

DataFrameIndexing&Loading

January 25, 2022

```
[1]: !cat resources/week-1/datasets/Admission_Predict.csv
```



```
[2]: import pandas as pd
```

```
[3]: df = pd.read_csv('resources/week-1/datasets/Admission_Predict.csv')
```

```
[4]: df.head()
```

```
[4]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	\
0	1	337	118	4	4.5	4.5	9.65	
1	2	324	107	4	4.0	4.5	8.87	
2	3	316	104	3	3.0	3.5	8.00	
3	4	322	110	3	3.5	2.5	8.67	
4	5	314	103	2	2.0	3.0	8.21	

	Research	Chance of Admit
0	1	0.92
1	1	0.76
2	1	0.72
3	1	0.80

4	0	0.65
---	---	------

```
[ ]: #a indexação do dataframe começa com 0 mas a identificação do aluno começa por 1, daí temos que usar
      #a coluna número 0 como referência para indexação, passamos essa referência no read_csv
```

```
[5]: df = pd.read_csv('resources/week-1/datasets/Admission_Predict.csv', index_col = 0)
      df.head()
```

```
[5]:
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	\
Serial No.							
1	337	118	4	4.5	4.5	9.65	
2	324	107	4	4.0	4.5	8.87	
3	316	104	3	3.0	3.5	8.00	
4	322	110	3	3.5	2.5	8.67	
5	314	103	2	2.0	3.0	8.21	

	Research	Chance of Admit
Serial No.		
1	1	0.92
2	1	0.76
3	1	0.72
4	1	0.80
5	0	0.65

```
[15]: #agora vamos mudar o nome das colunas SPO e LOR pq ninguém sabe o que isso significa a não ser quem esta usando o DF
      #no pandas usamos o rename() que aceita um parâmetro chamado columns e precisamos passar um dicionário onde as chaves dele são os antigos nomes e os valores o novo nome da coluna
      #old_name = new_name

      new_df = df.rename(columns = {
          'GRE Score': 'GRE Score', 'TOEFL Score': 'TOEFL Score', 'University Rating': 'University Rating',
          'SOP': 'Statement of Pourpose', 'LOR ': 'Letter of Recomendation', 'CGPA': 'CGPA',
          'Research': 'Research', 'Chance of Admit ': 'Chance of Admit'
      })
      new_df.head()
```

```
[15]:
```

	GRE Score	TOEFL Score	University Rating	Statement of Pourpose	\
Serial No.					
1	337	118	4	4.5	
2	324	107	4	4.0	
3	316	104	3	3.0	
4	322	110	3	3.5	

5	314	103	2	2.0
---	-----	-----	---	-----

	Letter of Recomendation	CGPA	Research	Chance of Admit
Serial No.				
1	4.5	9.65	1	0.92
2	4.5	8.87	1	0.76
3	3.5	8.00	1	0.72
4	2.5	8.67	1	0.80
5	3.0	8.21	0	0.65

```
[16]: new_df.columns
```

```
[16]: Index(['GRE Score', 'TOEFL Score', 'University Rating',
          'Statement of Pourpose', 'Letter of Recomendation', 'CGPA', 'Research',
          'Chance of Admit'],
          dtype='object')
```

```
[21]: #da pra usar um dicionário contendo somente o nome da coluna que queremos
      ↳ alterar
new_df = new_df.rename(columns = {'Letter of Recomendation': 'Letter Of
      ↳ Recomendation'})
new_df.head()
```

```
[21]:
```

	GRE Score	TOEFL Score	University Rating	Statement of Pourpose	\
Serial No.					
1	337	118	4	4.5	
2	324	107	4	4.0	
3	316	104	3	3.0	
4	322	110	3	3.5	
5	314	103	2	2.0	

	Letter Of Recomendation	CGPA	Research	Chance of Admit
Serial No.				
1	4.5	9.65	1	0.92
2	4.5	8.87	1	0.76
3	3.5	8.00	1	0.72
4	2.5	8.67	1	0.80
5	3.0	8.21	0	0.65

```
[23]: #colocar um espaço no nome antigo funciona, mas e se fosse dois epsaços, um tab
      ↳ ou enter
#outra forma de arrumar o label das colunas é criar funções que fazem isso de
      ↳ forma automática
#renomeando as colunas de acordo. python vem com uma função mão na roda chamada
      ↳ strtip()
#quando usamos ela, nós passamos a função como parâmetro mapper e então
      ↳ indicamos se o eixo deve ser a coluna ou a linha
new_df = new_df.rename(mapper = str.strip, axis = 'columns')
new_df.head()
```

[23]:	GRE Score	TOEFL Score	University Rating	Statement of Pourpose	\
Serial No.					
1	337	118	4	4.5	
2	324	107	4	4.0	
3	316	104	3	3.0	
4	322	110	3	3.5	
5	314	103	2	2.0	

Serial No.	Letter Of Recomendation	CGPA	Research	Chance of Admit
1	4.5	9.65	1	0.92
2	4.5	8.87	1	0.76
3	3.5	8.00	1	0.72
4	2.5	8.67	1	0.80
5	3.0	8.21	0	0.65

```
[26]: #também podemos usar o df.columns atribuido-o á lista de nomes de colunas que
      ↪renomeará as colunas
      #isso vai mudar diretamente o df original, e é bom quando você tem várias
      ↪colunas e quer usar só algumas

      #como exemplo, vamos mudar o nome das colunas para letras minúsculas

      cols = list(df.columns) #criando uma lista de labes das colunas
      cols = [x.lower().strip() for x in cols] #transforme x em minúsculas e faça o
      ↪strip para cada item x na lista cols
      cols
```

```
[26]: ['gre score',
       'toefl score',
       'university rating',
       'sop',
       'lor',
       'cgpa',
       'research',
       'chance of admit']
```

```
[29]: df.columns = cols #colunas do dataframe = lista com novos nomes
df.head()
```

```
[29]:
```

	gre score	toefl score	university rating	sop	lor	cgpa	\
Serial No.							
1	337	118	4	4.5	4.5	9.65	
2	324	107	4	4.0	4.5	8.87	
3	316	104	3	3.0	3.5	8.00	
4	322	110	3	3.5	2.5	8.67	
5	314	103	2	2.0	3.0	8.21	
	research	chance of admit					

Serial No.

1	1	0.92
2	1	0.76
3	1	0.72
4	1	0.80
5	0	0.65

[]:

[]:

[]:

[]:

[]: