

| |
|--|
| Guia Seaborn |
| Curso: UFCD 10809 |
| UFCD/Módulo/Temática: Visualização de dados em Python |
| Ação: 10809_1L Visualização de dados em Python |
| Formador/a: Sandra Liliana Meira de Oliveira |

Guia Seaborn: Funções Essenciais com Exemplos Explicados

O Seaborn é uma biblioteca de visualização estatística em Python baseada no Matplotlib. Fornece uma interface de alto nível para a criação de gráficos informativos e esteticamente agradáveis, particularmente útil quando se trabalha com estruturas de dados do pandas. Este documento apresenta as principais funções do Seaborn e exemplos explicativos usando dataset Tips incluído no Seaborn.

Dataset Utilizado: Tips

Neste exemplo usaremos o dataset `tips`, que vem incluído com o Seaborn.

Este dataset contém informações sobre gorjetas num restaurante, incluindo o valor da conta, o valor da gorjeta, o sexo do cliente, se ele era fumador, o dia da semana e o turno (almoço ou jantar).

Colunas principais:

- `total_bill`: valor total da conta.
- `tip`: valor da gorjeta.
- `sex`: género do cliente.
- `smoker`: se é fumador.
- `day`: dia da semana.
- `time`: almoço ou jantar.
- `size`: número de pessoas na mesa.

Funções Mais Importantes do Seaborn

- `sns.scatterplot()` – Gráfico de dispersão entre duas variáveis numéricas.
- `sns.lineplot()` – Gráfico de linhas para séries ou tendências.
- `sns.barplot()` – Gráfico de barras com estatísticas agregadas (ex: média).

- `sns.countplot()` – Gráfico de barras com contagem de ocorrências.
- `sns.boxplot()` – Boxplot para distribuição estatística (mediana, quartis, outliers).
- `sns.violinplot()` – Combinação de boxplot com densidade de distribuição.
- `sns.histplot()` – Histograma para distribuição de frequência.
- `sns.heatmap()` – Mapa de calor para matrizes de correlação ou categorias cruzadas.
- `sns.pairplot()` – Grelha de gráficos de dispersão entre todas as variáveis numéricas.

Exemplo: `sns.scatterplot()`

Visualiza a relação entre `total_bill` (conta total) e `tip` (gorjeta), distinguindo por género:

Explicação passo a passo:

1. `sns.scatterplot(...)` : cria um gráfico de dispersão.
2. `data=df` : indica o dataset a ser usado.
3. `x="total_bill"` : define o eixo X com o valor da conta.
4. `y="tip"` : define o eixo Y com o valor da gorjeta.
5. `hue="sex"` : colore os pontos com base no género.
6. `plt.title()`, `plt.xlabel()`, `plt.ylabel()` : definem os textos do gráfico.

Código:

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

# Carregar dataset
df = sns.load_dataset("tips")

# Gráfico de dispersão
sns.scatterplot(data=df, x="total_bill", y="tip", hue="sex")
plt.title("Valor da Conta vs Gorjeta por Género")
plt.xlabel("Total da Conta ($)")
plt.ylabel("Gorjeta ($)")
plt.show()
```

Exemplo: sns.lineplot()

Gráfico de linha entre `size` e `tip`, com média suavizada.

Explicação passo a passo:

1. `sns.lineplot(...)` cria um gráfico de linha.
2. `ci='sd'` mostra o desvio padrão como área sombreada.
3. Útil para mostrar tendências de forma agregada.

Código:

```
sns.lineplot(data=df, x="size", y="tip", ci='sd')
plt.title("Gorjeta Média em Função do Tamanho da Mesa")
plt.xlabel("Número de Pessoas")
plt.ylabel("Gorjeta ($)")
plt.show()
```

Exemplo: sns.barplot()

Média de gorjetas por dia da semana.

Explicação passo a passo:

1. `sns.barplot(...)` representa valores agregados (por defeito, a média).
2. Mostra facilmente as diferenças entre categorias.

Código:

```
sns.barplot(data=df, x="day", y="tip")
plt.title("Gorjeta Média por Dia da Semana")
plt.xlabel("Dia")
plt.ylabel("Gorjeta ($)")
plt.show()
```

Exemplo: `sns.countplot()`

Contagem de clientes por dia.

Explicação passo a passo:

1. ``sns.countplot(...)`` conta a frequência de cada categoria.
2. Útil para ver distribuições categóricas.

Código:

```
sns.countplot(data=df, x="day")
plt.title("Número de Clientes por Dia")
plt.xlabel("Dia")
plt.ylabel("Contagem")
plt.show()
```

Exemplo: `sns.boxplot()`

Distribuição de gorjetas por género.

Explicação passo a passo:

1. ``sns.boxplot(...)`` mostra mediana, quartis e outliers.
2. Bom para comparar a variabilidade de grupos.

Código:

```
sns.boxplot(data=df, x="sex", y="tip")
plt.title("Distribuição da Gorjeta por Género")
plt.xlabel("Género")
plt.ylabel("Gorjeta ($)")
plt.show()
```

Exemplo: `sns.violinplot()`

Distribuição e densidade de gorjetas por género.

☑ Explicação passo a passo:

1. `sns.violinplot(...)` combina boxplot com curva de densidade.
2. Permite ver a forma da distribuição para cada grupo.

Código:

```
sns.violinplot(data=df, x="sex", y="tip")
plt.title("Violinplot das Gorjetas por Género")
plt.xlabel("Género")
plt.ylabel("Gorjeta ($)")
plt.show()
```



Exemplo: `sns.histplot()`

Distribuição dos valores totais das contas.



Explicação passo a passo:

1. `sns.histplot(...)` mostra a distribuição de uma variável numérica.
2. `bins=20` define o número de barras (intervalos).
3. `kde=True` adiciona a curva de densidade.

Código:

```
sns.histplot(data=df, x="total_bill", bins=20, kde=True)
plt.title("Distribuição do Total da Conta")
plt.xlabel("Total da Conta ($)")
plt.ylabel("Frequência")
plt.show()
```



Exemplo: `sns.heatmap()`

Mapa de calor das correlações entre variáveis numéricas.



Explicação passo a passo:

1. `df.corr()` calcula a correlação entre variáveis numéricas.
2. `sns.heatmap(...)` visualiza estas correlações com cores.
3. `annot=True` mostra os valores numéricos.

Código:

```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="YlGnBu")
plt.title("Mapa de Calor de Correlações")
plt.show()
```



Exemplo: `sns.pairplot()`

Matriz de dispersão entre todas as variáveis numéricas.



Explicação passo a passo:

1. `sns.pairplot(...)` mostra relações bivariadas entre todas as variáveis numéricas.
2. `hue='sex'` distingue as observações por género.

Código:

```
sns.pairplot(df, hue="sex")
plt.suptitle("Gráficos de Pares com Seaborn", y=1.02)
plt.show()
```



Outras Funcionalidades Importantes

- ◆ ****Plotting Functions**** – Todas as funções do Seaborn seguem uma lógica similar: fornecem dados com `data=`, mapeiam variáveis com `x=`, `y=`, e categorizam com `hue=`, `style=` ou `size=`.
- ◆ ****Statistical Operations**** – Por padrão, o Seaborn agrega estatísticas como médias, medianas e desvios padrão com base nas categorias (ex: `barplot`, `lineplot`).
- ◆ ****Multi-plot Grids**** – Com `FacetGrid`, `catplot`, `pairplot` e `jointplot`, é possível criar grelhas de gráficos facilmente.

◆ ****Figure Aesthetics**** – Personalizações com `sns.set_theme()`, `sns.set_style()`, `sns.set_context()` e `sns.set_palette()` permitem adaptar os gráficos para publicações, apresentações ou relatórios.

◇ Exemplo: Plotting Functions

A maioria das funções de visualização do Seaborn utiliza o mesmo padrão de parâmetros:

- `data=` : define o DataFrame.
- `x=` e `y=` : mapeiam as variáveis para os eixos.
- `hue=`, `style=`, `size=` : permitem categorização visual adicional.

☑ Exemplo:

```
sns.scatterplot(data=df, x="total_bill", y="tip", hue="sex", style="smoker", size="size")
plt.title("Exemplo de Mapping com Seaborn")
plt.show()
```

Neste gráfico de dispersão:

- `hue="sex"` aplica cores distintas para género.
- `style="smoker"` muda o marcador para distinguir fumadores.
- `size="size"` altera o tamanho dos pontos conforme o número de pessoas.

◇ Exemplo: Statistical Operations

Funções como `barplot` e `lineplot` calculam médias ou desvios por grupo.

☑ Exemplo:

```
sns.barplot(data=df, x="day", y="tip", ci="sd")
plt.title("Gorjeta Média e Desvio Padrão por Dia")
plt.show()
```

Neste exemplo:

- O Seaborn calcula automaticamente a média das gorjetas por dia.
- O argumento `ci="sd"` mostra barras de erro com o desvio padrão.

◇ Exemplo: Multi-plot Grids

Com `FacetGrid` ou `catplot`, é possível criar grelhas de gráficos.

☑ Exemplo com catplot:

```
sns.catplot(data=df, x="day", y="total_bill", col="sex", kind="box")
plt.suptitle("Boxplot de Conta Total por Dia e por Género", y=1.05)
plt.show()
```

Neste exemplo:

- Criamos dois gráficos (colunas) para 'male' e 'female'.
- Usamos `kind="box"` para mostrar a distribuição de valores em cada grupo.

◇ Exemplo: Figure Aesthetics

O Seaborn permite alterar tema, paleta de cores, contexto e estilo dos gráficos.

☑ Exemplo:

```
sns.set_theme(style="whitegrid", palette="muted", context="talk")

sns.violinplot(data=df, x="day", y="total_bill")
plt.title("Violinplot com Estilo Personalizado")
plt.show()
```

Neste exemplo:

- `style="whitegrid"` aplica fundo com grelha.
- `palette="muted"` define cores suaves.
- `context="talk"` aumenta o tamanho de texto e linhas para apresentações.

🔍 Temas Avançados com Seaborn

📊 Exemplo: Regressão Linear com sns.lmplot()

A função `sns.lmplot()` permite ajustar uma linha de regressão entre duas variáveis.

☑ Exemplo:

```
sns.lmplot(data=df, x="total_bill", y="tip", hue="sex")
plt.title("Relação entre Conta e Gorjeta com Linha de Regressão")
plt.show()
```

Explicação:

- `lmplot()` cria uma grelha de regressão linear.
- `x` e `y` definem as variáveis independentes e dependentes.
- `hue="sex"` mostra uma linha para cada género.
- A linha representa a tendência da relação entre conta e gorjeta.

Exemplo: Visualização Temporal

Embora o dataset `tips` não inclua datas, podemos simular uma variável temporal para efeitos demonstrativos.

A visualização temporal é especialmente útil em séries cronológicas com `lineplot()`.

Exemplo:

```
# Criar variável temporal simulada
df['dia_simulado'] = range(len(df))

sns.lineplot(data=df, x="dia_simulado", y="total_bill")
plt.title("Evolução Simulada do Total da Conta ao Longo do Tempo")
plt.xlabel("Dia Simulado")
plt.ylabel("Total da Conta ($)")
plt.show()
```

Explicação:

- Criamos uma coluna `dia_simulado` para representar o tempo.
- `lineplot()` liga os pontos em sequência simulando evolução temporal.
- Útil para observar padrões de crescimento, tendência ou sazonalidade.