

• **Dataframe to numpy**

df.as\_matrix

• **columns (列) 中 value\_counts()**

# 在choice\_description中，下单次数最多的商品是什么？ chipo['choice\_description'].value\_counts()

• **Dataframe 中插入一行数据**

df['newcol'] = arr.toarray().tolist() col\_name.insert(1,'D') # 数字定位插入  
col\_name.insert(col\_name.index('B'),'D') # 在 B 列前面插入

• **numpy 和 Dataframe中取消科学记数法**

**pandas**

import pandas as pd pd.set\_option('display.float\_format',lambda x : '%.2f' %  
x) ----- pd.set\_option('display.max\_columns',  
10000, 'display.max\_rows', 10000)

pd.set\_option

pd.set\_option(pat, value)

参数：

display.[max\_categories, max\_columns, max\_colwidth, max\_info\_columns, max\_info\_rows, max\_rows,  
max\_seq\_items, memory\_usage, multi\_sparse, notebook\_repr\_html, pprint\_nest\_depth, precision,  
show\_dimensions]

**numpy**

import numpy as np np.set\_printoptions(suppress=True, threshold=np.nan)  
suppress=True 取消科学记数法  
threshold=np.nan 完整输出（没有省略号）

• **Dataframe 中行与列重命名 (index/ columns)**

data.rename(columns={0:'id\_yr\_mn'}, inplace= True)  
column\_indices = [1,4,5,6]  
new\_names = ['a','b','c','d']  
old\_names = df.columns[column\_indices]  
df.rename(columns=dict(zip(old\_names, new\_names)), inplace=True)

• **dataframe 按列统计nan值个数**

num = df.isna().sum()

• **Pandas对DataFrame单列/多列进行运算 (map, apply, transform, agg)**

1.单列运算 在Pandas中， DataFrame的一列就是一个Series, 可以通过map来对一行进行操作： df['col2'] =  
df['col1'].map(lambda x: x\*\*2) 其中lambda函数中的x代表当前元素。可以使用另外的函数来代替lambda函  
数， 例如： define square(x): return (x \*\* 2) df['col2'] = df['col1'].map(square) 2.多列运算 要对DataFrame  
的多个列同时进行运算， 可以使用apply， 例如col3 = col1 + 2 \* col2: df['col3'] = df.apply(lambda x:  
x['col1'] + 2 \* x['col2'], axis=1) 其中x代表当前行， 可以通过下标进行索引。 3.分组运算 可以结合groupby与  
transform来方便地实现类似SQL中的聚合运算的操作： df['col3'] = df.groupby('col1')  
['col2'].transform(lambda x: (x.sum() - x) / x.count()) 在transform函数中x.sum()与x.count()与SQL类似， 计  
算的是当前group中的和与数量， 还可以将transform的结果作为一个一个映射来使用， 例如： sumcount =  
df.groupby('col1')['col2'].transform(lambda x: x.sum() + x.count()) df['col1'].map(sumcount) 对col1进行  
一个map， 得到对应的col2的运算值。

• **Pandas中DataFrame将index设置为column**

df['index1'] = df.index or, .reset\_index: df.reset\_index(level=0, inplace=True)

• **pandas.read\_csv函数主要参数设置**

• **pandas 创建空dataframe**

df = pd.DataFrame()

• **pandas 重命名行或列**

xdat.rename(columns={0:'staNum'}, inplace= True)

• **pandas 删除列与apply 应用**

station.drop(columns=['F2','F3','F4','F6','F7','F8','x','y'], axis=1,inplace=True) station['staNum'] =  
station['staNum'].apply(lambda x:'CH0000'+str(x))