从此：使用ix

df.iloc[1:3,1:5] 等价于df.ix[1:3,1:5]

但ix还可以:df.ix[1:3,[‘a’,’b’]]或 df.ix[1:3,‘a’:’d’]

SQL的where ==>>df[df[column] boolean expr]

例如：

print (df[df['sex'] == 'Female'])

print (df[df['total\_bill'] > 20])

#where子句中常常会搭配and, or, in, not关键词

df[(条件1) 逻辑符 (条件2)] (条件)为：(df[column] boolean expr)

**逻辑符：$ 、|、(条件).isin([])、-(条件)**

# and print df[(df['sex'] == 'Female') & (df['total\_bill'] > 20)] # or print df[(df['sex'] == 'Female') | (df['total\_bill'] > 20)] # in print df[df['total\_bill'].isin([21.01, 23.68, 24.59])] # not print df[-(df['sex'] == 'Male')] print df[-df['total\_bill'].isin([21.01, 23.68, 24.59])] # string function print df = df[(-df['app'].isin(sys\_app)) & (-df.app.str.contains('^微信\d+$'))]

**distinct**

[drop\_duplicates](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.drop_duplicates.html)根据某列对dataframe进行去重：

df.**drop\_duplicates**(subset=['sex'], keep='first', inplace=True)

包含参数：

subset:为选定的列做distinct，默认为所有列；

keep：值选项{'first', 'last', False}，保留重复元素中的第一个、最后一个，或全部删除；

inplace :默认为False，返回一个新的dataframe；若为True，则返回去重后的原dataframe

## group

group一般会配合**合计函数**（Aggregate functions）使用，比如：count、avg等。Pandas对合计函数的支持有限，有count和size函数实现SQL的count：

print df.groupby('sex').size()

print df.groupby('sex').count()

print df.groupby('sex')['tip'].count()

对于多合计函数，

select sex, max(tip), sum(total\_bill) as total

from tips\_tb

group by sex;

实现在agg()中指定dict：

print df.groupby('sex').agg({'tip': np.max, 'total\_bill': np.sum})

# count(distinct \*\*)

print df.groupby('tip').agg({'sex': pd.Series.nunique})

## join

Pandas中join的实现也有两种：

# 1.

df.join(df2, how='left'...)

# 2.

pd.merge(df1, df2, how='left', left\_on='app', right\_on='app')

第一种方法是按DataFrame的index进行join的，而第二种方法才是按on指定的列做join。Pandas满足left、right、inner、full outer四种join方式。

## order

Pandas中支持多列order，并可以调整不同列的升序/降序，有更高的排序自由度：

print df.sort\_values(['total\_bill', 'tip'], ascending=[False, True])

## top

对于全局的top：

print df.nlargest(3, columns=['total\_bill'])

对于分组top，MySQL的实现（采用自join的方式）：

select a.sex, a.tip

from tips\_tb a

where (

select count(\*)

from tips\_tb b

where b.sex = a.sex and b.tip > a.tip

) < 2

order by a.sex, a.tip desc;

Pandas的等价实现，思路与上类似：

# 1.

df.assign(rn=df.sort\_values(['total\_bill'], ascending=False)

.groupby('sex')

.cumcount()+1)\

.query('rn < 3')\

.sort\_values(['sex', 'rn'])

# 2.

df.assign(rn=df.groupby('sex')['total\_bill']

.rank(method='first', ascending=False)) \

.query('rn < 3') \

.sort\_values(['sex', 'rn'])