Python数据科学速查表

导入数据

更多Python数据科学www.DataCamp.com

Python导入数据

通常使用**NumPy**or**pandas**导入数据：

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

帮助

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

Text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)
```

以读方式打开文件
读文件内容
检查文件是否关闭
关闭文件

使用上下文管理器

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

读单行

表数据: Flat Files

numpy导入Flat Files

一种数据类型文件

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

逗号分隔
跳过前2行
读第1、3列
结果数组类型

混合数据类型文件

```
>>> filename = 'titanic.csv'
>>> data = np.genfromtxt(filename,
                         delimiter=',',
                         names=True,
                         dtype=None)
```

查找列标题

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

The default `dtype` of the `np.recfromcsv()` function is `None`.

pandas导入Flat Files

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

读取行数
行数当列名
分隔符
注释符
识别为NA/NaN

Excel电子表格

```
>>> 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'])
```

To access the sheet names, use the `sheet_names` attribute:

```
>>> 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

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

Use the `table_names()` method to fetch a list of table names:

```
>>> 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)
```

Exploring Your Data

NumPy数组

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

数组元素数据类型
数组维度
数组长度

pandas数据框架

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

返回第一行
返回最后一行
描述索引
描述列
数据描述信息
数组转为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)
```

Exploring字典

数据函数访问数据

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

打印字典Keys
Print dictionary keys

返回字典值
列表项格式（键：值）

通过关键词访问数据

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

Explore the HDF5 structure

通过Key访问值

浏览文件系统

魔法命令

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

列出目录文件和子目录
改变当前工作路径
返回当前工作路径

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")
```

将当前目录名称存储在字符串中
列出目录路径
改变当前工作路径
重命名文件

删除已存在的文件
创建新目录