大数据基础与应用
BIG DATA FOUNDATION AND APPLICATION

CSV和Excel数据交换

北京理工大学计算机学院　高玉金

2019年3月

结构化文本

- 简单的结构化文本文件，唯一的结构层次是间隔的行
- 结构化的文本格式
  - 分隔符，如tab（‘\t’）、逗号（‘,’）或者竖线（‘|’）
  - 逗号分隔，如CSV
  - ‘<’和’>’标签，如XML和HTML
  - 标点符号，如Json文件
  - 缩进，如YAML
  - 混合类型，如各种配置文件

CSV文件格式

- CSV即Comma Separate Values，这种文件格式经常用来作为不同程序之间的数据交互的格式
- 1. 每条记录占一行 以逗号为分隔符
  2. 逗号前后的空格会被忽略
  3. 字段中包含有逗号，该字段必须用双引号括起来
  4. 字段中包含有换行符，该字段必须用双引号括起来
  5. 字段前后包含有空格，该字段必须用双引号括起来
  6. 字段中的双引号用两个双引号表示
  7. 字段中如果有双引号，该字段必须用双引号括起来
  8. 第一条记录，可以是字段名

"Joan ""the bone"", Anne",Jet,"9th, at Terrace plc",Desert City,CO,00123

大数据基础与应用 BIG DATA FOUNDATION AND APPLICATION

手工处理CSV文件

- 使用Python的open函数读入CSV文件
- 使用readlines()方法全部读入一个列表中，处理行尾的回车符
- 使用字符串的split()进行切割，得到切割后的列表

```
1 with  open("price.csv", "r") as fr:
2     ls = []
3     for line in fr:
4         line = line.strip()
5         ls.append(line.split(","))
6     print(ls)
```

使用CSV标准库

- 使用CSV标准库的优势
  - 除了逗号，还有其他可代替的分隔符：‘|’和’\t’很常见
  - 有些数据会有转义字符序列，如果分隔符出现在一块区域内，则整块都要加上引号或者在它之前加上转义字符
  - 文件可能有不同的换行符，Unix系统的文件使用’\n’，Microsoft使用’\r\n’，Apple之前使用’\r’而现在使用’\n’
  - 在第一行可以加上列名

使用CSV库读写文件

- 列表与CSV

```
with open('data.csv','r') as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
```

```
import csv
keys=['a','b','c','d']
data=[[1,2,3,4],[5,6,7,8],[9,10,11,12]]
with open('data.csv','w', newline='') as file:
    writer=csv.writer(file)
    writer.writerow(keys)
    #writer.writerows(data) #一次写入多个
    for row in data:
        writer.writerow(row)
```

使用CSV库读写文件

- 字典与csv
- 提取某行数据

```
If row[“a”] == “4”:
    print(row)
```

```
data=[]
with open('data.csv','r') as file:
    reader=csv.DictReader(file)
    fieldnames=reader.fieldnames
    print(fieldnames)
    for row in reader:
        data.append(dict(row))
    print(data)
```

```
data=[{'a':1,'b':2,'c':3},{'a':4,'b':5,'c':6},{'a':7,'b':8,'c':9}]
fieldnames=['a','b','c']
with open('data.csv','w',newline='') as file:
    writer=csv.DictWriter(file,fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(data)
```

读取Excel文件

- 导入读取Excel库：import xlrd
- 打开文件：data = xlrd.open_workbook('excel.xls’)
- 获取工作表
  - table = data.sheets()[0] #通过索引顺序获取
  - table = data.sheet_by_index(0) #通过索引顺序获取
  - table = data.sheet_by_name(u'Sheet1’) #通过名称获取
- 获取行列数据 table.row_values(i)或table.col_values(i)
- 获取行数和列数：table.nrows和table.ncols
- 获取单元格：table.cell(2,3).value

写入Excel文件

- 加载模块：import xlwt
- 创建workbook：workbook = xlwt.Workbook(encoding = 'ascii’)
- 创建表：worksheet = workbook.add_sheet('My Worksheet’)
- 在单元格写入数据：worksheet.write(0, 0,’53.6’)
- 保存数据：workbook.save('Excel_Workbook.xls’)
- 一般处理数据时不建议直接操作excel，可以通过pandas的excel读取写入函数进行处理
- 对于特殊的excel文件，或者需要生成Excel格式报告，可以花时间和精力研究如何更好地使用xlrd和xlwt库

读取Excel实例

```
a,b,c
1.0,2.0,3.0
4.0,5.0,6.0
7.0,8.0,9.0
[Finished in 0.6s]
```

```
import xlrd
workbook = xlrd.open_workbook('data.xlsx')
sheet = workbook.sheet_by_index(0)
for row in sheet.get_rows():
    for col in row[:-1]:
        print(col.value,end=",")
    print(row[-1].value)
```

BIG DATA FOUNDATION AND APPLICATION