第5章、pandas入门

5.1 数据结构

5.1.1 Series

- s.values、s.index
- s[索引列表]取值
- 类似字典，key in s
- 检查缺失，pd.isnull(s)、pd.notnull(s)
- s.name
- s.index.name

5.1.2 DataFrame

- df.values、df.index、df.columns
- df.index.name、df.columns.name
- del df[列名]
- pd.DataFrame(字典/嵌套字典)
  - 外部字典键作为columns
  - 内部字典键作为index
- df选取列是视图，不是复制　　复制用copy()
- df的列有不同的dtype，df.values会自动选择适合所有列的类型
- 创建

类型	说明
二维ndarray	数据矩阵，还可以传入行标和列标
由数组、列表或元组组成的字典	每个序列会变成DataFrame的一列。所有序列的长度必须相同
NumPy的结构化/记录数组	类似于“由数组组成的字典”
由Series组成的字典	每个Series会成为一列。如果没有显式指定索引，则各Series的索引会被合并成结果的行索引
由字典组成的字典	各内层字典会成为一列，键会被合并成结果的行索引，跟“由Series组成的字典”的情况一样
字典或Series的列表	各项将会成为DataFrame的一行。字典键或Series索引的并集将会成为DataFrame的列标
由列表或元组组成的列表	类似于“二维ndarray”
另一个DataFrame	该DataFrame的索引将会被沿用，除非显式指定了其他索引
NumPy的MaskedArray	类似于“二维ndarray”的情况，只是掩码值在结果DataFrame会变成NA/缺失值

5.1.3 索引对象

- 存储轴名称、标签
- index = pd.Index()
  - 索引index不能被修改
  - 可以重复　　index = pd.Index(['foo', 'foo'])
- 名称 in df.index/df.columns
- 常用方法

方法	说明
append	连接另一个Index对象，产生一个新的Index
difference	计算差集，并得到一个Index
intersection	计算交集
union	计算并集
isin	计算一个指示各值是否都包含在参数集合中的布尔型数组
delete	删除索引出的元素，并得到新的Index
drop	删除传入的值，并得到新的Index
insert	将元素插入到索引处，并得到新的Index
is_monotonic	当各元素均大于等于前一个元素时，返回True
is_unique	当Index没有重复值时，返回True
unique	计算Index中唯一值的数组

5.2 基本功能

5.2.1 重建索引

- obj.reindex(重建后索引序列)
- df.loc[行列表、列列表]也有效果

参数	说明
index	用作索引的新序列。既可以是Index实例，也可以是其他序列型的Python数据结构。Index会被完全使用，就像没有任何复制一样
columns	
method	插值（填充）方式，具体参数请参见表5-4
fill_value	在重新索引的过程中，需要引入缺失值时使用的替代值
limit	前向或后向填充时的最大填充量
tolerance	向前后向后填充时，填充不准确匹配项的最大间距（绝对值距离）
level	在MultiIndex的指定级别上匹配简单索引，否则选取其子集
copy	默认为True，无论如何都复制，如果为False，则新旧相等就不复制

5.2.2 删除

obj.drop(标签列, axis=, inplace=)

5.2.3 索引、选择、过滤

- Series的索引值可以是整数，代表行，也可以是index的值
- s['b':'c']包含尾部c
- df[列标签列表]选择列
- df[数字/数字切片]选择行
- 使用布尔值DataFrame进行索引　　df < 5 得到布尔值DataFrame

类型	说明
df[val]	从 DataFrame 选取单列或一组列；在特殊情况下比较便利：布尔型数组（过滤行）、切片（行切片）、或布尔型 DataFrame（根据条件设置值）
df.loc[val]	通过标签，选取 DataFrame 的单个行或一组行
df.loc[:, val]	通过标签，选取单列或列子集
df.loc[val, val2]	通过标签，同时选取行和列
df.iloc[where]	通过整数位置，从 DataFrame 选取单个行或行子集
df.iloc[:, where]	通过整数位置，从 DataFrame 选取单个列或列子集
df.iloc[where_i, where_j]	通过整数位置，同时选取行和列
df.at[label_i, label_j]	通过行和列标签，选取单一的标量
df.iat[i, j]	通过行和列的位置（整数），选取单一的标量
reindex	通过标签选取行或列
get_value, set_value	通过行和列标签选取单一值

5.2.3.1 loc、iloc选择数据

5.2.4 整数索引

如果轴索引是整数，推荐使用标签/iloc/loc选择数据

5.2.5 算术和数据对齐

- 算术会导致索引的自动外连接（outer join）

5.2.5.1 使用填充值

- 参数fill_value=

方法	说明
add, radd	用于加法（+）的方法
sub, rsub	用于减法（-）的方法
div, rdiv	用于除法（/）的方法
floordiv, rfloordiv	用于底除（//）的方法
mul, rmul	用于乘法（*）的方法
pow, rpow	用于指数（**）的方法

5.2.5.2 df和s之间的算术操作

数学操作默认axis='columns'

- s的索引和df的列进行匹配　　行上操作
- 要想在df的行上匹配
  - 列上操作
  - axis='index'

5.2.6 函数应用、映射

df.apply(func, axis=0)

- 默认df每一列调用func
- 想在行上调用func，设置axis=1
- 不一定返回标量，可返回多个值的Series

```
def f(x):
    return pd.Series([x.min(), x.max()], index=['min', 'max'])
```

返回df，index是min/max，columns是列名

逐元素应用

- s.map(func)
- df.applymap(func)

5.2.7 排序、排名

- 按索引排序　　df.sort_index(axis=, ascending=False)
- 按值排序
  - df.sort_values(by=[列列表])
  - df.sort_values(by=[行索引列表], axis=1)
- 排名
  - obj.rank(axis=, ascending=, method=)
  - method参数

方法	说明
'average'	默认：在相等分组中，为各个值分配平均排名
'min'	使用整个分组的最小排名
'max'	使用整个分组的最大排名
'first'	按值在原始数据中的出现顺序分配排名
'dense'	类似于'min'方法，但是排名总是在组间递加1，而不是组中相同的元素数

5.2.8 轴索引重复

obj.index.is_unique

5.3 描述性统计

方法	说明
count	非NA值的数量
describe	针对Series或各DataFrame列计算汇总统计
min、max	计算最小值和最大值
argmin、argmax	计算能够获取到最小值和最大值的索引位置（整数）
idxmin、idxmax	计算能够获取到最小值和最大值的索引值
quantile	计算样本的分位数（0到1）
sum	值的总和
mean	值的平均数
median	值的算术中位数（50%分位数）
mad	根据平均值计算平均绝对离差
var	样本值的方差
std	样本值的标准差
skew	样本值的偏度（三阶矩）
kurt	样本值的峰度（四阶矩）
cumsum	样本值的累计和
cummin、cummax	样本值的累计最大值和累计最小值
cumprod	样本值的累计积
diff	计算一阶差分（对时间序列很有用）
pct_change	计算百分数变化

常用函数

5.3.1 相关性、协方差

- s1.corr(s2)　　返回标量
- s1.cov(s2)
- df.corr()　　返回df
- df.cov()
- df.corrwith(s/df)　　df的行/列与另一个s/df的相关性

5.3.2 Series的唯一值、计数、成员属性

- 唯一值　　s.unique()
- 计数　　s.value_counts()
- 成员属性　　s.isin([列表])
- 非唯一值数组与唯一值数组联系

```
to_match = pd.Series(list('cabbcd'))
unique_vals = pd.Series(list('cba'))
pd.Index(unique_vals).get_indexer(to_match)
# array([ 0,  2,  1,  1,  0, -1])
```

get_indexer(待匹配数组)　　根据已有的Index作为基础，返回待匹配数组在Index的下标