# **pandas学习（一）——数据的导入和导出**

[](https://www.jianshu.com/u/10a6dd100718)

[骑着炮弹进城](https://www.jianshu.com/u/10a6dd100718) 关注

2017.04.25 22:07\* 字数 458 阅读 418评论 0喜欢 1

要学数据挖掘与分析第一步当然是要导入数据到程序当中或者从程序中导出数据到本地文件当中，这里我使用pandas库提供的函数来举例导入和导出数据。本文所用的环境：python ：3.5 pandas：0.19.2 numpy：1.12.1，sqlalchemy 1.1.9 如果你的环境和这样不一样可能会有 细微差别。

## **pandas支持的数据格式**

pandas作为一个强大的数据处理包，支持比较多的数据处理格式，下面是一些常见格式数据的读取方法，更多请参考：[链接](https://link.jianshu.com/?t=http://pandas.pydata.org/pandas-docs/stable/api.html" \l "input-output" \t "https://www.jianshu.com/p/_blank)

| **函数** | **描述** |
| --- | --- |
| read\_table(filepath\_or\_buffer[, sep, ...]) | 读取普通分隔的数据 |
| read\_csv(filepath\_or\_buffer[, sep, ...]) | 读取csv格式的数据 |
| read\_excel(io[, sheetname, header, ...]) | 读取excel格式的数据 |
| read\_json([path\_or\_buf, orient, typ, dtype, ...]) | 读取json格式的数据 |
| read\_html(io[, match, flavor, header, ...]) | 读取html格式的 数据 |
| read\_sql(sql, con[, index\_col, ...]) | 读取数据库中的数据 |

前面两个一般用的比较多。

## **常见格式读取示例**

### **read\_table举例**

example.csv是一个用逗号隔开的数据格式。所以可以用read\_table读取，需要指定间隔符为逗号。

import pandas as pd

data\_csv = pd.read\_table('example.csv',sep=',')

print("data\_csv:")

print(data\_csv)

a b c d message

0 1 2 3 4 hello

1 5 6 7 8 world

2 9 10 11 12 foo

但是有时隔符是不定个数的空格，这时可以用正则表达式。

import pandas as pd

data\_txt = pd.read\_table('example.txt',sep='\s+')

print("data\_txt:")

print(data\_txt)

此处由于数据中列名比数据列少1，read\_table会推断第一行为列名。

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

### **read\_csv举例**

import pandas as pd

data\_csv2 = pd.read\_csv('example.csv')

print("data\_csv2:")

print(data\_csv2)

a b c d message

0 1 2 3 4 hello

1 5 6 7 8 world

2 9 10 11 12 foo

### **read\_exel示例**

import pandas as pd

data\_xlsx = pd.read\_excel('example.xlsx')

print("data\_xlsx:")

print(data\_xlsx)

默认是读取第一个 sheet表格的，如果要制定读取sheet表格则需要指定 sheetname参数

data\_xlsx2 = pd.read\_excel('example.xlsx',sheetname="Sheet2")

print("data\_xlsx2:")

print(data\_xlsx2)

a b c d message

0 11 12 13 4 hello

1 15 16 17 18 world

2 19 20 21 12 foo

更多参数可以参考官方手册

### **read\_json示例**

data\_json = pd.read\_json('example.json')

print(data\_json)

直接打开json文件的内容如下：

[{"a": 1, "b": 2, "c": 3},

{"a": 4, "b": 5, "c": 6},

{"a": 7, "b": 8, "c": 9}]

读取后 输出的格式如下

a b c

0 1 2 3

1 4 5 6

2 7 8 9

### **read\_sql、read\_sql\_table和read\_sql\_query示例**

import pymysql

import pandas as pd

con = pymysql.connect(host="127.0.0.1",user="root",password="password",db="world")

data\_sql=pd.read\_sql("select \* from city limit 10",con)

print(data\_sql)

数据库用的是mysql，数据是里面自带的测试数据。

ID Name CountryCode District Population

0 1 Kabul AFG Kabol 1780000

1 2 Qandahar AFG Qandahar 237500

2 3 Herat AFG Herat 186800

3 4 Mazar-e-Sharif AFG Balkh 127800

4 5 Amsterdam NLD Noord-Holland 731200

在使用read\_sql\_table和read\_sql\_query时需要使用sqlalchemy对数据库进行连接。这里仍然使用mysql为例，其他数据库的链接方式有细微差别。

import pandas as pd

import pymysql

from sqlalchemy import create\_engine

con = create\_engine('mysql+pymysql://root:password@localhost:3306/world')

data\_sql2 = pd.read\_sql\_table("city", con)

print(data\_sql2)

data\_sql3 = pd.read\_sql\_query("select \* from city limit 5", con)

print(data\_sql3)