Data Loading, Storage,

In [1]:
```
import numpy as np
import pandas as pd
np.random.seed(12345)
import matplotlib.pyplot as plt
plt.rc('figure', figsize=(10, 6))
np.set_printoptions(precision=4, suppress=True)
```

Reading and Writing Data in Text Format

In [8]:
```
!cat examples/ex1.csv
```

```
'cat' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

In [9]:
```
df = pd.read_csv('examples/ex1.csv')
df
```

Out[9]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [10]:
```
pd.read_table('examples/ex1.csv', sep=',')
```

Out[10]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [ ]:
```
!cat examples/ex2.csv
```

In [11]:
```
pd.read_csv('examples/ex2.csv', header=None)
```

Out[11]:

	0	1	2	3	4
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [12]:
```
pd.read_csv('examples/ex2.csv', names=['a', 'b', 'c', 'd', 'message'])
```

Out[12]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [13]:
```
names = ['a', 'b', 'c', 'd', 'message']
pd.read_csv('examples/ex2.csv', names=names, index_col='message')
```

Out[13]:

	a	b	c	d
message				
hello	1	2	3	4
world	5	6	7	8
foo	9	10	11	12

In [14]:
```
!cat examples/csv_mindex.csv
parsed = pd.read_csv('examples/csv_mindex.csv',
                     index_col=['key1', 'key2'])
parsed
```

```
'cat' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

Out[14]:

		value1	value2
key1	**key2**		
one	**a**	1	2
	b	3	4
	c	5	6
	d	7	8
two	**a**	9	10
	b	11	12
	c	13	14
	d	15	16

In [15]:
```
list(open('examples/ex3.txt'))
```

Out[15]:
```
['            A         B         C\n',
 'aaa -0.264438 -1.026059 -0.619500\n',
 'bbb  0.927272  0.302904 -0.032399\n',
 'ccc -0.264273 -0.386314 -0.217601\n',
 'ddd -0.871858 -0.348382  1.100491\n']
```

In [16]:
```
result = pd.read_table('examples/ex3.txt', sep='\s+')
result
```

Out[16]:

	A	B	C
aaa	-0.264438	-1.026059	-0.619500
bbb	0.927272	0.302904	-0.032399
ccc	-0.264273	-0.386314	-0.217601
ddd	-0.871858	-0.348382	1.100491

In [19]:
```
pd.read_csv('examples/ex4.csv')
```

Out[19]:

					# hey!
	a	b	c	d	message
# just wanted to make things more difficult for you	NaN	NaN	NaN	NaN	
# who reads CSV files with computers	anyway?	NaN	NaN	NaN	
1	2	3	4	hello	
5	6	7	8	world	
9	10	11	12	foo	

In [20]:
```
pd.read_csv('examples/ex4.csv', skiprows=[0, 2, 3])
```

Out[20]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [22]:
```
result = pd.read_csv('examples/ex5.csv')
result
```

Out[22]:

	something	a	b	c	d	message
0	one	1	2	3.0	4	NaN
1	two	5	6	NaN	8	world
2	three	9	10	11.0	12	foo

In [23]:
```
pd.isnull(result)
```

Out[23]:

	something	a	b	c	d	message
0	False	False	False	False	False	True
1	False	False	False	True	False	False
2	False	False	False	False	False	False

In [24]:
```
result = pd.read_csv('examples/ex5.csv', na_values=['NULL'])
result
```

Out[24]:

	something	a	b	c	d	message
0	one	1	2	3.0	4	NaN
1	two	5	6	NaN	8	world
2	three	9	10	11.0	12	foo

In [25]:
```
sentinels = {'message': ['foo', 'NA'], 'something': ['two']}
pd.read_csv('examples/ex5.csv', na_values=sentinels)
```

Out[25]:

	something	a	b	c	d	message
0	one	1	2	3.0	4	NaN
1	NaN	5	6	NaN	8	world
2	three	9	10	11.0	12	NaN

Reading Text Files in Pieces

In [26]:
```
pd.options.display.max_rows = 10
```

In [27]:
```
result = pd.read_csv('examples/ex6.csv')
result
```

Out[27]:

	one	two	three	four	key
0	0.467976	-0.038649	-0.295344	-1.824726	L
1	-0.358893	1.404453	0.704965	-0.200638	B
2	-0.501840	0.659254	-0.421691	-0.057688	G
3	0.204886	1.074134	1.388361	-0.982404	R
4	0.354628	-0.133116	0.283763	-0.837063	Q
...	...	...	...	...	...
9995	2.311896	-0.417070	-1.409599	-0.515821	L
9996	-0.479893	-0.650419	0.745152	-0.646038	E
9997	0.523331	0.787112	0.486066	1.093156	K
9998	-0.362559	0.598894	-1.843201	0.887292	G
9999	-0.096376	-1.012999	-0.657431	-0.573315	0

10000 rows × 5 columns

In [28]:
```
pd.read_csv('examples/ex6.csv', nrows=5)
```

Out[28]:

	one	two	three	four	key
0	0.467976	-0.038649	-0.295344	-1.824726	L
1	-0.358893	1.404453	0.704965	-0.200638	B
2	-0.501840	0.659254	-0.421691	-0.057688	G
3	0.204886	1.074134	1.388361	-0.982404	R
4	0.354628	-0.133116	0.283763	-0.837063	Q

In [29]:
```
chunker = pd.read_csv('examples/ex6.csv', chunksize=1000)
chunker
```

Out[29]:
```
<pandas.io.parsers.readers.TextFileReader at 0x11a5fcdee20>
```

In [30]:
```
chunker = pd.read_csv('examples/ex6.csv', chunksize=1000)

tot = pd.Series([])
for piece in chunker:
    tot = tot.add(piece['key'].value_counts(), fill_value=0)

tot = tot.sort_values(ascending=False)
```

```
C:\Users\Usuário\AppData\Local\Temp\ipykernel_18392\3012506434.py:3: FutureWarnin
g: The default dtype for empty Series will be 'object' instead of 'float64' in a f
uture version. Specify a dtype explicitly to silence this warning.
  tot = pd.Series([])
```

In [31]:
```
tot[:10]
```

Out[31]:
```
E    368.0
X    364.0
L    346.0
O    343.0
Q    340.0
M    338.0
J    337.0
F    335.0
K    334.0
H    330.0
dtype: float64
```

Writing Data to Text Format

In [32]:
```
data = pd.read_csv('examples/ex5.csv')
data
```

Out[32]:

	something	a	b	c	d	message
0	one	1	2	3.0	4	NaN
1	two	5	6	NaN	8	world
2	three	9	10	11.0	12	foo

In [33]:
```
data.to_csv('examples/out.csv')
```

In [34]:
```
import sys
data.to_csv(sys.stdout, sep='|')
```

```
|something|a|b|c|d|message
0|one|1|2|3.0|4|
1|two|5|6||8|world
2|three|9|10|11.0|12|foo
```

In [35]:
```
data.to_csv(sys.stdout, na_rep='NULL')
```

```
,something,a,b,c,d,message
0,one,1,2,3.0,4,NULL
1,two,5,6,NULL,8,world
2,three,9,10,11.0,12,foo
```

In [36]:
```
data.to_csv(sys.stdout, index=False, header=False)
```

```
one,1,2,3.0,4,
two,5,6,,8,world
three,9,10,11.0,12,foo
```

In [37]:
```
data.to_csv(sys.stdout, index=False, columns=['a', 'b', 'c'])
```

```
a,b,c
1,2,3.0
5,6,
9,10,11.0
```

In [39]:
```
dates = pd.date_range('1/1/2000', periods=7)
ts = pd.Series(np.arange(7), index=dates)
ts.to_csv('examples/tseries.csv')

data2 = pd.read_csv('examples/tseries.csv')
data2
```

Out[39]:

	Unnamed: 0	0
0	2000-01-01	0
1	2000-01-02	1
2	2000-01-03	2
3	2000-01-04	3
4	2000-01-05	4
5	2000-01-06	5
6	2000-01-07	6

Working with Delimited Formats

In [ ]:
```
!cat examples/ex7.csv
```

In [40]:
```
import csv
f = open('examples/ex7.csv')

reader = csv.reader(f)
```

In [41]:
```
for line in reader:
    print(line)
```

```
['a', 'b', 'c']
['1', '2', '3']
['1', '2', '3']
```

In [42]:
```
with open('examples/ex7.csv') as f:
    lines = list(csv.reader(f))
```

In [43]:
```
header, values = lines[0], lines[1:]
```

In [44]:
```
data_dict = {h: v for h, v in zip(header, zip(*values))}
data_dict
```

Out[44]:
```
{'a': ('1', '1'), 'b': ('2', '2'), 'c': ('3', '3')}
```

In [45]:
```
class my_dialect(csv.Dialect):
    lineterminator = '\n'
    delimiter = ';'
    quotechar = '"'
    quoting = csv.QUOTE_MINIMAL
```

In [46]:
```
reader = csv.reader(f, dialect=my_dialect)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_18392\394569316.py in <module>
----> 1 reader = csv.reader(f, dialect=my_dialect)

ValueError: I/O operation on closed file.
```

In [47]:
```
reader = csv.reader(f, delimiter='|')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_18392\1872722919.py in <module>
----> 1 reader = csv.reader(f, delimiter='|')

ValueError: I/O operation on closed file.
```

In [48]:
```
with open('mydata.csv', 'w') as f:
    writer = csv.writer(f, dialect=my_dialect)
    writer.writerow(('one', 'two', 'three'))
    writer.writerow(('1', '2', '3'))
    writer.writerow(('4', '5', '6'))
    writer.writerow(('7', '8', '9'))
```

JSON Data

In [49]:
```
obj = """
{"name": "Wes",
 "places_lived": ["United States", "Spain", "Germany"],
 "pet": null,
 "siblings": [{"name": "Scott", "age": 30, "pets": ["Zeus", "Zuko"]},
              {"name": "Katie", "age": 38,
               "pets": ["Sixes", "Stache", "Cisco"]}]
}
"""
```

In [50]:
```
import json
result = json.loads(obj)
result
```

Out[50]:
```
{'name': 'Wes',
 'places_lived': ['United States', 'Spain', 'Germany'],
 'pet': None,
 'siblings': [{'name': 'Scott', 'age': 30, 'pets': ['Zeus', 'Zuko']},
  {'name': 'Katie', 'age': 38, 'pets': ['Sixes', 'Stache', 'Cisco']}]}
```

In [52]:
```
asjson = json.dumps(result)
asjson
```

Out[52]:
```
'{"name": "Wes", "places_lived": ["United States", "Spain", "Germany"], "pet": nul
l, "siblings": [{"name": "Scott", "age": 30, "pets": ["Zeus", "Zuko"]}, {"name":
"Katie", "age": 38, "pets": ["Sixes", "Stache", "Cisco"]}]}'
```

In [53]:
```
siblings = pd.DataFrame(result['siblings'], columns=['name', 'age'])
siblings
```

Out[53]:

	name	age
0	Scott	30
1	Katie	38

In [54]:
```
!cat examples/example.json
```

```
'cat' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

In [55]:
```
data = pd.read_json('examples/example.json')
data
```

Out[55]:

	a	b	c
0	1	2	3
1	4	5	6
2	7	8	9

In [56]:
```
print(data.to_json())
```

```
{"a":{"0":1,"1":4,"2":7},"b":{"0":2,"1":5,"2":8},"c":{"0":3,"1":6,"2":9}}
```

In [57]:
```
print(data.to_json(orient='records'))
```

```
[{"a":1,"b":2,"c":3},{"a":4,"b":5,"c":6},{"a":7,"b":8,"c":9}]
```

XML and HTML: Web Scraping

conda install lxml pip install beautifulsoup4 html5lib

In [58]:
```
tables = pd.read_html('examples/fdic_failed_bank_list.html')
len(tables)
```

Out[58]:
```
1
```

In [59]:
```
failures = tables[0]
```

In [60]:
```
failures.head()
```

Out[60]:

	Bank Name	City	ST	CERT	Acquiring Institution	Closing Date	Updated Date
0	Allied Bank	Mulberry	AR	91	Today's Bank	September 23, 2016	November 17, 2016
1	The Woodbury Banking Company	Woodbury	GA	11297	United Bank	August 19, 2016	November 17, 2016
2	First CornerStone Bank	King of Prussia	PA	35312	First-Citizens Bank & Trust Company	May 6, 2016	September 6, 2016
3	Trust Company Bank	Memphis	TN	9956	The Bank of Fayette County	April 29, 2016	September 6, 2016
4	North Milwaukee State Bank	Milwaukee	WI	20364	First-Citizens Bank & Trust Company	March 11, 2016	June 16, 2016

In [61]:
```
close_timestamps = pd.to_datetime(failures['Closing Date'])
close_timestamps.dt.year.value_counts()
```

Out[61]:
```
2010    157
2009    140
2011     92
2012     51
2008     25
       ...
2004      4
2001      4
2007      3
2003      3
2000      2
Name: Closing Date, Length: 15, dtype: int64
```

Parsing XML with lxml.objectify

373889 Metro-North Railroad Escalator Availability Percent of the time that escalators are operational systemwide. The availability rate is based on physical observations performed the morning of regular business days only. This is a new indicator the agency began reporting in 2009. 2011 12 Service Indicators M U % 1 97.00 97.00

In [62]:
```
from lxml import objectify

path = 'datasets/mta_perf/Performance_MNR.xml'
parsed = objectify.parse(open(path))
root = parsed.getroot()
```

In [63]:
```
data = []

skip_fields = ['PARENT_SEQ', 'INDICATOR_SEQ',
               'DESIRED_CHANGE', 'DECIMAL_PLACES']

for elt in root.INDICATOR:
    el_data = {}
    for child in elt.getchildren():
        if child.tag in skip_fields:
            continue
        el_data[child.tag] = child.pyval
    data.append(el_data)
```

In [64]:
```
perf = pd.DataFrame(data)
perf.head()
```

Out[64]:

	AGENCY_NAME	INDICATOR_NAME	DESCRIPTION	PERIOD_YEAR	PERIOD_MONTH	CATEGORY
0	Metro-North Railroad	On-Time Performance (West of Hudson)	Percent of commuter trains that arrive at thei...	2008	1	Service Indicators
1	Metro-North Railroad	On-Time Performance (West of Hudson)	Percent of commuter trains that arrive at thei...	2008	2	Service Indicators
2	Metro-North Railroad	On-Time Performance (West of Hudson)	Percent of commuter trains that arrive at thei...	2008	3	Service Indicators
3	Metro-North Railroad	On-Time Performance (West of Hudson)	Percent of commuter trains that arrive at thei...	2008	4	Service Indicators
4	Metro-North Railroad	On-Time Performance (West of Hudson)	Percent of commuter trains that arrive at thei...	2008	5	Service Indicators

In [65]:
```
from io import StringIO
tag = '<a href="http://www.google.com">Google</a>'
root = objectify.parse(StringIO(tag)).getroot()
```

In [66]:
```
root
```

Out[66]:
```
<Element a at 0x11a60fc36c0>
```

In [67]:
```
root.get('href')
```

Out[67]:
```
'http://www.google.com'
```

In [68]:
```
root.text
```

Out[68]:
```
'Google'
```

Binary Data Formats

In [69]:
```
frame = pd.read_csv('examples/ex1.csv')
frame
```

Out[69]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [70]:
```
frame.to_pickle('examples/frame_pickle')
```

In [71]:
```
pd.read_pickle('examples/frame_pickle')
```

Out[71]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [72]:
```
!rm examples/frame_pickle
```

```
'rm' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

Using HDF5 Format

In [73]:
```
frame = pd.DataFrame({'a': np.random.randn(100)})
store = pd.HDFStore('mydata.h5')
store['obj1'] = frame
store['obj1_col'] = frame['a']
store
```

Out[73]:
```
<class 'pandas.io.pytables.HDFStore'>
File path: mydata.h5
```

In [74]:
```
store['obj1']
```

Out[74]:

	a
0	-0.204708
1	0.478943
2	-0.519439
3	-0.555730
4	1.965781
...	...
95	0.795253
96	0.118110
97	-0.748532
98	0.584970
99	0.152677

100 rows × 1 columns

In [76]:
```
store.put('obj2', frame, format='table')
store.select('obj2', where=['index >= 10 and index <= 15'])
```

Out[76]:

	a
10	1.007189
11	-1.296221
12	0.274992
13	0.228913
14	1.352917
15	0.886429

In [77]:
```
store.close()
```

In [78]:
```
frame.to_hdf('mydata.h5', 'obj3', format='table')
pd.read_hdf('mydata.h5', 'obj3', where=['index < 5'])
```

Out[78]:

	a
0	-0.204708
1	0.478943
2	-0.519439
3	-0.555730
4	1.965781

In [79]:
```
os.remove('mydata.h5')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_18392\782371490.py in <module>
----> 1 os.remove('mydata.h5')

NameError: name 'os' is not defined
```

Reading Microsoft Excel Files

In [81]:
```
xlsx = pd.ExcelFile('examples/ex1.xlsx')
xlsx
```

Out[81]:
```
<pandas.io.excel._base.ExcelFile at 0x11a60fdb520>
```

In [82]:
```
pd.read_excel(xlsx, 'Sheet1')
```

Out[82]:

	Unnamed: 0	a	b	c	d	message
0	0	1	2	3	4	hello
1	1	5	6	7	8	world
2	2	9	10	11	12	foo

In [83]:
```
frame = pd.read_excel('examples/ex1.xlsx', 'Sheet1')
frame
```

Out[83]:

	Unnamed: 0	a	b	c	d	message
0	0	1	2	3	4	hello
1	1	5	6	7	8	world
2	2	9	10	11	12	foo

In [84]:
```
writer = pd.ExcelWriter('examples/ex2.xlsx')
frame.to_excel(writer, 'Sheet1')
writer.save()
```

In [85]:
```
frame.to_excel('examples/ex2.xlsx')
```

In [86]:
```
!rm examples/ex2.xlsx
```

```
'rm' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

In [88]:
```
%load examples/ex2.xlsx
```

```
Traceback (most recent call last):

  File "C:\PythonDSA\anaconda3\lib\tokenize.py", line 330, in find_cookie
    line_string = line.decode('utf-8')

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xf8 in position 14: invalid s
tart byte


During handling of the above exception, another exception occurred:

Traceback (most recent call last):

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y", line 3457, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "C:\Users\Usuário\AppData\Local\Temp\ipykernel_18392\2603545630.py", line
1, in <module>
    get_ipython().run_line_magic('load', 'examples/ex2.xlsx')

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y", line 2364, in run_line_magic
    result = fn(*args, **kwargs)

  File "C:\PythonDSA\anaconda3\lib\site-packages\decorator.py", line 232, in fun
    return caller(func, *(extras + args), **kw)

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\core\magic.py", line 187,
in <lambda>
    call = lambda f, *a, **k: f(*a, **k)

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\core\magics\code.py", lin
e 353, in load
    contents = self.shell.find_user_code(args, search_ns=search_ns)

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y", line 3761, in find_user_code
    return openpy.read_py_file(tgt, skip_encoding_cookie=skip_encoding_cookie)

  File "C:\PythonDSA\anaconda3\lib\site-packages\IPython\utils\openpy.py", line 7
5, in read_py_file
    with open(filename) as f:   # the open function defined in this module.

  File "C:\PythonDSA\anaconda3\lib\tokenize.py", line 394, in open
    encoding, lines = detect_encoding(buffer.readline)

  File "C:\PythonDSA\anaconda3\lib\tokenize.py", line 371, in detect_encoding
    encoding = find_cookie(first)

  File "C:\PythonDSA\anaconda3\lib\tokenize.py", line 335, in find_cookie
    raise SyntaxError(msg)

  File "<string>", line unknown
SyntaxError: invalid or missing encoding declaration for 'examples/ex2.xlsx'
```

Interacting with Web APIs

In [89]:

```
import requests
url = 'https://api.github.com/repos/pandas-dev/pandas/issues'
resp = requests.get(url)
resp
```

Out[89]:
```
<Response [200]>
```

In [90]:
```
data = resp.json()
data[0]['title']
```

Out[90]:
```
'BUILD: Python Docker Build Issues'
```

In [91]:
```
issues = pd.DataFrame(data, columns=['number', 'title',
                                     'labels', 'state'])
issues
```

Out[91]:

	number	title	labels	state
0	59504	BUILD: Python Docker Build Issues	[{'id': 129350, 'node_id': 'MDU6TGFiZWwxMjkzNT...	open
1	59503	TYP: Fix NatType.combine type definition	[{'id': 2822342, 'node_id': 'MDU6TGFiZWwyODIyM...	open
2	59502	DEPR: future.no_silent_downcasting option	[{'id': 31404521, 'node_id': 'MDU6TGFiZWwzMTQw...	open
3	59501	REF (string): move ArrowStringArrayNumpySemant...	[]	open
4	59500	Fix valerror dataframe array	[]	open
...	...	...	...	...
25	59459	DOC: Development on Gitpod have problems	[{'id': 129350, 'node_id': 'MDU6TGFiZWwxMjkzNT...	open
26	59458	DOC: fix docstring validation errors for `pand...	[{'id': 134699, 'node_id': 'MDU6TGFiZWwxMzQ2OT...	open
27	59455	ENH: Add an option to prevent stripping extra ...	[{'id': 76812, 'node_id': 'MDU6TGFiZWw3NjgxMg=...	open
28	59454	BUG: escapechar=',' Causes Double Commas in Ou...	[{'id': 76811, 'node_id': 'MDU6TGFiZWw3NjgxMQ=...	open
29	59453	DOC: clarify the documentation for DataFrame.t...	[{'id': 134699, 'node_id': 'MDU6TGFiZWwxMzQ2OT...	open

30 rows × 4 columns

Interacting with Databases

In [92]:
```
import sqlite3
query = """
CREATE TABLE test
(a VARCHAR(20), b VARCHAR(20),
 c REAL,        d INTEGER
);"""
con = sqlite3.connect('mydata.sqlite')
con.execute(query)
con.commit()
```

In [93]:
```
data = [('Atlanta', 'Georgia', 1.25, 6),
        ('Tallahassee', 'Florida', 2.6, 3),
        ('Sacramento', 'California', 1.7, 5)]
stmt = "INSERT INTO test VALUES(?, ?, ?, ?)"
```

```
con.executemany(stmt, data)
con.commit()
```

In [94]:
```
cursor = con.execute('select * from test')
rows = cursor.fetchall()
rows
```

Out[94]:
```
[('Atlanta', 'Georgia', 1.25, 6),
 ('Tallahassee', 'Florida', 2.6, 3),
 ('Sacramento', 'California', 1.7, 5)]
```

In [95]:
```
cursor.description
```

Out[95]:
```
(('a', None, None, None, None, None, None),
 ('b', None, None, None, None, None, None),
 ('c', None, None, None, None, None, None),
 ('d', None, None, None, None, None, None))
```

In [96]:
```
pd.DataFrame(rows, columns=[x[0] for x in cursor.description])
```

Out[96]:

	a	b	c	d
0	Atlanta	Georgia	1.25	6
1	Tallahassee	Florida	2.60	3
2	Sacramento	California	1.70	5

In [97]:
```
import sqlalchemy as sqla
db = sqla.create_engine('sqlite:///mydata.sqlite')
pd.read_sql('select * from test', db)
```

Out[97]:

	a	b	c	d
0	Atlanta	Georgia	1.25	6
1	Tallahassee	Florida	2.60	3
2	Sacramento	California	1.70	5

In [98]:
```
!rm mydata.sqlite
```

```
'rm' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.
```

Conclusion