

Carregando e armazenando dados em vários formatos de arquivos

Escrevendo dados em formato texto

Carregando e armazenando dados

Escrevendo dados em formato texto

Podemos exportar dados de um DataFrame para arquivos de texto usando o método `to_csv()` do DataFrame.

Vamos começar importando o Pandas e o Numpy.

```
import pandas as pd
import numpy as np
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Agora vamos gerar um DataFrame com dados aleatórios, contendo 100.000 linhas e 26 colunas. Vamos usar as letras do alfabeto para identificar as colunas.

```
df = pd.DataFrame(np.random.randn(100000, 26),  
                  columns=list('ABCDEFGHIJKLMNOPQRSTUVWXYZ'))
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Como você poderá ver na imagem a seguir, ao visualizar o conteúdo do Dataframe veremos apenas 8 linhas, sendo, quatro do início e quatro do final do DataFrame.

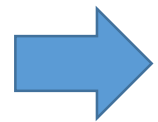
Isso porque foi definido o número máximo de linhas para oito com o comando:

```
pd.options.display.max_rows=8.
```

Carregando e armazenando dados

Escrevendo dados em formato texto

df												
	A	B	C	D	E	F	G	H	I	J	...	Q
0	0.485033	-0.345434	-0.393798	1.614686	-0.001967	0.100296	0.166128	0.652852	0.640477	-1.639274	...	-0.633519
1	-1.016669	0.649413	-0.369399	0.945271	0.040730	1.099865	-0.601038	-1.014448	1.110866	0.069572	...	0.581508
2	1.070810	-2.666222	-1.113743	-0.067586	-0.124944	0.008870	0.589807	0.108029	-0.264044	0.183298	...	-0.833616
3	0.093775	1.623235	2.905946	-0.042831	0.413616	0.764938	0.653329	1.082971	0.633434	1.216340	...	-0.177359
...
99996	1.320003	0.575220	0.822213	0.208702	1.086502	0.784380	0.202163	0.182784	0.568002	1.041433	...	0.209622
99997	-0.858624	-0.286410	1.084309	0.204710	0.094744	-0.055045	-0.594217	0.847468	0.077236	-1.485857	...	-1.321139
99998	0.843541	1.301511	0.441663	-1.363562	0.475709	0.450311	0.022528	-2.243839	-0.603900	-0.587130	...	-0.601389
99999	-1.797786	-0.110297	-1.156075	-0.601392	-0.775781	-0.107749	0.357712	-0.388636	-1.389150	-0.002033	...	-0.276535
100000 rows × 26 columns												



Carregando e armazenando dados

Escrevendo dados em formato texto

Agora vamos exportar o conteúdo de nosso DataFrame para um arquivo chamado “dados1.csv”.

```
df.to_csv('dados1.csv')
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Enquanto o comando estiver sendo executado, você verá um asterisco em vez do número da célula do Jupyter Notebook.

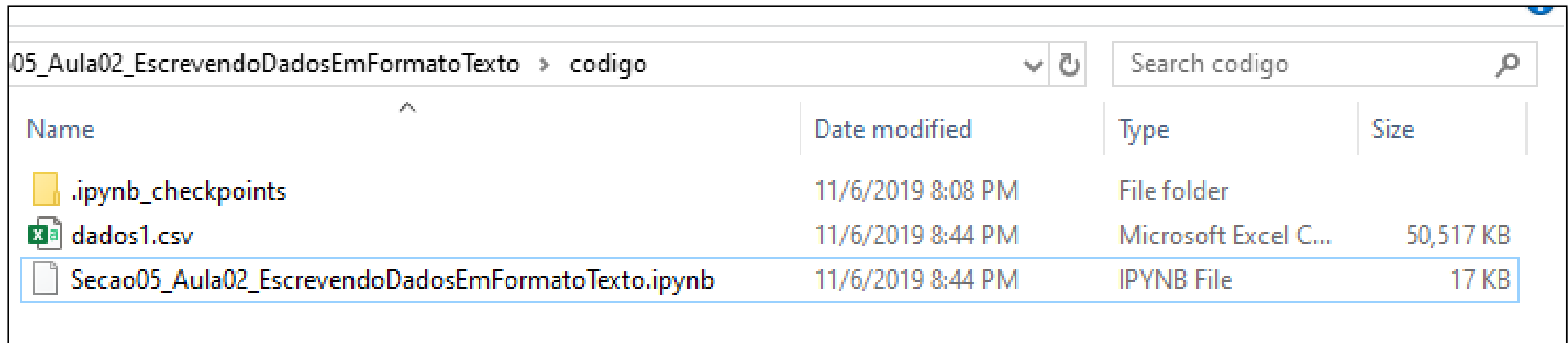
```
In [*]: df.to_csv('dados1.csv')
```



Carregando e armazenando dados

Escrevendo dados em formato texto

Verifique na pasta onde está sendo executado o Jupyter Notebook o arquivo gerado.



05_Aula02_EscrevendoDadosEmFormatoTexto > codigo				Search codigo
Name	Date modified	Type	Size	
.ipynb_checkpoints	11/6/2019 8:08 PM	File folder		
dados1.csv	11/6/2019 8:44 PM	Microsoft Excel C...	50,517 KB	
Secao05_Aula02_EscrevendoDadosEmFormatoTexto.ipynb	11/6/2019 8:44 PM	IPYNB File	17 KB	

Carregando e armazenando dados

Escrevendo dados em formato texto

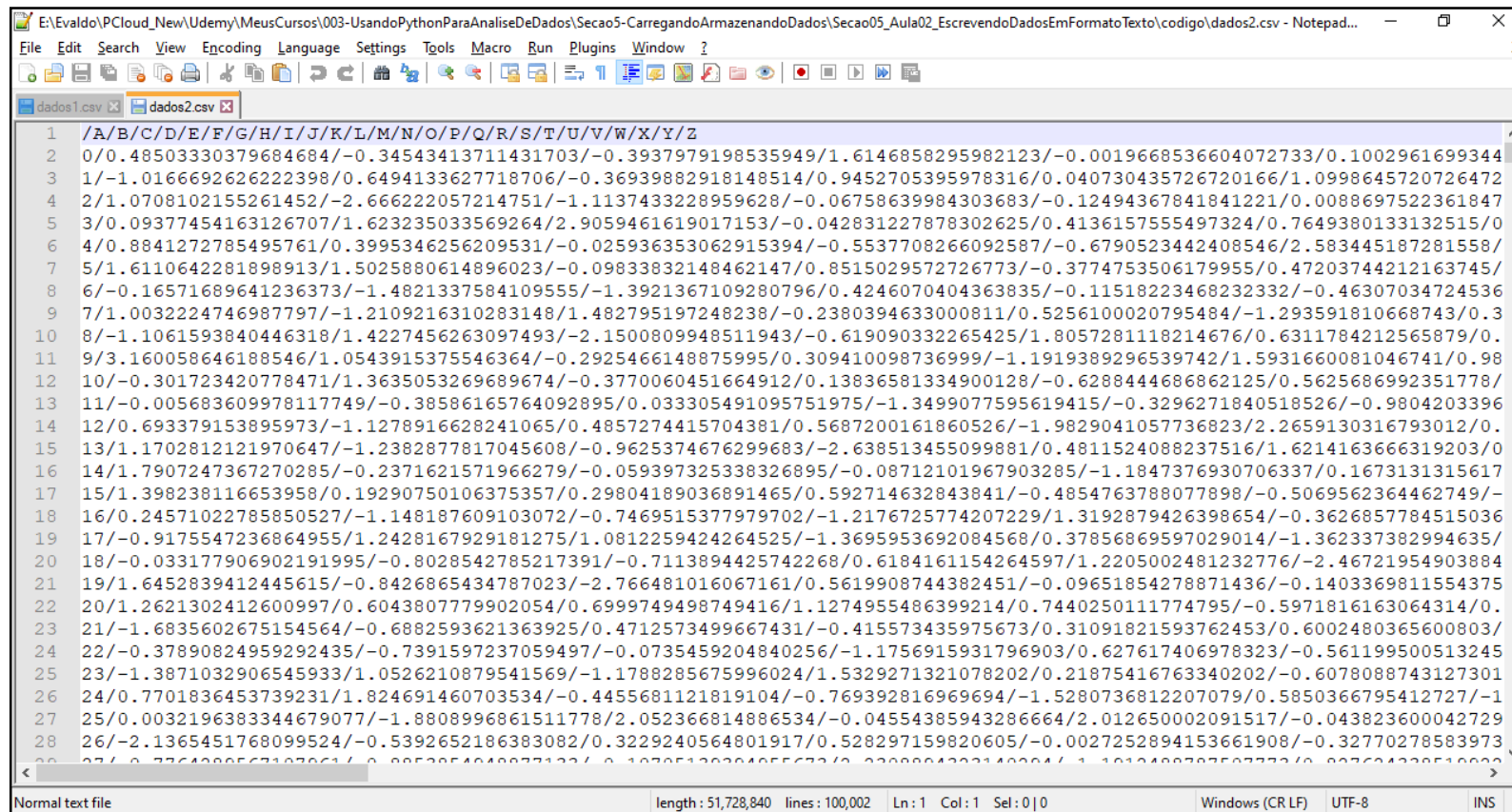
Por padrão o delimitador utilizado é a vírgula, porém, podemos definir um delimitador diferente utilizando `sep=delimitador`.

```
df.to_csv('dados2.csv', sep='/')
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Veja o resultado:



```
E:\Evaldo\PCloud_New\Udemy\MeusCursos\003-UsandoPythonParaAnaliseDeDados\Secao5- CarregandoArmazenandoDados\Secao05_Aula02_EscrevendoDadosEmFormatoTexto\codigo\dados2.csv - Notepad...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
dados1.csv dados2.csv
1 /A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z
2 0/0.48503330379684684/-0.34543413711431703/-0.3937979198535949/1.6146858295982123/-0.0019668536604072733/0.1002961699344
3 1/-1.0166692626222398/0.6494133627718706/-0.36939882918148514/0.9452705395978316/0.040730435726720166/1.0998645720726472
4 2/1.0708102155261452/-2.666222057214751/-1.1137433228959628/-0.06758639984303683/-0.12494367841841221/0.0088697522361847
5 3/0.09377454163126707/1.623235033569264/2.9059461619017153/-0.042831227878302625/0.4136157555497324/0.7649380133132515/0
6 4/0.8841272785495761/0.3995346256209531/-0.025936353062915394/-0.5537708266092587/-0.6790523442408546/2.583445187281558/
7 5/1.6110642281898913/1.5025880614896023/-0.09833832148462147/0.8515029572726773/-0.3774753506179955/0.47203744212163745/
8 6/-0.16571689641236373/-1.4821337584109555/-1.3921367109280796/0.4246070404363835/-0.11518223468232332/-0.46307034724536
9 7/1.0032224746987797/-1.2109216310283148/1.482795197248238/-0.2380394633000811/0.5256100020795484/-1.293591810668743/0.3
10 8/-1.1061593840446318/1.4227456263097493/-2.1500809948511943/-0.619090332265425/1.8057281118214676/0.6311784212565879/0.
11 9/3.160058646188546/1.0543915375546364/-0.2925466148875995/0.309410098736999/-1.1919389296539742/1.5931660081046741/0.98
12 10/-0.301723420778471/1.3635053269689674/-0.3770060451664912/0.13836581334900128/-0.6288444686862125/0.5625686992351778/
13 11/-0.005683609978117749/-0.38586165764092895/0.033305491095751975/-1.3499077595619415/-0.3296271840518526/-0.9804203396
14 12/0.693379153895973/-1.1278916628241065/0.4857274415704381/0.5687200161860526/-1.9829041057736823/2.2659130316793012/0.
15 13/1.1702812121970647/-1.2382877817045608/-0.9625374676299683/-2.638513455099881/0.4811524088237516/1.6214163666319203/0
16 14/1.7907247367270285/-0.2371621571966279/-0.059397325338326895/-0.08712101967903285/-1.1847376930706337/0.1673131315617
17 15/1.398238116653958/0.19290750106375357/0.29804189036891465/0.592714632843841/-0.4854763788077898/-0.5069562364462749/-
18 16/0.24571022785850527/-1.148187609103072/-0.7469515377979702/-1.2176725774207229/1.3192879426398654/-0.3626857784515036
19 17/-0.9175547236864955/1.2428167929181275/1.0812259424264525/-1.3695953692084568/0.37856869597029014/-1.362337382994635/
20 18/-0.033177906902191995/-0.8028542785217391/-0.7113894425742268/0.6184161154264597/1.2205002481232776/-2.46721954903884
21 19/1.6452839412445615/-0.8426865434787023/-2.766481016067161/0.5619908744382451/-0.09651854278871436/-0.1403369811554375
22 20/1.2621302412600997/0.6043807779902054/0.6999749498749416/1.1274955486399214/0.7440250111774795/-0.5971816163064314/0.
23 21/-1.6835602675154564/-0.6882593621363925/0.4712573499667431/-0.415573435975673/0.31091821593762453/0.6002480365600803/
24 22/-0.37890824959292435/-0.7391597237059497/-0.0735459204840256/-1.1756915931796903/0.627617406978323/-0.561199500513245
25 23/-1.3871032906545933/1.0526210879541569/-1.1788285675996024/1.5329271321078202/0.21875416763340202/-0.6078088743127301
26 24/0.7701836453739231/1.824691460703534/-0.4455681121819104/-0.769392816969694/-1.5280736812207079/0.5850366795412727/-1
27 25/0.0032196383344679077/-1.8808996861511778/2.052366814886534/-0.04554385943286664/2.012650002091517/-0.043823600042729
28 26/-2.1365451768099524/-0.5392652186383082/0.322940564801917/0.528297159820605/-0.0027252894153661908/-0.32770278583973
29 27/0.7364288567318783/0.8853854088777333/0.1382854088777333/0.2288842323142884/1.1032488287587773/0.827624238518823
30 28/0.7364288567318783/0.8853854088777333/0.1382854088777333/0.2288842323142884/1.1032488287587773/0.827624238518823
31 29/0.7364288567318783/0.8853854088777333/0.1382854088777333/0.2288842323142884/1.1032488287587773/0.827624238518823
32 30/0.7364288567318783/0.8853854088777333/0.1382854088777333/0.2288842323142884/1.1032488287587773/0.827624238518823
length: 51,728,840 lines: 100,002 Ln: 1 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Observe que os rótulos, tanto das linhas, quanto das colunas foram exportados.

df

	A	B	C	D	E	F	G	H	I	J	...	Q
0	0.485033	-0.345434	-0.393798	1.614686	-0.001967	0.100296	0.166128	0.652852	0.640477	-1.639274	...	-0.633519
1	-1.016669	0.649413	-0.369399	0.945271	0.040730	1.099865	-0.601038	-1.014448	1.110866	0.069572	...	0.581508
2	1.070810										...	0.837616
3	0.093775										...	
...	
99996	1.320003										...	0.209622
99997	-0.858624										...	-1.321139
99998	0.843541										...	0.601389
99999	-1.797786										...	0.276535

100000 rows x 26 columns

Carregando e armazenando dados

Escrevendo dados em formato texto

Para gerar o arquivo sem o rótulo das linhas podemos especificar o parâmetro `index=False`.

```
df.to_csv('dados_sem_rotulos_das_linhas.csv', index=False)
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Veja o arquivo gerado sem os rótulos das linhas.

	A	B	C	D	E	F	G	H	I	J	K	L
1	A	B	C	D	E	F	G	H	I	J	K	L
2	0.485033	-0.34543	-0.3938	1.614686	-0.00197	0.100296	0.166128	0.652852	0.640477	-1.63927	0.373376	
3	-1.01667	0.649413	-0.3694	0.945271	0.04073	1.099865	-0.60104	-1.01445	1.110866	0.069572	0.868292	
4	1.07081	-2.66622	-1.11374	-0.06759	-0.12494	0.00887	0.589807	0.108029	-0.26404	0.183298	-0.79842	
5	0.093775	1.623235	2.905946	-0.04283	0.413616	0.764938	0.653329	1.082971	0.633434	1.21634	-1.14833	
6	0.884127	0.399535	-0.02594	-0.55377	-0.67905	2.583445	-0.90981	-0.50481	-1.22965	0.469037	-1.52207	
7	1.611064	1.502588	-0.09834	0.851503	-0.37748	0.472037	0.050856	-1.70312	0.497343	-0.42849	0.358522	
8	-0.16572	-1.48213	-1.39214	0.424607	-0.11518	-0.46307	-1.07884	0.60352	-1.79175	1.144023	1.46721	
9	1.003222	-1.21092	1.482795	-0.23804	0.52561	-1.29359	0.326276	0.162173	0.19107	0.572869	0.317788	
10	1.18616	1.182715	0.15888	0.61888	1.885788	0.681178	0.857561	0.17688	0.756817	0.11786	0.888788	

Carregando e armazenando dados

Escrevendo dados em formato texto

Para gerar o arquivo sem o rótulo das colunas podemos especificar o parâmetro `header=False`.

```
df.to_csv('dados_sem_rotulos_das_colunas.csv', header=False)
```


Carregando e armazenando dados

Escrevendo dados em formato texto

Veja o arquivo gerado sem os rótulos das colunas.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0.485033	-0.34543	-0.3938	1.614686	-0.00197	0.100296	0.166128	0.652852	0.640477	-1.63927	0.373376	-0.5753
2	1	-1.01667	0.649413	-0.3694	0.945271	0.04073	1.099865	-0.60104	-1.01445	1.110866	0.069572	0.868292	0.952364
3	2	1.07081	-2.66622	-1.11374	-0.06759	-0.12494	0.00887	0.589807	0.108029	-0.26404	0.183298	-0.79842	-0.3135
4	3	0.093775	1.623235	2.905946	-0.04283	0.413616	0.764938	0.653329	1.082971	0.633434	1.21634	-1.14833	0.277991
5	4	0.884127	0.399535	-0.02594	-0.55377	-0.67905	2.583445	-0.90981	-0.50481	-1.22965	0.469037	-1.52207	0.566381
6	5	1.611064	1.502588	-0.09834	0.851503	-0.37748	0.472037	0.050856	-1.70312	0.497343	-0.42849	0.358522	0.82168
7	6	-0.16572	-1.48213	-1.39214	0.424607	-0.11518	-0.46307	-1.07884	0.60352	-1.79175	1.144023	1.46721	-0.6989
8	7	1.003222	-1.21092	1.482795	-0.23804	0.52561	-1.29359	0.326276	0.162173	0.19107	0.572869	0.317788	0.876496
9	8	-1.10616	1.422746	-2.15008	-0.61909	1.805728	0.631178	0.957564	-0.47632	0.776817	2.44706	0.823722	-1.61492
10	9	3.160059	1.054392	-0.29255	0.30941	-1.19194	1.593166	0.986535	1.585831	-1.22012	-0.73862	-0.57536	-0.95804
11	10	-0.30172	1.363505	-0.37701	0.138366	-0.62884	0.562569	-0.14865	1.125008	2.615064	-0.02688	0.52468	0.749607
12	11	-0.00568	-0.38586	0.033305	-1.34991	-0.32963	-0.98042	-0.52471	1.251425	2.154576	-0.10353	0.714514	-1.96366

Carregando e armazenando dados

Escrevendo dados em formato texto

Podemos também especificar as colunas que queremos exportar e sua ordem.

```
df.to_csv('dados_colunas_especificas.csv', columns=['A', 'K', 'X', 'B', 'W'])
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Veja o arquivo gerado.

	A	B	C	D	E	F	G
1		A	K	X	B	W	
2	0	0.485033	0.373376	0.364107	-0.34543	0.803969	
3	1	-1.01667	0.868292	-0.17356	0.649413	0.883211	
4	2	1.07081	-0.79842	1.282296	-2.66622	2.279548	
5	3	0.093775	-1.14833	-0.12561	1.623235	-0.39745	
6	4	0.884127	-1.52207	-0.51297	0.399535	-0.38855	
7	5	1.611064	0.358522	0.845694	1.502588	-0.88397	
8	6	-0.16572	1.46721	0.767377	-1.48213	1.490818	
9	7	1.003222	0.317788	-0.92182	-1.21092	-1.57623	
10	8	-1.10616	0.823722	-0.79954	1.422746	-0.14047	
11	9	3.160059	-0.57536	0.396826	1.054392	0.043516	
12	10	-0.30172	0.52468	0.573465	1.363505	0.702675	

Carregando e armazenando dados

Escrevendo dados em formato texto

Podemos também exportar os dados de uma Series pois este objeto também possui o método `to_csv()`.

Vamos utilizar o `date_range` do Pandas para gerar uma sequência de datas para utilizarmos como índice de nossa Serie. O índice conterá as datas de 01/01/2019 a 31/01/2019, para isso informamos os parâmetros `start` e `end`.

Carregando e armazenando dados

Escrevendo dados em formato texto

```
datas = pd.date_range(start='01/01/2019', end='31/01/2019')
```

```
datas
```

```
DatetimeIndex(['2019-01-01', '2019-01-02', '2019-01-03', '2019-01-04',  
               '2019-01-05', '2019-01-06', '2019-01-07', '2019-01-08',  
               '2019-01-09', '2019-01-10', '2019-01-11', '2019-01-12',  
               '2019-01-13', '2019-01-14', '2019-01-15', '2019-01-16',  
               '2019-01-17', '2019-01-18', '2019-01-19', '2019-01-20',  
               '2019-01-21', '2019-01-22', '2019-01-23', '2019-01-24',  
               '2019-01-25', '2019-01-26', '2019-01-27', '2019-01-28',  
               '2019-01-29', '2019-01-30', '2019-01-31'],  
              dtype='datetime64[ns]', freq='D')
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Agora vamos gerar uma sequência com 31 valores, utilizando a variável `datas` como índice.

```
serie = pd.Series(np.arange(31), index=datas)
```

serie

2019-01-01	0
2019-01-02	1
2019-01-03	2
2019-01-04	3

..

2019-01-28	27
2019-01-29	28
2019-01-30	29
2019-01-31	30

Freq: D, Length: 31, dtype: int32

Carregando e armazenando dados

Escrevendo dados em formato texto

Vamos agora exportar os dados da Serie para um arquivo.

```
serie.to_csv('serie.csv')
```

```
C:\Users\Wolkers\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: The signature of `Series.to_csv` was aligned to that of `DataFrame.to_csv`, and argument 'header' will change its default value from False to True: please pass an explicit value to suppress this warning.
```

```
"""Entry point for launching an IPython kernel.
```

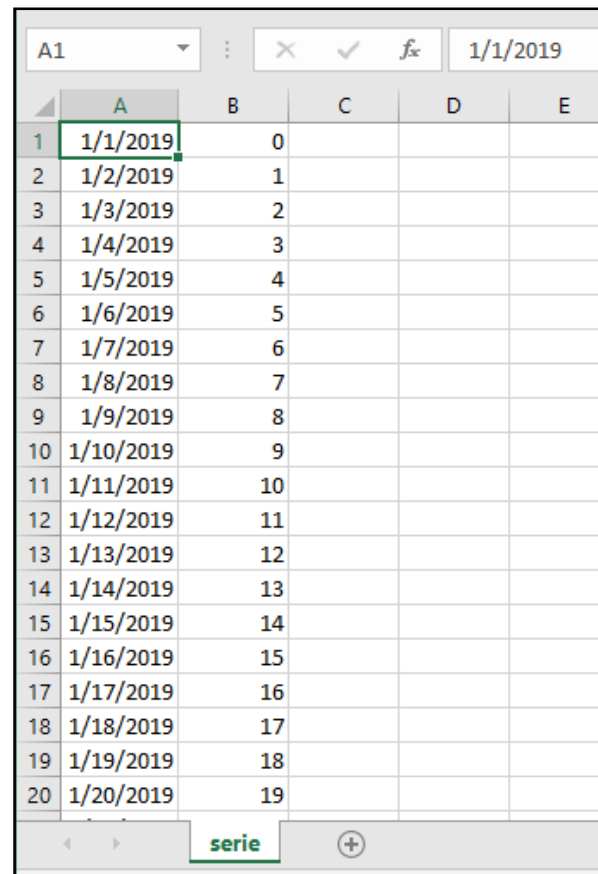
Este aviso diz que a assinatura do método `to_csv` das Series foi alinhada com a dos DataFrames e o argumento `header` terá seu valor padrão alterado de `False` para `True`. Agora, precisamos definir o valor do `header` para suprimir este aviso.

```
serie.to_csv('serie.csv', header=False)
```

Carregando e armazenando dados

Escrevendo dados em formato texto

Arquivo gerado:



The screenshot shows a spreadsheet with two columns of data. Column A contains dates from 1/1/2019 to 1/20/2019, and Column B contains a numerical series from 0 to 19. The spreadsheet interface includes a formula bar at the top showing '1/1/2019' and a status bar at the bottom with the word 'serie'.

	A	B	C	D	E
1	1/1/2019	0			
2	1/2/2019	1			
3	1/3/2019	2			
4	1/4/2019	3			
5	1/5/2019	4			
6	1/6/2019	5			
7	1/7/2019	6			
8	1/8/2019	7			
9	1/9/2019	8			
10	1/10/2019	9			
11	1/11/2019	10			
12	1/12/2019	11			
13	1/13/2019	12			
14	1/14/2019	13			
15	1/15/2019	14			
16	1/16/2019	15			
17	1/17/2019	16			
18	1/18/2019	17			
19	1/19/2019	18			
20	1/20/2019	19			

FIM