Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

Pandas – Manipulação de datas

Aula 2

Código da aula: [DADOS]ANO1C2B4S27A2

Mapa da Unidade 5 Componente 3

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

semana 23

Pandas: Acesso e seleção

semana 25

Pandas: Transformando *data frame*

Você está aqui!

semana 27

Pandas: Manipulação de datas

Mapa da Unidade 5 Componente 3

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

Você está aqui!

Pandas: Manipulação de datas

27

Aula 2

Código da aula: [DADOS]ANO1C2B4S27A2

Objetivos da Aula

- Conhecer o conceito de agrupar e resumir DataFrames usando o método *groupby* da biblioteca Pandas do Python e diferentes funções de agregação.

Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet;
- Software Anaconda/Jupyter Notebook instalado ou similar.

Duração da Aula

50 minutos.

Competências Técnicas

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados;
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências.

Competências Socioemocionais

- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados;
- Trabalhar em equipes multifuncionais, colaborando com colegas, gestores e clientes.

Construindo
o **conceito**

Groupby

O método *groupby* permite dividir um DataFrame em grupos menores com base em colunas **categóricas** e aplicar **funções de agregação**, transformação ou filtragem em cada grupo.

Com isso, vamos conhecer melhor os métodos mais comuns de agrupamento com *groupby*? Eles são de Agregação (*Aggregation*).

Construindo o **conceito**

Agregação (*Aggregation*)

Funções	Conceitos
sum()	Soma dos valores em cada grupo.
mean()	Média dos valores em cada grupo.
count()	Contagem de elementos em cada grupo.
min()	Valor mínimo em cada grupo.
max()	Valor máximo em cada grupo.
median()	Mediana dos valores em cada grupo.
std()	Desvio padrão dos valores em cada grupo.
var()	Variância dos valores em cada grupo.
Nunique()	Quantidade de valores únicos (distintos) em cada grupo.

Construindo o **conceito**

Groupby

Imagine que temos **três times** de um esporte e eles jogaram várias partidas e suas pontuações estão abaixo:

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Construindo o **conceito**

Groupby

Quantas vezes cada time teve pontuação?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Quantidade de Pontuação
A	2
B	3
C	1

Construindo o **conceito**

Groupby

Veja como é no Python:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time').size()
```

```
Time
A    2
B    3
C    1
dtype: int64
```

```
df.groupby('Time').count()
```

	Pontuação
Time	
A	2
B	3
C	1

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

E qual é a **pontuação total** de cada time?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação
A	15 + 18
B	11 + 17 + 10
C	13

Construindo o **conceito**

Groupby

E qual é a **pontuação total** de cada time no Python?

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].sum()
```

```
Time
A    33
B    38
C    13
Name: Pontuação, dtype: int64
```

- Observe que agrupamos os times e somamos sua pontuação.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Qual é a pontuação **média** de cada time?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação Média
A	(15 + 18)/2
B	(11 + 17 + 10)/3
C	13

Construindo o **conceito**

Groupby

Pontuação **média** no Python:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].mean()
```

```
Time
A    16.500000
B    12.666667
C    13.000000
Name: Pontuação, dtype: float64
```

- Observe que agrupamos os times e tiramos a média da pontuação.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Qual é a pontuação **mínima** de cada time?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação Mínima
A	Menor entre 15 e 18
B	Menor entre 11, 17 e 10
C	13

Construindo
o **conceito**

Groupby

Mínimo no Python:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].min()
```

```
Time
A    15
B    10
C    13
Name: Pontuação, dtype: int64
```

- Observe que agrupamos os times e selecionamos a menor pontuação.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Qual é a pontuação **máxima** de cada time?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação Máxima
A	Maior entre 15 e 18
B	Maior entre 11, 17 e 10
C	13

Construindo
o **conceito**

Groupby

Veja como é a pontuação **máxima** no Python:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].max()
```

```
Time
A    18
B    17
C    13
Name: Pontuação, dtype: int64
```

- Observe que agrupamos os times e selecionamos a maior pontuação.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Qual é a **variação padrão da pontuação** de cada time?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação
A	Variância dos valores entre 15 e 18
B	Variância dos valores entre 11, 17 e 10
C	13

Construindo o **conceito**

Groupby

Veja como é a **variação padrão** em Python:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].std()
```

```
Time
A    2.121320
B    3.785939
C         NaN
Name: Pontuação, dtype: float64
```

Observe que agrupamos os times e calculamos o desvio padrão entre os valores.

Não se preocupe com o conceito de desvio padrão agora.

Note que apareceu um **NaN** que significa *Not a Number* (não é um número). Não calculamos desvios de um valor apenas.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Qual é a quantidade de **pontuações que cada time** teve?

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação
A	15 e 18 (2)
B	11, 17 e 10 (3)
C	13 (1)

Construindo o **conceito**

Groupby

No Python com **count():**

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].count()
```

```
Time
A    2
B    3
C    1
Name: Pontuação, dtype: int64
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Groupby

Quantas pontuações diferentes cada time teve?
(Quantos valores únicos de pontuação.)

Time	Pontuação
A	15
A	18
B	11
B	17
B	10
C	13

Time	Pontuação
A	15 e 18 (2)
B	11, 17 e 10 (3)
C	13 (1)

Construindo o **conceito**

Groupby

No Python, as pontuações diferentes ficam da seguinte forma:

	Time	Pontuação
0	A	15
1	A	18
2	B	11
3	B	17
4	B	10
5	C	13

```
df.groupby('Time')["Pontuação"].nunique()
```

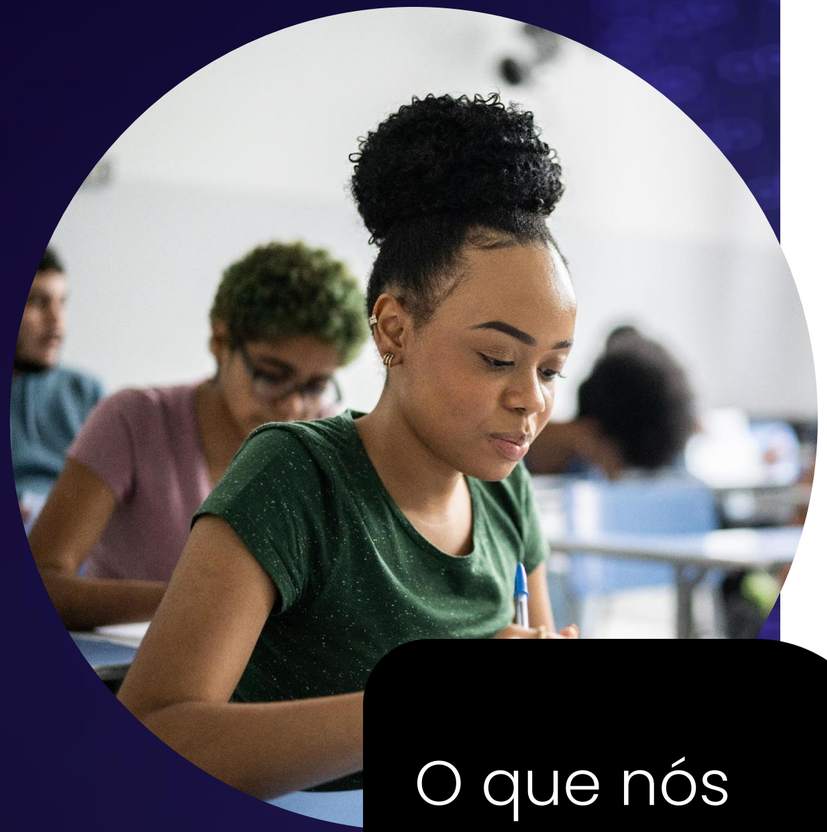
```
Time
A    2
B    3
C    1
Name: Pontuação, dtype: int64
```

- Observe que agrupamos os times e verificamos quantos valores diferentes de pontuação cada time teve.

  Nesse caso, todos os valores são diferentes.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

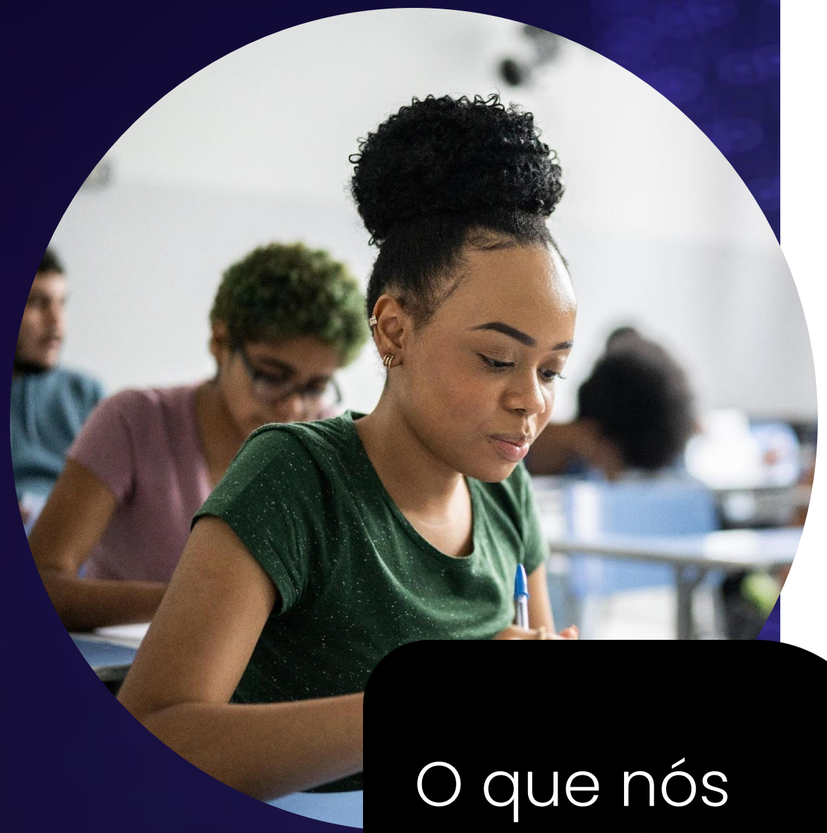
Diálogo



© Getty Images

O que nós **aprendemos hoje?**

Nuvem de palavras



© Getty Images

O que nós **aprendemos hoje?**

Então ficamos assim...

1. Ao filtrar dados de um DataFrame selecionando linhas e/ou colunas, pode-se usar o método .loc.

2. O método *groupby* do Pandas é amplamente utilizado para agrupar dados em um DataFrame.

3. As agregações mais comuns são: sum(), mean(), count(), min(), max(), median(), std(), var(), nunique().

Saiba mais

Que tal encarar o desafio de aprender Pandas em 10 minutos?

Acesse o guia abaixo e traduza para o português para você saber tudo de Pandas em 10 minutos!

PANDAS. *User Guide: 10 minutes to pandas*, [s.d.]. Disponível em: https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html#viewing-data/. Acesso em: 30 jul. 2024.

Referências da aula

MCKINNEY, W. *Python para análise de dados*: tratamento de dados com Pandas, NumPy & Jupyter. São Paulo: Novatec, 2023.

PANDAS. *User Guide*: merge, join, concatenate and compare, [s.d.]. Disponível em: https://pandas.pydata.org/docs/user_guide/merging.html. Acesso em: 13 ago. 2024.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO