



Dados e Estatística

Resgatando as variáveis

```
filmes.head()  
# variável => filmes = pd.read_csv('../arquivos/movies.csv')  
filmes.head(2) # mostras só os 2 primeiros elementos  
notas.head()  
# notas = pd.read_csv('../arquivos/ratings.csv')
```

Analisando os dois primeiros filmes

```
notas_do_toy_story = notas.query('filmeId==1')  
notas_do_jumanji = notas.query('filmeId==2')  
print(len(notas_do_toy_story), len(notas_do_jumanji))
```

Calculando a média

```
print(f'Notas médias do Toy Story: {notas_do_toy_story.nota.mean():.2f}')  
# usando f strings  
# print('Notas do Toy Story: %.2f' % notas_do_toy_story.nota.mean())  
# usando interpolação
```

```
print(f'Notas médias do Jumanji: {notas_do_jumanji.nota.mean():.2f}')
# usando f strings
```

```
Notas médias do Toy Story: 3.92
Notas médias do Jumanji: 3.43
```

- a média não representa muitas coisas como por exemplo:
 - quantas pessoas odiaram ou amaram o filme
 - quantas pessoas deram nota 5, ou nota 1 por exemplo

Calculando a mediana

```
print(f'Notas medianas do Toy Story: {notas_do_toy_story.nota.median():.2f}')
print(f'Notas medianas Jumanji: {notas_do_jumanji.nota.median():.2f}')
```

```
Notas medianas do Toy Story: 4.00
Notas medianas Jumanji: 3.50
```

- a mediana expressa apenas o valor de corte onde 50% está acima e 50% está abaixo
- novamente não apresenta muita informação
- tentam apenas representar o valor central do todo

Utilizando o Numpy

- o Pandas pode, em algum momento rodar o Numpy por baixo dos panos

```
import numpy as np
```

Criando um array com Numpy

```
print([2.5] * 10) # sem numpy
np.array([2.5] * 10) # com numpy
```

```
[2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5]
array([2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5])
```

Juntando dois arrays

```
np.append(np.array([2.5] * 10), np.array([3.5] * 10))
```

```
array([2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 3.5, 3.5, 3.5,  
      3.5, 3.5, 3.5, 3.5, 3.5, 3.5])
```

Comparando média e mediana com Numpy

```
filme1 = np.append(np.array([2.5] * 10), np.array([3.5] * 10))  
# supondo que essas sejam as notas de um filme...  
filme2 = np.append(np.array([5] * 10), np.array([1] * 10))  
# e essas de um outro filme  
print(filme1.mean(), filme2.mean()) # média com numpy  
print(np.median(filme1), np.median(filme2))  
# não é possível chamar a mediana com variável.median()
```

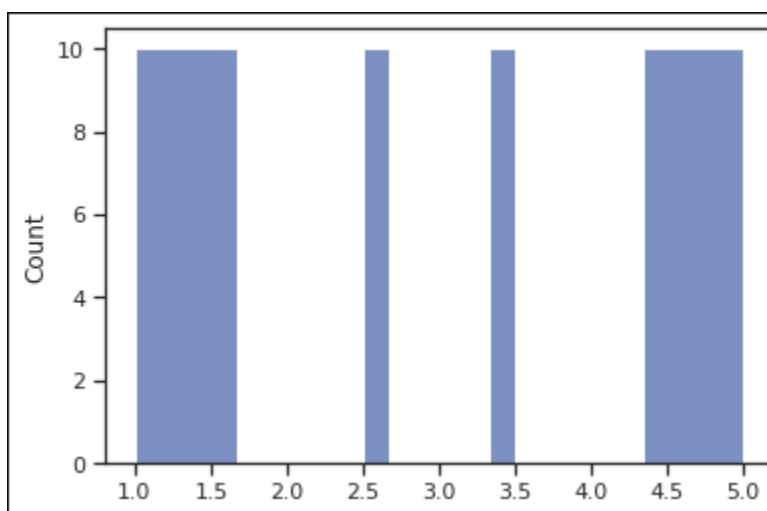
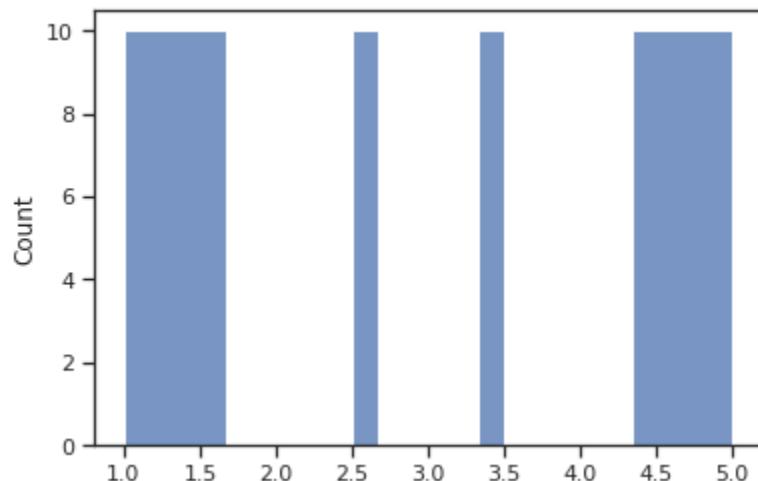
```
3.0 3.0  
3.0 3.0
```

- a média e mediana tiveram o mesmo valor para os dois conjuntos de dados
- porém, sabemos que cada conjunto tem um comportamento completamente diferente do outro:
 - no primeiro as notas se aproximam muito de um valor central
 - no segundo os valores são muito mais distribuídos, extremos
- portanto, cada filme refletiu um comportamento diferente, que não foi possível ver com as funções

Visualizando os dados

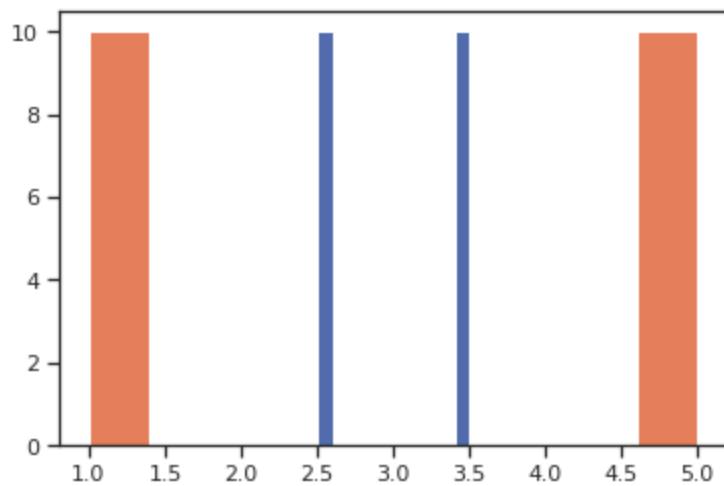
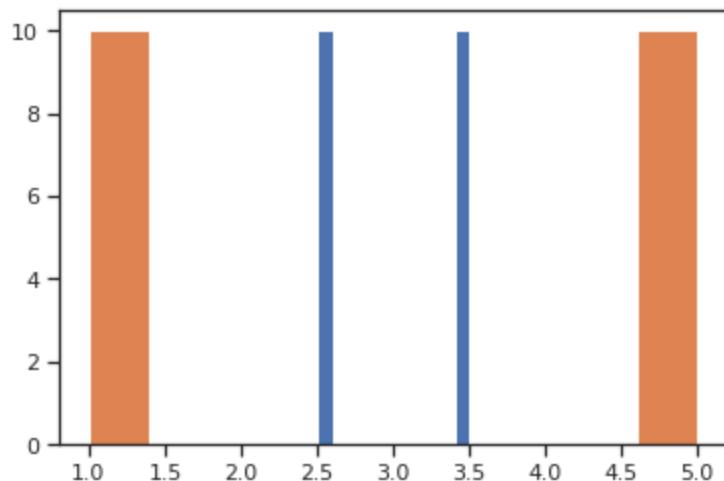
Com Seaborn

```
sns.histplot(filme1)  
sns.histplot(filme2)
```

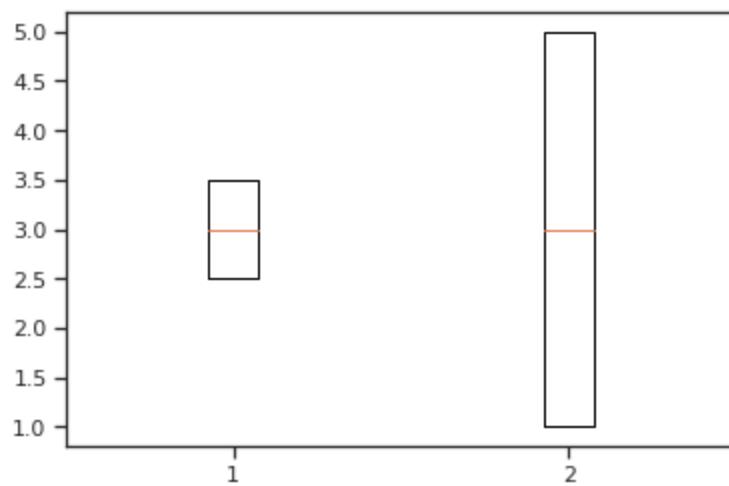


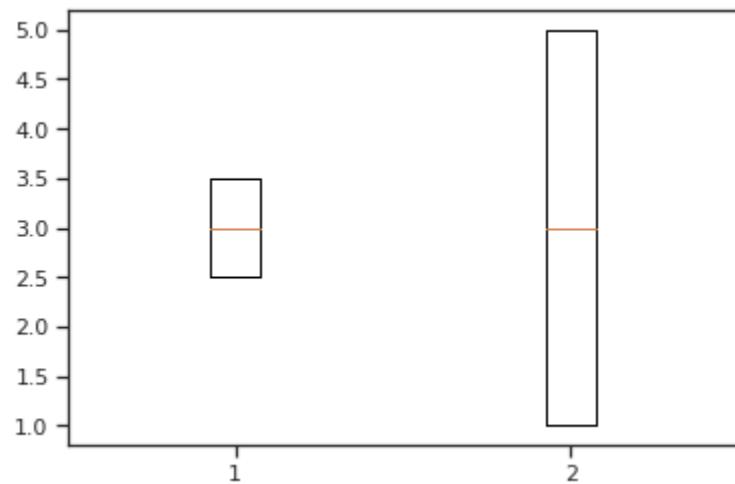
Com Matplot

```
plt.hist(filme1)  
plt.hist(filme2)
```



```
plt.boxplot([filme1, filme2]) # para ver mais de um conj de dados, passamos por um array
```

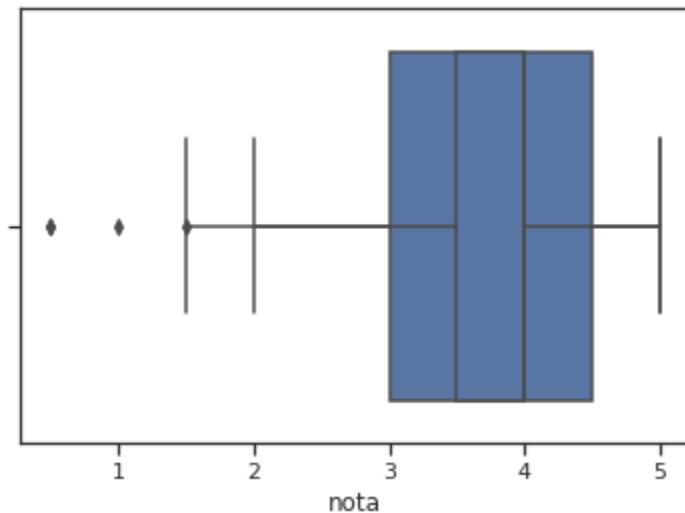


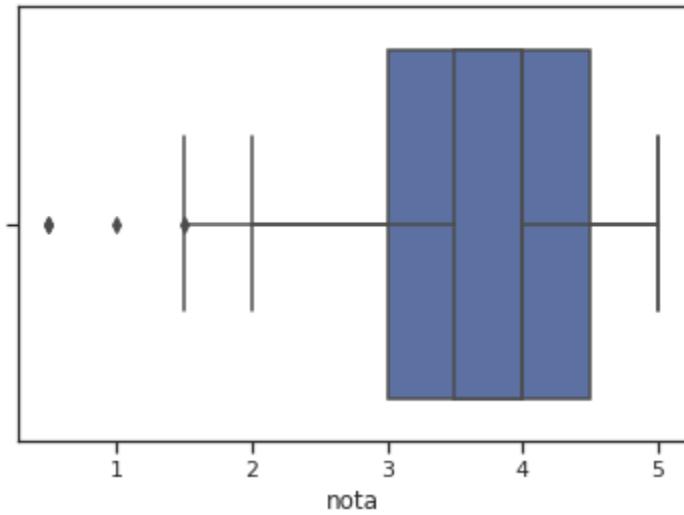


Voltando aos filmes

Boxplot com Seaborn

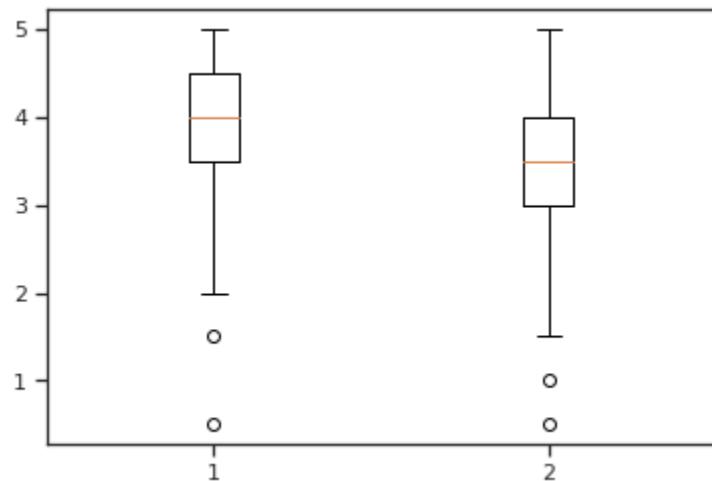
```
sns.boxplot(x=notas_do_toy_story.nota)
sns.boxplot(x=notas_do_jumanji.nota)
```

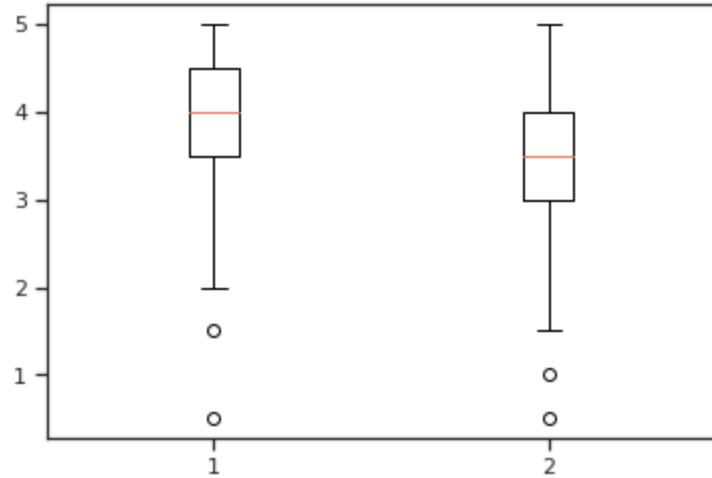




Boxplot com Matplotlib

```
plt.boxplot([notas_do_toy_story.nota, notas_do_jumanji.nota])
```

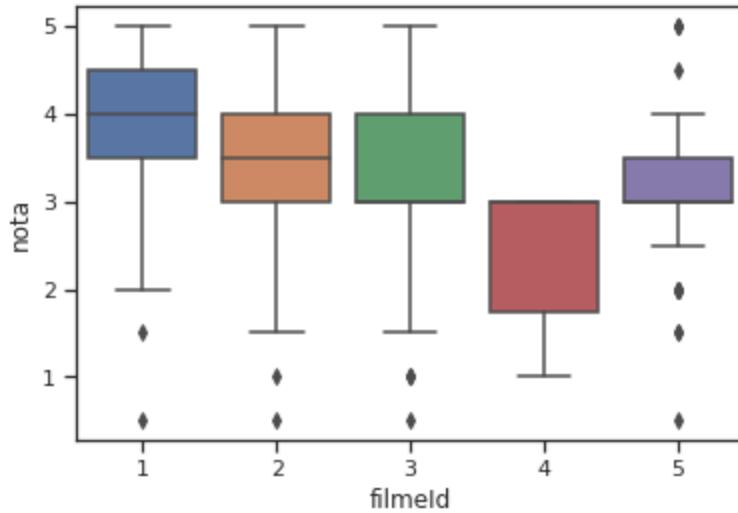


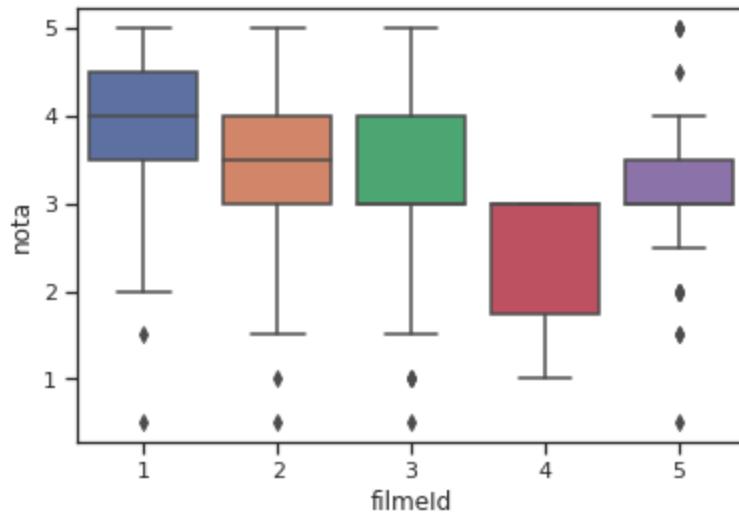


Explorando os dados com Seaborn

- podemos explorar os dados mesmo quando não temos eles separados

```
# sns.boxplot(x='filmeId', y='nota', data = notas)
# nesse caso ele criaria um boxplot para cada filme
# para reduzir o nº de filmes, podemos passar uma query
sns.boxplot(x='filmeId',y='nota',data=notas.query('filmeId in [1, 2, 3, 4, 5]'))
# podemos passar valores da query em um array
```





Cada boxplot representa um filme

Dispersão de Dados e Desvio Padrão

Dispersão de Dados

- Em estatística, dispersão de dados pode ser chamada de variabilidade ou espalhamento
- são dados estatísticos para mostrar uma determinada variação de dados
- com esses dados é possível obter a variância, o desvio padrão e a amplitude interquartil

Desvio Padrão

- mostra o quanto os dados fogem de uma tendência central
- resume em um nº o quanto os dados estão desviando por padrão
- em inglês **standard deviation**
- buscando a documentação do pandas: pandas standard deviation ⇒ pandas.DataFrame.std
- buscando a documentação do numpy ⇒ numpy.std

No Pandas

```
print(notas_do_toy_story.nota.std(), notas_do_jumanji.nota.std())
```

```
0.8348591407114047 0.8817134921476455
```

No Numpy

```
# utilizando as variáveis filme1 e filme2, criadas anteriormente
print(filme1.mean(), filme2.mean())
print(np.std(filme1), np.std(filme2))
print(np.median(filme1), np.median(filme2))
```

```
3.0 3.0
0.5 2.0
3.0 3.0
```

Fontes de Dados

Kaggle

Kaggle: Your Machine Learning and Data Science Community

Kaggle is the world's largest data science community with powerful tools and resources to help you achieve your data science goals.

 <http://www.kaggle.com>

Google Dataset Search

Dataset Search

 <https://datasetsearch.research.google.com/>