Análise exploratória

Bibliotecas usadas

- utiliza-se várias bibliotecas para análise de dados:
  - matplotlib
  - pandas
  - seaborn
- ambas 2 últimas rodam matplot por baixo

Importação

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

Comandos

Ler dados

- arquivo csv

```
pd.read_csv('arquivo.csv')
```

- dados html

```
pd.read_html('URL')
```

- pode-se colocar dentro de uma variável e aplicar funções e plotar gráficos em cima dela

```
variavel = pd.read_csv('arquivo.csv')
```

Funções

```
variavel.head() # mostra 5 primeira linhas
variavel.shape # mostra o formato do dataframe em linhas e colunas
variavel.columns = ['coluna1', ...] # alterando títulos das colunas
variavel['coluna1'] # mostrando apenas uma coluna específica
variavel['coluna1'].unique() # mostra os valores presentes na coluna sem repetir
variavel['coluna1'].value_counts() # conta e ordena os valores da coluna
variavel['coluna1'].mean() # média da coluna
variavel.coluna1.median() # mediana da coluna... esse formato, .série, é mais usado
variavel.coluna1.describe()
# exibe valores como total de dados, média, max, min, mediana (50%)
# e ainda valor de corte (25%) => 1/4 das notas são inferiores a 3
# (75%) => 1/4 das notas são maiores que 4
```

Plotando Gráficos

```
variavel.nota.plot() # plota um gráfico padrão
variavel.nota.plot(kind='hist') # plota um gráfico do tipo histograma
sns.boxplot(x=variavel.coluna1) # gráfico do tipo caixa do seaborn
# apresenta as mesmas informações da função describe, mas de maneira visual
# para plotar verticalmente, usar eixo y ao invés do x
sns.displot(variavel) # grafico de distribuição, parecido com hist
sns.histplot(variavel) # histograma do seaborn
plt.hist(variavel) # plotando com o matplot
plt.title('Histograma das médias dos filmes') # dando um título ao gráfico
sns.histplot(variavel, bins=10) # ajustando número de agrupamentos do gráfico
plt.figure(figsize=(5,8)) # tamanho alterado com matplot
sns.boxplot(y=variavel) # o padrão é horizontal, ou seja, eixo x
```

Filtrando Dados

```
variavel.query('coluna == valor')
# função query seleciona as linhas em que a coluna tem o valor desejado
variavel.query('coluna == valor').mean()
# é possível aplicar funções à query
variavel.drop(columns='coluna3') # remove a coluna especificada
# caso queira mais de uma coluna, separar os valores por vírgula e usar colchetes
variavel.groupby('coluna2').coluna1.mean()
# a função grupby agrega valores iguais de uma mesma coluna
# nesse caso ainda foi aplicada a função mean() de média
```