

[AI共學社群](#) > [Python資料科學](#) > [D09 使用 Pandas 讀寫各種常用的檔案格式](#)

# D09 使用 Pandas 讀寫各種常用的檔案格式



[簡報閱讀](#)



[範例與作業](#)



[問題討論](#)



[學習心得\(完成\)](#)



## 重要知識點



- 讀寫excel
- 讀寫json
- 讀寫SQL資料庫

## 讀寫 CSV

CSV 的資料由很多文本資料組成，資料之間以逗點隔開。

首先我們使用 Pandas 的套件中 `pd.read_csv` 讀取一個 CSV 檔案夾中的 `iris.csv`，讀取之後即為 `DataFrame` 的資料型態。有時候資料太多不想一次讀取這麼多行可以使用 `usecols` 參數指定讀取的行名稱。

```
iris_data = pd.read_csv('iris.csv')
iris_data
```

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

有時候資料太多不想一次讀取這麼多行可以使用 `usecols` 參數指定讀取的行名稱。



0	1.4	0.2	0
1	1.4	0.2	0
2	1.3	0.2	0
3	1.5	0.2	0
4	1.4	0.2	0
5	1.7	0.4	0

可以再資料圖取中利用 **names** 參數指定行的名稱，因為有指定行名稱所以必須以 **header=0** 參數跳過檔案裡放置行名稱的列。

```
iris_data = pd.read_csv('iris.csv',header=0,names=['featue1','featue2','featue3','featue4','target'])
iris_data
```

	featue1	featue2	featue3	featue4	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0

輸出 csv 檔案使用 **.to\_csv()** 的方法就可以將資料存到指定路徑下。

```
iris_data.to_csv('my_iris.csv')
```

## 讀寫 excel

Pandas 利用 **pd.read\_excel()** 函數支援讀取 Excel 2003 之後的格式資料，此方法內部是使用 XLRD 或 OpenPyXL 套件，所以使用前確認至少安裝其中一項。



	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

`pd.read_excel()` 只會讀第一個工作簿，如今天 excel 工作簿不只一個，我們可以使用 `sheetname` 參數傳入要讀取頁面的名稱。

```
boston_data = pd.read_excel('data.xls',sheet_name='boston')
boston_data
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33

`pd.read_excel` 和 `pd.read_csv` 一樣有 `usecols`、`header`、`names` 可以使用

```
boston_data = pd.read_excel('data.xls',sheet_name='boston',header=0 \
,usecols=['TAX','PTRATIO','B','LSTAT'])
boston_data
```

	TAX	PTRATIO	B	LSTAT
0	296	15.3	396.90	4.98
1	242	17.8	396.90	9.14
2	242	17.8	392.83	4.03
3	222	18.7	394.63	2.94
4	222	18.7	396.90	5.33

輸出 excel 檔案使用 `.to_excel()` 的方法就可以  
料存到指定路徑下，也可以使用 `sheet_name` 更改  
工作簿名稱。

```
boston_data.to_excel('my_boston.xlsx',sheet_name='boston')
```

利用 `pd.read_json()` 函數支援讀取 JavaScript 物件表示法 (JSON) 格式資料，`json` 具有跨平台與程式語言的可攜性。

```
boston_data = pd.read_json('boston.json')
boston_data
```

	TAX	PTRATIO	B	LSTAT
0	296	15.3	396.90	4.98
1	242	17.8	396.90	9.14
2	242	17.8	392.83	4.03
3	222	18.7	394.63	2.94
4	222	18.7	396.90	5.33

輸出 `json` 檔案使用 `.to_json()` 的方法就可以將資料存到指定路徑下。

```
boston_data.to_json('boston.json')
```

## 讀寫 SQL 資料庫

任何 SQL 資料庫如果支援遵守 Python DB-API 都可以被 Pandas 讀取。以下先用 `boston.csv` 的資料寫入 SQLite3 資料庫中並且命名 `boston`，由 `if_exists` 參數判斷是否存在資料庫，如果檔案不存在會立即被建立，如果存在 `if_exists='replace'` 將會取  
本資料，`if_exists='append'` 將會繼續寫在原始  
下。

重要知識點



讀寫 csv



讀寫 excel



讀寫 json



讀寫 SQL 資料庫



知識點回顧



補充教材



延伸閱讀



```
connection = sqlite3.connect('sql_db.sqlite')
boston_data.to_sql('boston', connection, if_exists='replace')
connection.commit()
connection.close()
```

讀取 SQLite3 資料庫可以使用 `pd.io.sql.read_sql`，可以直接下 SQL 指令對 `sql_db` 中的 `boston` 做搜尋。

```
connection = sqlite3.connect('sql_db.sqlite')
boston_data_sql = pd.io.sql.read_sql("select * from boston",connection)
connection.close()
boston_data_sql
```

	index	TAX	PTRATIO	B	LSTAT
0	0	296	15.3	396.90	4.98
1	1	242	17.8	396.90	9.14
2	2	242	17.8	392.83	4.03
3	3	222	18.7	394.63	2.94
4	4	222	18.7	396.90	5.33

## 知識點回顧

- 資料型態多元，我們可以使用 Pandas 針對不同資料型態讀取，讀寫 csv(`read_csv`、`to_csv`)、讀寫 excel(`read_excel`、`to_excel`)、讀寫 json(`read_json`、`to_json`)、讀寫 SQL 資料庫(`io.sql.read_sql`、`to_sql`)，讀取進來後都是 `DataFrame` 的型態，方便之後進行資料的分析，也可以藉由 Pandas 做資料型態的轉換。

## 補充教材

python pandas 讀取檔案

網站：[itread01](https://itread01.com)



其他

## python pandas 讀取檔案——讀取具有明顯分隔符的資料

其他 · 發表 2018-12-18

關於這篇博文提到的所有函式都可以在 [IO Tools \(Text, CSV, HDF5, ...\)](#) 找到。首先我們來列舉一下 pandas 處理檔案的函式<sup>1</sup>：

- `pd.read_csv(filepath)`：從 CSV 檔案匯入資料
- `pd.read_table(filepath)`：從限定分隔符的文字檔案匯入資料
- `pd.read_excel(filepath)`：從 Excel 檔案匯入資料
- `pd.read_sql(query, connection_object)`：從 SQL 表 / 庫匯入資料
- `pd.read_json(json_string)`：從 JSON 格式的字串匯入資料
- `pd.read_html(url)`：解析 URL、字串或者 HTML 檔案，抽取其中的 tables 表格
- `pd.read_clipboard()`：從你的貼上板獲取內容，並傳給 `read_table()`
- `pd.DataFrame(dict)`：從字典物件匯入資料，Key 是列名，Value 是資料

想要讀取具有明顯分隔符的資料，我們可以採用的函式是：

- `pd.read_csv(filepath)`
- `pd.read_table(filepath)`

與此問有關的相關引數有三個：sep, delimiter, delim\_whitespace。具體含義以 `pandas.read_table` 為例，網站：[yltang](#)



### 資料科學與機器學習

#### 第 9 章 資料載入、儲存、與檔案格式

序言

- 1. Python
- 2. Python 簡介
- 3. 資料與模型
- 4. 資料結構
- 5. 決策樹與函式
- 6. 物件與類別

#### II. 資料科學

- 7. 資料科學簡介
- 8. NumPy
- 9. pandas
- 10. 資料載入
- 11. 圖表

#### III. 機器學習

- 12. 機器學習簡介
- 13. 監督式學習
- 14. 非監督式學習

資料載入的主要來源

- 文字檔
- 資料庫
- 網頁

(I) 文字格式資料 (Text format)

• pandas 提供讀取檔案並存為 DataFrame 的函式：

函法：pd.read\_csv(filepath), 例如 pd.read\_csv("data/pa1to")

函式	說明
read_csv()	從檔案或 URL 讀取無以逗點分隔的資料
read_table()	從檔案或 URL 讀取無以逗點分隔的資料
read_fwf()	讀取固定寬度欄位格式 (fixed-width column format, 沒有分隔符號)
read_excel()	從 XLS 或 XLSX 檔案讀取表格資料
read_hdf()	讀取 HDF5 檔
read_html()	讀取 HTML 檔案裡的所有表格資料
read_json()	讀取 JSON 檔
read_msgpack()	讀取 MessagePack 二元格式檔
read_pickle()	讀取以 pickle 格式儲存的 Python 物件
read_sas()	讀取 SAS 檔
read_sql()	讀取 SQL 查詢結果
read_sqlite()	讀取 SQLite 格式檔

## 延伸閱讀

使用 python 讀取資料科學最常用的檔案格式

網站：[itread01](#)

其他

## 使用python讀取資料科學最常用的檔案格式

其他 · 發表 2019-02-17

Author : kevinelstri

DateTime : 2017/3/13

### 1、什麼是檔案格式？

檔案格式是在檔案中儲存資訊的一種標準方法。首先，檔案格式指定檔案是一個二進位制或ASCII檔案。其次，檔案展示了檔案的組織形式。例如，逗號分隔值 (CSV) 檔案格式儲存在純文字的表格資料。

CSV				
#	A	B	C	D
1	ID	Gender	City	Monthly_Income_Range
2	100000020	Female	Delhi	20000
3	100000040	Male	Mumbai	35000
4	100000070	Male	Panchkula	22500
5	100000090	Male	Saharsa	35000
6	100000095	Male	Bengaluru	100000
7	100000100	Male	Bengaluru	45000
8	100000110	Female	Sindhubur	70000
9	100000120	Male	Bengaluru	20000
10	100000130	Male	Kochi	75000
11	100000140	Female	Mumbai	30000
12	100000160	Male	Mumbai	25000
13	100000185	Female	Surat	25000
14	100000190	Female	Pune	24000
15	100000210	Male	Bhubanes	27000
16	100000220	Female	Howrah	28000

### JSON

```
{
  "Employee": [
    {
      "id": "1",
      "Name": "Ankit",
      "Sal": "1000",
    },
    {
      "id": "2",
      "Name": "Caitu"
    }
  ]
}
```

```
<?xml version="1.0"?>
<contact-info>
  <name>Ankit</name>
  <company>Analytics Vidhya</company>
  <phone>+9187654321</phone>
</contact-info>
```

## Python 數據分析之 Pandas 讀寫外部數據文件

網站：[kknews](http://kknews.net)

## Python數據分析之Pandas讀寫外部數據文件

2019-09-07 由 編程python新視野 發表于程式開發

### 1 引言

數據分析、數據挖掘、可視化是Python的眾多強項之一，但無論是這幾項中的哪一項都必須以數據作為基礎，數據通常都存儲在外部文件中，例如txt、csv、excel、資料庫。本篇中，我們來捋一捋Python中那些外部數據文件讀取、寫入的常用方法。

undefined

[下一步：閱讀範例與完成作業](#)





[AI共學社群](#)

[我的](#)

