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

数据结构DataFrame

北京理工大学计算机学院　高玉金

2019年3月

数据结构：DataFrame对象

- DataFrame在Pandas中用于表示矩阵数据
- DataFrame包含了已排序的列集合
- 每一列可以是不同的值类型（数值、字符串、布尔值等）
- DataFrame既有行索引index，也有列索引columns
- DataFrame可以理解为一个**共享相同索引**的Series的字典

pd.Series(['Diego','Anna','Eugene'])

```
0      Diego
1       Anna
2     Eugene
dtype: object
```

pd.Series(['M','F','M'])

```
0     M
1     F
2     M
dtype: object
```

pd.Series([12,23,34])

```
0     12
1     23
2     34
dtype: int64
```

```
      name gender   age
0    Diego      M    12
1     Anna      F    23
2   Eugene      M    34
```

如何构建DataFrame

- 利用包含等长度列表/NumPy数组/Series的字典

```
data={'name':['Diego','Anna','Eugene'],\
      'gender':['M','F','M'],
      'age':[12,23,34]}
df = pd.DataFrame(data)
```

- df1== { ‘age’ : np.arange(3)}
- df2 = { ‘name’ :df[‘name’][:-1],}
- df3 = { ‘name’ :pd.Series([‘D’ ,’ A’ ,’ E’])}

```
>>> df
   age gender   name
0   12      M   Diego
1   23      F    Anna
2   34      M  Eugene
```

```
>>> df2
    name gender  age
0  Diego      M   12
1   Anna      F   23
2 Eugene      M   34
```

用嵌套字典构建DataFrame

- 嵌套字典构建DataFrame时，Pandas将字典的键做为列索引，将内部字典的键作为行进行索引

```
dd= {'name':{1:'Diego',2:'Anna',3:'Eugene'},
     'gender':{1:'M',2:'F'},
     'age':{1:12,2:23,3:34}}
df4=pd.DataFrame(dd)
```

```
>>> df4
  age gender   name
1  12      M  Diego
2  23      F   Anna
3  34    NaN Eugene
```

检索DataFrame

- 若DataFrame数据量太大，可以用df.head()筛选出最前面的五行
- 选择一列
  - 类似字典的标记，如df[‘name’]（任意列名）
  - 列名为属性，如df.name(列名是有效的Python变量名)
- 选择一行，可以使用属性loc返回一个Series对象。如df.loc[1]，其索引的名字即为当前行的列索引“1”，索引名字为各列的名字

```
>>> ds= df.loc[1]
>>> ds
age        23
gender      F
name      Anna
Name: 1, dtype: object
```

```
>>> type(ds)
<class 'pandas.core.series.Series'>
```

通过列修改数据

- 标量值
- 值数组（长度必须匹配）
- Series对象（其索引按DataFrame的索引重新排列，空缺处填充NaN）

```
df3.score=pd.Series(['22','33','44’ ], index = [2,1,0])
```

```
>>> df3
    name gender  age score
0   Diego      M   12    44
1    Anna      F   23    33
2  Eugene      M   34    22
```

通过列修改数据

```
>>> df3.score=60
>>> df3
    name gender  age  score
0   Diego      M   12     60
1    Anna      F   23     60
2  Eugene      M   34     60
```

```
>>> df3.score=[70,80,90]
>>> df3
    name gender  age  score
0   Diego      M   12     70
1    Anna      F   23     80
2  Eugene      M   34     90
```

重新索引

- 重新索引方法reindex()
- df1=pd.DataFrame(np.arange(9).reshape(3,3),index=['a','c','d'],columns=['one','two','four'])
- 默认对行进行重新索引，如df1.reindex(['a','b','c','d'])
- 可以同时对列和行进行索引，如df1.reindex(index=['a','b','c','d'],columns=['one','two','three','four'])
- 缺失值自动用NaN填充
- **使用参数fill_value=n，**用n代替缺失值

	one	two	four
a	0	1	2
c	3	4	5
d	6	7	8

索引对象

- 索引对象用于存储轴标签和其他元数据
- 用pd.index()生成索引对象，或从Series和DataFrame的行列索引

```
labels=pd.index(np.arange(3))
labels=df.index或 labels=df.columns
obj = pd.Series([1.3,-3.4,0], index = labels)
```

- 索引对象不可变（只读）
- 同一个索引对象可以被不同数据结构共享
- 索引对象是一个容器，可以使用in和not in进行元素判断