

1. 文本文件

用 Python 导入数据: 大多数情况下, 都是用 Numpy 或 Pandas 导入数据

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

纯文本文件

操作与读写 - 手动打开关闭

```
> filename = 'huck_finn.txt'
> file = open(filename, mode='r') # 以只读方式读取文件
> text = file.read() # 读取文件内容
> print(file.closed) # 查看文件是否已经关闭
> file.close() # 关闭文件
> print(text)
```

操作与读写 - 使用上下文管理器 with

```
> with open('huck_finn.txt', 'r') as file:
    print(file.readline()) # 读取一行
    print(file.readline())
    print(file.readline())
```

表格数据: 文本文件

用 Numpy 导入文本文件

单数据类型文件

```
> filename = 'mnist.txt'

# 用于分割各列值的字符, 跳过前两行, 读取并使用第 1 列和第 3 列使用的数据类型
> data = np.loadtxt(filename, delimiter=',', skiprows=2, usecols=[0,2], dtype=str)
```

多数据类型文件

```
> filename = 'titanic.csv'
> data = np.genfromtxt(filename, delimiter=',', names=True, dtype=None) # 导入时查找列名
```

`np.recfromcsv()`

```
> data_array = np.recfromcsv(filename) # 函数的 dtype 默认值为 None
```

用 Pandas 导入文本文件

```
> filename = 'winequality-red.csv'
```

文件名, 读取的行数, 用哪一行做列名, 用于分隔各列的字符, 用于分割注释的字符, 读取时哪些值为 NA/NaN

```
> data = pd.read_csv(filename, nrows=5, header=None, sep='\t', comment='#', na_values=[""])
```

2. 其他文件

SAS 文件

读取成 Dataframe 格式

```
> from sas7bdat import SAS7BDAT
> with SAS7BDAT('urbanpop.sas7bdat') as file:
> df_sas = file.to_dataframe()
```

Stata 文件

使用 pandas 读取

```
> data = pd.read_stata('urbanpop.dta')
```

Pickled 文件

使用 pickle 工具库打开读取

```
> import pickle
> with open('pickled_fruit.pkl', 'rb') as file:
> pickled_data = pickle.load(file)
```

Excel 文件

读写文件

```
> file = 'urbanpop.xlsx'
> data = pd.ExcelFile(file)
> df_sheet2 = data.parse('1960-1966', skiprows=[0], names=['Country', 'AAM: War(2002)'])
> df_sheet1 = data.parse(0, parse_cols=[0], skiprows=[0], names=['Country'])
```

使用 sheet_names 属性访问表单名称

```
> data.sheet_names
```

HDF5 文件

使用 h5py 工具库打开读取

```
> import h5py
> filename = 'H-H1_LOSC_4_v1-815411200-4096.hdf5'
> data = h5py.File(filename, 'r')
```

Matlab 文件

scipy 工具库读取

```
> import scipy.io
> filename = 'workspace.mat'
> mat = scipy.io.loadmat(filename)
```

3. Array 与 Dataframe 数据

Numpy 数组

```
> data_array.dtype # 查看数组元素的数据类型
> data_array.shape # 查看数组维度
> len(data_array) # 查看数组长度
```

Pandas 数据帧

```
> df.head() # 返回数据帧的前几行，默认为 5 行
> df.tail() # 返回数据帧的后几行，默认为 5 行
> df.index # 查看数据帧的索引
> df.columns # 查看数据帧的列名
> df.info() # 查看数据帧各列的信息
> data_array = data.values # 将数据帧转换为 Numpy 数组
```

4. 字典数据

通过函数访问数据元素

```
> print(mat.keys()) # 输出字典的键值 (Key)
> for key in data.keys(): # 输出字典的键值 (Key)
> print(key)
meta
quality
strain

> pickled_data.values() # 返回字典的值
> print(mat.items()) # 返回由元组构成字典键值对列表
```

通过键访问数据

```
# 探索 HDF5 的结构
> for key in data ['meta'].keys()
> print(key)
Description
DescriptionURL
Detector
Duration
GPSstart
Observatory
Type
UTCstart

# 提取某个键对应的值
> print(data['meta']['Description'].value)
```

6. 文件系统与操作

魔法命令

```
!ls # 列出目录里的子目录和文件夹
%cd .. # 改变当前工作目录
%pwd # 返回当前工作目录的路径
```

os 库

```
> import os
> path = "/usr/tmp"
> wd = os.getcwd() # 将当前工作目录保存为字符串
> os.listdir(wd) # 将目录里的内容输出为列表
> os.chdir(path) # 改变当前的工作目录
> os.rename("test1.txt", "test2.txt") # 重命名文件
> os.remove("test1.txt") # 删除现有文件
> os.mkdir("newdir") # 新建文件夹
```

5. 数据库

关系型数据库

使用 sqlalchemy 库

```
> from sqlalchemy import create_engine
> engine = create_engine('sqlite:///Northwind.sqlite')
```

使用 table_names() 方法获取表名列表:

```
> table_names = engine.table_names()
```

查询关系型数据库

执行 SQL 语句查询


```
> con = engine.connect()
> rs = con.execute("SELECT * FROM Orders")
> df = pd.DataFrame(rs.fetchall())
> df.columns = rs.keys()
> con.close()
```

使用上下文管理器 with

```
> with engine.connect() as con:
    rs = con.execute("SELECT OrderID FROM Orders")
    df = pd.DataFrame(rs.fetchmany(size=5))
    df.columns = rs.keys()
```

使用 Pandas 查询关系型数据库

```
> df = pd.read_sql_query("SELECT * FROM Orders", engine)
```

SQLQuery1.sql - Q2...ROD\BPetrovi (53) - 

```
1 CREATE VIEW vTop3SalesByQuantity
2 AS
3 SELECT TOP 3 --will only return first 3 records from query
4 Sales.ProductID,
5 Name AS ProductName,
6 SUM(Sales.Quantity) AS TotalQuantity
7 FROM Sales
8 JOIN Products ON Sales.ProductID = Products.ProductID
9 GROUP BY Sales.ProductID,
10 Name
11 ORDER BY SUM(Sales.Quantity) DESC;
```

Results Messages

	ProductID	ProductName	TotalQuantity
1	1	Long-Sleeve Logo Jersey, S	4
2	3	Long-Sleeve Logo Jersey, L	3
3	2	Long-Sleeve Logo Jersey, M	3

作者 | 韩信子 @ShowMeAI
设计 | 南乔 @ShowMeAI
参考 | datacamp cheatsheet



扫码回复“速查表”

下载最新全套资料

SHOW ME AI