

E-BOOK **DO ESTUDANTE**

»»» **Matplotlib e Seaborn**

Ver “diz” tudo!



Todos os direitos reservados
©2023 Resilia Educação

RESILIA

SUMÁRIO



CONTEXTUALIZANDO	2
MATPLOTLIB	3
SEABORN	7
PARA REFLETIR	11
ATIVIDADE	12
PARA IR ALÉM	12
RESUMO DE MATPLOTLIB E SEABORN	13

Matplotlib e Seaborn

Ver “diz” tudo!



A **visualização de dados** é uma parte essencial do processo de análise de dados, e o Python é uma das linguagens de programação mais populares para esse fim. Duas das bibliotecas mais usadas para visualização de dados em Python são o Matplotlib e o Seaborn.

O **Matplotlib** é uma biblioteca de visualização de dados que permite criar gráficos em 2D e 3D facilmente. Ele possui uma ampla gama de opções de visualização, desde gráficos básicos de linhas e barras até gráficos mais complexos, como gráficos de dispersão e histogramas. Já o **Seaborn** é uma biblioteca mais avançada, que é construída sobre o Matplotlib. Ela oferece uma ampla variedade de gráficos estatísticos, como gráficos de distribuição, gráficos de dispersão com regressão linear e gráficos de correlação. Neste e-book, vamos abordar considerações importantes sobre o uso dessas duas bibliotecas de visualização de dados em Python, apresentando os códigos e os gráficos gerados como resultado.

CONTEXTUALIZANDO

Atualmente, a visualização de dados se tornou uma habilidade essencial para qualquer profissional que trabalhe com análise de dados. Isso porque a capacidade de interpretar dados e transformá-los em insights valiosos é fundamental para tomar decisões e melhorar a eficácia das empresas. Permite ainda que as informações sejam comunicadas de forma clara e concisa, o que possibilita aos usuários compreender rapidamente as principais conclusões dos dados.

Nesse contexto, o Python se tornou uma das linguagens mais populares para a visualização de dados, e bibliotecas como o **Matplotlib** e o **Seaborn** desempenham um papel crucial nesse processo.

A habilidade de gerar visualização de dados em Python com bibliotecas como o Matplotlib e o Seaborn é imprescindível aos profissionais de análise de dados nos dias de hoje. Além de permitir às empresas melhorarem sua eficácia através de decisões baseadas em dados, a visualização de dados também desempenha um papel fundamental no desenvolvimento de estratégias de negócios bem-sucedidas. Portanto, os profissionais que dominam essas ferramentas têm uma vantagem competitiva no mercado de trabalho.

MATPLOTLIB



Matplotlib é uma biblioteca de visualização de dados em Python. Ela permite a criação de gráficos de linhas, barras, dispersão, histogramas, entre outros, além de ser extremamente personalizável e flexível. Neste texto, vamos explorar alguns dos recursos do Matplotlib, dar exemplos de gráficos e explicar como criá-los e personalizá-los.

Para começar, é necessário importar a biblioteca. A convenção para importar o Matplotlib é usar o alias "plt", como mostrado abaixo:

```
import matplotlib.pyplot as plt
```

O "pyplot" é um submódulo do Matplotlib, que oferece uma interface descomplicada para a criação de gráficos.

Vamos começar com um exemplo simples: um **gráfico de linhas**. Este gráfico é adequado para visualizar dados que **variam continuamente** ao longo do tempo, sendo particularmente útil para mostrar tendências e padrões. O código a seguir cria um gráfico que mostra a evolução da receita de uma empresa:

```
import matplotlib.pyplot as plt

# Dados
ano = [2015, 2016, 2017, 2018, 2019, 2020]
receita = [100000, 120000, 150000, 180000, 200000, 220000]

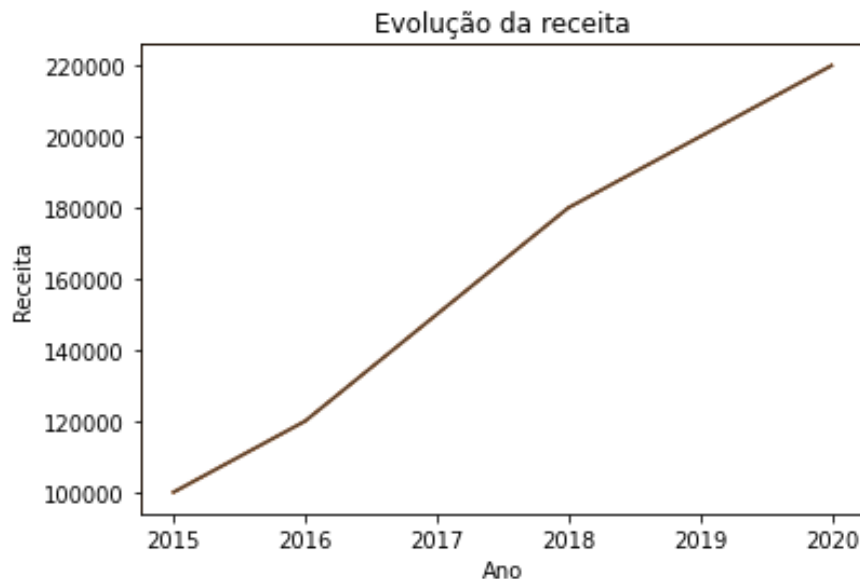
# Criação do gráfico
plt.plot(ano, receita)

# Personalização do gráfico
plt.title("Evolução da receita")
plt.xlabel("Ano")
plt.ylabel("Receita")
plt.show()
```

O código começa definindo duas listas: "ano" e "receita", que armazenam os valores dos dados a serem plotados. Em seguida, a função "plot" é chamada com as listas de dados como argumentos. Essa função cria um gráfico de linhas com base nos dados fornecidos.

Depois, personalizamos o gráfico usando as funções "title", "xlabel" e "ylabel". Por fim, chamamos a função "show" para exibir o gráfico.

Eis o gráfico gerado pelo código acima:



FONTE: O Autor

Outro tipo de gráfico muito comum é o **gráfico de barras**. Este gráfico é adequado para visualizar **dados categóricos**, ou seja, dados que são divididos em categorias ou grupos. Ele é particularmente útil para comparar quantidades entre diferentes categorias. O código abaixo cria um gráfico de barras que mostra a receita de três empresas diferentes:

```
import matplotlib.pyplot as plt

# Dados
empresas = ["Empresa A", "Empresa B", "Empresa C"]
receita = [100000, 120000, 150000]

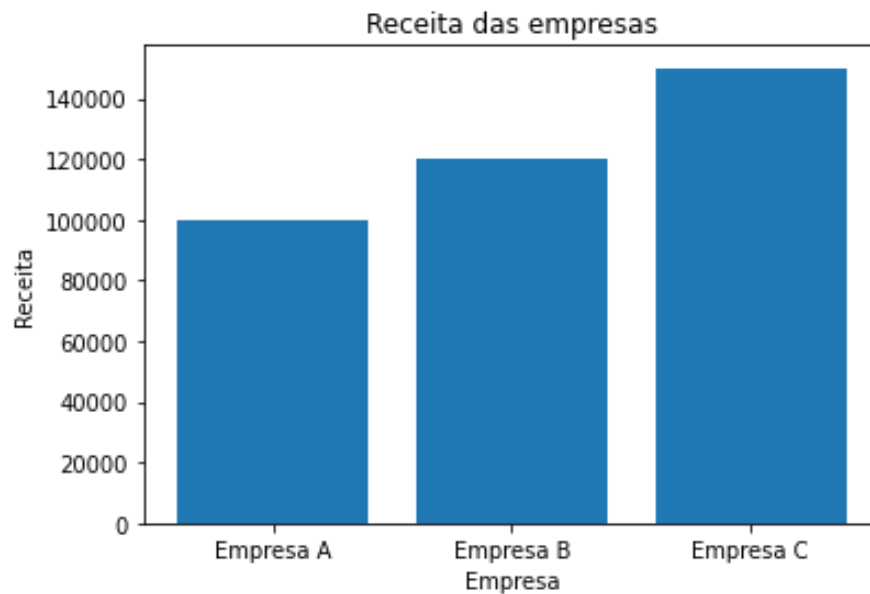
# Criação do gráfico
plt.bar(empresas, receita)

# Personalização do gráfico
plt.title("Receita das empresas")
plt.xlabel("Empresa")
plt.ylabel("Receita")
plt.show()
```

O código começa definindo duas listas: "empresas" e "receita", que armazenam os valores dos dados a serem plotados. Em seguida, a função "bar" é chamada com as listas de dados como argumentos. Essa função cria um gráfico de barras com base nos dados fornecidos.

Em seguida, personalizamos o gráfico usando as funções "title", "xlabel" e "ylabel". Por fim, chamamos a função "show" para exibir o gráfico.

Eis o gráfico gerado:



FONTE: O Autor

Outro tipo de gráfico muito útil é o **gráfico de dispersão**. Ele é usado para mostrar a relação entre duas variáveis e é particularmente útil na análise de dados de vendas, por exemplo. Ele pode ser usado para explorar a relação entre duas variáveis, como a quantidade de vendas e o preço de um produto, ajudando as empresas a identificar padrões e tendências nas vendas, bem como a definir preços competitivos. O código abaixo cria um gráfico de dispersão que mostra a relação entre a idade e o salário de um grupo de pessoas:

```
import matplotlib.pyplot as plt

# Dados
idade = [25, 30, 35, 40, 45, 50, 55, 60]
salario = [3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500]

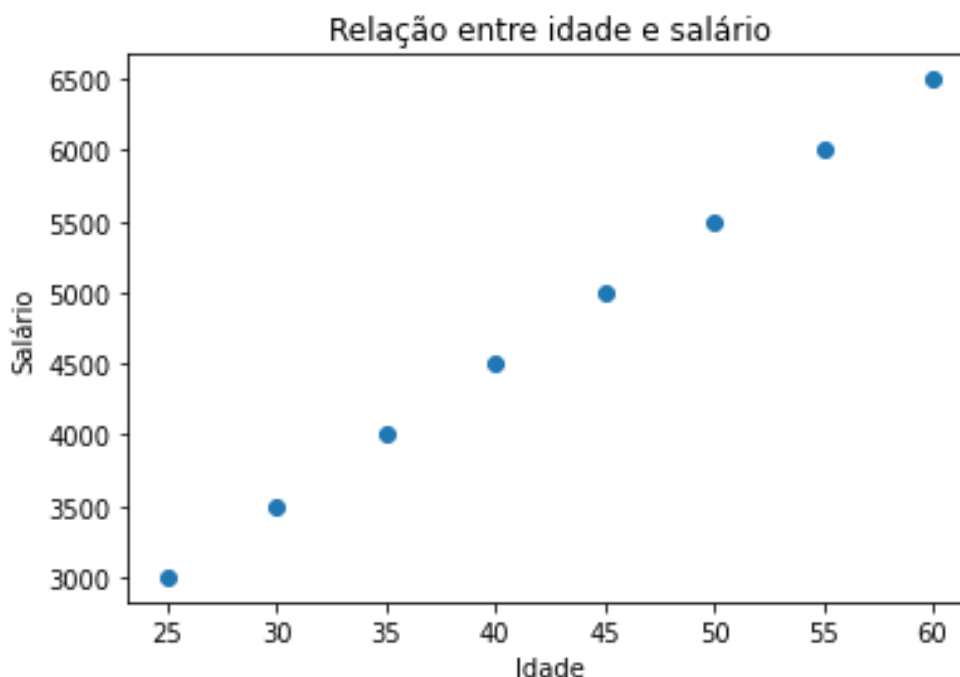
# Criação do gráfico
plt.scatter(idade, salario)

# Personalização do gráfico
plt.title("Relação entre idade e salário")
plt.xlabel("Idade")
plt.ylabel("Salário")
plt.show()
```

Esse código começa definindo duas listas: "idade" e "salario", que armazenam os valores dos dados a serem plotados. Em seguida, a função "scatter" é chamada com as listas de dados como argumentos. Essa função cria um gráfico de dispersão com base nos dados fornecidos.

Depois, personalizamos o gráfico usando as funções "title", "xlabel" e "ylabel". Por fim, chamamos a função "show" para exibir o gráfico.

Segue o gráfico gerado por esse código:



FONTE: O Autor

Ao observar o gráfico, vemos que há uma tendência geral de aumento de salário com a idade. Isso pode indicar que as pessoas com mais experiência e qualificações recebem salários mais altos.

Além desses, o Matplotlib oferece muitos outros tipos de gráficos, como histogramas, gráficos de pizza, gráficos de área, entre outros. A biblioteca também permite a personalização de praticamente todos os aspectos dos gráficos, como cores, tipos de linha, legendas etc.

Para concluir, o Matplotlib é uma biblioteca muito poderosa e flexível para a criação de gráficos em Python. Com a sua ajuda, é possível visualizar dados de forma clara e eficiente, além de permitir personalizar os gráficos de acordo com as necessidades do usuário.

Se você está trabalhando com análise de dados em Python, o Matplotlib é definitivamente uma biblioteca que vale a pena conhecer e explorar. Agora, vamos conferir outra biblioteca bastante útil de visualização de dados: a Seaborn.

SEABORN



Seaborn é uma biblioteca em Python para visualização de dados baseada na popular biblioteca Matplotlib. Ela oferece uma variedade de tipos de gráficos que são visualmente atraentes e que possuem recursos adicionais para a análise de dados.

Para começar a utilizar a Seaborn, é preciso primeiro instalar a biblioteca no seu ambiente Python. Isso pode ser feito utilizando o gerenciador de pacotes pip:

```
!pip install seaborn
```

Depois de instalar a Seaborn, importe a biblioteca no seu código Python:

```
import seaborn as sns
```

Vamos agora começar a explorar alguns dos tipos de gráficos disponíveis na biblioteca Seaborn. Vamos começar pelo gráfico Histograma.

O **histograma** é uma representação gráfica de distribuição de frequência. Ele é utilizado para mostrar como os valores de uma variável estão distribuídos. O Seaborn possui uma função chamada `distplot`, que permite criar histogramas com facilidade.

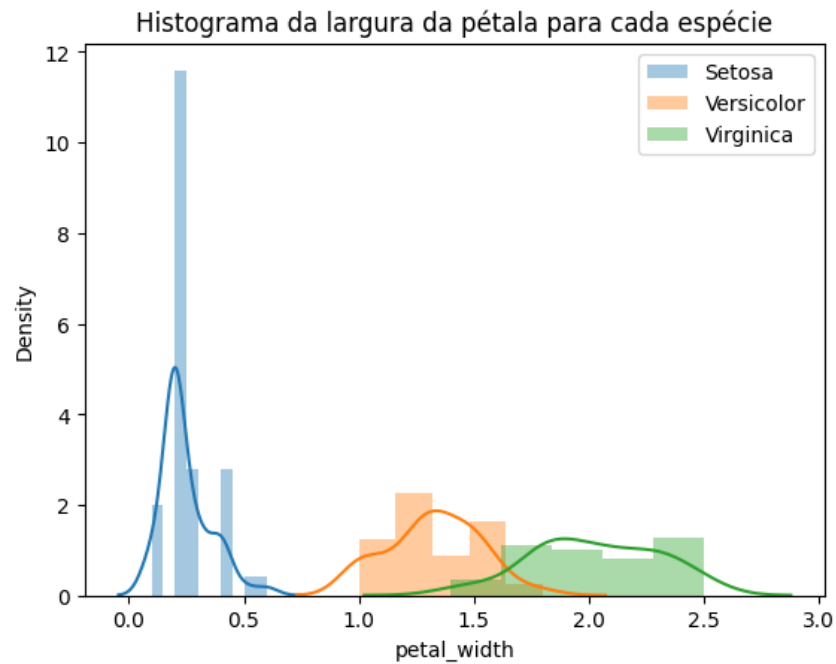
```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset iris do Seaborn
iris = sns.load_dataset('iris')

# Cria um histograma da largura da pétala para cada espécie
sns.distplot(iris[iris.species=='setosa'].petal_width, label='Setosa')
sns.distplot(iris[iris.species=='versicolor'].petal_width,
label='Versicolor')
sns.distplot(iris[iris.species=='virginica'].petal_width,
label='Virginica')

# Adiciona legenda e título
plt.legend()
plt.title('Histograma da largura da pétala para cada espécie')
plt.show()
```


Veja o gráfico gerado:



FONTE: O Autor

Já o **gráfico de barras** é um dos mais populares utilizados para exibir dados categóricos. O Seaborn oferece a função `countplot` para criar gráficos de barras. Veja um exemplo a seguir:

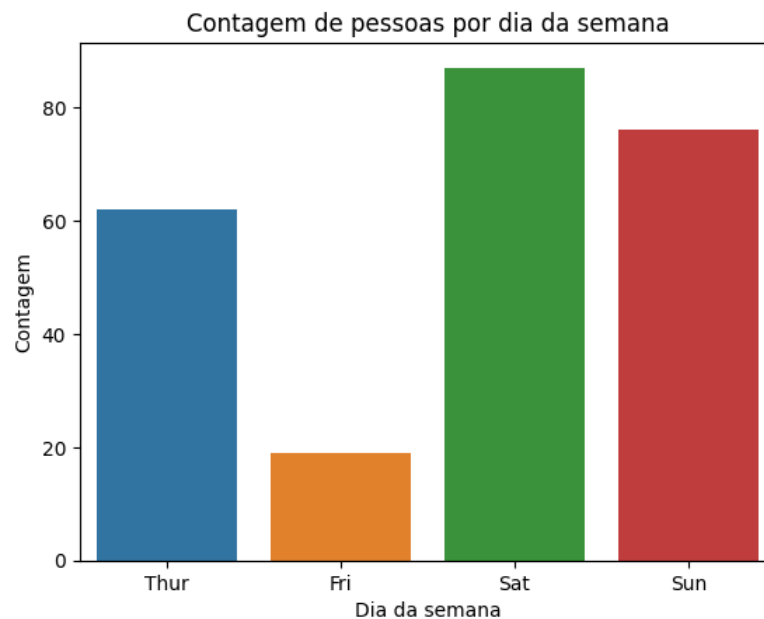
```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset tips do Seaborn
tips = sns.load_dataset('tips')

# Cria um gráfico de barras da contagem de pessoas por dia da
semana
sns.countplot(x='day', data=tips)

# Adiciona título e rótulos dos eixos
plt.title('Contagem de pessoas por dia da semana')
plt.xlabel('Dia da semana')
plt.ylabel('Contagem')
plt.show()
```

Veja o gráfico gerado:



FONTE: O Autor

Outro gráfico interessante é o de **dispersão**. Ele utiliza coordenadas cartesianas para exibir valores de duas variáveis diferentes. E mostra a relação entre essas duas variáveis. O Seaborn possui a função `scatterplot` para criar gráficos de dispersão. Veja o código a seguir:

```
import seaborn as sns
import matplotlib.pyplot as plt

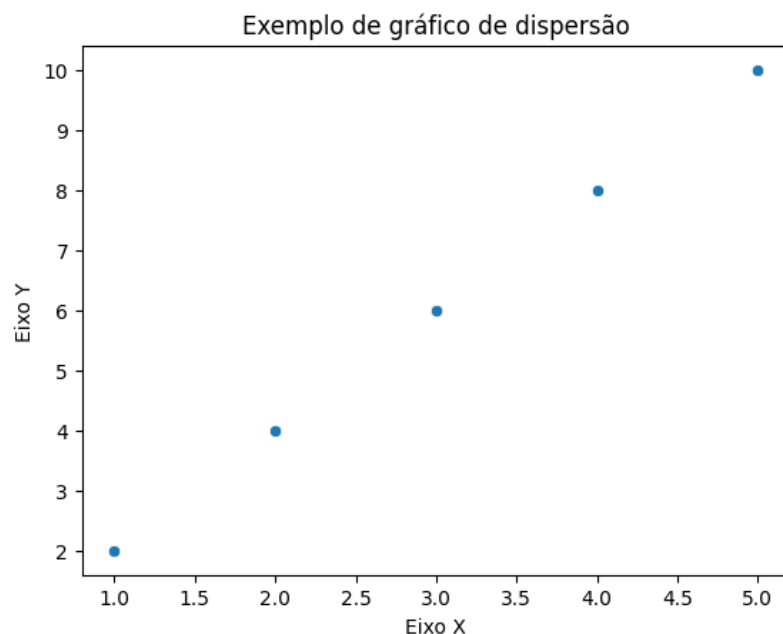
# Carregar os dados
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Criar o gráfico de dispersão
sns.scatterplot(x=x, y=y)

# Adicionar título e rótulos dos eixos
plt.title("Exemplo de gráfico de dispersão")
plt.xlabel("Eixo X")
plt.ylabel("Eixo Y")

# Mostrar o gráfico
plt.show()
```

E o gráfico resultante:



FONTE: O Autor

O gráfico de dispersão mostra cada valor de x e y como um ponto no gráfico. No exemplo dado, x e y são duas listas com 5 elementos cada, representando os valores [1, 2, 3, 4, 5] e [2, 4, 6, 8, 10], respectivamente. Ao observar o gráfico gerado, podemos ver que os pontos estão perfeitamente dispostos em uma linha reta. Isso ocorre porque os valores de y são sempre o dobro dos valores de x.

Por fim, podemos citar ainda o gráfico Boxplot. O box plot é um gráfico que permite visualizar a distribuição de uma variável numérica por categoria. Ele mostra a mediana, o quartil inferior e superior, e os valores mínimo e máximo. O Seaborn possui a função `boxplot` para criar box plots. Veja o código abaixo:

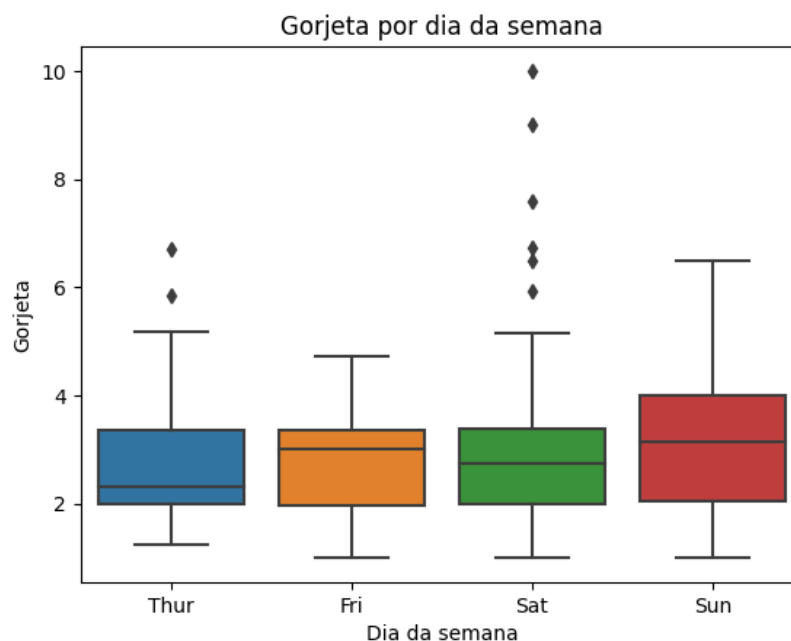
```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset tips do Seaborn
tips = sns.load_dataset('tips')

# Cria um box plot da gorjeta por dia da semana
sns.boxplot(x='day', y='tip', data=tips)

# Adiciona título e rótulos dos eixos
plt.title('Gorjeta por dia da semana')
plt.xlabel('Dia da semana')
plt.ylabel('Gorjeta')
plt.show()
```

Como resultado, ele gera o seguinte gráfico:



FONTE: O Autor

Esses são apenas alguns exemplos de gráficos disponíveis na biblioteca Seaborn. Além deles, ela possui várias outras funções para visualização de dados, como gráficos de barras empilhados, gráficos de linhas, gráficos de violino etc.

Em resumo, o Seaborn é uma biblioteca muito útil para a visualização de dados em Python, oferecendo recursos adicionais aos já presentes no Matplotlib. Com ela, é possível criar gráficos mais atraentes visualmente e mais informativos para a análise de dados.



Para refletir

- Como a biblioteca Matplotlib pode ajudar a criar visualizações de dados de alta qualidade em Python, e quais são alguns dos recursos mais poderosos da biblioteca?
- De que maneira o Seaborn difere do Matplotlib, e como essa biblioteca pode ser usada para criar visualizações de dados mais complexas e informativas?
- Como a escolha de diferentes tipos de gráficos (como gráficos de barras, gráficos de dispersão e histogramas) pode afetar a maneira como as informações são comunicadas em uma visualização de dados, e quais são algumas das considerações importantes ao escolher um tipo de gráfico?



Atividade: Análise Loja de Produtos

Você é o gerente de uma loja de produtos naturais e deseja analisar a preferência dos clientes em relação aos tipos de alimentos vendidos. Para isso, você coletou os dados de vendas de uma semana de cada categoria de alimento em uma lista chamada "dados". Cada elemento da lista representa uma categoria de alimento e o número de unidades vendidas naquela semana.

O objetivo dessa atividade é realizar uma análise exploratória desses dados e gerar visualizações com as bibliotecas matplotlib e seaborn, para melhor compreensão.

Passo a passo:

- Definir a lista de dados contendo as categorias de alimentos e as quantidades vendidas na semana. Exemplo:

```
dados = [('Grãos', 56), ('Frutas', 78), ('Legumes', 32),  
        ('Proteínas', 44), ('Bebidas', 68)]
```

- Identificar o número total de unidades vendidas na semana;
- Utilizar o gráfico de barras para visualizar a quantidade vendida de cada categoria de alimentos. No eixo x, colocar as categorias de alimentos e, no eixo y, a quantidade vendida. Utilize a biblioteca matplotlib para criar o gráfico;
- Utilizar o histograma para visualizar a distribuição dos dados. No eixo x, colocar as quantidades vendidas e, no eixo y, a frequência de cada quantidade. Utilize a biblioteca seaborn para criar o gráfico;
- Utilizar o boxplot para visualizar a distribuição dos dados e identificar a presença de outliers. Utilize a biblioteca matplotlib para criar o gráfico.

Para ir além



- Vimos que Matplotlib é uma biblioteca de visualização de dados em Python que permite a criação de gráficos, histogramas, dispersões, diagramas de caixa, entre outros. No site abaixo, você encontra exemplos de como criar gráficos com essa tecnologia.
<https://www.w3schools.com/python/matplotlib_pyplot.asp>
- Vimos que o Seaborn é uma biblioteca de visualização de dados em Python, construída sobre o Matplotlib, que permite a criação de gráficos estatísticos mais atraentes e informativos. No site abaixo, você confere outros exemplos práticos de uso dessa biblioteca.
<https://www.w3schools.com/python/numpy/numpy_random_seaborn.asp>

RESUMO DE MATPLOTLIB E SEABORN

MATPLOTLIB

Matplotlib é uma biblioteca de visualização de dados em Python que permite a criação de gráficos de linhas, barras, dispersão, histogramas dentre outros.

Importação

```
import matplotlib.pyplot as plt
```

Gráfico de Linhas

```
import matplotlib.pyplot as plt

# Dados
ano = [2015, 2016, 2017, 2018, 2019, 2020]
receita = [100000, 120000, 150000, 180000, 200000, 220000]

# Criação do gráfico
plt.plot(ano, receita)

# Personalização do gráfico
plt.title("Evolução da receita")
plt.xlabel("Ano")
plt.ylabel("Receita")
plt.show()
```

Gráfico de Barras

```
import matplotlib.pyplot as plt

# Dados
empresas = ["Empresa A", "Empresa B", "Empresa C"]
receita = [100000, 120000, 150000]

# Criação do gráfico
plt.bar(empresas, receita)

# Personalização do gráfico
plt.title("Receita das empresas")
plt.xlabel("Empresa")
plt.ylabel("Receita")
plt.show()
```

Gráfico de Dispersão

```
import matplotlib.pyplot as plt

# Dados
idade = [25, 30, 35, 40, 45, 50, 55, 60]
salario = [3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500]

# Criação do gráfico
plt.scatter(idade, salario)

# Personalização do gráfico
plt.title("Relação entre idade e salário")
plt.xlabel("Idade")
plt.ylabel("Salário")
plt.show()
```

SEABORN

Seaborn é uma biblioteca em Python para visualização de dados baseada na popular biblioteca Matplotlib

Instalação/Importação

```
!pip install seaborn
import matplotlib.pyplot as plt
```

Gráfico Histograma

```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset iris do Seaborn
iris = sns.load_dataset('iris')

# Cria um histograma da largura da pétala para cada espécie
sns.distplot(iris[iris.species=='setosa'].petal_width,
label='Setosa')
sns.distplot(iris[iris.species=='versicolor'].petal_width,
label='Versicolor')
sns.distplot(iris[iris.species=='virginica'].petal_width,
label='Virginica')

# Adiciona legenda e título
plt.legend()
plt.title('Histograma da largura da pétala para cada espécie')
plt.show()
```

Gráfico Dispersão

```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset iris do Seaborn
iris = sns.load_dataset('iris')

# Cria um histograma da largura da pétala para cada espécie
sns.distplot(iris[iris.species=='setosa'].petal_width,
label='Setosa')
sns.distplot(iris[iris.species=='versicolor'].petal_width,
label='Versicolor')
sns.distplot(iris[iris.species=='virginica'].petal_width,
label='Virginica')

# Adiciona legenda e título
plt.legend()
plt.title('Histograma da largura da pétala para cada espécie')
plt.show()
```

Gráfico Barras

```
import seaborn as sns
import matplotlib.pyplot as plt

# Carregar os dados
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Criar o gráfico de dispersão
sns.scatterplot(x=x, y=y)

# Adicionar título e rótulos dos eixos
plt.title("Exemplo de gráfico de dispersão")
plt.xlabel("Eixo X")
plt.ylabel("Eixo Y")

# Mostrar o gráfico
plt.show()
```

Gráfico Boxplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Carrega o dataset tips do Seaborn
tips = sns.load_dataset('tips')

# Cria um box plot da gorjeta por dia da semana
sns.boxplot(x='day', y='tip', data=tips)

# Adiciona título e rótulos dos eixos
plt.title('Gorjeta por dia da semana')
plt.xlabel('Dia da semana')
plt.ylabel('Gorjeta')
plt.show()
```



**Até a próxima e
#confianoprocesso**

