



escola
britânica de
artes criativas
& tecnologia

Módulo | Análise de Dados: Visualização de Dados II

Caderno de **Aula**

Professor [André Perez](#)

Tópicos

1. Distribuições: Histograma, KDE e Box Plot;
 2. Correlação: Gráfico de Dispersão e Mapa de Calor.
-

Aulas

0. Estruturas de dados

- ****Não estruturado****: texto, imagem, áudio, etc.
- **Semi estruturado**: html, json, etc.
- **Estruturado**: tabelas, planilhas, etc.

1. Distribuições

1.1. Histograma

O **histograma** representa a distribuição de uma variável numérica. A variável numérica é segmentada em intervalos representados por uma barra. Já a altura da barra indica a contagem dos valores presentes na segmentação.

O método do pacote Seaborn que constrói este gráfico é o `histplot` ([docs](#)).

Algumas dicas:

- A variação do tamanho da segmentação por gerar diferentes insights.

Vamos utilizar a base de dados do **titanic**:

```
In [ ]: import seaborn as sns
```

```
In [ ]: data = sns.load_dataset("titanic")
data.head()
```

- **Exemplo:** Valor da passagem por sobrevivência:

```
In [ ]: titanic = data[["fare", "survived"]]
titanic = titanic.query("fare < 100")

with sns.axes_style('whitegrid'):

    grafico = sns.histplot(data=titanic, x="fare")
    grafico.set(title='Valor da passagem por sobrevivência', \
                  xlabel='Valor (USD)', ylabel='Contagem');
    # grafico.get_legend().set_title("Sobreviveu?");
```

1.2. KDE

O **KDE** (*kernel density estimator*), assim como o **histograma**, representa a distribuição de uma variável numérica, mas em uma distribuição contínua. A variável numérica é segmentada em intervalos representados por uma função contínua estimada. Ajuda a evidenciar a distribuição da variável numérica.

O método do pacote Seaborn que constrói este gráfico é o `histplot` ([docs](#)).

Algumas dicas:

- A variação do tamanho da segmentação por gerar diferentes insights.

Vamos utilizar a base de dados do **titanic**:

```
In [ ]: import seaborn as sns
```

```
In [ ]: data = sns.load_dataset("titanic")
data.head()
```

- **Exemplo:** Idade por sobrevivência:

```
In [ ]: titanic = data[["age", "survived"]]

with sns.axes_style('whitegrid'):

    grafico = sns.histplot(data=titanic, x="age", hue="survived", kde=True)
    grafico.set(title='Idade por sobrevivência', xlabel='Idade', \
                  ylabel='Contagem');
    grafico.get_legend().set_title("Sobreviveu?");
```

- **Exemplo:** Idade por sobrevivência por classe:

```
In [ ]: titanic = data[["age", "survived", "class"]]
```

```
with sns.axes_style('whitegrid'):

    grafico = sns.FacetGrid(data=titanic, col="class", hue="survived")
    grafico.map(sns.histplot, "age", kde=True)
```

1.3. Box Plot

O **Box Plot** representa um resumo da distribuição de uma variável numérica. Numa mesmo gráfico mostra a mediana, quartis e *outliers*. É um dos melhores gráficos para representar a distribuição de uma variável numérica.

O método do pacote Seaborn que constrói este gráfico é o `boxplot` ([docs](#)).

Algumas dicas:

- O box plot esconde a distribuição dos grupos na variável, caso existam.

Vamos utilizar a base de dados do **titanic**:

```
In [ ]: import seaborn as sns
```

```
In [ ]: data = sns.load_dataset("titanic")
data.head()
```

- **Exemplo:** Distribuição de idade:

```
In [ ]: with sns.axes_style('whitegrid'):

    grafico = sns.boxplot(x=data["age"])
    grafico.set(title='Distribuição de Idade', xlabel='Idade');
```

- **Exemplo:** Distribuição de idade por classe:

```
In [ ]: with sns.axes_style('whitegrid'):

    grafico = sns.boxplot(x=data["age"], y=data["class"])
    grafico.set(title="Distribuição de idade por classe", xlabel="Idade", \
                  ylabel="Classe");
```

2. Correlação

2.1. Gráfico de Dispersão

O **gráfico de dispersão** representa a correlação entre duas variáveis numéricas. Cada valor é representado um ponto $P(x, y)$. É útil para observar a variação conjunta de duas variáveis.

O método do pacote Seaborn que constrói este gráfico é o `scatterplot` ([docs](#)).

Algumas dicas:

- Evidencie grupos (uma terceira variável categórica), se houverem.

Vamos utilizar a base de dados **iris**:

```
In [ ]: import seaborn as sns
```

```
In [ ]: data = sns.load_dataset("iris")
data.head()
```

- **Exemplo:** Comprimento da pétala por comprimento da sépala:

```
In [ ]: iris = data[["sepal_length", "petal_length", "species"]]
iris.head()
```

```
In [ ]: with sns.axes_style('whitegrid'):

    grafico = sns.scatterplot(data=iris, x="petal_length", y="sepal_length", \
                              hue="species", palette="pastel")
    grafico.set(title='Comprimento da pétala por comprimento da sépala', \
                xlabel='Pétala (cm)', ylabel='Sépala (cm)');
    grafico.get_legend().set_title("Espécie");
```

- **Exemplo:** Largura da pétala por largura da sépala:

```
In [ ]: iris = data[["petal_width", "sepal_width", "species"]]
iris.head()
```

```
In [ ]: with sns.axes_style('whitegrid'):

    grafico = sns.scatterplot(data=iris, x="petal_width", y="sepal_width", \
                              hue="species", palette="pastel")
    grafico.set(title='Largura da pétala por largura da sépala', \
                xlabel='Pétala (cm)', ylabel='Sépala (cm)');
    grafico.get_legend().set_title("Espécie");
```

2.2. Mapa de Calor

O **mapa de calor** representação a correlação entre três variáveis, essencialmente uma representação 2D de um gráfico 3D. Cada valor é representado por um ponto com três coordenadas: **x** e **y** indicam a sua posição e o **z** (necessariamente numérica) a sua intensidade. É útil para observar a distribuição geral dos dados.

O método do pacote Seaborn que constrói este gráfico é o **heatmap** ([doc](#)).

Algumas dicas:

- A paleta de cores é muito importante.

Vamos utilizar a base de dados de **vôos**:

```
In [ ]: import seaborn as sns
```

```
In [ ]:
```

```
data = sns.load_dataset("flights")
data.head()
```

- **Exemplo:** Distribuição de passageiros por mês por ano:

```
In [ ]: flights = data.pivot("month", "year", "passengers")
flights.head()
```

```
In [ ]: grafico = sns.heatmap(data=flights, cmap="Spectral")
grafico.set(title='Passageiros por mês por ano', xlabel='Ano', \
            ylabel='Mês');
```

```
In [ ]: grafico = sns.heatmap(data=flights, cmap="Spectral", annot=True, \
                               fmt="d")
grafico.set(title='Passageiros por mês por ano', xlabel='Ano', \
            ylabel='Mês');
grafico.figure.set_size_inches(w=20/2.54, h=10/2.54)
```

```
In [ ]: data = sns.load_dataset("flights")
```

```
In [ ]: flights = data.pivot("year", "month", "passengers")
flights.head()
```

```
In [ ]: grafico = sns.heatmap(data=flights, cmap="Spectral")
grafico.set(title='Passageiros por mês por ano', xlabel='Mês', \
            ylabel='Ano');
```