

Carregando e armazenando dados em vários formatos de arquivos

Lendo dados em formato texto

Carregando e armazenando dados

Lendo dados em formato texto

Podemos carregar dados de arquivos de texto separados por vírgula utilizando o *read_csv*. O *read_csv* retorna um objeto DataFrame com os dados do arquivo.

Carregando e armazenando dados

Lendo dados em formato texto

```
In [128]: import pandas as pd
```

```
In [129]: arq = pd.read_csv('snv_201807a_2.csv')  
type(arq)
```

```
Out[129]: pandas.core.frame.DataFrame
```

```
In [130]: arq
```

```
Out[130]:
```

	UF	Unnamed: 1	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB- TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB- TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17,930.8
5	AL	Alagoas	100.5	-	49.0	-	-	4.4	53.4	473.6	182.7	91.5	747.8	901.7

Carregando e armazenando dados

Lendo dados em formato texto

Veja o que acontece se carregarmos um arquivo que tenha outro caracter como separador (ponto e vírgula, por exemplo). O *read_csv* entende a linha inteira como pertencendo a uma única coluna.

Carregando e armazenando dados

Lendo dados em formato texto

```
In [3]: arq = pd.read_csv('snv_201807a_2_pontoevirgula.csv')
arq
```

Out[3]:

	UF;;PLANEJADA;TRAVESSIA;LEITO NATUAL;EM OBRAS IMP;IMPLANT;EM OBRAS PAV;SUB-TOTAL;PISTA SIMPLES;EM OBRAS DUP;PISTA DUPLA;SUB-TOTAL;TOTAL
0	DF ; Distrito Federal ; 170.2 ; - ; - ; -...
1	GO ; Goiás ;" 2;636.8 " ; - ; 100.7 ; - ; ...
2	MS ; Mato Grosso do Sul ; 562.6 ; - ; 225.5...
3	MT ; Mato Grosso ;" 1;528.0 " ; - ; 427.7 ; ...
4	Sub-Total;" 4;897.6 " ; - ; 753.9 ; - ; 40...
5	AL ; Alagoas ; 100.5 ; - ; 49.0 ; - ; - ...
6	BA ; Bahia ;" 4;006.4 " ; 39.8 ; 497.7 ; - ; ...
7	CE ; Ceará ;" 1;096.3 " ; - ; 42.0 ; 80.9 ; ...
8	MA ; Maranhão ;" 1;062.9 " ; - ; - ; - ; ...
9	PB ; Paraíba ; 388.0 ; - ; 18.3 ; - ; 0.5...
10	PE ; Pernambuco ; 683.4 ; - ; - ; - ; 9...
11	PI ; Piauí ;" 1;633.7 " ; 52.7 ; 1...

Carregando e armazenando dados

Lendo dados em formato texto

Neste caso, temos que informar o caracter separador usando "sep".

```
In [4]: arq = pd.read_csv('snv_201807a_2_pontoevirgula.csv', sep=';')
arq
```

Out[4]:

	UF	Unnamed: 1	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17,930.8

Carregando e armazenando dados

Lendo dados em formato texto

Como nosso arquivo está muito grande, vamos limitar o número de linhas exibidas. Para isso, usaremos o `nrows=numero_linhas`.

```
In [5]: arq = pd.read_csv('snv_201807a_2.csv', nrows=5)
arq
```

Out[5]:

	UF	Unnamed: 1	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17,930.8

Carregando e armazenando dados

Lendo dados em formato texto

Podemos configurar o pandas para limitar o número máximo de linhas exibida. Fazemos isso utilizando `pd.options.display.max_rows=X`, onde X é o número de linhas. Vamos definir 8, assim, serão exibidas as 4 primeiras linhas e as 4 últimas linhas.

```
In [131]: pd.get_option("display.max_rows")
```

```
Out[131]: 50
```

```
In [132]: pd.options.display.max_rows=8
```

```
In [133]: pd.get_option("display.max_rows")
```

```
Out[133]: 8
```


Carregando e armazenando dados

Lendo dados em formato texto

Veja agora como será exibido o conteúdo da variável `arq`.

```
In [134]: arq = pd.read_csv('snv_201807a_2.csv')
          arq
```

Out[134]:

	UF	Unnamed: 1	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
...
29	RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1
30	SC	Santa Catarina	1,203.3	1.2	-	-	-	28.0	29.2	1,867.8	11.1	467.7	2,346.6	3,579.1
31	Sub-Total	NaN	6,519.3	7.7	-	-	141.9	149.4	299.0	9,818.4	370.6	1,593.8	11,782.8	18,601.1
32	NaN	NaN	44,201.7	147.1	1,965.2	140.6	6,067.7	2,108.9	10,429.5	57,865.0	1,268.8	6,793.2	65,927.0	120,558.2

Carregando e armazenando dados

Lendo dados em formato texto

Nosso arquivo possui um cabeçalho, mas podemos nos deparar com arquivos que não tenham um cabeçalho. Eu fiz uma cópia do arquivo retirando a primeira linha, que é o cabeçalho. Veja como ficará ao carregarmos este novo arquivo.

Carregando e armazenando dados

Lendo dados em formato texto

```
In [8]: arq = pd.read_csv('snv_201807a_2_semcabecalho.csv')
arq
```

Out[8]:

	DF	Distrito Federal	170.2	-	-.1	-.2	-.3	-.4	-.5	82.5	-.6	129.0	211.5	381.7
0	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
1	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
2	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
3	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17,930.8
...
28	RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1
29	SC	Santa Catarina	1,203.3	1.2	-	-	-	28.0	29.2	1,867.8	11.1	467.7	2,346.6	3,579.1
30	Sub-Total	NaN	6,519.3	7.7	-	-	141.9	149.4	299.0	9,818.4	370.6	1,593.8	11,782.8	18,601.1
31	NaN	NaN	44,201.7	147.1	1,965.2	140.6	6,067.7	2,108.9	10,429.5	57,865.0	1,268.8	6,793.2	65,927.0	120,558.2

32 rows × 14 columns

Carregando e armazenando dados

Lendo dados em formato texto

A primeira linha com os dados será utilizada como cabeçalho. Para isso, podemos usar o parâmetro `header=None`. Assim, o pandas vai gerar um cabeçalho, usando todas as linhas como conteúdo do arquivo.

Carregando e armazenando dados

Lendo dados em formato texto

```
In [9]: arq = pd.read_csv('snv_201807a_2_semcabecalho.csv', header=None)
arq
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
...
29	RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1
30	SC	Santa Catarina	1,203.3	1.2	-	-	-	28.0	29.2	1,867.8	11.1	467.7	2,346.6	3,579.1
31	Sub-Total	NaN	6,519.3	7.7	-	-	141.9	149.4	299.0	9,818.4	370.6	1,593.8	11,782.8	18,601.1
32	NaN	NaN	44,201.7	147.1	1,965.2	140.6	6,067.7	2,108.9	10,429.5	57,865.0	1,268.8	6,793.2	65,927.0	120,558.2

33 rows × 14 columns

Carregando e armazenando dados

Lendo dados em formato texto

Podemos também especificar o nome dos cabeçalhos das colunas.

```
names = ['Sigla UF', 'Descrição UF', 'PLANEJADA', 'TRAVESSIA', 'LEITO NATURAL', 'EM OBRAS IMP', 'IMPLANT',  
         'EM OBRAS PAV', 'SUB-TOTAL', 'PISTA SIMPLES', 'EM OBRAS DUP', 'PISTA DUPLA', 'SUB-TOTAL.1', 'TOTAL']  
arq = pd.read_csv('snv_201807a_2_semcabecalho.csv', names=names)  
arq
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATURAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
...
29	RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1

Carregando e armazenando dados

Lendo dados em formato texto

Vamos agora utilizar a coluna com a descrição dos estados (Descrição UF) como índice do DataFrame. Para isso, podemos informar a coluna no índice 1 com `index_col=1` ou de nome 'Descrição UF' com `index_col='Descrição UF'`.

Carregando e armazenando dados

Lendo dados em formato texto

```
names = ['Sigla UF', 'Descrição UF', 'PLANEJADA', 'TRAVESSIA', 'LEITO NATUAL', 'EM OBRAS IMP', 'IMPLANT',  
         'EM OBRAS PAV', 'SUB-TOTAL', 'PISTA SIMPLES', 'EM OBRAS DUP', 'PISTA DUPLA', 'SUB-TOTAL.1', 'TOTAL']  
arq = pd.read_csv('snv_201807a_2_semcabecalho.csv', names=names, index_col=1)  
arq
```

	Sigla UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
Descrição UF													
Distrito Federal	DF	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
Goiás	GO	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
Mato Grosso do Sul	MS	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
Mato Grosso	MT	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
...
Rio Grande do Sul	RS	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1
Santa													

Carregando e armazenando dados

Lendo dados em formato texto

```
names = ['Sigla UF', 'Descrição UF', 'PLANEJADA', 'TRAVESSIA', 'LEITO NATUAL', 'EM OBRAS IMP', 'IMPLANT',  
         'EM OBRAS PAV', 'SUB-TOTAL', 'PISTA SIMPLES', 'EM OBRAS DUP', 'PISTA DUPLA', 'SUB-TOTAL.1', 'TOTAL']  
arq = pd.read_csv('snv_201807a_2_semcabecalho.csv', names=names, index_col='Descrição UF')  
arq
```

	Sigla UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
Descrição UF													
Distrito Federal	DF	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
Goiás	GO	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
Mato Grosso do Sul	MS	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
Mato Grosso	MT	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
...
Rio Grande do Sul	RS	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1

Carregando e armazenando dados

Lendo dados em formato texto

Podemos definir mais de uma coluna como índice. Para exemplificar, vamos usar o arquivo "snv_201807a_2_DoisIndices.csv", onde copiei a linha do DF para termos 3 linhas referentes à Sigla UF.

Carregando e armazenando dados

Lendo dados em formato texto

```
names = ['Sigla UF', 'Descrição UF', 'PLANEJADA', 'TRAVESSIA', 'LEITO NATUAL', 'EM OBRAS IMP', 'IMPLANT',  
         'EM OBRAS PAV', 'SUB-TOTAL', 'PISTA SIMPLES', 'EM OBRAS DUP', 'PISTA DUPLA', 'SUB-TOTAL.1', 'TOTAL']  
arq = pd.read_csv('snv_201807a_2_DoisIndices.csv', names=names, index_col=['Sigla UF', 'Descrição UF'])  
arq
```

		PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
Sigla UF	Descrição UF												
UF	NaN	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL	TOTAL
DF	Distrito Federal 1	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
	Distrito Federal 2	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
	Distrito Federal 3	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
...
RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1

Carregando e armazenando dados

Lendo dados em formato texto

Vamos agora aumentar a quantidade máxima de linhas para 50. E depois vamos carregar o arquivo “snv_201807a_2_semcabecalho.csv” novamente para uma variável denominada df.

```
In [27]: pd.options.display.max_rows=50
```

```
In [28]: df = pd.read_csv('snv_201807a_2_semcabecalho.csv', names=names)
```

Carregando e armazenando dados

Lendo dados em formato texto

Vamos utilizar ordenação e classificação. Assuntos vistos na aula anterior. Primeiro vamos ordenar o DataFrame pelo índice no eixo 0.

```
df.sort_index(axis=0)
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4,602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17,930.8
5	AL	Alagoas	100.5	-	49.0	-	-	4.4	53.4	473.6	182.7	91.5	747.8	901.7
6	BA	Bahia	4,006.4	39.8	497.7	-	272.4	258.7	1,068.6	6,029.7	69.8	125.9	6,225.4	11,300.4

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos ordenar pelo índice no eixo 1.

```
df.sort_index(axis=1)
```

	Descrição UF	EM OBRAS DUP	EM OBRAS IMP	EM OBRAS PAV	IMPLANT	LEITO NATUAL	PISTA DUPLA	PISTA SIMPLES	PLANEJADA	SUB- TOTAL	SUB- TOTAL.1	Sigla UF	TOTAL	TRAVESSIA
0	Distrito Federal	-	-	-	-	-	129.0	82.5	170.2	-	211.5	DF	381.7	-
1	Goiás	18.7	-	178.7	-	100.7	791.9	2,597.3	2,636.8	279.4	3,407.9	GO	6,324.1	-
2	Mato Grosso do Sul	-	-	36.2	1.4	225.5	70.1	3,706.9	562.6	263.1	3,777.0	MS	4,602.7	-
3	Mato Grosso	206.8	-	246.0	402.0	427.7	273.6	3,538.2	1,528.0	1,075.7	4,018.6	MT	6,622.3	-
4	NaN	225.5	-	460.9	403.4	753.9	1,264.6	9,924.9	4,897.6	1,618.2	11,415.0	Sub- Total	17,930.8	-
5	Alagoas	182.7	-	4.4	-	49.0	91.5	473.6	100.5	53.4	747.8	AL	901.7	-
6	Bahia	69.8	-	258.7	272.4	497.7	125.9	6,029.7	4,006.4	1,068.6	6,225.4	BA	11,300.4	39.8

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos ordenar pelos dados da coluna total em ordem crescente. Observe que a coluna será considerada como texto. Vai ser ordenado como texto.

```
df.sort_values(by="TOTAL")
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
17	AP	Amapá	193.0	-	-	-	542.5	11.5	554.0	467.4	-	-	467.4	1,214.4
15	AC	Acre	503.1	0.2	-	-	-	6.4	6.6	1,140.0	-	8.8	1,148.8	1,658.5
9	PB	Paraíba	388.0	-	18.3	-	0.5	7.7	26.5	998.6	2.9	273.3	1,274.8	1,689.3
23	ES	Espírito Santo	617.6	-	50.9	-	-	24.7	75.6	940.7	-	61.1	1,001.8	1,695.0
12	RN	Rio Grande do Norte	253.5	-	-	-	32.0	-	32.0	1,352.4	16.7	147.4	1,516.5	1,802.0
20	RR	Roraima	184.7	-	-	-	607.8	15.5	623.3	1,033.4	-	17.2	1,050.6	1,858.6
6	BA	Bahia	4,006.4	39.8	497.7	-	272.4	258.7	1,068.6	6,029.7	69.8	125.9	6,225.4	11,300.4
32	NaN	NaN	44,201.7	147.1	1,965.2	140.6	6,067.7	2,108.9	10,429.5	57,865.0	1,268.8	6,793.2	65,927.0	120,558.2

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos ordenar pelos dados da coluna total em ordem decrescente. Observe que a coluna será considerada como texto. Vai ser ordenado como texto.

```
df.sort_values(by="TOTAL", ascending=False)
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
5	AL	Alagoas	100.5	-	49.0	-	-	4.4	53.4	473.6	182.7	91.5	747.8	901.7
29	RS	Rio Grande do Sul	2,844.6	5.7	-	-	141.9	28.2	175.8	4,958.8	268.1	398.8	5,625.7	8,646.1
18	PA	Pará	2,558.6	64.7	109.0	-	1,504.3	721.0	2,399.0	2,652.4	-	70.6	2,723.0	7,680.6
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6,622.3
26	SP	São Paulo	5,427.0	-	-	-	-	-	-	486.6	-	635.8	1,122.4	6,549.4
28	PR	Paraná	2,471.4	0.8	-	-	-	93.2	94.0	2,991.8	91.4	727.3	3,810.5	6,375.9
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6,324.1
16	AM	Amazonas	3,803.0	30.5	-	-	1,546.7	86.2	1,663.4	700.3	-	2.8	703.1	6,169.5
13	SE	Sergipe	100.4	-	-	-	-	-	-	161.5	77.6	79.7	318.8	419.2

Carregando e armazenando dados

Lendo dados em formato texto

Vamos remover os espaços no campo TOTAL, por exemplo, onde está assim " 6,324.1" vai ficar assim "6,324.1".

```
df['TOTAL'] = df['TOTAL'].str.strip()
```

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos remover as vírgulas do conteúdo do campo TOTAL.

```
df['TOTAL'].replace(',', '', regex=True, inplace=True)
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
1	GO	Goiás	2,636.8	-	100.7	-	-	178.7	279.4	2,597.3	18.7	791.9	3,407.9	6324.1
2	MS	Mato Grosso do Sul	562.6	-	225.5	-	1.4	36.2	263.1	3,706.9	-	70.1	3,777.0	4602.7
3	MT	Mato Grosso	1,528.0	-	427.7	-	402.0	246.0	1,075.7	3,538.2	206.8	273.6	4,018.6	6622.3
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17930.8

Carregando e armazenando dados

Lendo dados em formato texto

Agora vou converter a coluna TOTAL para float.

```
df["TOTAL"] = pd.to_numeric(df["TOTAL"].astype(float))
```

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos ordenar novamente pelos valores da coluna TOTAL. Primeiro em ordem crescente.

```
df.sort_values(by="TOTAL")
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
0	DF	Distrito Federal	170.2	-	-	-	-	-	-	82.5	-	129.0	211.5	381.7
13	SE	Sergipe	100.4	-	-	-	-	-	-	161.5	77.6	79.7	318.8	419.2
5	AL	Alagoas	100.5	-	49.0	-	-	4.4	53.4	473.6	182.7	91.5	747.8	901.7
17	AP	Amapá	193.0	-	-	-	542.5	11.5	554.0	467.4	-	-	467.4	1214.4
15	AC	Acre	503.1	0.2	-	-	-	6.4	6.6	1,140.0	-	8.8	1,148.8	1658.5
9	PB	Paraíba	388.0	-	18.3	-	0.5	7.7	26.5	998.6	2.9	273.3	1,274.8	1689.3
23	ES	Espírito Santo	617.6	-	50.9	-	-	24.7	75.6	940.7	-	61.1	1,001.8	1695.0
12	RN	Rio Grande do Norte	253.5	-	-	-	32.0	-	32.0	1,352.4	16.7	147.4	1,516.5	1802.0
20	RR	Roraima	184.7	-	-	-	607.8	15.5	623.3	1,033.4	-	17.2	1,050.6	1858.6
19	RO	Rondônia	165.0	1.2	-	-	182.4	50.7	234.3	1,795.9	-	88.7	1,884.6	2283.9
25	RJ	Rio de Janeiro	820.5	-	-	-	0.5	0.2	47.0	4,073.4	26.0	593.5	4,694.6	2540.0

Carregando e armazenando dados

Lendo dados em formato texto

Depois em ordem decrescente.

```
df.sort_values(by="TOTAL", ascending=False)
```

	Sigla UF	Descrição UF	PLANEJADA	TRAVESSIA	LEITO NATUAL	EM OBRAS IMP	IMPLANT	EM OBRAS PAV	SUB-TOTAL	PISTA SIMPLES	EM OBRAS DUP	PISTA DUPLA	SUB-TOTAL.1	TOTAL
32	NaN	NaN	44,201.7	147.1	1,965.2	140.6	6,067.7	2,108.9	10,429.5	57,865.0	1,268.8	6,793.2	65,927.0	120558.2
14	Sub-Total	NaN	9,324.1	39.8	659.7	80.9	730.1	315.2	1,825.7	18,599.9	512.2	1,279.5	20,391.6	31541.4
27	Sub-Total	NaN	15,427.1	1.1	242.9	-	345.9	207.6	797.5	10,086.6	160.5	2,404.3	12,651.4	28876.0
22	Sub-Total	NaN	8,033.6	98.5	308.7	59.7	4,446.4	975.8	5,889.1	9,435.2	-	251.0	9,686.2	23608.9
31	Sub-Total	NaN	6,519.3	7.7	-	-	141.9	149.4	299.0	9,818.4	370.6	1,593.8	11,782.8	18601.1
24	MG	Minas Gerais	8,543.0	1.1	192.0	-	337.3	173.6	704.0	7,587.2	134.5	1,113.9	8,835.6	18082.6
4	Sub-Total	NaN	4,897.6	-	753.9	-	403.4	460.9	1,618.2	9,924.9	225.5	1,264.6	11,415.0	17930.8

Carregando e armazenando dados

Lendo dados em formato texto

Vamos agora carregar um arquivo contendo a lista de times que ganharam o Brasileirão desde o primeiro campeonato até o de 2018.

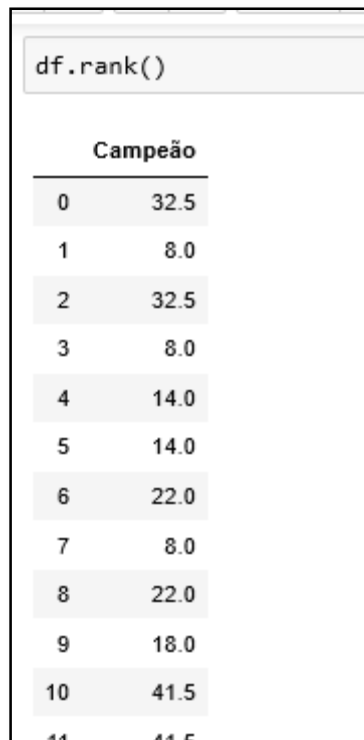
```
df = pd.read_csv('BrasileiraoTimes.csv')  
df
```

	Campeão
0	Palmeiras
1	Corinthians
2	Palmeiras
3	Corinthians
4	Cruzeiro
5	Cruzeiro
6	Fluminense
7	Corinthians
8	Fluminense
9	Flamengo
10	São Paulo
11	São Paulo

Carregando e armazenando dados

Lendo dados em formato texto

Agora vamos gerar o ranking de acordo com a quantidade de vezes que um determinado time aparece na lista.



```
df.rank()
```

	Campeão
0	32.5
1	8.0
2	32.5
3	8.0
4	14.0
5	14.0
6	22.0
7	8.0
8	22.0
9	18.0
10	41.5
11	41.5

Carregando e armazenando dados

Lendo dados em formato texto

Vamos agora carregar os dados de um arquivo cujas colunas são separadas por espaços, porém, não há um padrão, o número de espaços é variado.

Veja o conteúdo do arquivo:

NOME	IDADE	SEXO	UF
EVALDO	41	M	ES
MARIA	30	F	SP
JOAO	45	M	RJ
JOSE	15	M	MG

```
df = pd.read_csv('colunas_espaco_variado.txt')  
df
```

	NOME	IDADE	SEXO	UF
0	EVALDO	41	M	ES
1	MARIA	30	F	SP
2	JOAO	45	M	RJ
3	JOSE	15	M	MG

Carregando e armazenando dados

Lendo dados em formato texto

Ao tentar visualizar os dados da coluna NOME, um erro vai ocorrer.

```
df['NOME']

-----
KeyError                                Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2896         try:
-> 2897             return self._engine.get_loc(key)
    2898         except KeyError:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'NOME'

During handling of the above exception, another exception occurred:
```

Carregando e armazenando dados

Lendo dados em formato texto

Para resolver este problema vamos informar uma expressão regular como separador. A expressão "\s" indica um espaço, adicionando o sinal de "+", indica um ou mais espaços.

```
df = pd.read_csv('colunas_espaco_variado.txt', sep='\s+')  
df
```

	NOME	IDADE	SEXO	UF
0	EVALDO	41	M	ES
1	MARIA	30	F	SP
2	JOAO	45	M	RJ
3	JOSE	15	M	MG

```
df['NOME']
```

```
0    EVALDO  
1    MARIA  
2    JOAO  
3    JOSE  
Name: NOME, dtype: object
```

Carregando e armazenando dados

Lendo dados em formato texto

Agora temos um arquivo denominado "ignorar_linhas.txt" que tem 4 linhas que devem ser ignoradas:

```
# Lista contendo nome,  
# idade, sexo e uf  
#  
NOME,IDADE,SEXO,UF  
EVALDO,41,M,ES  
MARIA,30,F,SP  
JOAO,45,M,RJ  
JOSE,15,M,MG  
# Fim da lista
```

Para ignorar estas linhas podemos utilizar skiprows=lista, onde lista é uma lista de linhas a ignorar.

```
df = pd.read_csv('ignorar_linhas.txt', skiprows=[0,1,2,8])  
df
```

	NOME	IDADE	SEXO	UF
0	EVALDO	41	M	ES
1	MARIA	30	F	SP
2	JOAO	45	M	RJ
3	JOSE	15	M	MG

Carregando e armazenando dados

Lendo dados em formato texto

O `read_csv` também permite ler arquivos através de uma URL. A url que vamos utilizar retorna o arquivo `gapminder-FiveYearData.csv` que também está disponível junto aos arquivos desta aula. Se no momento que estiver assistindo a esta aula, o arquivo não estiver mais online, troque a URL pelo endereço local do arquivo e execute o exemplo normalmente. Esse arquivo tem 1705 linhas. Quando você for trabalhar com arquivos muito grandes, pode ser que prefira quebrar o arquivo em "pedaços", pegando assim, seu conteúdo, bloco por bloco. Vamos percorrer o arquivo deste exemplo, separando de 5 em 5 linhas. Para isso podemos usar o parâmetro `chunksize` (em tradução literal, tamanho do pedaço) do método `read_csv`.

Carregando e armazenando dados

Lendo dados em formato texto

```
url='http://bit.ly/2cLzoxH'
```

```
for pedaco in pd.read_csv(url, chunksize=5):  
    print("-----")  
    print(pedaco)  
    print("-----")
```

```
21  Albania  1997  3428038  Europe  72.950  3193.054604  
22  Albania  2002  3508512  Europe  75.651  4604.211737  
23  Albania  2007  3600523  Europe  76.423  5937.029526  
24  Algeria  1952  9279525  Africa  43.077  2449.008185
```

```
-----  
-----  
      country  year      pop  continent  lifeExp  gdpPercap  
25  Algeria  1957  10270856  Africa  45.685  3013.976023  
26  Algeria  1962  11000948  Africa  48.303  2550.816880  
27  Algeria  1967  12760499  Africa  51.407  3246.991771  
28  Algeria  1972  14760787  Africa  54.518  4182.663766  
29  Algeria  1977  17152804  Africa  58.014  4910.416756
```

```
-----  
-----  
      country  year      pop  continent  lifeExp  gdpPercap  
30  Algeria  1982  20033753  Africa  61.368  5745.160213  
31  Algeria  1987  23254956  Africa  65.799  5681.358539  
32  Algeria  1992  26298373  Africa  67.744  5023.216647
```

FIM