Python数据科学入门

第2章 数据科学的python核心

4单元 理解基本的字符串函数

- lower()函数将所有字符转换为小写；
- upper()函数将所有字符转换为大写；
- capitalize() 函数将第一个字符转换为大写，同时将其他所有字符转换为小写
- 判定（predict）函数根据字符串s是否属于适当的类而返回True 或False：
- islower()函数检查所有字母字符是否为小写；
- isupper()函数检查所有字母字符是否为大写；
- isspace()函数检查所有字符是否为空格；
- isdigit()函数检查所有字符是否为范围0～9中的十进制数字；
- isalpha()函数检查所有字符是否为a～z或A～Z范围内的字母字符。
- bin.decode()将二进制数组转换为字符串，
- s.encode()将字符串转换为二进制数组。
- lstrip()（left strip）、rstrip()（right strip）和strip()分别在字符串的开始处、结束处或对整个字符串删除所有空格（不删除字符串内部空格）
- split(delim='')使用delim作为分隔符，将字符串s分割为子字符串组成的一个列表
- 连接函数join(ls)——分割函数的姐妹函数——将字符串列表ls连接 在一起，形成一个字符串，并使用特定的对象字符串作为连接符 `", ".join(["alpha", "bravo", "charlie", "delta"])`
- find(needle)返回对象字符串中子字符串needle第一次出现的索引值，当子字符串不存在时，返回-1

  `"www.networksciencelab.com".find(".com")`
- count(needle)返回对象字符串中子字符串needle非重叠出现的 次数，该函数也区分大小写

5单元 选择合适的数据结构

- 列表的搜索时间是线性的，因此用列表来存储大量可搜索的数据是不切实际的。
- 元组是不可变的列表，创建后就无法再更改。元组的搜索时间也是线性的。
- 集合不是序列：集合项不存在索引。集合最多只能存储一个项的副本，集合非常适合于成员查找和消除重复项（如果将包含重复项的列表转换为集合，则重复项将会消失）——`myList = list(set(myList))`
- 字典构建了从键到值的映射。任何可哈希的数据类型（数字、布尔、字符串、元组）的对象都可以作为键，且同一字典中的不同键可以属于不同的数据类型
  - `seq = ["alpha", "bravo", "charlie", "delta"]`

    `dict(enumerate(seq))`
  - `kseq = "abcd"`

    `vseq = ["alpha", "bravo", "charlie", "delta"]`

```
dict(zip(kseq, vseq))
```

6单元 列表推导式

(1) 表达式遍历数据集并访问集合中的每一项。

(2) 为每一项计算可选的布尔表达式（默认值为True）。

(3) 如果布尔表达式为True，则计算当前项目的循环表达式，并将其值附加到结果列表中。

(4) 如果布尔表达式为False，则忽略该项。

```
# 复制myList；等同于myList.copy()或者myList[:]，但二者的效率都没有列表推导式高
[x for x in myList]
# 提取非负项
[x for x in myList if x >= 0]
# 用Mylist各项的平方构建一个新列表
[x**2 for x in** myList]
# 用Mylist非零项的倒数构建一个新列表
[1/x for x in myList if x != 0]
# 从打开的infile文件中选出所有的非空行，
# 并删除这些行开头和结尾的空格
[l.strip() for l in infile if l.strip()]
```

7单元 使用计数器

```
from collections import Counter
phrase = "a man a plan a canal panama"
cntr = Counter(phrase.split())
cntr.most_common()

⇒ [('a', 3), ('canal', 1), ('panama', 1), ('plan', 1), ('man', 1)]
```

8单元 使用文件

- 打开文件进行读取（默认模式，定义为"r"）、（覆盖）写入（"w"）或追加写入（"a"）。
- with语句允许显式地打开一个文件，同时保证在退出Python后能自动关闭文件，从而避免了跟踪那些已打开却不再需要的文件。
- ```
  f.read() # 以字符串或二进制的方式读入所有数据
  f.read(n) # 以字符串或二进制的方式读入前n字节的数据
  f.readline() # 以字符串的方式读入下一行
  f.readlines() # 以字符串的方式读入所有行

  f.write(line) # 写字符串数据或二进制数据
  f.writelines(ines) # 写字符串数据列表
  ```
  ```

11单元 通配符（globbing）

- 通配符可以包含特殊符号'*'（表示零个或多个字符）和'?'（表示正好一个字符）
- ```
  glob.glob("*.txt")
  ⇒ ['public.policy.txt', 'big.data.txt']
  ```

12单元 Picking和Unpicking

```
# 将一个对象转存（dump）到文件
with open("myData.pickle", "wb") as oFile:
    pickle.dump(object, oFile)

# 重新加载相同的对象
with open("myData.pickle", "rb") as iFile:
    object = pickle.load(iFile)
```
```