escola britânica de artes criativas & tecnologia

Analista de Dados

Visualização de dados I

Módulo | Análise de Dados: Visualização de Dados I

Caderno de Aula

Professor André Perez

Tópicos

1. Pacote Seaborn;
2. Categorias: Gráficos de Barras e de Setores;
3. Tendências: Gráficos de Linha e de Área.

Aulas

0. Estruturas de dados

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

1. Pacote Seaborn

Seaborn é um dos pacotes Python mais utilizados para visualização de dados. A documentação pode ser encontrada neste link. Possui diversas opções gráficos (barra, setores, linha, área, etc.) e uma excelente integração com os DataFrames do pacote Python Pandas.

> O **Seaborn** usa o **Matplotlib** (link da documentação), outro pacote bastante utilizada para visualização de dados.

1.1. Visualização

Exemplo: Valor da gorgeta em um restaurante (exemplo padrão do pacote).

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

In [ ]:
```
sns.get_dataset_names()
```

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

In [ ]:
```
data.head()
```

In [ ]:
```
sns.relplot(data=data,x="total_bill", y="tip", hue="day", style="day")
```

In [ ]:
```
sns.relplot(data=data,x="total_bill", y="tip", hue="day", style="day", \
            col="time")
```

In [ ]:
```
sns.relplot(data=data,x="total_bill", y="tip", hue="day", style="day", \
            col="time", row="sex")
```

1.2. Elementos

In [ ]:
```
grafico = sns.relplot(data=data,x="total_bill", y="tip", hue="day", \
                      style="day")
```

- **Títulos e Eixos:**

> Documentação completa com todas as opções de manipulação do texto neste link.

In [ ]:
```
grafico.ax.set_title("Gorjetas", fontsize=12, fontweight="bold");
grafico.set_xlabels("Valor da conta (USD)", fontsize=10);
grafico.set_ylabels("Valor da gorjeta (USD)", fontsize=10);
```

In [ ]:
```
grafico.fig
```

- **Legenda**

In [ ]:
```
grafico.legend.set_title("Dias da semana", \
                         prop={"size": 10, "weight": "bold"});
```

In [ ]:
```
grafico.fig
```

- **Paleta de Cores**

> As paletas de cores podem ser conferidas no link.

In [ ]:
```
grafico = sns.relplot(data=data,x="total_bill", y="tip", hue="day", \
                      style="day", palette='pastel')
```

```
grafico.ax.set_title("Gorjetas", fontsize=12, fontweight="bold");
grafico.set_xlabels("Valor da conta (USD)", fontsize=10);
grafico.set_ylabels("Valor da gorjeta (USD)", fontsize=10);
grafico.legend.set_title("Dias da semana", \
                         prop={"size": 10, "weight": "bold"});
```

In [ ]:
```
grafico = sns.relplot(data=data,x="total_bill", y="tip", hue="day", \
                      style="day", palette='dark')

grafico.ax.set_title("Gorjetas", fontsize=12, fontweight="bold");
grafico.set_xlabels("Valor da conta (USD)", fontsize=10);
grafico.set_ylabels("Valor da gorjeta (USD)", fontsize=10);
grafico.legend.set_title("Dias da semana", \
                         prop={"size": 10, "weight": "bold"});
```

- **Figura**

> Conversão: 1 polegada = 2.54 cm

In [ ]:
```
altura  = 10 / 2.54
largura = 10 / 2.54

grafico.fig.set_size_inches(w=largura, h=altura)
```

In [ ]:
```
grafico.fig
```

In [ ]:
```
altura  = 20 / 2.54
largura = 20 / 2.54

grafico.fig.set_size_inches(w=largura, h=altura)
```

In [ ]:
```
grafico.fig
```

In [ ]:
```
grafico.fig.savefig(fname="gorjetas.png", bbox_inches="tight")
```

In [ ]:
```
grafico.fig.savefig(fname="gorjetas.pdf", bbox_inches="tight")
```

2. Categorias

2.1. Gráfico de Barras

O **gráfico de barras** representa a relação entre uma variável categórica com uma variável numérica. Cada entidade da categoria é representada por uma barra, já a altura das barras representam os seus correspondente valor numérico. Útil para entender a distribuição de uma variável categórica.

> O método do pacote Seaborn que constrói este gráfico é o `barplot` (doc).

Algumas dicas:

- Ordernar as barras pode gerar *insights*;
- Barras horizontais podem facilitar a visualização.

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

In [ ]:
```
import seaborn as sns

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

- **Exemplo**: Valor da conta por dia da semana:

In [ ]:
```
tips = data[["day", "total_bill"]].groupby("day").agg("sum").reset_index()
tips.head()
```

In [ ]:
```
grafico = sns.barplot(data=tips, x="day", y="total_bill", ci=None, \
                      palette="pastel")
grafico.set(title='Valor da conta por dia da semana', \
            xlabel='Dia da semana', ylabel='Valor da conta (USD)');
```

In [ ]:
```
grafico = sns.barplot(data=tips, y="day", x="total_bill", ci=None, \
                      palette="pastel")
grafico.set(title='Valor da conta por dia da semana', \
            ylabel='Dia da semana', xlabel='Valor da conta (USD)');
```

- **Exemplo**: Valor da conta por dia da semana por período:

In [ ]:
```
tips = data[["day", "total_bill", "time"]]
tips = tips.groupby(["day", "time"]).agg("sum").reset_index()
tips.head(10)
```

In [ ]:
```
grafico = sns.barplot(data=tips, x="day", y="total_bill", hue="time", \
                      palette="pastel")
grafico.set(title='Valor da conta por dia da semana por período', \
            xlabel='Dia da semana', ylabel='Valor da conta (USD)');
```

2.2. Gráfico de Setores

O **gráfico de setores**, também conhecido como **gráfico de pizza**, representa a proporção entre as entidades de uma variável categórica. Cada entidade da categoria é representada por uma setor de tamanho proporcional a sua respectiva proporção no todo.

> O pacote Seaborn **não possui suporte para gráficos de setores** e recomenda o uso de gráficos de barras.

> O seu uso **não é recomendado**, em geral humanos não são bons para relacionar angulos com proporções.

Mas se for usar, seguem algumas dicas:

- Se a proporção for em porcentagem, garanta que elas somem 100%;
- Procure colocar a legenda no gráfico;
- Não use gráficos 3D.

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

In [ ]:
```
import seaborn as sns

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

- **Exemplo**: Proporção das gorjetas por dia da semana **com Pandas**:

In [ ]:
```
tips = data[["tip", "day"]].groupby("day").agg("sum").reset_index()
tips["tip_percent"] = 100 * tips["tip"] / tips["tip"].sum()
tips.head()
```

In [ ]:
```
tips.plot.pie(y="tip_percent", labels=tips["day"]);
```

3. Tendências

3.1. Gráfico de Linha

O **gráfico de linha** representa a evolução de uma variável numérica (eixo `y`), geralmente ao longo do tempo (eixo `x`), formando assim uma `série temporal`. Cada valor numérico é representado por pontos conectados por uma linha reta.

> O método do pacote Seaborn que constrói este gráfico é o `lineplot` (doc).

Algumas dicas:

- Se uma das colunas for temporal (anos, meses, dias, horas, etc.) garanta a ordenação cronológica;
- Muitas linhas em um mesmo gráfico por dificultar a visualização.

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

In [ ]:
```
import seaborn as sns

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

- **Exemplo**: Número de passageiros por ano:

In [ ]:
```
flights = data[["year", "passengers"]]
flights = flights.groupby("year").agg("sum").reset_index()
flights.head()
```

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

  grafico = sns.lineplot(data=flights, x="year", \
```

```
                          y="passengers", palette="pastel")
  grafico.set(title='Passageiros por ano', xlabel='Ano', \
              ylabel='Passageiros');
```

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

In [ ]:
```
flights = data

with sns.axes_style('whitegrid'):

  grafico = sns.lineplot(data=data, x="month", y="passengers", \
                         hue="year", palette="pastel")
  grafico.set(title='Passageiros por mês por ano', xlabel='Mês',\
              ylabel='Passageiros');
  grafico.get_legend().set_title("Ano");
```

In [ ]:
```
flights = data.query("1955 <= year < 1960")

with sns.axes_style('whitegrid'):

  grafico = sns.lineplot(data=flights, x="month", y="passengers", \
                         hue="year", palette="pastel")
  grafico.set(title='Passageiros por mês por ano', xlabel='Mês', \
              ylabel='Passageiros');
  grafico.get_legend().set_title("Ano");
```

3.2. Gráfico de Área

O **gráfico de área** é similar ao **gráfico de linha** e representa a evolução de uma variável numérica (eixo `y`), geralmente ao longo do tempo (eixo `x`), formando assim uma `série temporal`. Cada valor numérico é representado por pontos conectados por uma linha reta tendo ainda a área entre a linha e o eixo `x` preenchido por uma cor.

> O método do pacote Seaborn que constrói este gráfico é o `lineplot` (doc).

Algumas dicas:

- Se uma das colunas for temporal (anos, meses, dias, horas, etc.) garanta a ordenação cronológica;
- Muitas áreas em um mesmo gráfico por dificultar a visualização.

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

In [ ]:
```
import seaborn as sns

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

- **Exemplo**: Número de passageiros por ano:

In [ ]:
```
flights = data[["year", "passengers"]]
flights = flights.groupby("year").agg("sum").reset_index()
flights.head()
```

In [ ]:

```
import matplotlib.pyplot as plt

with sns.axes_style('whitegrid'):

  grafico = sns.FacetGrid(data=flights, palette="pastel")
  grafico.map(sns.lineplot, "year", "passengers")
  grafico.map(plt.fill_between, 'year', 'passengers', alpha=0.3)
  grafico.set(title='Passageiros por ano', xlabel='Ano', \
              ylabel='Passageiros');
  grafico.fig.set_size_inches(w=15/2.54, h=7.5/2.54)
```