82.09 set_index() 和 reset_index()

set_index()

官方定义:

使用一个或多个现有列设置索引， ***默认情况下生成一个新对象***

DataFrame.set_index(*keys*, *drop=True*, *append=False*, *inplace=False*, *verify_integrity=False*)

drop:默认为true，表示是否删除列作为新索引。

append：是否增加列到原来的索引上。

inplace：是否创建一个新的dataframe

Parameters:	**keys** : *column label or list of column labels / arrays* **drop** : *boolean, default True* Delete columns to be used as the new index **append** : *boolean, default False* Whether to append columns to existing index **inplace** : *boolean, default False* Modify the DataFrame in place (do not create a new object) **verify_integrity** : *boolean, default False* Check the new index for duplicates. Otherwise defer the check until necessary. Setting to False will improve the performance of this method
Returns:	**dataframe** : *DataFrame*

单索引:

```
In [307]: data
Out[307]:
     a    b  c    d
0  bar  one  z  1.0
1  bar  two  y  2.0
2  foo  one  x  3.0
3  foo  two  w  4.0

In [308]: indexed1 = data.set_index('c')

In [309]: indexed1
Out[309]:
     a    b    d
c
z  bar  one  1.0
y  bar  two  2.0
x  foo  one  3.0
w  foo  two  4.0
```

复合索引：

```
In [310]: indexed2 = data.set_index(['a', 'b'])

In [311]: indexed2
Out[311]:
         c    d
a   b
bar one  z  1.0
    two  y  2.0
foo one  x  3.0
    two  w  4.0
```

---

reset_index()

知道了set_index()后，再看reset_index()。

reset_index可以还原索引，从新变为默认的整型索引
DataFrame.reset_index(level=None, drop=False, inplace=False, col_level=0, col_fill=”)
level控制了具体要还原的那个等级的索引

drop为False则索引列会被还原为普通列，否则会丢失

DataFrame.reset_index(*level=None*, *drop=False*, *inplace=False*, *col_level=0*, *col_fill=''*)

http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.reset_index.html

示例：

```
In [3]: df1
Out[3]:
          0         1         2         3         4
0 -0.127085 -0.538321  0.641609 -0.020957  0.003503
2 -0.239710  1.235562  0.917208 -0.964571 -1.120331
3 -0.176416 -0.216204  0.433998 -0.366355 -0.261724


# reset_index，原行索引作为一列保留，列名为index
In [4]: df2 = df1.reset_index()

In [5]: df2
Out[5]:
   index         0         1         2         3         4
0      0 -0.127085 -0.538321  0.641609 -0.020957  0.003503
1      2 -0.239710  1.235562  0.917208 -0.964571 -1.120331
2      3 -0.176416 -0.216204  0.433998 -0.366355 -0.261724
```

```
# reset_index，通过函数 drop=True 删除原行索引
In [6]: df3 = df1.reset_index(drop=True)

In [7]: df3
Out[7]:
          0         1         2         3         4
0 -0.127085 -0.538321  0.641609 -0.020957  0.003503
1 -0.239710  1.235562  0.917208 -0.964571 -1.120331
2 -0.176416 -0.216204  0.433998 -0.366355 -0.261724
```

https://github.com/Bifzivkar/Boutique-Travel-Services-Predict/blob/master/feature/3_extract_feature.py 57

count_1 = orderHistory_1.groupby(orderHistory_1.userid)['orderid'].count().reset_index()