呆鸟译

Python 数据科学 *速查表*
导入数据

天善智能 商业智能与大数据社区 www.hellobi.com

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
```

调用帮助

```
>>> np.info(np.ndarray.dtype)
>>> help(pd.read_csv)
```

文本文件

纯文本文件

```
>>> filename = 'huck_finn.txt'
>>> file = open(filename, mode='r')
>>> text = file.read()
>>> print(file.closed)
>>> file.close()
>>> print(text)
```

以只读方式读取文件
读取文件内容
查看文件是否已经关闭
关闭文件

使用上下文管理器 with

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

读取一行

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'
>>> data = np.loadtxt(filename,
                      delimiter=',',
                      skiprows=2,
                      usecols=[0,2],
                      dtype=str)
```

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

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

```
>>> data_array = np.recfromcsv(filename)
```

np.recfromcsv() 函数的 dtype 默认值为 None。

用 Pandas 导入文本文件

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

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

Excel表

```
>>> 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
```

SAS 文件

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

Stata 文件

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

关系型数据库文件

```
>>> 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

```
>>> 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)
```

探索数据

Numpy 数组

```
>>> data_array.dtype
>>> data_array.shape
>>> len(data_array)
```

查看数组元素的数据类型
查看数组维度
查看数组长度

Pandas 数据框

```
>>> df.head()
>>> df.tail()
>>> df.index
>>> df.columns
>>> df.info()
>>> data_array = data.values
```

返回数据框的前几行，默认为5行
放回数据框的后几行，默认为5行
查看数据框的索引
查看数据框的列名
查看数据框各列的信息
将数据框转换为 Numpy 数组

Pickled 文件

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

HDF5 文件

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

Matlab 文件

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

探索字典

通过函数访问数据元素

```
>>> print(mat.keys())
>>> for key in data.keys():
        print(key)
meta
quality
strain
>>> pickled_data.values()
>>> print(mat.items())
```

输出字典的键值 (Key)
输出字典的键值 (Key)
返回字典的值
返回由元组构成字典键值对列表

通过键访问数据

```
>>> for key in data ['meta'].keys()
        print(key)
Description
DescriptionURL
Detector
Duration
GPSstart
Observatory
Type
UTCstart
>>> print(data['meta']['Description'].value)
```

探索 HDF5 的结构
提取某个键对应的值

探寻文件系统

魔法命令

```
!ls
%cd ..
%pwd
```

列出目录里的子目录和文件夹
改变当前工作目录
返回当前工作目录的路径

os 库

```
>>> import os
>>> path = "/usr/tmp"
>>> wd = os.getcwd()
>>> os.listdir(wd)
>>> os.chdir(path)
>>> os.rename("test1.txt",
              "test2.txt")
>>> os.remove("test1.txt")
>>> os.mkdir("newdir")
```

将当前工作目录存为字符串
将目录里的内容输出为列表
改变当前的工作目录
重命名文件
删除现有文件
新建文件夹

原文作者 DataCamp
Learn R for Data Science Interactively