Pandas MultiIndex

E se pudessemos ter mais de uma coluna como no índice do DataFrame? O recurso de **índice multinível** em Pandas permite que fazer exatamente isso.

Um DataFrame pandas regular tem uma única coluna que age como um identificador de linha único, ou em outras palavras, um "índice". Esses valores de índice podem ser números, de 0 ao infinito. Eles também podem ser mais detalhados, como ter "Dish Name" como valor de índice para uma tabela de todos os alimentos de uma franquia do McDonald's.

Mas e se tivéssemos as duas franquias do McDonald's, e quisesse comparar as vendas de um prato em ambas as franquias?

Enquanto a função em Pandas funcionaria, este caso também é um exemplo de onde um MultiIndex poderia ser útil.`groupby()`

Um **MultiIndex**, também conhecido como **índice de vários níveis** ou **índice hierárquico**, permite que você tenha várias colunas agindo como um identificador de linha, enquanto tem cada coluna de índice relacionada a outra através de uma relação pai/filho.

No final desta matéria, responderemos às seguintes perguntas criando e selecionando a partir de um DataFrame com um índice hierárquico:

- **Quais personagens falam no primeiro capítulo de "A Sociedade do Anel"?** (respondeu com `.loc`)
- **Quem são os três primeiros elfos a falar no "A Sociedade do Anel"?** (respondeu com `.loc`)
- **Quanto Gandalf e Saruman falam em "As Duas Torres"?** (respondeu com `.loc`)
- **Quanto isildur fala em todos os filmes?** (respondeu com `.xs`)

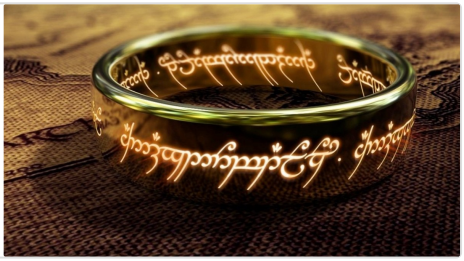
- **Quais hobbits falam mais em cada filme e em todos os três filmes?** (respondeu com uma mesa pivô e `.loc`)

Podemos os dados usados neste artigo aqui:

Lord Of The Rings Data

Character and Movie Data

k https://www.kaggle.com/mokosan/lord-of-the-rings-character-data?select=WordsByCharacter.csv



Usaremos dados dos filmes "O Senhor dos Anéis", especificamente o arquivo "WordsByCharacter.csv" no conjunto de dados. Este arquivo terá o número de palavras de cada personagem falado em cada cena de cada filme.

Como sempre, não se esqueça de importar pandas antes de tentar qualquer um dos códigos.

```
import pandas as pd # load data
df = pd.read_csv('WordsByCharacter.csv')
```

Out[1]:

	Film	Chapter	Character	Race	Words
0	The Fellowship Of The Ring	01: Prologue	Bilbo	Hobbit	4
1	The Fellowship Of The Ring	01: Prologue	Elrond	Elf	5
2	The Fellowship Of The Ring	01: Prologue	Galadriel	Elf	460
3	The Fellowship Of The Ring	01: Prologue	Gollum	Gollum	20
4	The Fellowship Of The Ring	02: Concerning Hobbits	Bilbo	Hobbit	214
...	...	...	...	...	...
726	The Return Of The King	76: The Grey Havens	Elrond	Elf	6
727	The Return Of The King	76: The Grey Havens	Frodo	Hobbit	132
728	The Return Of The King	76: The Grey Havens	Galadriel	Elf	17
729	The Return Of The King	76: The Grey Havens	Gandalf	Ainur	42
730	The Return Of The King	76: The Grey Havens	Sam	Hobbit	14

731 rows × 5 columns

Vamos mergulhar em como índices multiníníveis podem ser usados para análise de dados.

Uma introdução rápida ao MultiIndex

Um índice hierárquico significa que **o DataFrame terá duas ou mais dimensões que podem ser usadas para identificar cada linha**.

Para obter o rótulo de índice original do DataFrame, podemos usar este código:

```
df.index.names
```

```
Out[2]: FrozenList([None])
```

Isso produz um "FrozenList", que é apenas uma construção específica pandas usada para mostrar o (s) rótulo de índice de um DataFrame. Aqui, vemos que o valor é "Nenhum", pois este é o valor padrão do índice de um DataFrame.

Para criar um MultiIndex com nosso DataFrame original, tudo o que precisamos fazer é passar uma lista de colunas para a função Pandas como esta:

```
multi = df.set_index(['Film', 'Chapter', 'Race', 'Character'])
```

Out[3]:

				Words
Film	Chapter	Race	Character	
The Fellowship Of The Ring	01: Prologue	Hobbit	Bilbo	4
		Elf	Elrond	5
			Galadriel	460
		Gollum	Gollum	20
	02: Concerning Hobbits	Hobbit	Bilbo	214
...	...	...	...	...
The Return Of The King	76: The Grey Havens	Elf	Elrond	6
		Hobbit	Frodo	132
		Elf	Galadriel	17
		Ainur	Gandalf	42
		Hobbit	Sam	14

731 rows × 1 columns

Aqui, já podemos ver que o novo DataFrame chamado "multi" foi organizado para que agora existam quatro colunas que

compõem o índice.

Podemos verificar isso olhando os nomes dos índices mais uma vez:

```
multi.index.names
```

```
Out[4]: FrozenList(['Film', 'Chapter', 'Race', 'Character'])
```

Agora vemos que o valor "Nenhum" anteriormente foi substituído pelos nomes das quatro colunas que designamos para ser nosso novo índice.

Cada valor de índice no DataFrame regular e sem tempo seria apenas um número de 0 a 730 (porque o DataFrame tem 731 linhas). Para mostrar qual é o valor de cada índice em nosso Recém-criado MultiIndex, podemos usar esta linha de código:

```
multi.index.values
```

```
Out[5]: array([('The Fellowship Of The Ring', '01: Prologue', 'Hobbit', 'Bilbo'),
               ('The Fellowship Of The Ring', '01: Prologue', 'Elf', 'Elrond'),
               ('The Fellowship Of The Ring', '01: Prologue', 'Elf', 'Galadriel'),
               ('The Fellowship Of The Ring', '01: Prologue', 'Gollum', 'Gollum'),
               ('The Fellowship Of The Ring', '02: Concerning Hobbits', 'Hobbit', 'Bilbo'),
               ('The Fellowship Of The Ring', '03: The Shire', 'Hobbit', 'Bilbo'),
               ('The Fellowship Of The Ring', '03: The Shire', 'Hobbit', 'Frodo'),
               ('The Fellowship Of The Ring', '03: The Shire', 'Ainur', 'Gandalf'),
               ('The Fellowship Of The Ring', '03: The Shire', 'Hobbit', 'Hobbit Kids'),
               ('The Fellowship Of The Ring', '03: The Shire', 'Hobbit', 'Hobbits'),
               ('The Fellowship Of The Ring', '04: Very Old Friends', 'Hobbit', 'Bilbo'),
               ('The Fellowship Of The Ring', '04: Very Old Friends', 'Ainur', 'Gandalf'),
               ('The Fellowship Of The Ring', '04: Very Old Friends', 'Hobbit', 'Lobelia Sackville-Baggins'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Hobbit', 'Bilbo'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Hobbit', 'Frodo'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Ainur', 'Gandalf'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Hobbit', 'Hobbits'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Hobbit', 'Lobelia Sackville-Baggins'),
               ('The Fellowship Of The Ring', '05: A Long Expected Party', 'Hobbit', 'Merry'),
```

Matriz de valores de índice do MultiIndex DataFrame

Agora, vemos que cada valor de linha na coluna "Palavras" pode ser identificado por qual filme veio, a que capítulo se refere, a qual raça o personagem que falou a palavra, e o nome desse personagem.

Uma coisa a notar antes de mergulharmos em alguma análise é a função Pandas. Ao criar um DataFrame com um MultiIndex, certifique-se de anexar isso ao final da linha de código como este: `.sort_index()`

```
multi = df.set_index(['Film', 'Chapter', 'Race', 'Character']).sort_index()
```

A indexação funcionará mesmo que os dados não sejam classificados, mas será bastante ineficiente (e mostrará um Aviso de Desempenho).

Ele também retornará uma cópia dos dados em vez de uma exibição.

Também vale a pena notar que podemos remover o índice hierárquico simplesmente passando para o DataFrame, assim: `.reset_index()`

```
multi.reset_index()
```

Isso apenas retornará o DataFrame original sem as relações pai/filho em várias colunas de índice.

---

1. Quais personagens falam no primeiro capítulo de "A Sociedade do Anel"?

Aqui, estamos interessados em dois componentes do nosso índice: "Filme" e "Capítulo". Em nosso seletor, escrevemos o seguinte: `.loc`

```
multi.loc[('The Fellowship Of The Ring', '01: Prologue'), :]
```

Out[6]:

		Words
Race	**Character**	
Elf	Elrond	5
	Galadriel	460
Gollum	Gollum	20
Hobbit	Bilbo	4

Saída do seletor simples .loc no MultiIndex DataFrame

Você pode selecionar o valor desejado de cada parte do MultiIndex do seu DataFrame, passando-o para uma tupla.

Na primeira parte do seletor que acabamos de escrever, passamos pela tupla ('A Sociedade do Anel', '01: Prólogo'). Não precisávamos passar valores para "Raça" ou "Personagem", porque ainda não sabemos quem falou no primeiro capítulo.

Observe que as colunas de índice **"Filme" e "Capítulo"** não estão mais presentes na visão. Isso porque já inserimos **"A Sociedade do Anel"** para "Filme" e **"01: Prólogo"** como valores na tupla, de modo que a saída não precisa duplicar esses valores na visão.

Além disso, o é inserido como o argumento final no seletor para indicar que queremos que todas as colunas sejam exibidas. No nosso caso, o DataFrame tem apenas uma coluna chamada "Palavras", já que todas as outras colunas do DataFrame foram convertidas em uma coluna de índice.

No entanto, se você tivesse várias colunas no DataFrame e quisesse apenas ver um subconjunto, você os nomes das colunas em vez de .`:.loc:`

Agora, sabemos que Elrond, Galadriel, Gollum e Bilbo falaram na primeira cena do primeiro filme. Vamos passar para um exemplo um pouco mais complicado.

Quem são os três primeiros elfos a falar no "A Sociedade do Anel"?

Aqui, ainda vamos usar, mas agora vamos usar alguma sintaxe que nos dará alguma flexibilidade ao selecionar a partir de nossos valores MultiIndex.`.loc`

Para responder a esta pergunta, estamos interessados nas colunas de índice "Filme" e "Raça". Desta vez, estamos "pulando" uma coluna de índice, já que "Capítulo" está entre "Filme" e "Raça". Ainda não sabemos em que elfos "Capítulos" falam pela primeira vez, então precisamos deixar isso em branco.

Para isso, vamos usar esta linha de código:

```
multi.loc[(‘The Fellowship Of The Ring’, slice(None),’Elf’), :].head(3)
```

Out[7]:

				Words
Film	Chapter	Race	Character	
The Fellowship Of The Ring	01: Prologue	Elf	Elrond	5
			Galadriel	460
	21: Flight To The Ford	Elf	Arwen	131

Mais uma vez, passamos uma tupla com os valores de índice desejados, mas em vez de adicionar valores para "Capítulo", `passamos slice(None)`.

Este é o comando de slice padrão em Pandas para selecionar todos os conteúdos do nível MultiIndex. Então aqui, estamos selecionando todos os valores possíveis do "Capítulo".

Agora, sabemos que Elrond, Galadriel e Arwen são os três primeiros elfos a falar no primeiro livro. Mas até agora, tudo o que temos feito é passar um valor por coluna de índice para os seletores. E se quiséssemos ter mais de um valor?`.loc`

Quanto Gandalf e Saruman falam em cada capítulo de "As Duas Torres"?

Nesta pergunta, estamos interessados em todos os "Capítulos" do segundo "Filme", ambos os quais já sabemos como selecionar. Mas desta vez, estamos interessados em **dois** valores para o índice "Caráter".

```
multi.loc[('The Two Towers',slice(None),slice(None),['Gandalf','Saruman']), :]
```

Out[8]:

Film	Chapter	Race	Character	Words
The Two Towers	01: The Foundations Of Stone	Ainur	Gandalf	39
	06: The Burning of the Westfold	Ainur	Saruman	187
	15: The White Rider	Ainur	Gandalf	298
	17: The Heir Of Númenor	Ainur	Gandalf	226
	20: The King Of The Golden Hall	Ainur	Gandalf	151
	22: Simbelmynë on the Burial Mounds	Ainur	Gandalf	28
	23: The King's Decision	Ainur	Gandalf	165
	25: The Ring Of Barahir	Ainur	Saruman	68
	27: Exodus From Edoras	Ainur	Saruman	4
	36: Isengard Unleashed	Ainur	Saruman	50
	58: Forth Eorlingas	Ainur	Gandalf	21
	65: The Battle For Middle Earth Is About To Begin	Ainur	Gandalf	36

Tudo o que precisamos fazer aqui é passar uma lista dos nossos valores de índice desejados para a tupla que inserimos. Como não sabemos qual "Capítulo" ou "Raça" obter, passamos por esses dois valores de índice antes de chegar a "Personagem". `slice(None)`

Até agora, tivemos múltiplas condições em nossas seleções, e é por isso que usar foi útil. Mas e se quiséssemos especificar um nível do índice? Seria cansativo escrever três vezes, então vamos recorrer a outro método. `.loc` `slice(None)`

Quanto isildur fala em todos os filmes?

Para responder a essa pergunta, basta especificar um valor de nível de índice para "Caráter". Usaremos o método (seção transversal) em Pandas, que permite especificar em qual parte do MultiIndex você deseja pesquisar. `xs()`

Usamos o seguinte código:

```
multi.xs('Isildur', level='Character').sum()
```

```
Out[9]: Words    1
        dtype: int64
```

No seu mais básico, requer uma entrada para o valor do índice que você deseja procurar e o nível que você deseja pesquisar. Neste caso, queremos o nível "Personagem", por

isso passamos isso para , e queremos que o valor para "Personagem" seja "Isildur". Então, para agregar todos os valores, adicionamos ao final da linha.`xs()` `level=` `.sum()`

Eu trapaceei um pouco para que eu pudesse verificar a precisão dos dados, já que eu sabia a resposta para isso (a única palavra que Isildur diz em toda a trilogia é "Não"). Mas você também poderia usar essa mesma linha de código para descobrir o quanto qualquer personagem falou, simplesmente substituindo seu nome como parâmetro em .`xs()`

Quais hobbits falam mais em cada filme e em todos os três filmes?

Para responder a essa pergunta, seria ótimo se tivéssemos uma mesa com os valores de "Palavras" agregados para cada personagem em cada filme. Este é um ótimo lugar para criar uma mesa pivô!

Vamos usar a função do Pandas, mas você verá que passamos uma lista para a configuração do parâmetro para criar um MultiIndex novamente. Nosso código para criar a tabela pivô será assim:`.pivot_table()` `index=`

```
pivoted = df.pivot_table(index = ['Race','Character'],
                         columns = 'Film',
                         aggfunc = 'sum',
                         margins = True, # total column
                         margins_name = 'All Films',
                         fill_value = 0).sort_index()
order = [('Words', 'The Fellowship Of The Ring'),
         ('Words', 'The Two Towers'),
         ('Words', 'The Return Of The King'),
         ('Words', 'All Films')]
pivoted = pivoted.sort_values(by=('Words', 'All Films'), ascending=False)
pivoted = pivoted.reindex(order, axis=1)
```

Out[9]:

		Words			
	Film	The Fellowship Of The Ring	The Two Towers	The Return Of The King	All Films
Race	Character				
All Films		11225	11169	9575	31969
Ainur	Gandalf	2360	964	1504	4828
Hobbit	Sam	557	1044	924	2525
Men	Aragorn	920	822	580	2322
	Frodo	967	664	650	2281
Hobbit	...	...	...	...	...
	Mrs. Bracegirdle	2	0	0	2
Orc	Mauhur	0	2	0	2
Men	Eothain	0	2	0	2
Hobbit	Proudfoot	1	0	0	1
Men	Isildur	1	0	0	1

75 rows × 4 columns

Aqui, nosso índice multiníniveis tem dois componentes em vez dos quatro do exemplo anterior. Isso porque queremos que os valores do "Filme" também sejam uma coluna e não um índice. Também não precisamos ver os valores do "Capítulo", então isso é excluído. Como estamos interessados no número total de palavras, queremos que nosso parâmetro (função agregada) seja "soma". `aggfunc`

Pandas também tem uma coluna total incorporada para a função. Tudo o que você precisa fazer é passar para habilitá-lo e, opcionalmente, definir o nome da coluna total no parâmetro. `.pivot_table()` `margins=True` `margins_name`

Como queremos ver o número total de palavras para cada "Personagem" em que tem um valor "Raça" como "Hobbit", podemos selecionar esta condição com uma aplicação muito fácil de . `.loc`

```
pivoted.loc['Hobbit']
```

Out[10]:

	Words			
Film	The Fellowship Of The Ring	The Two Towers	The Return Of The King	All Films
Character				
Sam	557	1044	924	2525
Frodo	967	664	650	2281
Bilbo	1310	0	56	1366
Pippin	274	359	628	1261
Merry	323	396	398	1117
Hobbits	88	0	4	92
Ted	55	0	0	55
Gaffer	21	0	0	21
Farmer Maggot	21	0	0	21
Sandyman	17	0	0	17
Deagol	0	0	13	13
Hobbit Kids	10	0	0	10
Lobelia Sackville-Baggins	9	0	0	9
Rosie	3	0	2	5
Mrs. Bracegirdle	2	0	0	2
Proudfoot	1	0	0	1

Agora temos uma mesa encomendada pela qual Hobbit falou mais em todos os filmes. Também podemos ver o quanto cada personagem falou em cada filme. Alguns insights úteis incluem como Bilbo não falou em "As Duas Torres", mas ainda tem mais diálogo total do que Pippin e Merry, que falaram nos três filmes.

O que é ótimo em ter essa mesa é que podemos fazer o mesmo tipo de análise para "Elfo", "Homens", "Ainur", e o resto das corridas em O Senhor dos Anéis, tudo substituindo um argumento na função acima. `.loc`