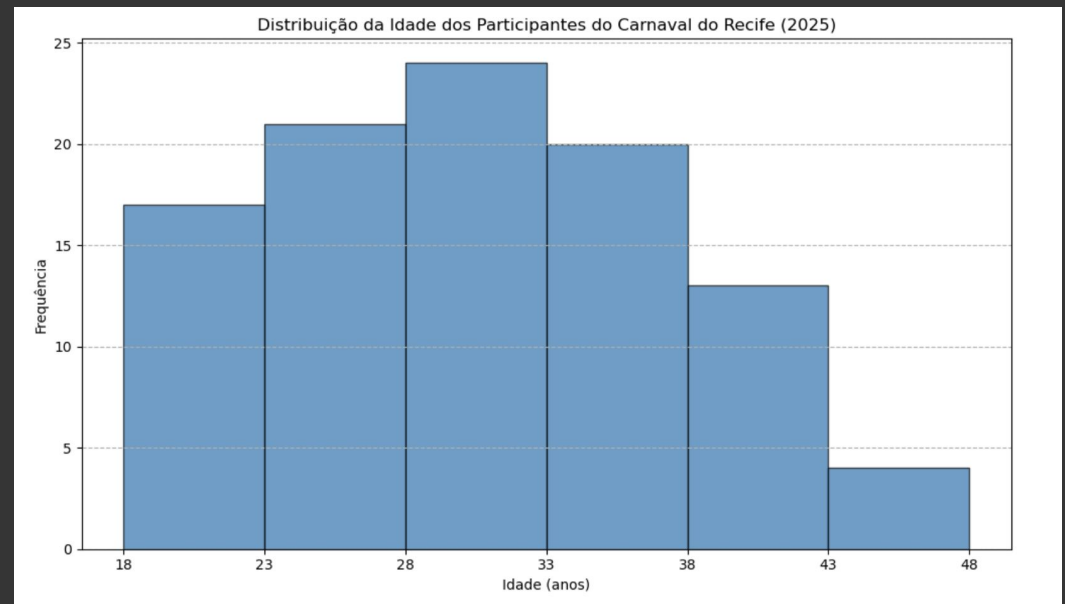


Gráfico de Linhas

Ferramenta essencial para visualizar a **evolução de uma ou mais variáveis ao longo de um período contínuo**. O gráfico de linhas é perfeito para revelar tendências, padrões e mudanças ao longo do tempo.

Um gráfico de linhas utiliza:

- **Eixo Horizontal:** Representa a variável contínua, geralmente o tempo (dias, semanas, meses, anos).
- **Eixo Vertical:** Representa a escala de valores da variável que está sendo rastreada
- **Pontos de Dados:** Cada ponto no gráfico representa um valor da variável em um ponto específico no tempo.
- **Linhas Conectando os Pontos:** Os pontos de dados são conectados por linhas retas, formando uma ou mais linhas que mostram a trajetória da variável ao longo do tempo.

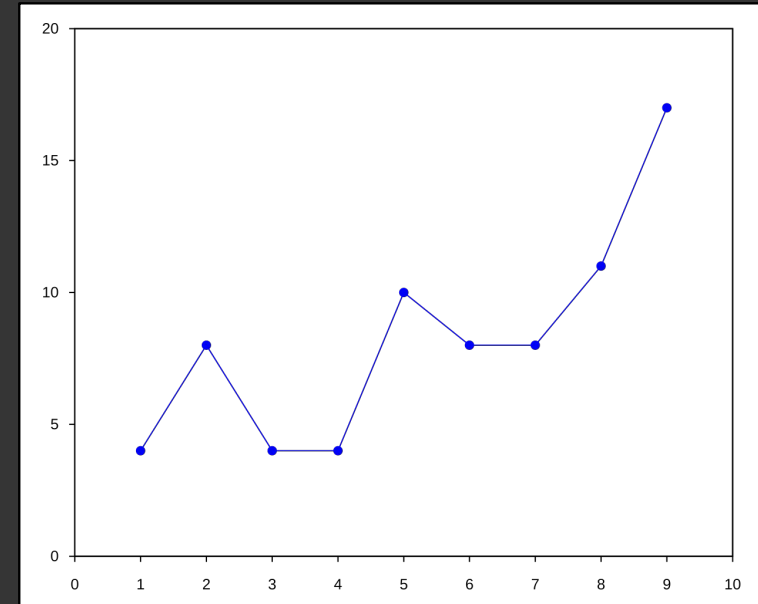


Gráfico de Linhas

Vantagens:

- Visualização de Tendências
- Continuidade: A linha que conecta os pontos enfatiza a natureza contínua da variável ao longo do tempo, sugerindo uma progressão ou fluxo.
- Comparação de Múltiplas Séries: Várias linhas podem ser plotadas no mesmo gráfico para comparar a evolução de diferentes variáveis ao longo do mesmo período
- Identificação de Outliers e Eventos: Picos ou quedas acentuadas nas linhas podem indicar eventos significativos ou valores atípicos nos dados.

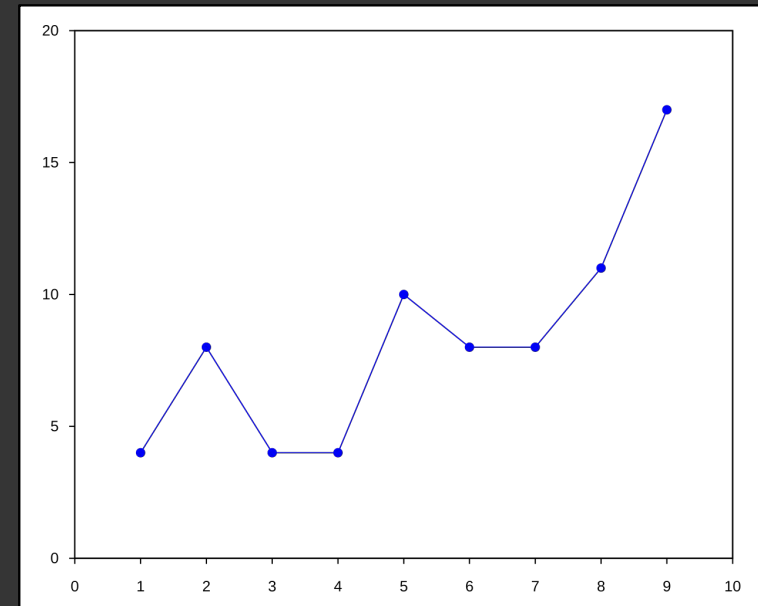
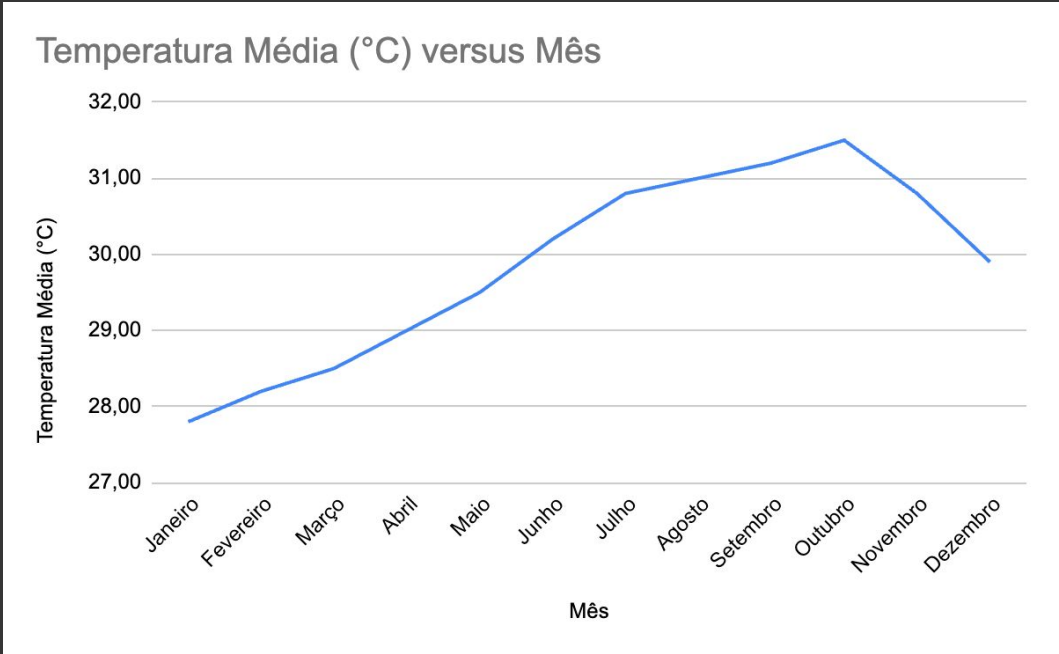


Gráfico de Linhas - Exemplo

Imagine que a Agência Pernambucana de Águas e Clima (APAC) coletou dados sobre a temperatura média mensal na cidade do Recife durante o ano de 2024. Os dados foram os seguintes



Mês	Temperatura Média (°C)
Janeiro	27.8
Fevereiro	28.2
Março	28.5
Abril	29.0
Maio	29.5
Junho	30.2
Julho	30.8
Agosto	31.0
Setembro	31.2
Outubro	31.5
Novembro	30.8
Dezembro	29.9

Gráfico de Dispersão (Scatter Plot)

O gráfico de dispersão, é uma ferramenta visual poderosa para investigar a relação entre duas variáveis numéricas. Em vez de mostrar a evolução ao longo do tempo ou comparar categorias, o gráfico de dispersão **revela se existe alguma correlação ou padrão na forma como duas variáveis se movem juntas.**

Um gráfico de dispersão utiliza:

- Eixo Horizontal (Eixo X): Representa uma das variáveis numéricas
- Eixo Vertical (Eixo Y): Representa a outra variável numérica
- Pontos: Cada ponto no gráfico representa uma única observação ou item do seu conjunto de dados. A posição do ponto é determinada pelos valores das duas variáveis para essa observação.

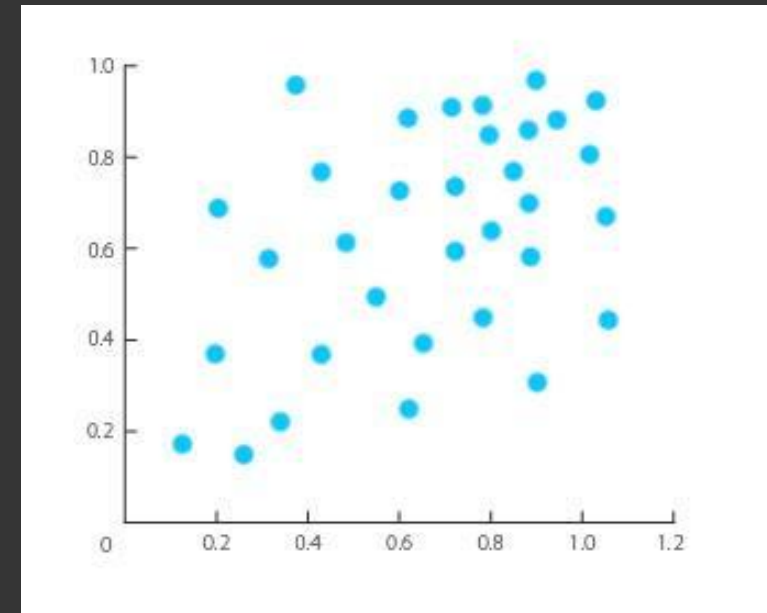


Gráfico de Dispersão (Scatter Plot)

Vantagens:

- **Visualização de Correlação:** permite observar se os pontos tendem a se agrupar ao longo de uma linha, formar uma curva ou se estão espalhados sem um padrão claro.
- **Identificação de Padrões:** Além da correlação linear, o gráfico pode revelar outros padrões, como agrupamentos (clusters) de pontos, que podem indicar diferentes subgrupos dentro dos seus dados.
- **Detecção de Outliers:** Pontos que estão significativamente distantes do padrão geral podem ser erros de coleta de dados ou observações incomuns que merecem investigação.

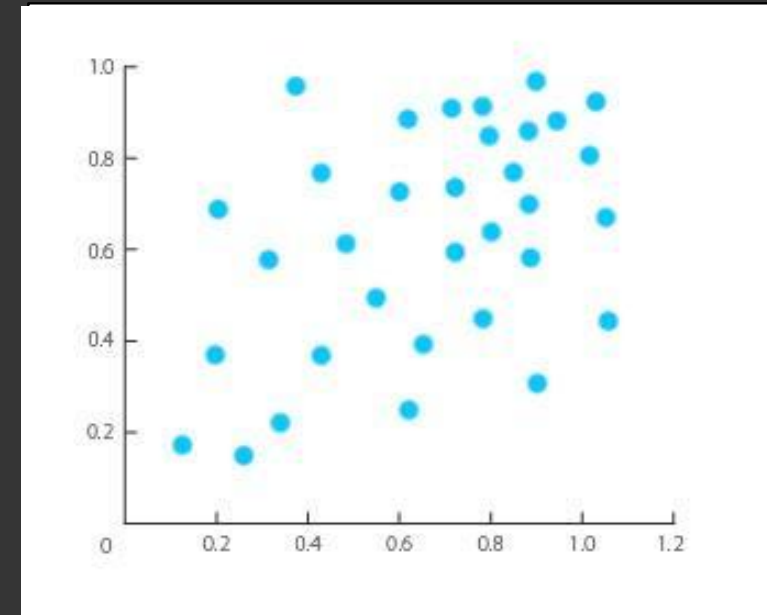
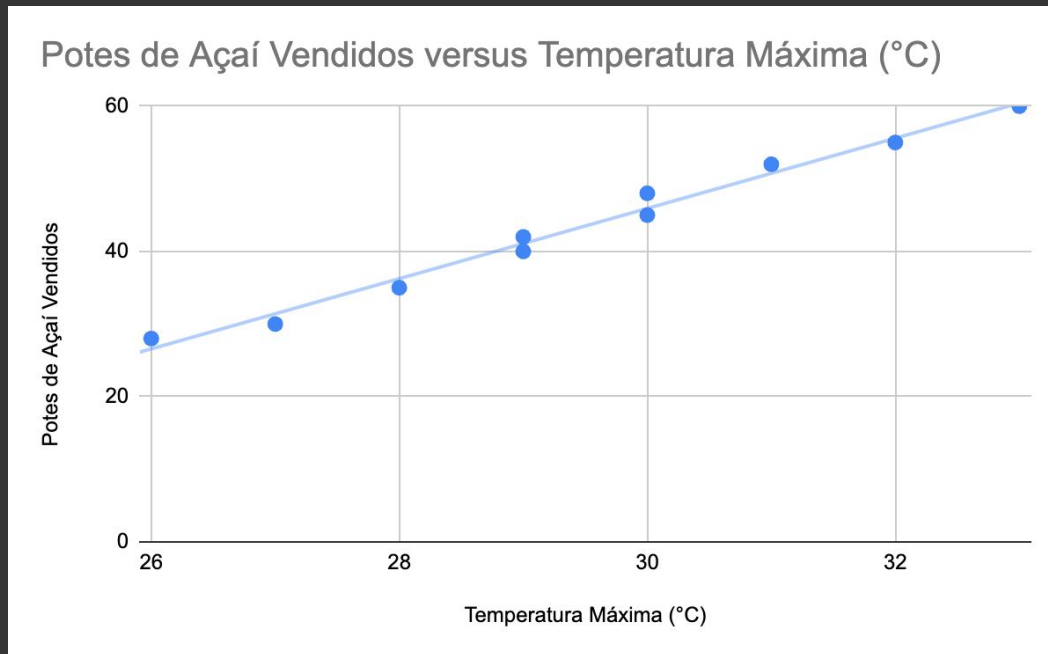


Gráfico de Dispersão - Exemplo 1

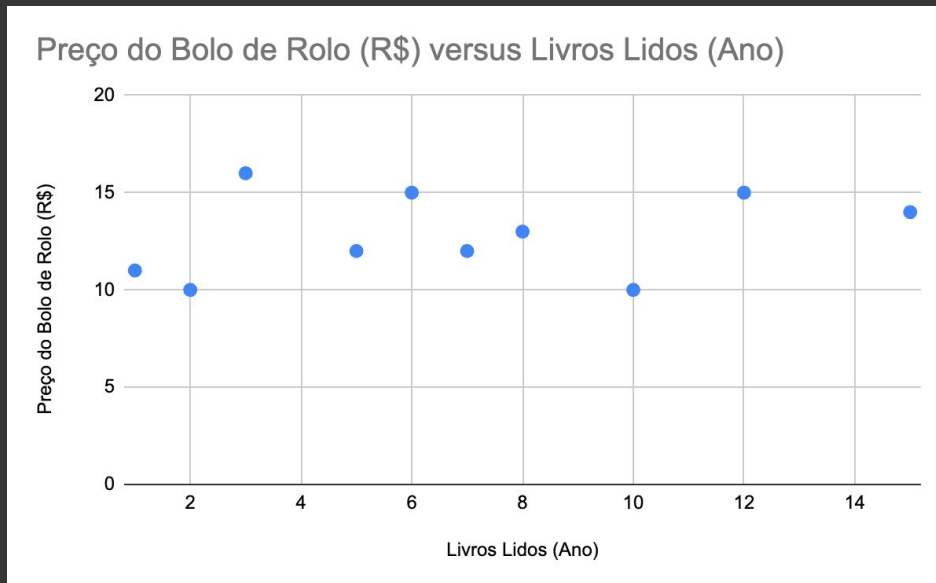
Imagine que um vendedor de açaí na Praia de Boa Viagem registrou diariamente a temperatura máxima (em graus Celsius) e o número de potes de açaí vendidos durante um mês de verão.



Temperatura Máxima (°C)	Potes de Açaí Vendidos
28	35
30	48
27	30
32	55
29	40
31	52
26	28
33	60
29	42
30	45

Gráfico de Dispersão - Exemplo 2

Imagine que um pesquisador está tentando encontrar alguma relação entre o número de livros que diferentes frequentadores de uma livraria leram no último ano e o preço do seu bolo de rolo favorito.



Frequentedor	Livros Lidos (Ano)	Preço do Bolo de Rolo (R\$)
João	5	12
Maria	12	15
Pedro	2	10
Ana	8	13
Carlos	1	11
Sofia	15	14
Lucas	7	12
Beatriz	3	16
Gabriel	10	10
Manuela	6	15

Gráfico de Pizza (Pie Chart)

O gráfico de pizza é um tipo de gráfico circular que divide um todo em fatias proporcionais para representar as partes desse todo. Cada fatia representa uma categoria, e o tamanho da fatia é proporcional à porcentagem ou proporção dessa categoria em relação ao total.

Um gráfico de pizza utiliza:

- Um círculo: Representa o todo
- Setores (fatias): O círculo é dividido em setores, cada um representando uma categoria diferente.
- Cores e Rótulos: Cada setor geralmente tem uma cor diferente e é rotulado com o nome da categoria.

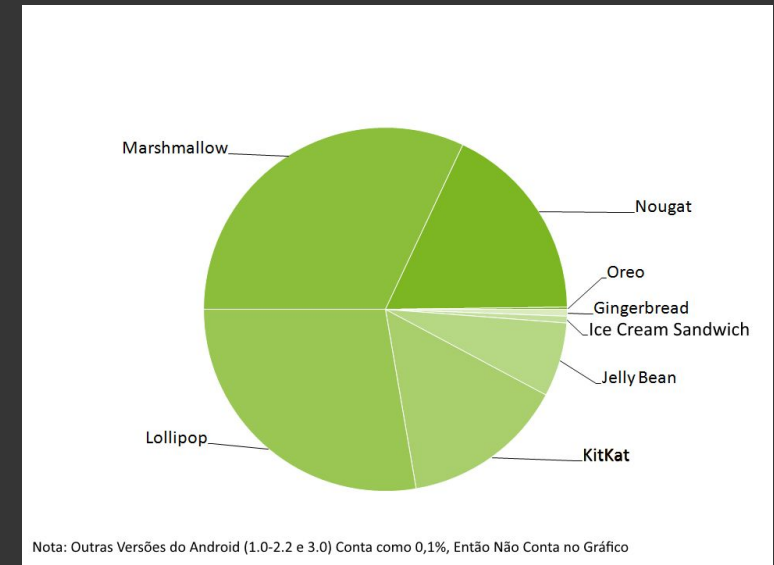


Gráfico de Pizza (Pie Chart)

Vantagens:

- **Visualização de Proporções:** A principal força do gráfico de pizza é mostrar a relação das partes com o todo de forma visualmente intuitiva
- **Simplicidade Conceitual:** O conceito de um círculo dividido em partes proporcionais é fácil de entender para um público amplo.
- **Destaque de Partes Dominantes:** Pode ser eficaz para destacar uma ou duas categorias que representam uma porção significativamente maior do todo.

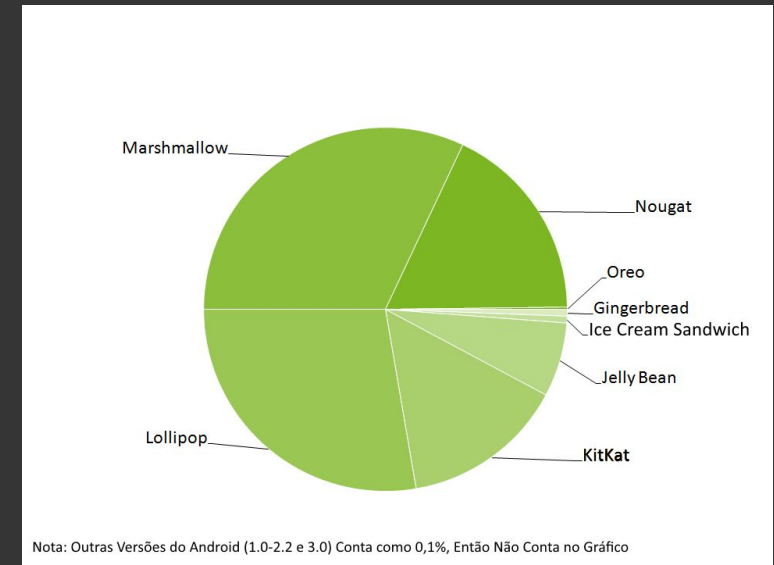


Gráfico de Pizza (Pie Chart)

Desvantagens do Gráfico de Pizza:

- Dificuldade em Comparar Tamanhos de Setores Similares
- Dificuldade com Muitas Categorias: Se houver muitas categorias, o gráfico de pizza fica cheio de fatias pequenas e difíceis de distinguir
- Dificuldade em Ler Valores Exatos: A menos que as porcentagens estejam claramente rotuladas, é difícil determinar os valores exatos representados por cada fatia.

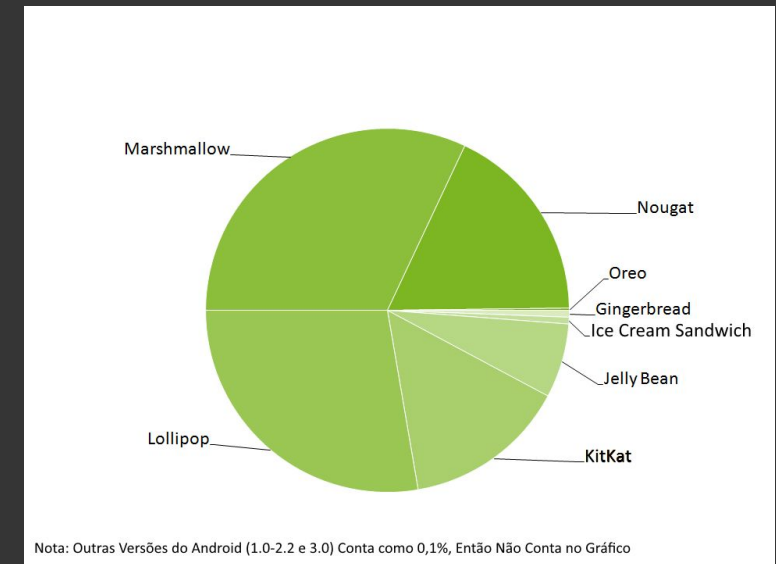
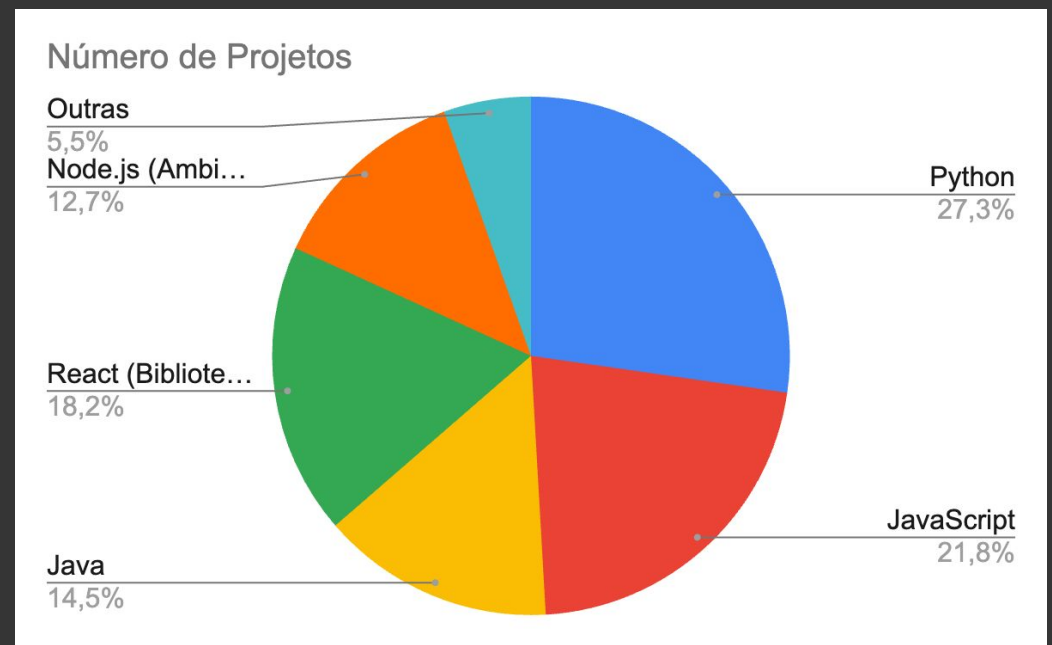


Gráfico de Pizza - Exemplo 1

Imagine uma startup de tecnologia localizada no Porto Digital do Recife que analisou a distribuição das linguagens de programação utilizadas em seus projetos ativos. Os dados levantados foram os seguintes:

Linguagem de Programação	Número de Projetos
Python	15
JavaScript	12
Java	8
React (Biblioteca JS)	10
Node.js (Ambiente JS)	7
Outras	3



Biblioteca - Matplotlib



É a biblioteca fundamental para a criação de gráficos e visualizações de dados em Python.
É a base a qual muitas outras bibliotecas de visualização mais sofisticadas são construídas.

Vantagens:

- Altamente Personalizável: Quase todos os aspectos de um gráfico Matplotlib podem ser personalizados, desde cores, estilos de linha, marcadores, rótulos, títulos, legendas, anotações.
- Controle Granular: Você tem controle direto sobre os elementos de baixo nível do gráfico, como os objetos Figure (a tela ou janela do gráfico) e Axes (a área onde os dados são plotados).
- Integração com NumPy e Pandas: O Matplotlib funciona perfeitamente com as estruturas de dados do NumPy (arrays) e do Pandas (Series e DataFrames), tornando a visualização de dados científicos e tabulares muito conveniente.

Biblioteca - Matplotlib



Pyplot

- submódulo da biblioteca Matplotlib
- fazem o Matplotlib funcionar de uma forma semelhante ao MATLAB

https://www.w3schools.com/python/matplotlib_intro.asp

Biblioteca - Matplotlib



Instalação `pip install matplotlib`

Importação `import matplotlib.pyplot as plt`

Biblioteca - Matplotlib



Criando um gráfico de linha

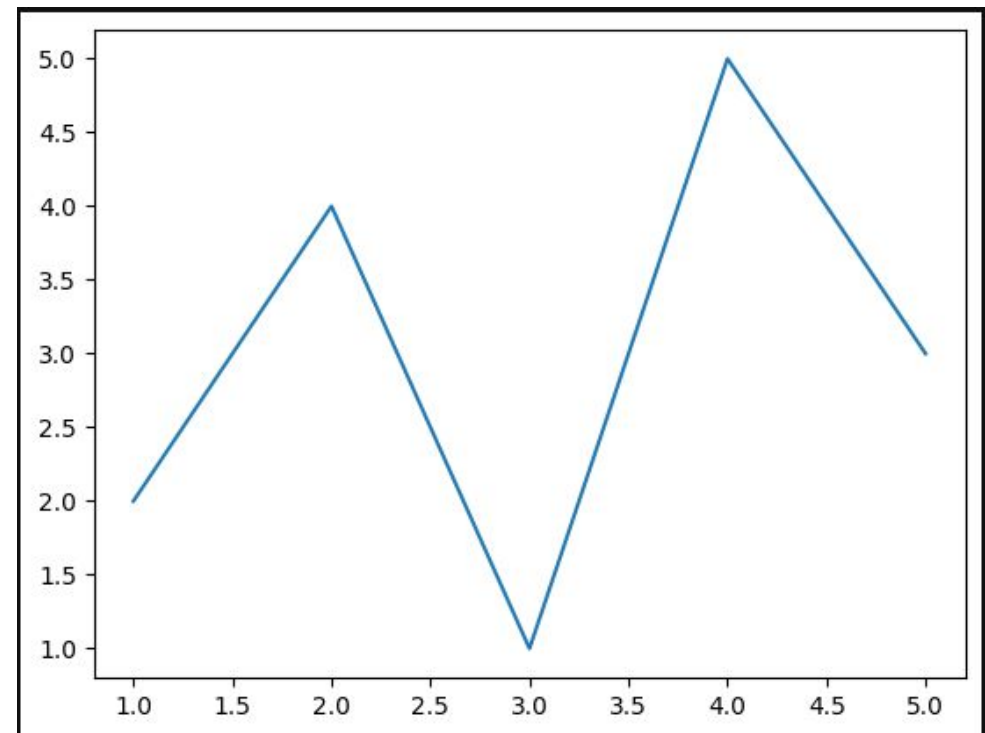
Dados

```
x = [1, 2, 3, 4, 5]
```

```
y = [2, 4, 1, 5, 3]
```

```
plt.plot(x, y)
```

```
plt.show()
```



Biblioteca - Matplotlib



criando um gráfico de barras

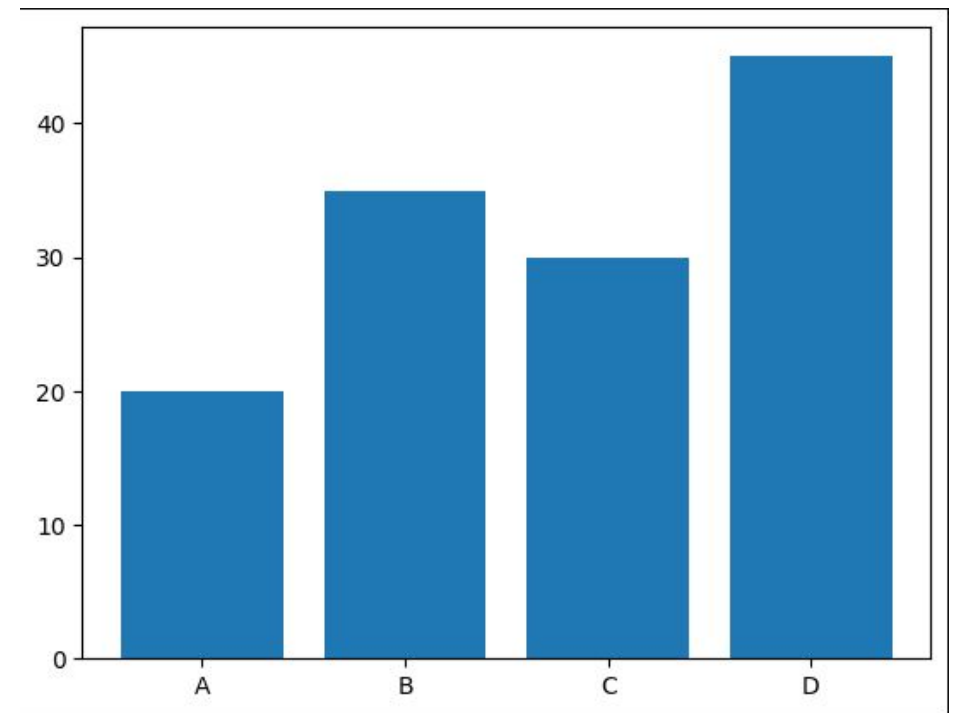
Dados

```
categorias = ['A', 'B', 'C', 'D']
```

```
valores = [20, 35, 30, 45]
```

```
plt.bar(categorias, valores)
```

```
plt.show()
```



Biblioteca - Matplotlib

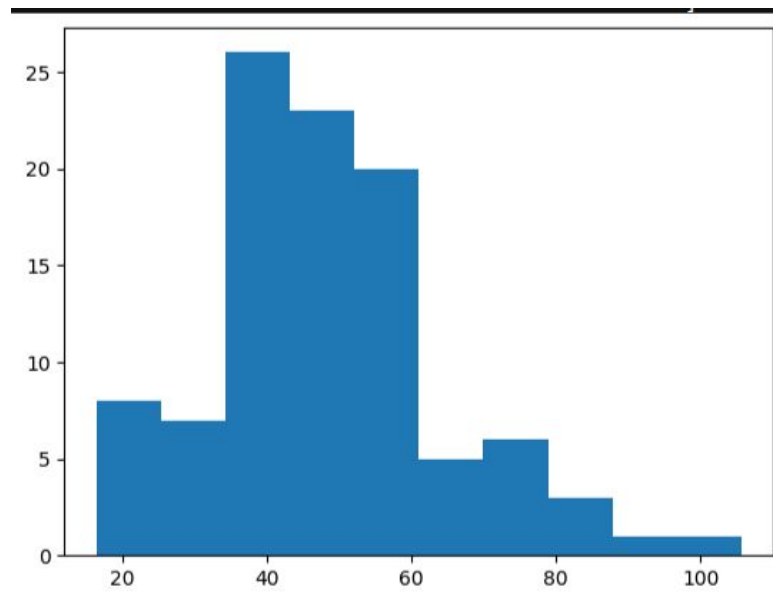


criando um histograma

```
dados_histograma = np.random.normal(loc=50, scale=15, size=100)
```

```
plt.hist(dados_histograma, bins=10)
```

```
plt.show()
```



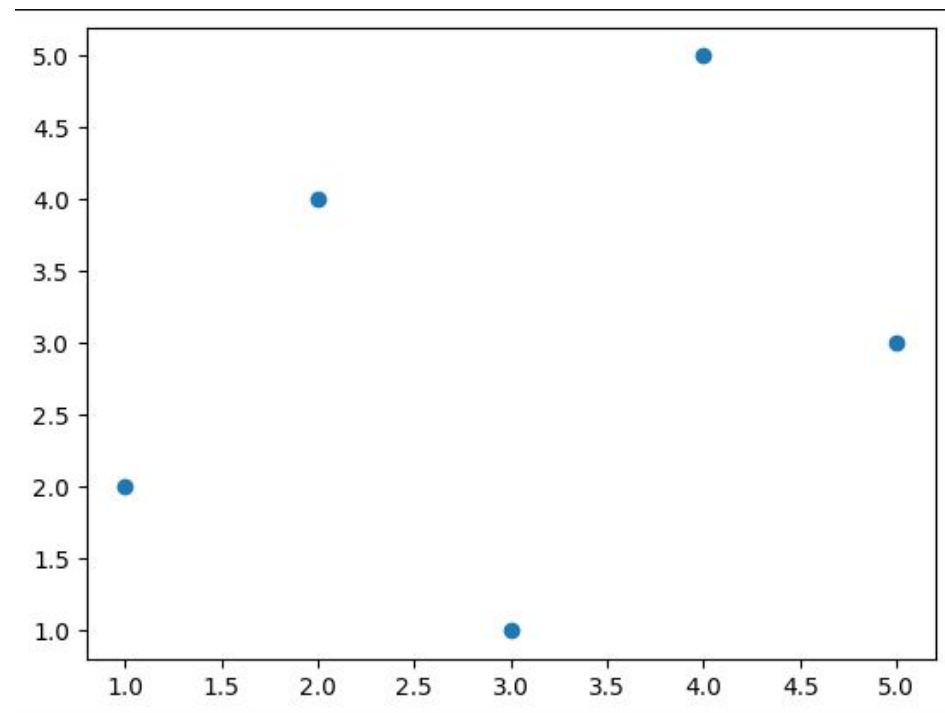
Biblioteca - Matplotlib



criando um gráfico de dispersão

```
plt.scatter(x, y)
```

```
plt.show()
```

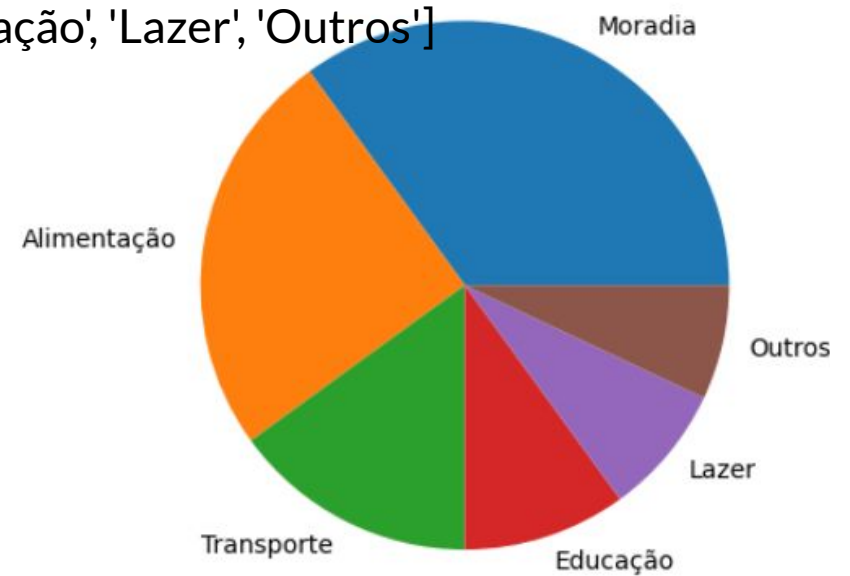


Biblioteca - Matplotlib



criando um gráfico de pizza

```
import matplotlib.pyplot as plt  
#dados  
categorias = ['Moradia', 'Alimentação', 'Transporte', 'Educação', 'Lazer', 'Outros']  
porcentagens = [35, 25, 15, 10, 8, 7]  
  
plt.pie(porcentagens, labels=categorias)  
plt.show()
```



Biblioteca - Matplotlib



Tipo de Gráfico	Comando Principal (plt.)	Argumentos Essenciais	Personalizações Comuns
Linhas	<code>plot(x, y)</code>	x (dados do eixo x), y (dados do eixo y)	marker (estilo dos pontos), linestyle (estilo da linha), color, label, xlabel, ylabel, title, legend(), grid(True)
Barras (Verticais)	<code>bar(x, height)</code>	x (categorias), height (alturas das barras)	color, edgecolor, alpha (transparência), label, xlabel, ylabel, title, legend()
Barras (Horizontais)	<code>barh(y, width)</code>	y (categorias), width (larguras das barras)	color, edgecolor, alpha, label, xlabel, ylabel, title, legend()
Dispersão	<code>scatter(x, y)</code>	x (coordenadas x), y (coordenadas y)	s (tamanho dos pontos), c (cores dos pontos), marker (formato dos pontos), alpha, label, xlabel, ylabel, title, legend()
Histograma	<code>hist(x)</code>	x (dados numéricos)	bins (número de barras), color, edgecolor, alpha, density (normalizar), xlabel, ylabel, title
Pizza	<code>pie(sizes, labels)</code>	sizes (tamanhos das fatias), labels (rótulos)	colors, autopct (formato das porcentagens), startangle (ângulo inicial), shadow (sombra), explode (afastar fatias), title, axis('equal') (manter proporção circular)

Exemplo 1 - Matplotlib



Você é um analista de dados trabalhando com informações do IBGE sobre a taxa de desemprego na Região Metropolitana do Recife (RMR) ao longo dos anos de 2023 e 2024. Os dados trimestrais coletados foram os seguintes:

Trimestre	Ano	Taxa de Desemprego (%)
1	2023	14.2
2	2023	13.8
3	2023	13.5
4	2023	13.2
1	2024	12.9
2	2024	12.5
3	2024	12.2
4	2024	12.0

Seu objetivo é criar um gráfico de linhas informativo e visualmente agradável utilizando Matplotlib para apresentar a evolução da taxa de desemprego na RMR durante esse período

Exemplo 1 - Matplotlib



```
import numpy as np
import matplotlib.pyplot as plt

trimestres = np.array(['01/23', '02/23', '03/23', '04/23', '01/24', '02/24', '03/24', '04/24'])
taxa_desemprego = np.array([14.2, 13.8, 13.5, 13.2, 12.9, 12.5, 12.2, 12.0])

# Criando o gráfico de linhas
plt.plot(trimestres, taxa_desemprego, marker='o', linestyle='--', color='blue')

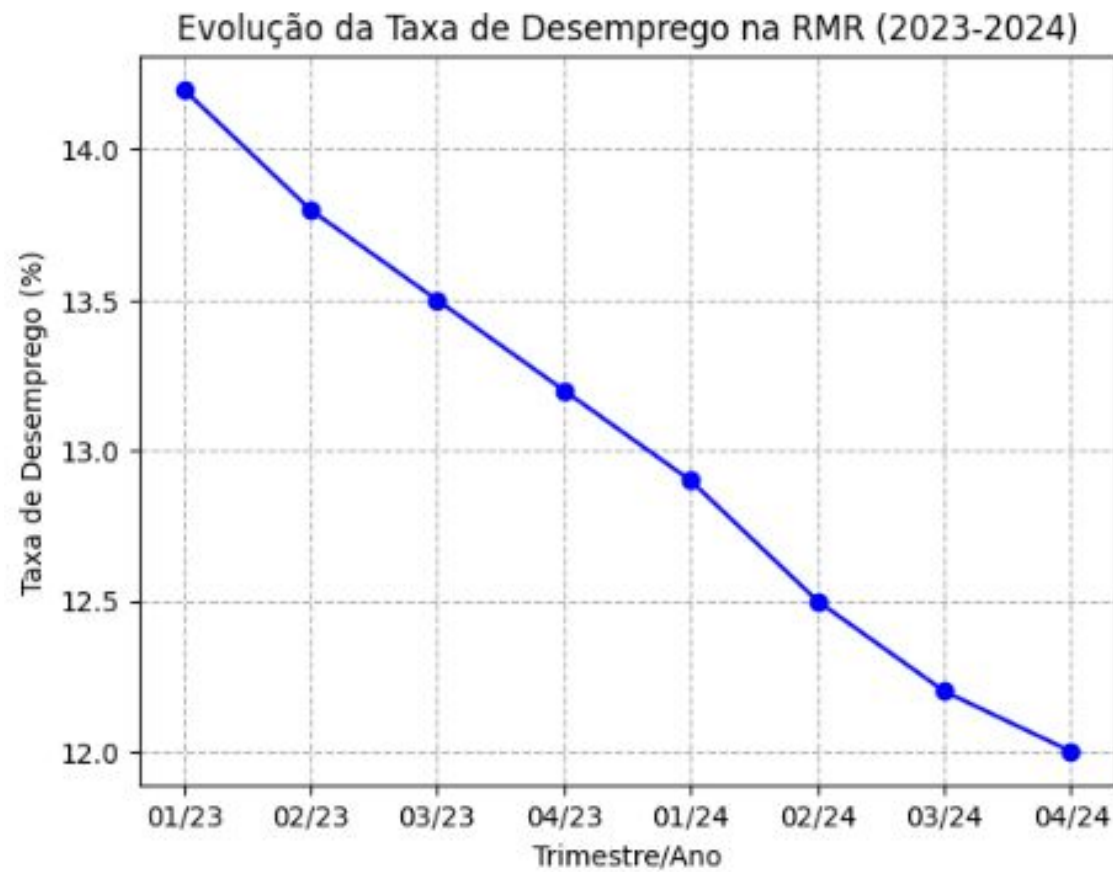
# Adicionando rótulos e título
plt.xlabel('Trimestre/Ano')
plt.ylabel('Taxa de Desemprego (%)')
plt.title('Evolução da Taxa de Desemprego na RMR (2023-2024)')

# Adicionando grade
plt.grid(True, linestyle='--')

# Salvando o gráfico (opcional)
plt.savefig('taxa_desemprego_rmr.png')

# Exibindo o gráfico
plt.show()
```


Exemplo 1 - Matplotlib



Exemplo 2 - Matplotlib



Você é responsável por analisar os dados de vendas de diferentes categorias de artesanato no famoso Mercado de São José, no Recife, durante o mês de Agosto de 2024.

Categoria de Artesanato	Total de Vendas (R\$)
Cerâmica	12500
Madeira	9800
Renda	7200
Pintura em Tela	11000
Esculturas em Argila	6500
Tecidos e Bordados	14000

Seu objetivo é criar um gráfico de barras vertical atraente e informativo utilizando Matplotlib para comparar o total de vendas por categoria de artesanato.

Exemplo 2 - Matplotlib



```
import matplotlib.pyplot as plt
import numpy as np

# Dados de vendas de artesanato
categorias = np.array(['Cerâmica', 'Madeira', 'Renda', 'Pintura em Tela', 'Esculturas em Argila', 'Tecidos e Bordados'])
vendas = np.array([12500, 9800, 7200, 11000, 6500, 14000])
cores = ['pink', '#dd8452', '#55a868', '#c44e52', '#8172b2', '#64b5cd'] # Cores diferentes para cada barra

# Criando o gráfico de barras
plt.bar(categorias, vendas, color=cores, edgecolor='black', label='Vendas (R$)')

# Adicionando rótulos e título
plt.xlabel('Categoria de Artesanato')
plt.ylabel('Total de Vendas (R$)')
plt.title('Total de Vendas de Artesanato por Categoria - Mercado de São José (Agosto/2024)')

# Adicionando grade no eixo y
plt.grid(axis='y', linestyle='--')

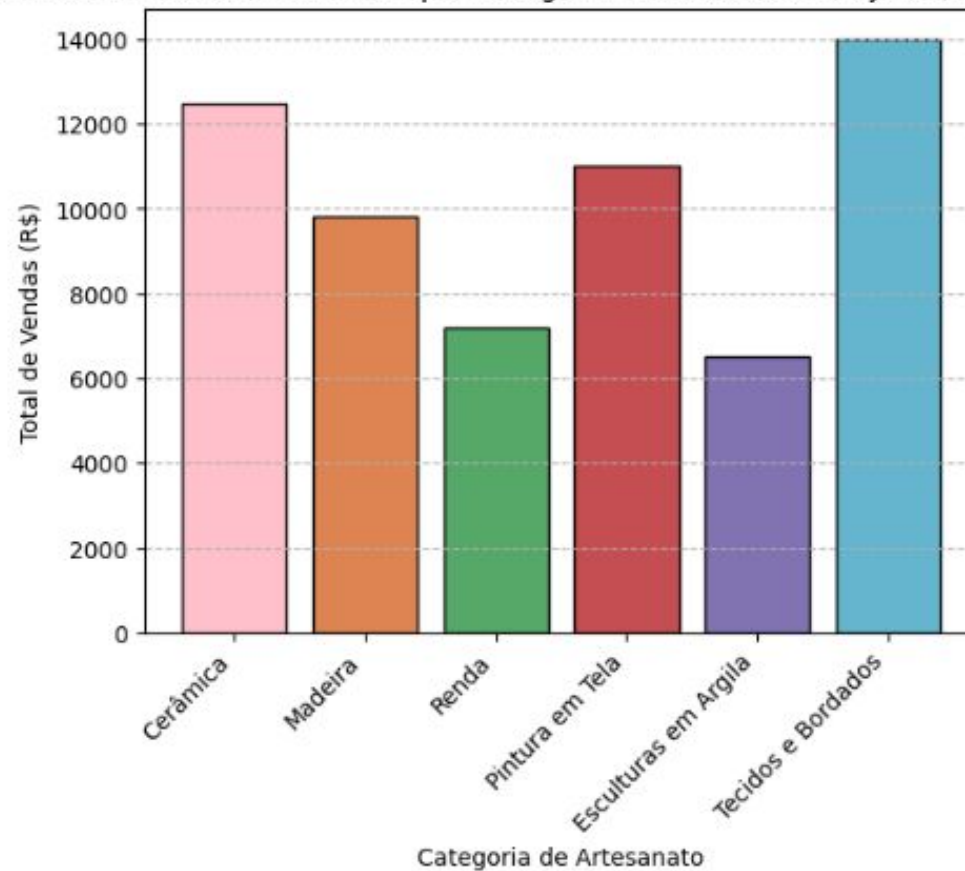
# rotaciona os rótulos das categorias no eixo x em 45 graus e os alinha à direita
plt.xticks(rotation=45, ha='right')

# Exibindo o gráfico e salvando
plt.show()
plt.savefig('vendas_artesanato_sao_jose.png')
```

Exemplo 2 - Matplotlib



Total de Vendas de Artesanato por Categoria - Mercado de São José (Agosto/2024)



Exemplo 3 - Matplotlib



Você é um urbanista analisando as características dos edifícios no bairro do Pina, um dos bairros com maior verticalização do Recife. Você coletou dados da altura (em metros) de uma amostra de 100 edifícios na região:

```
[75, 82, 90, 68, 78, 85, 95, 72, 80, 88, 70, 83, 92, 65,  
77, 86, 98, 75, 81, 89, 73, 84, 91, 67, 79, 87, 96, 71, 82,  
90, 69, 78, 85, 94, 76, 83, 93, 66, 77, 86, 97, 74, 80, 88,  
72, 81, 91, 68, 79, 87, 99, 70, 84, 92, 65, 76, 85, 95, 73,  
82, 90, 67, 78, 86, 98, 71, 83, 91, 69, 77, 87, 96, 74, 81,  
89, 66, 79, 88, 93, 72, 80, 92, 68, 76, 85, 97, 70, 82, 90,  
65, 78, 86, 99, 73, 84, 91, 67, 77, 87, 95]
```

Seu objetivo é criar um histograma utilizando Matplotlib para visualizar a distribuição da altura desses edifícios no bairro do Pina.

Exemplo 3 - Matplotlib



```
import matplotlib.pyplot as plt
import numpy as np

# Dados da altura dos edifícios no Pina
alturas = np.array([75, 82, 90, 68, 78, 85, 95, 72, 80, 88, 70, 83, 92, 65, 77, 86, 98, 75, 81, 89,
                    73, 84, 91, 67, 79, 87, 96, 71, 82, 90, 69, 78, 85, 94, 76, 83, 93, 66, 77, 86,
                    97, 74, 80, 88, 72, 81, 91, 68, 79, 87, 99, 70, 84, 92, 65, 76, 85, 95, 73, 82,
                    90, 67, 78, 86, 98, 71, 83, 91, 69, 77, 87, 96, 74, 81, 89, 66, 79, 88, 93, 72,
                    80, 92, 68, 76, 85, 97, 70, 82, 90, 65, 78, 86, 99, 73, 84, 91, 67, 77, 87, 95])

# Criando o histograma
plt.hist(alturas, bins=12, color='skyblue', edgecolor='black')

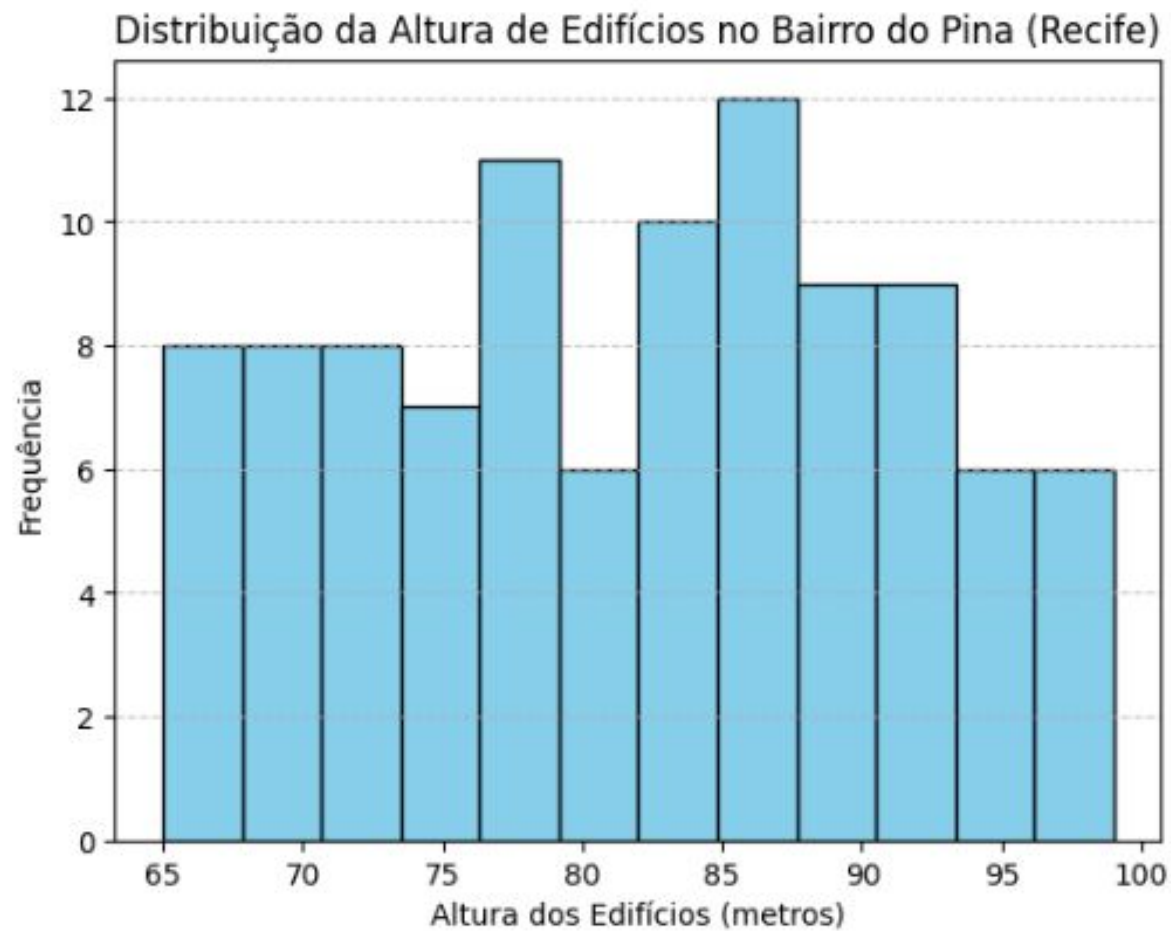
# Adicionando rótulos e título
plt.xlabel('Altura dos Edifícios (metros)')
plt.ylabel('Frequência')
plt.title('Distribuição da Altura de Edifícios no Bairro do Pina (Recife)')

# Adicionando grade no eixo y
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Exibindo o gráfico
plt.show()

# Salvando o gráfico (opcional)
plt.savefig('histograma_altura_pina.png')
```

Exemplo 3 - Matplotlib



Exemplo 4 - Matplotlib



Uma nutricionista de uma escola municipal no Recife realizou uma pesquisa com os alunos para entender quais frutas são mais consumidas durante o lanche escolar. Os resultados da pesquisa com uma amostra de alunos foram os seguintes:

Fruta	Número de Alunos que Consomem
Banana	85
Maçã	60
Melancia	45
Laranja	50
Manga	30
Outras Frutas	20

Seu objetivo é criar um gráfico de pizza utilizando Matplotlib para visualizar a distribuição do consumo de frutas entre os alunos da escola.

Exemplo 4 - Matplotlib



```
import matplotlib.pyplot as plt
import numpy as np

# Dados do consumo de frutas
frutas = np.array(['Banana', 'Maçã', 'Melancia', 'Laranja', 'Manga', 'Outras Frutas'])
consumo = np.array([85, 60, 45, 50, 30, 20])
cores = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99', '#c2c2f0', '#ffb3e6']

# Criando o gráfico de pizza
plt.pie(consumo, labels=frutas, colors=cores, shadow=True)

# Adicionando um título
plt.title('Distribuição do Consumo de Frutas no Lanche Escolar')

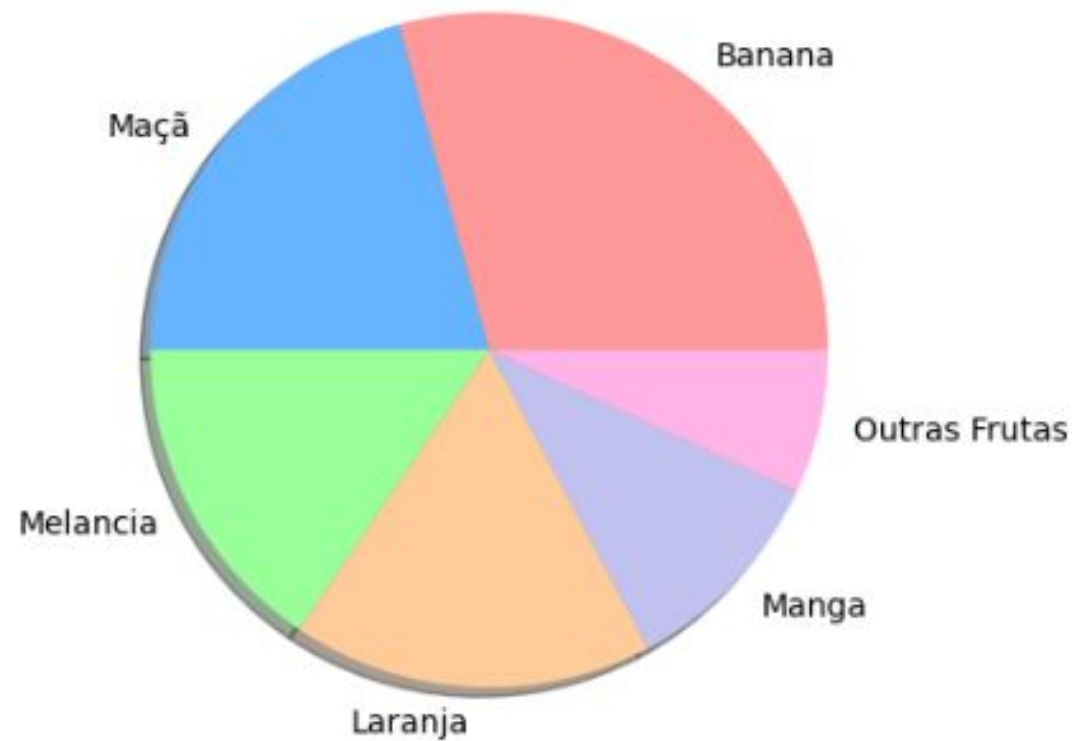
# Exibindo o gráfico
plt.show()

# Salvando o gráfico (opcional)
plt.savefig('distribuicao_frutas_escola.png')
```


Exemplo 4 - Matplotlib



Distribuição do Consumo de Frutas no Lanche Escolar



Exemplo 5 - Matplotlib

Você é um pequeno empreendedor que vende sorvetes em um quiosque na Praia de Boa Viagem, no Recife. Você suspeita que a temperatura do ar tem um impacto significativo nas suas vendas. Para investigar essa possível relação, você coletou dados diários durante um período do verão de 2025, registrando a temperatura máxima do dia (em graus Celsius) e o número de sorvetes vendidos. Os dados coletados foram os seguintes:

Seu objetivo é criar um gráfico de dispersão utilizando Matplotlib para visualizar a relação entre a temperatura máxima e o número de sorvetes vendidos na Praia de Boa Viagem.

Dia	Temperatura Máxima (°C)	Sorvetes Vendidos
1	30	120
2	32	150
3	28	100
4	33	160
5	29	115
6	31	135
7	27	90
8	34	170
9	30	125
10	32	145
11	28	105
12	33	165
13	31	140
14	29	110
15	34	175

Exemplo 5 - Matplotlib



```
import matplotlib.pyplot as plt
import numpy as np

# Dados de temperatura e vendas de sorvete
temperaturas = np.array([30, 32, 28, 33, 29, 31, 27, 34, 30, 32, 28, 33, 31, 29, 34])
vendas = np.array([120, 150, 100, 160, 115, 135, 90, 170, 125, 145, 105, 165, 140, 110, 175])

# Criando o gráfico de dispersão
plt.scatter(temperaturas, vendas, color='coral', label='Vendas Diárias')

# Adicionando rótulos e título
plt.xlabel('Temperatura Máxima (°C)')
plt.ylabel('Número de Sorvetes Vendidos')
plt.title('Relação entre Temperatura e Vendas de Sorvete - Praia de Boa Viagem (Verão 2025)')

# Adicionando grade (opcional)
plt.grid(True, linestyle='--')

# Adicionando legenda
plt.legend()

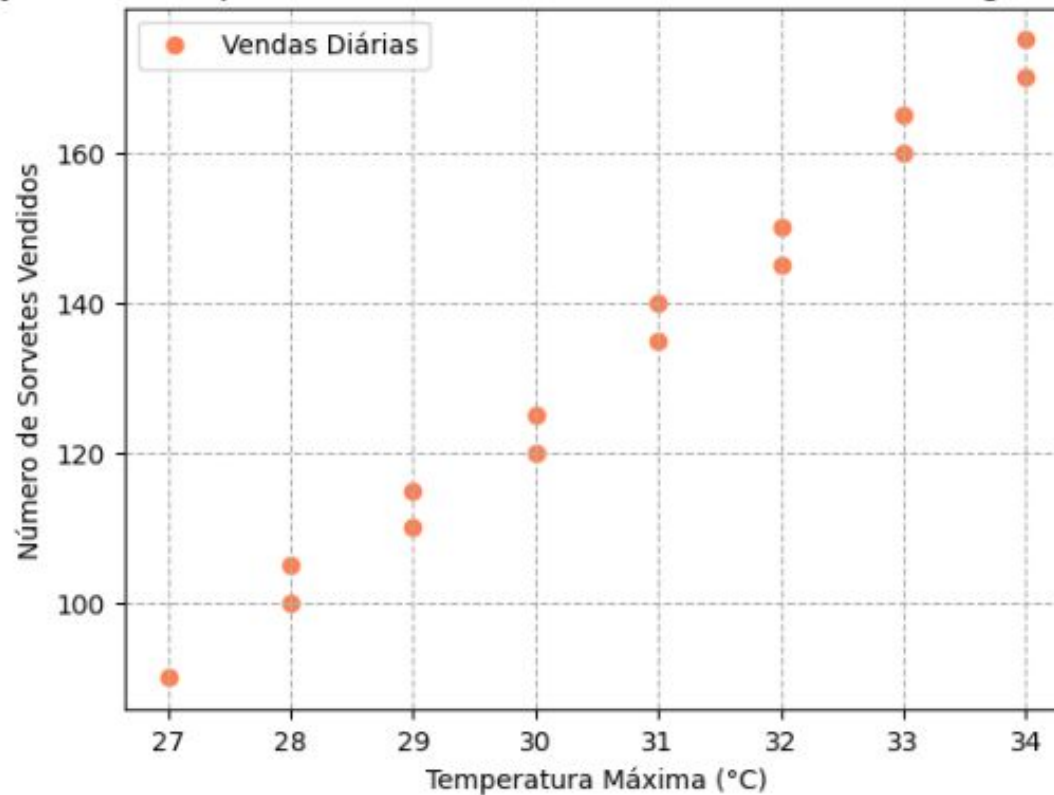
# Exibindo o gráfico
plt.show()

# Salvando o gráfico (opcional)
plt.savefig('dispersao_temperatura_sorvete.png')
```

Exemplo 5 - Matplotlib



Relação entre Temperatura e Vendas de Sorvete - Praia de Boa Viagem (Verão 2025)



Outliers

Em termos simples, outliers são pontos de dados que se desviam significativamente dos outros dados em um conjunto. Eles são valores que parecem estar "fora da curva", sendo incomumente grandes ou pequenos em comparação com o restante das observações.

—

Eles podem surgir por diversas razões, como:

- **Erros de medição ou coleta de dados:** Alguém pode ter digitado um valor incorretamente.
- **Variabilidade natural extrema:** Em alguns casos, os outliers podem ser valores reais, mas representam eventos raros ou extremos.

—
É importante identificar e analisar outliers, pois eles podem:

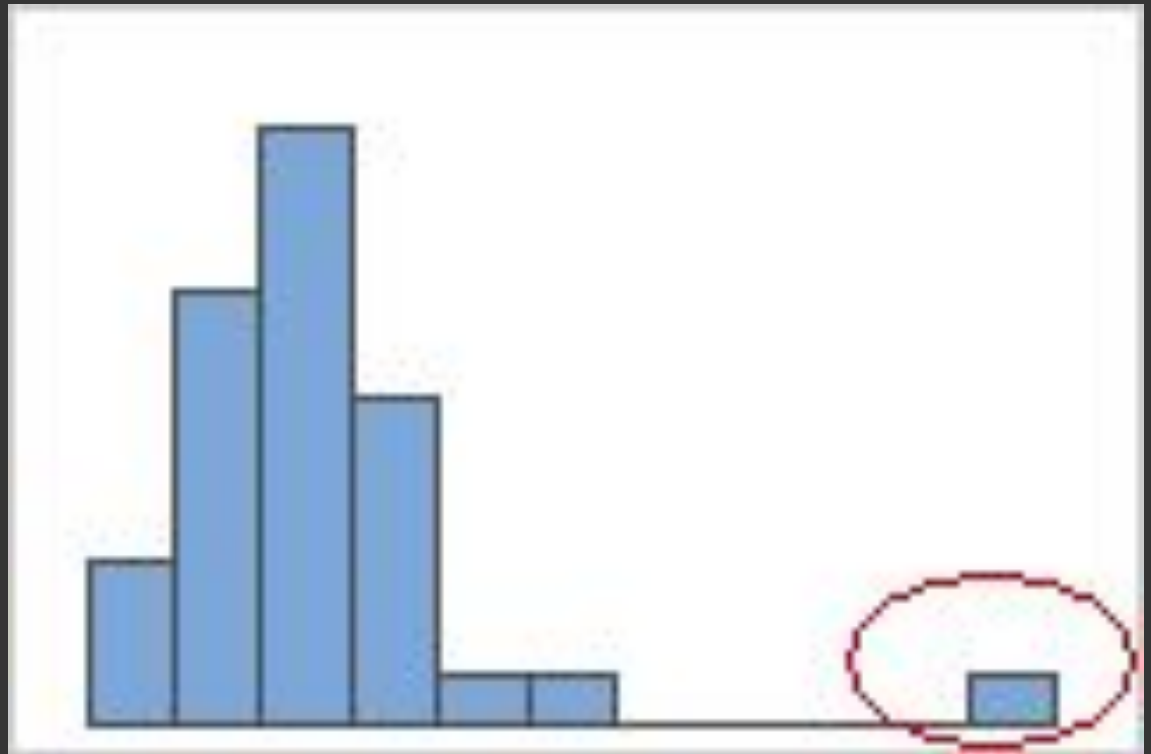
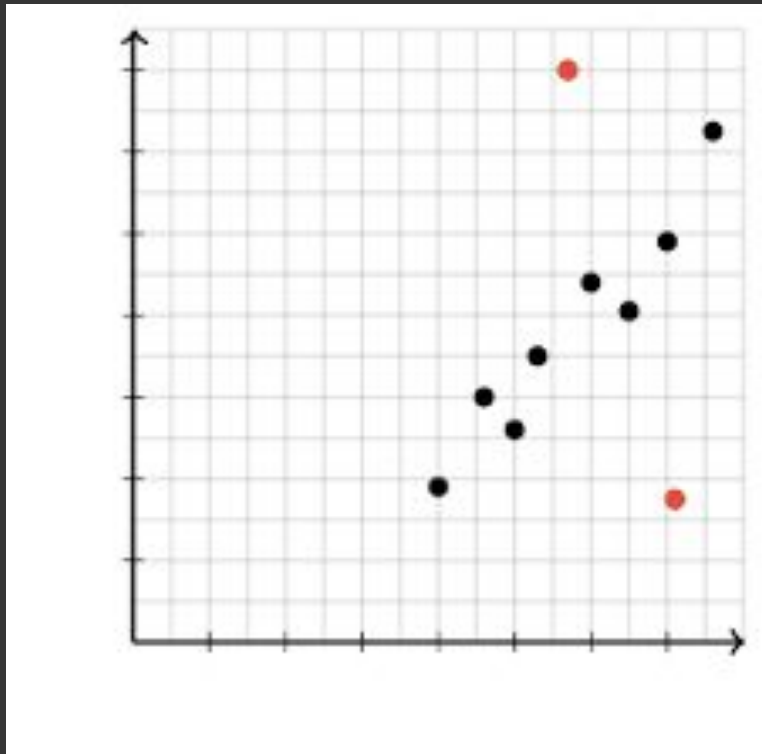
- **Distorcer estatísticas:** Podem influenciar fortemente a média, o desvio padrão e outros cálculos estatísticos, levando a conclusões errôneas sobre os dados.
- **Indicar problemas na qualidade dos dados:** Sua presença pode sinalizar erros que precisam ser corrigidos.
- **Representar informações importantes:** Em alguns casos, os outliers podem ser os pontos mais interessantes, indicando eventos raros ou descobertas significativas.

—

Métodos para detecção de Outliers

Métodos visuais

A visualização dos dados pode ser uma forma intuitiva de identificar outliers.



Métodos Estatísticos

Esses métodos utilizam propriedades estatísticas dos dados para identificar valores atípicos.

Z-Score (Pontuação Padrão)

É uma medida estatística que descreve a posição de um ponto de dado em relação à média de um conjunto de dados. Mais especificamente, ele indica quantos desvios padrão um determinado valor está acima ou abaixo da média.

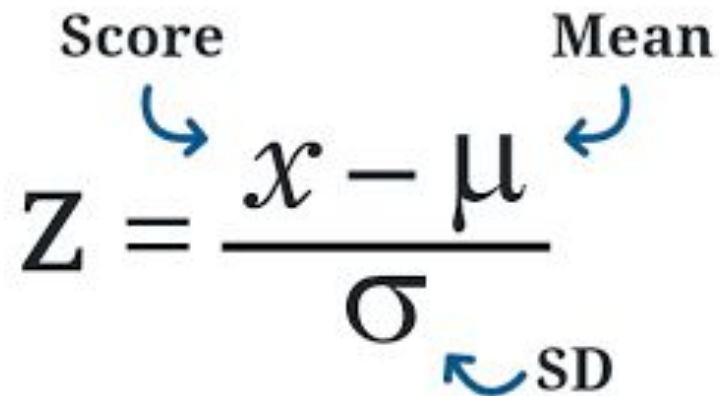
- Z-score = 0 : o valor é exatamente igual à média do conjunto de dados.
- Z-score > 0 : indica que o valor está acima da média. Quanto maior o valor, mais distante ele está da média.
- Z-score < 0 : indica que o valor está abaixo da média. Quanto menor (mais negativo) o valor, mais distante ele está da média, na direção dos valores mais baixos.

Z-Score (Pontuação Padrão)

$$Z = \frac{x - \mu}{\sigma}$$

Score Mean

SD

The diagram shows the Z-score formula $Z = \frac{x - \mu}{\sigma}$. Above the variable x is the label "Score" with a blue arrow pointing to it. Above the Greek letter μ is the label "Mean" with a blue arrow pointing to it. Below the Greek letter σ is the label "SD" (Standard Deviation) with a blue arrow pointing to it.

Valores com um Z-score maior que +3 ou menor que -3 são considerados outliers.

Exemplo

Imagine que você está estudando a altura de coqueiros em uma determinada área de Porto de Galinhas. Você coletou a altura (em metros) de 20 coqueiros:

[15, 16, 15.5, 17, 16.2, 15.8, 16.5, 17.2, 16, 15.7, 16.3, 16.8, 15.9, 16.1, 16.6, 17.1, 16.4, 15.6, 25, 16.7]

→ **Passo 1:** Calcular a média

media = **16.84 metros**

→ **Passo 2:** Calcular o Desvio Padrão

desvio_padrao = **2.04 metros**

Exemplo

→ **Passo 3:** Calcular o Z-score para cada altura

Altura: 15.0m, Z-score: -0.90
Altura: 16.0m, Z-score: -0.41
Altura: 15.5m, Z-score: -0.66
Altura: 17.0m, Z-score: 0.08
Altura: 16.2m, Z-score: -0.31
Altura: 15.8m, Z-score: -0.51
Altura: 16.5m, Z-score: -0.17
Altura: 17.2m, Z-score: 0.18
Altura: 16.0m, Z-score: -0.41

Altura: 15.7m, Z-score: -0.56
Altura: 16.3m, Z-score: -0.26
Altura: 16.8m, Z-score: -0.02
Altura: 15.9m, Z-score: -0.46
Altura: 16.1m, Z-score: -0.36
Altura: 16.6m, Z-score: -0.12
Altura: 17.1m, Z-score: 0.13
Altura: 16.4m, Z-score: -0.22
Altura: 15.6m, Z-score: -0.61
Altura: 25.0m, Z-score: 3.99
Altura: 16.7m, Z-score: -0.07

Exemplo

→ **Passo 4:** Identificar os outliers

Altura: 15.0m, Z-score: -0.90
Altura: 16.0m, Z-score: -0.41
Altura: 15.5m, Z-score: -0.66
Altura: 17.0m, Z-score: 0.08
Altura: 16.2m, Z-score: -0.31
Altura: 15.8m, Z-score: -0.51
Altura: 16.5m, Z-score: -0.17
Altura: 17.2m, Z-score: 0.18
Altura: 16.0m, Z-score: -0.41

Altura: 15.7m, Z-score: -0.56
Altura: 16.3m, Z-score: -0.26
Altura: 16.8m, Z-score: -0.02
Altura: 15.9m, Z-score: -0.46
Altura: 16.1m, Z-score: -0.36
Altura: 16.6m, Z-score: -0.12
Altura: 17.1m, Z-score: 0.13
Altura: 16.4m, Z-score: -0.22
Altura: 15.6m, Z-score: -0.61
Altura: 25.0m, Z-score: 3.99
Altura: 16.7m, Z-score: -0.07

Exemplo - Usando python

```
import numpy as np

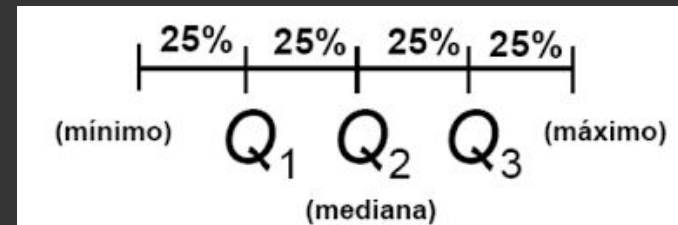
alturas = np.array([15, 16, 15.5, 17, 16.2, 15.8, 16.5, 17.2, 16, 15.7, 16.3,
                    16.8, 15.9, 16.1, 16.6, 17.1, 16.4, 15.6, 25, 16.7])

media = np.mean(alturas)
desvio_padrao = np.std(alturas)
z_score = (alturas - media)/desvio_padrao
for score in z_score:
    if score <= -3 or score >=3:
        print(score)
```


Intervalo Interquartil (IIQ ou IQR)

Relembrando...

$$IIQ = Q3 - Q1$$



Identificando Outliers

Limite Inferior: $Q1 - 1.5 \times IIQ$

Limite Superior: $Q3 + 1.5 \times IIQ$

Qualquer ponto de dado que seja menor que o Limite Inferior ou maior que o Limite Superior é sinalizado como um potencial outlier.

Exemplo

Imagine que você está coletando dados sobre o tempo de espera (em minutos) para atendimento em um determinado serviço público aqui no Recife, durante uma semana. Os dados coletados foram os seguintes:

[5, 7, 6, 8, 9, 6, 7, 10, 8, 7, 6, 5, 9, 7, 8, 6, 25, 7, 8, 9, 6, 7, 8, 7, 6]

→ **Passo 1:** Calcular os Quartis (Q1 e Q3)

$$Q1 = 6$$

$$Q3 = 8.5$$

→ **Passo 2:** Calcular o IIQ

$$IIQ = Q3 - Q1 = 8.5 - 6 = 2.5$$

Exemplo

→ **Passo 4:** Determinar os Limites para Outliers

$$\text{Limite inferior} = Q1 - 1.5 * IQR = 6 - 1.5 * 2.5 = 6 - 3.75 = 2.25$$

$$\text{Limite superior} = Q3 + 1.5 * IQR = 8.5 + 1.5 * 2.5 = 8.5 + 3.75 = 12.25$$

→ **Passo 5:** Identificar os outliers

Qualquer valor no nosso conjunto de dados que esteja **abaixo de 2.25** ou **acima de 12.25** será considerado um outlier. Olhando para os nossos dados ordenados:

[5, 5, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8, 9, 9, 9, 10, 25]

Vemos que o valor 25 está acima do nosso Limite Superior de 12.25. Portanto, 25 é um outlier potencial neste conjunto de dados de tempo de espera.

Exemplo - usando python

```
import numpy as np

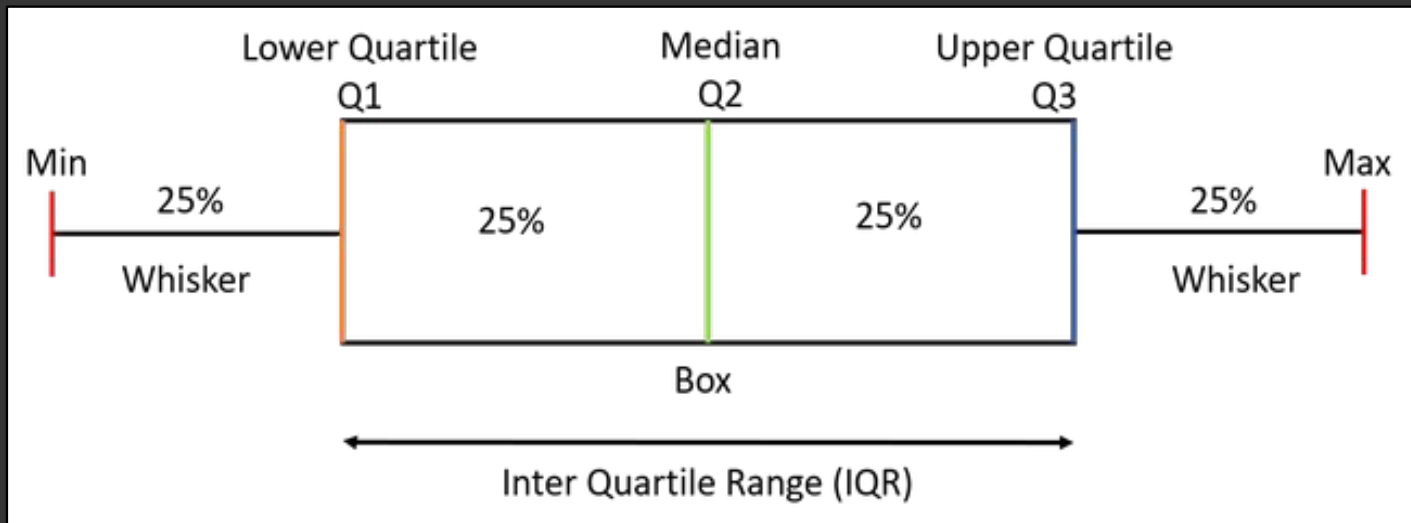
tempos = np.array([5, 7, 6, 8, 9, 6, 7, 10, 8, 7, 6, 5, 9, 7, 8, 6, 25, 7, 8, 9, 6, 7, 8, 7, 6])

q1 = np.percentile(tempos, 25)
q3 = np.percentile(tempos, 75)
iiq = q3 - q1
limite_inferior = q1 - 1.5*iiq
limite_superior = q3 + 1.5*iiq

for tempo in tempos:
    if tempo <= limite_inferior or tempo >= limite_superior:
        print(tempo)
```

Box Plot (Diagrama de Caixa)

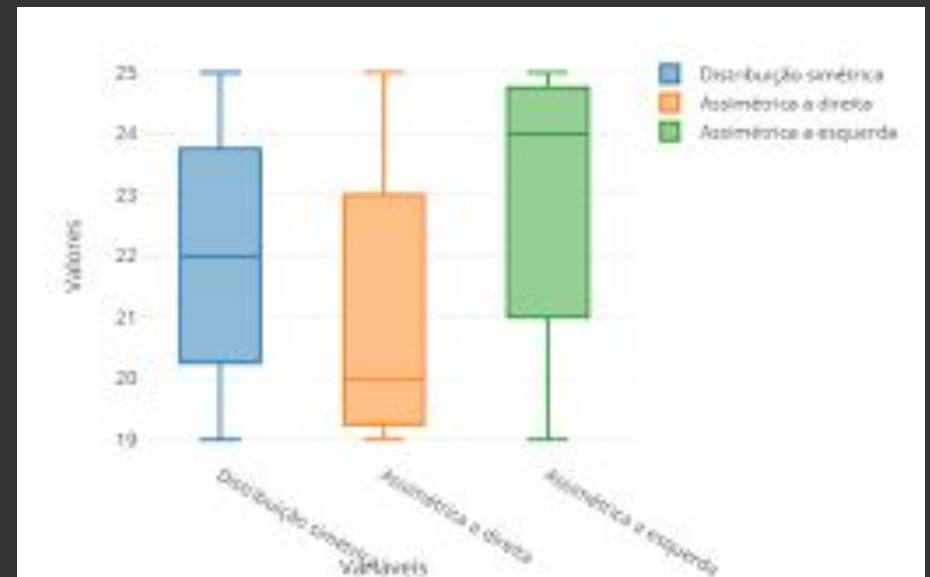
Fornece um resumo conciso da distribuição de uma variável numérica contínua através da exibição de seus quartis, mediana e potenciais outliers.



Box Plot (Diagrama de Caixa)

Vantagens:

- **Resumo da Distribuição:** Fornece uma visão compacta da tendência central (mediana), dispersão (IIQ e alcance), e assimetria dos dados.
- **Comparação entre Grupos:** Facilita a comparação da distribuição de uma variável entre diferentes categorias ou grupos, lado a lado.
- **Identificação de Outliers**



Exemplo

Imagine que você está analisando o tempo de viagem (em minutos) entre dois bairros específicos do Recife (digamos, Boa Viagem e o Centro) em diferentes horários do dia durante uma semana. Você coletou os seguintes dados:

[15, 18, 16, 20, 17, 19, 15, 22, 18, 17, 16, 19, 17, 20, 18, 16, 35, 17, 19, 21, 16, 18, 20, 17, 15]

→ **Passo 1:** Calcular os Quartis (Q1 e Q3) e IIQ

$$Q1 = 16$$

$$Q3 = 19$$

$$IIQ = Q3 - Q1 = 19 - 16 = 3$$

Exemplo

→ **Passo 2:** Detecção dos Whiskers (bigodes)

Limite inferior: $Q1 - 1.5 * IQR = 16 - 1.5 * 3 = 16 - 4.5 = 11.5$

Limite superior: $Q3 + 1.5 * IQR = 20 + 1.5 * 3 = 20 + 4.5 = 24.5$

→ **Passo 3:** Identificação dos Outliers

[15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 17, 18, 18, 18, 19, 19, 19, 20, 20, 20, 21, 22, 35]

→ **Passo 4:** Geração do boxplot

Exemplo

$Q1 = 16$

$Q2 = 18$

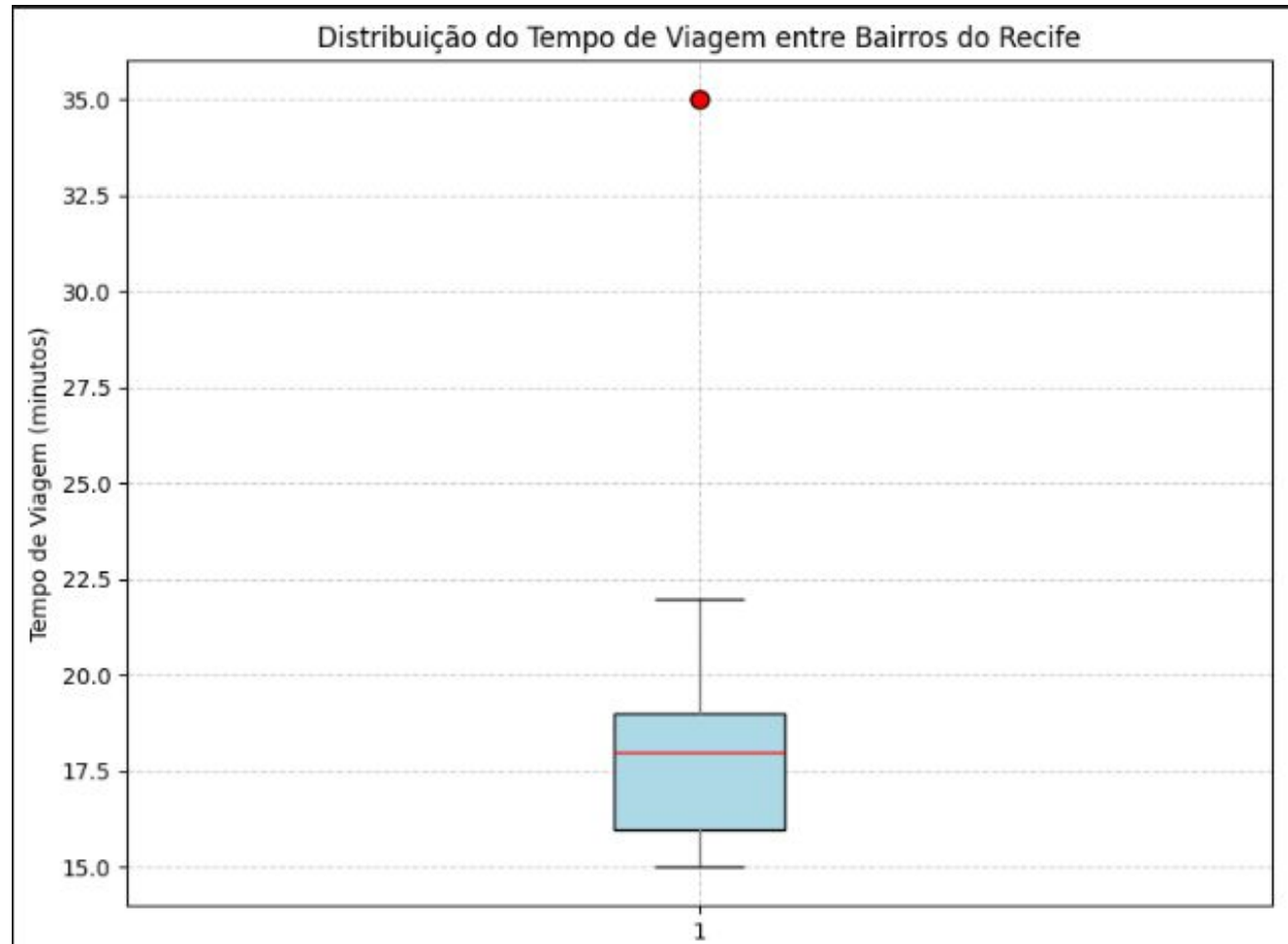
$Q3 = 19$

$IIQ = 3$

Limite inferior = **11.5**

Limite superior = **24.5**

outlier = **35**

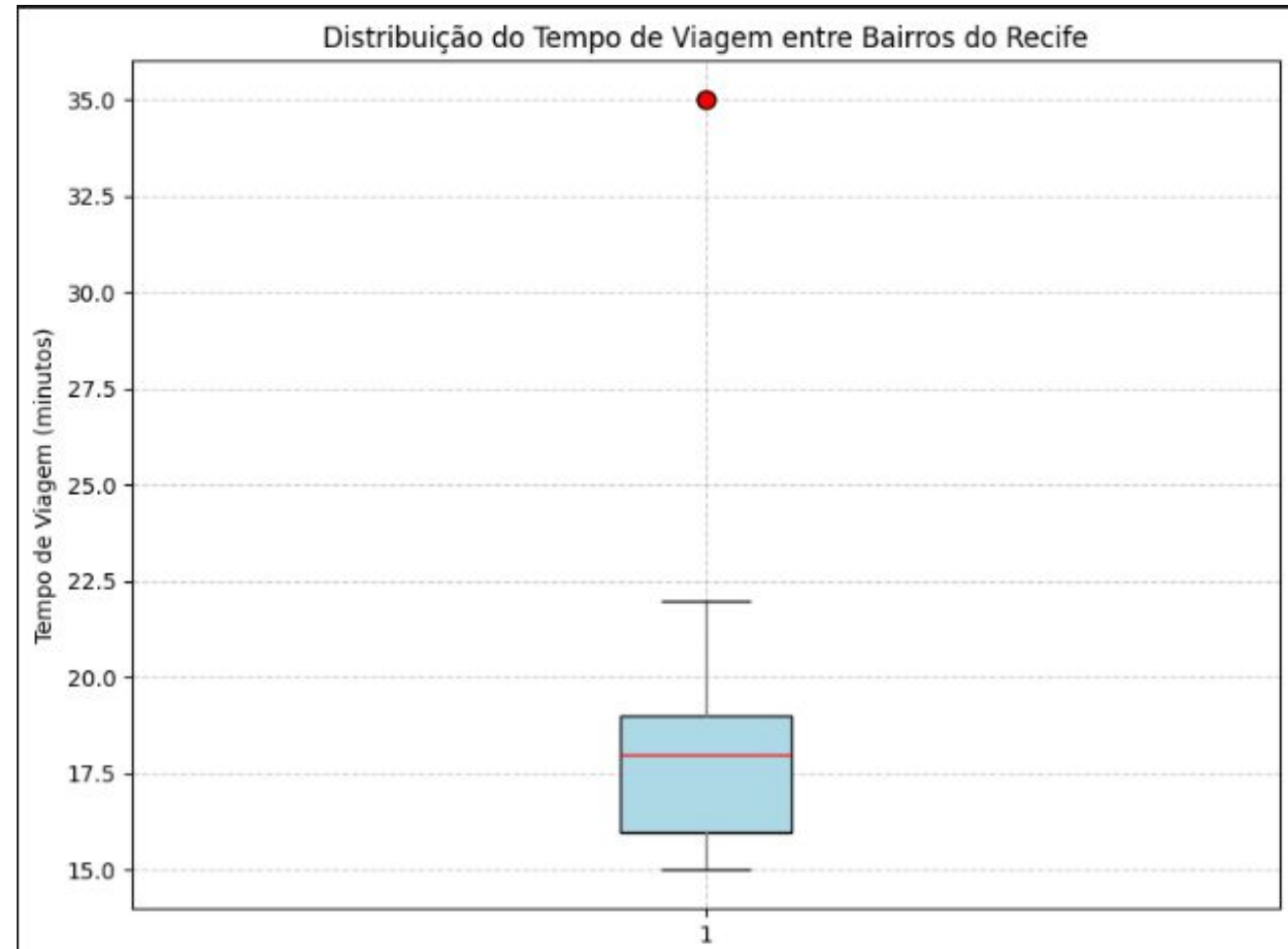


Exemplo

#Plotagem de gráfico

```
plt.boxplot(dados)
```

```
plt.show()
```



Exemplo

```
import matplotlib.pyplot as plt
import numpy as np

tempos_viagem = np.array([15, 18, 16, 20, 17, 19, 15, 22, 18, 17, 16, 19, 17, 20, 18, 16, 35, 17, 19, 21, 16, 18, 20, 17, 15])

plt.boxplot(tempos_viagem,
            patch_artist=True, # Preenche a caixa com cor
            boxprops={'facecolor': 'lightblue'}, # Define a cor da caixa
            whiskerprops={'color': 'gray'}, # Define a cor dos bigodes
            capprops={'color': 'black'}, # Define a cor das hastes dos bigodes
            medianprops={'color': 'red'}, # Define a cor da mediana
            flierprops={'marker': 'o', 'markerfacecolor': 'red', 'markersize': 8, 'linestyle': 'none', 'markeredgecolor': 'black'}) # Estilo dos outliers

plt.ylabel('Tempo de Viagem (minutos)')
plt.title('Distribuição do Tempo de Viagem entre Bairros do Recife')
plt.grid(True, linestyle='--') # Adiciona uma grade de fundo
plt.show()
```



Obrigada!
Bons estudos

