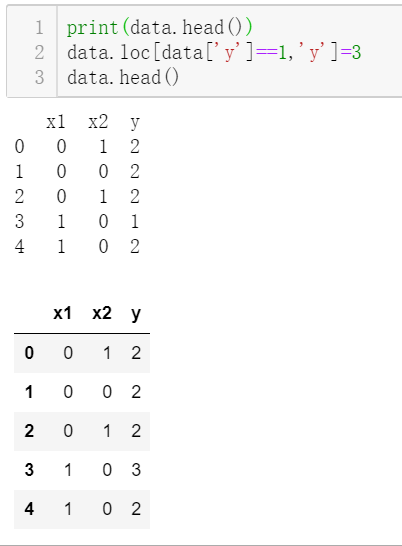
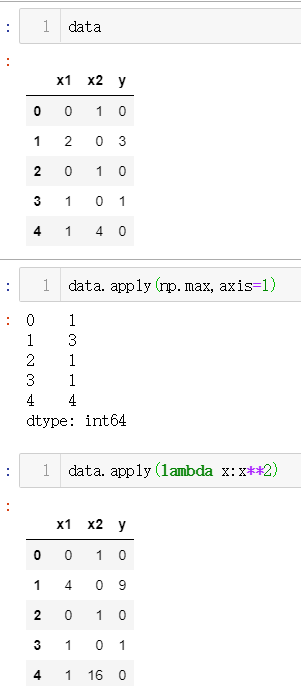
Python使用技巧

## 1、根据条件修改dataframe的值

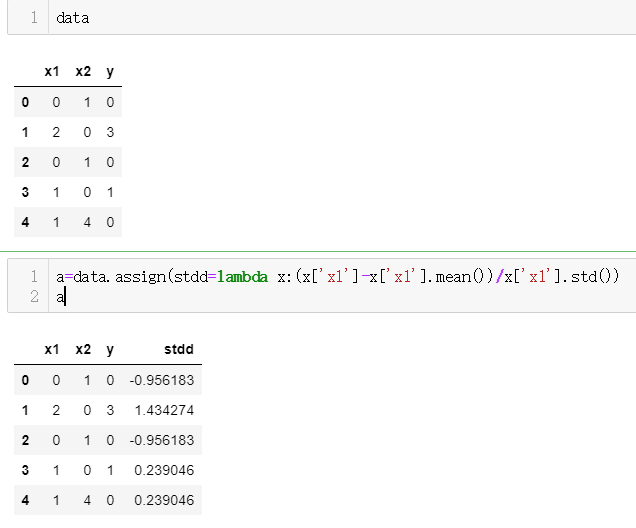
Data.loc[条件,修改列名]=修改的值



## 2、apply函数 ：Applying functions to the data



## 3、lambda函数，简单函数定义

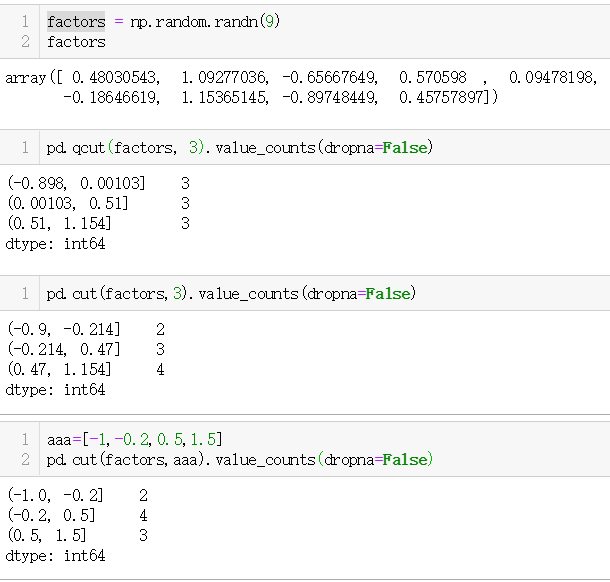


Apply lambda函数应用实例：<https://blog.csdn.net/anshuai_aw1/article/details/82347016>

## 4、Pandas — qcut( )与cut( )

<https://blog.csdn.net/starter_____/article/details/79327997>

Qcut等频分箱、cut等距分箱或，指定分箱距离



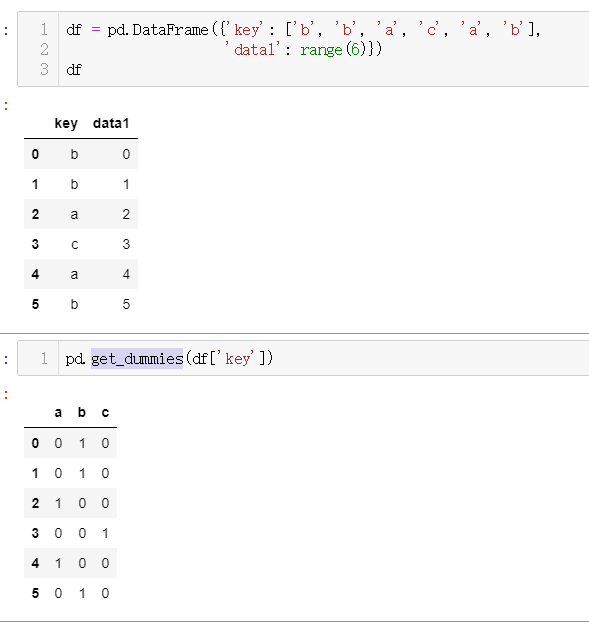
## 5、Map（）、applymap（）、apply（）比较

Apply用于dataframe上，对行或列操作，applymap用于dataframe上，对元素进行操作，map用于series上，对元素进行操作。



## 6、one-hot编码

Get\_dummies()



### 手动编码

from numpy import argmax

# define input string

data = 'hello world'

print(data)

# define universe of possible input values

alphabet = 'abcdefghijklmnopqrstuvwxyz '

# define a mapping of chars to integers

char\_to\_int = dict((c, i) for i, c in enumerate(alphabet))

int\_to\_char = dict((i, c) for i, c in enumerate(alphabet))

# integer encode input data

integer\_encoded = [char\_to\_int[char] for char in data]

print(integer\_encoded)

# one hot encode

onehot\_encoded = list()

for value in integer\_encoded:

letter = [0 for \_ in range(len(alphabet))]

letter[value] = 1

onehot\_encoded.append(letter)

print(onehot\_encoded)

# invert encoding

inverted = int\_to\_char[argmax(onehot\_encoded[0])]

print(inverted)

### 基于sklearn编码

from numpy import array

from numpy import argmax

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import OneHotEncoder

# define example

data = ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot']

values = array(data)

print(values)

# integer encode

label\_encoder = LabelEncoder()

integer\_encoded = label\_encoder.fit\_transform(values)

print(integer\_encoded)

# binary encode

onehot\_encoder = OneHotEncoder(sparse=False)

integer\_encoded = integer\_encoded.reshape(len(integer\_encoded), 1)

onehot\_encoded = onehot\_encoder.fit\_transform(integer\_encoded)

print(onehot\_encoded)

# invert first example

inverted = label\_encoder.inverse\_transform([argmax(onehot\_encoded[0, :])])

print(inverted)

### 基于keras

from numpy import array

from numpy import argmax

from keras.utils import to\_categorical

# define example

##对字符串

data = ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot']

values = array(data)

print(values)

# integer encode

label\_encoder = LabelEncoder()

integer\_encoded = label\_encoder.fit\_transform(values)

print(integer\_encoded)

##对数值

#data=[1, 3, 2, 0, 3, 2, 2, 1, 0, 1]

#data=array(data)

#print(data)

# one hot encode

encoded = to\_categorical(integer\_encoded)

print(encoded)

# invert encoding

inverted = argmax(encoded[0])

print(inverted)

## 7、缺失值、空值填充

### 固定值填充

data['灰度分'] = data['灰度分'].fillna('-99')

### 均值填充

Data[‘灰度分’]= data['灰度分'].fillna(data['灰度分'].mean())

### 众数填充

data['灰度分'] = data['灰度分'].fillna(data['灰度分'].mode()))

用上下数据填充

### 用前一个数据填充

data['灰度分'] = data['灰度分'].fillna(method='pad')

### 用后一个数据填充

data['灰度分'] = data['灰度分'].fillna(method='bfill')

### 用插值法填充

data['灰度分'] = data['灰度分'].interpolate()

用knn填充

from fancyimpute import KNN

dataset = KNN(k=3).complete(dataset)

### 参考资料

<https://blog.csdn.net/Katherine_hsr/article/details/80279963>

## 8、中文转编码

pd.get\_dummies(df, prefix=['col1', 'col2'])

df['size'] = df['size'].map(size\_mapping)

label\_mapping = {lab:idx for idx,lab in enumerate(set(df['label']))}

df['label'] = df['label'].map(label\_mapping)

## 9、缺失值统计

percent =(train.isnull().sum()/train.isnull().count()).sort\_values(ascending=False)

missing\_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

missing\_data.head(20)

## 10、merge、concat

<https://blog.csdn.net/LY_ysys629/article/details/73849543>

### merge 函数参数



### concat函数参数表格



## 11、json格式

json数据类型和python数据类型的对应关系

{} <——> dict

[] <——> list

"string" <——> "str"或u"unicode"

123.4 <——> int或float

true/false <——> True/False

null <——> None

把字典转换成json字符串：json.dumps(d)

把json字符串转成字典：json.loads(json\_str)

把字典转换成json字符串并写入文件：

with open('out.txt','w+') as f:

json.dump(d,f)

从文件中读取一个json字符串并转换为字典：

with open('out.txt','r') as f:

ret = json.load(f)

print ret

print type(ret)

## 12、数据探查

### 初略查看

train.head(5) #显示前5行数据

train.tail(5) #显示后5行

train.columns #查看列名

train.info() #查看各字段的信息

train.shape #查看数据集行列分布，几行几列

train.describe() #查看数据的大体情况

### 缺失率统计

percent =(train.isnull().sum()/train.isnull().count()).sort\_values(ascending=False)

missing\_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

missing\_data.head(20)

### 数据统计

#统计某一列中各个元素值出现的次数

train['MSSubClass'].value\_counts()

#

#列出数据的偏斜度

train['MSSubClass'].skew()

#列出数据的峰度

train['MSSubClass'].kurt()

#计算两个列的相关度

train['LotFrontage'].corr(train['LotArea'])

#观察两个列的值的二维图

x = 'GrLivArea'；y = 'SalePrice'

data = pd.concat([train[y], train[x]], axis=1)

data.plot.scatter(x=x, y=y, ylim=(0,800000));#这里800000为y的最大值

#计算所有特征值每两个之间的相关系数，并作图表示。

corrmat = train.corr()#得到相关系数

f,ax = plt.subplots(figsize = (12,9))

sns.heatmap(corrmat, vmax = .8, square = True)#热点图

#取出相关性最大的前十个，做出热点图表示

k = 10 #number of variables for heatmap

cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index

cm = np.corrcoef(train[cols].values.T)

sns.set(font\_scale=1.25)

hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot\_kws={'size': 10}, yticklabels=cols.values, xticklabels=cols.values)

plt.show()

### 数据分组

#按照用户分组---------------------一个特征值

train.groupby('userid',as\_index=False)

#按照用户与目的地分组---------------两个特征值

train.groupby(['userid','end\_loc'],as\_index=False)

#用户、起点、目的地-----------------三个特征值

train.groupby(['userid','start\_loc','end\_loc'],as\_index=False)

#跟MSSubClass进行分组，并求分组后的平均值

train[['MSSubClass', 'LotFrontage']].groupby(['MSSubClass'], as\_index=False).mean()

#选取特定的属性的某个值然后进行分类

train[train['date']=='2017-1-2'].groupby(['userid'],as\_index=False)

data.groupby(‘title’).size()

### 数据合并

#获得分组后，统计分组中'end\_loc'的数量返回为一列由‘userid’和‘user\_count’组成的新的DataFrame

user\_count = train.groupby('userid',as\_index=False)['end\_loc'].agg({'user\_count':'count'})

#将获得的新的DataFrame合并到train，更多的merge参数请查阅文档

train= pd.merge(train,user\_count,on=['userid'],how='left')

user\_eloc\_count = train.groupby(['userid','end\_loc'],as\_index=False)['userid'].agg({'user\_eloc\_count':'count'})

train= pd.merge(train,user\_eloc\_count,on=['userid','end\_loc'],how='left')

### 数据连接

#讲训练数据与测试数据连接起来，以便一起进行数据清洗。

#这里需要注意的是，如果没有后面的ignore\_index=True，那么index的值在连接后的这个新数据中是不连续的，如果要按照index删除一行数据，可能会发现多删一条。

merge\_data=pd.concat([train,test]，ignore\_index=True)

#另一种合并方式，按列名字进行合并。

all\_data = pd.concat((train.loc[:,'MSSubClass':'SaleCondition'], test.loc[:,'MSSubClass':'SaleCondition']))

数据转换、标准化、归一化

### 数值转换

#浮点型数值转换为整型

train['Age']=train['Age'].astype(int)

#字符串的替换--映射

train['MSZoning']=train['MSZoning'].map({'RL':1,'RM':2,'RR':3,}).astype(int)

train['Embarked'] = train['Embarked'].map( {'S': 0, 'C': 1, 'Q': 2} ).astype(int)

#一般建议将map拿出来

title\_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}

train['Title'] = train['Title'].map(title\_mapping)

train['Title'] = train['Title'].fillna(0)

#将字符串特征列中的内容分别提出来作为新的特征出现，表现为0、1。

train= pd.get\_dummies(houseprice)

#将连续型特征值分块，每一块用数字标识

train.loc[ train['Fare'] <= 7.91, 'Fare'] = 0

train.loc[(train['Fare'] > 7.91) & (train['Fare'] <= 14.454), 'Fare'] = 1

train.loc[(train['Fare'] > 14.454) & (train['Fare'] <= 31), 'Fare'] = 2

train.loc[ train['Fare'] > 31, 'Fare'] = 3

train['Fare'] = train['Fare'].astype(int

train['SalePrice'] = np.log(train['SalePrice'])

#将偏斜度大于0.75的数值列log转换，使之尽量符合正态分布。

skewed\_feats = train[numeric\_feats].apply(lambda x: skew(x.dropna())) #compute skewness

skewed\_feats = skewed\_feats[skewed\_feats > 0.75]

skewed\_feats = skewed\_feats.index

all\_data[skewed\_feats] = np.log1p(all\_data[skewed\_feats])

### 数据标准化和归一化（Standardization、Normalization）

标准化归一化概念不再赘述，实际使用时最主要的还是要了解什么时候需要标准化，什么时候用归一化，还需要清楚当前数据适合什么标准化方式等等。

在sklearn.preprocessing 介绍的标准化方式有：

1. preprocessing.scale()、preprocessing.StandardScaler()，使数据集呈现标准正态分布，即mean = 0，且标准差std = 1。

2. MinMaxScaler 、MaxAbsScaler，前者使数据集分布在[0,1]，后者分布在[-1,1]。这种方式通常在(1) 特征的标准差较小 (2) 可以使稀疏数据集中的0值继续为0，这两种情况下使用。

3. preprocessing.QuantileTransformer()，将数据映射到[0,1]之间均匀分布，会破坏原数据之间的相关特性。

4. 归一化方式：preprocessing.normalize()，将样本缩放成单位向量，(1)需要使用二次方程，比如点积或者其他核方法计算样本对之间的相似性(2)常用于文本分类和内容聚类的向量空间模型的基础。

### 相关系数和协方差corr(), cov()

Df.corr(df.b)

Df.corrwith(df.b) #b列与其他列的相关系数

### 参考资料

<https://blog.csdn.net/u011094454/article/details/77618604>

<https://blog.csdn.net/helen1313/article/details/53301198>

## 13、pivot\_table数据透视表

<https://blog.csdn.net/chencheng126/article/details/50119215>

<https://nbviewer.jupyter.org/url/pbpython.com/extras/Pandas-Pivot-Table-Explained.ipynb>

## 14、数据类型转换

Np.astype(dtype) 手动指定数据类型

Astype(data1.dtype) 继承其他数据的类型

数据跟列表最大的区别是，赋值操作，数据修改源数据，列表不会修改源数据。

需要得到ndarray一份复制数据数据，需要显示的制定，如data[5:8].copy() 否则只是源数据的一个镜像。

## 15、ndarray

data[2,3]等价data[2][3]

不等式条件筛选

data[names!=’bob’]或者data[-(names==’bob’)]

多条件用& 或者|联接

矩阵转置data.T

## 16、numpy.where(cond,xarr,yarr)

## 17、数组统计方法

argmin、argmax、cumsum累计求和、cumprod累计积

arr=np.array([[0,1,2],[3,4,5],[6,7,8]])

arr.cumsum(0)

arr.cumprod(1)

## 18、any和all

Any判断任一一个满足要求，all判断所有的满足要求

## 19、sort

Arr.sort()

## 20、unique,set唯一值

Np.unique(names)

## 21、in1d

测试一个数组中的值在另一个数组成员资格

## 22、集合运算

Unique(x) 返回唯一有序结构

intersect1d（x,y）x,y的公共元素

union1d(x,y) 并集

in1d(x,y) x的元素是否包含于y

setdiff1d(x,y) 集合的差，元素在x中且不在y中

setxor1d(x,y) 集合的对称差

## 23、np.save np.load读写磁盘数据

Np.save(‘some\_array’,arr)

Np.load(‘some\_array.npy’)

## 24、np.loadtxt np.savetxt

Arr=np.loadtxt(‘array\_ex.txt’,delimiter=’,’)

Np.genfromtxt读取数据

25、dot矩阵乘法

x.dot(y)

## 26、numpy.linalg矩阵分解运算或者求逆，行列式

From numpy.linalg import inv,qr

Diag 返回方阵的对角线元素

Dot 矩阵乘法

Trace 对角线元素的和

Det 矩阵行列式

Eig 特征及特征向量

Inv 逆矩阵

Pinv 矩阵的moore-penrose谓逆

Qr qr分解

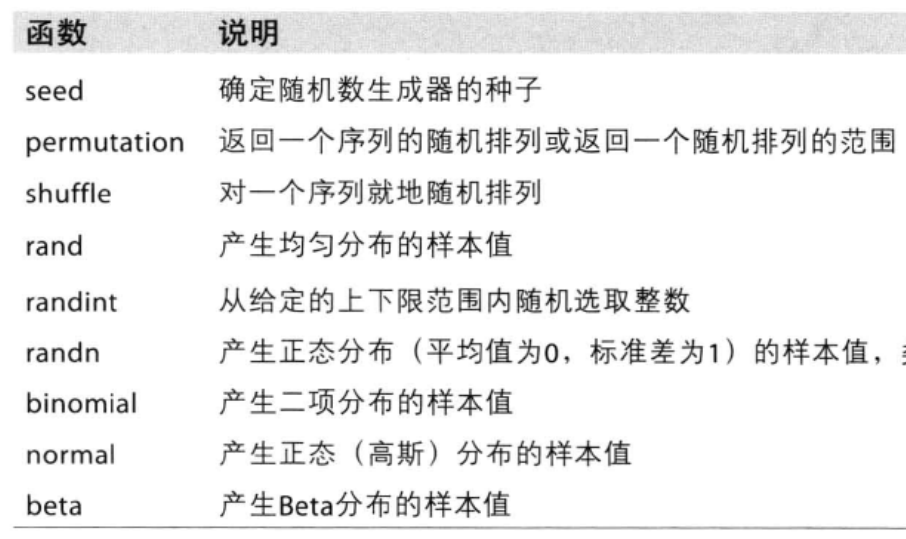
Svd 奇异值分解

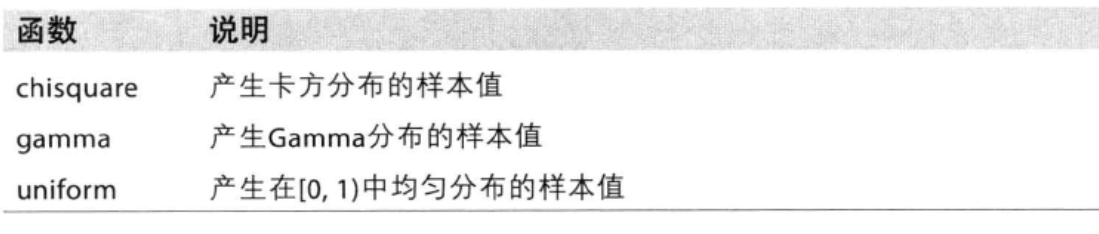
Solve 线性方程组求解

Lstsq 最小二乘解

## 27、随机生成

np.random.normal正态分布概率





## 28、缺失值检测

isnull、notnull判断

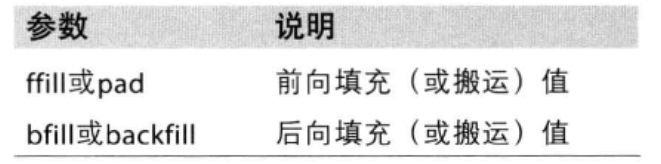
## 29、del 删除dataframe的某些列

## 30、reindex重建索引

Reindex会根据之前的索引重新排序,没有的序列会插入Na值

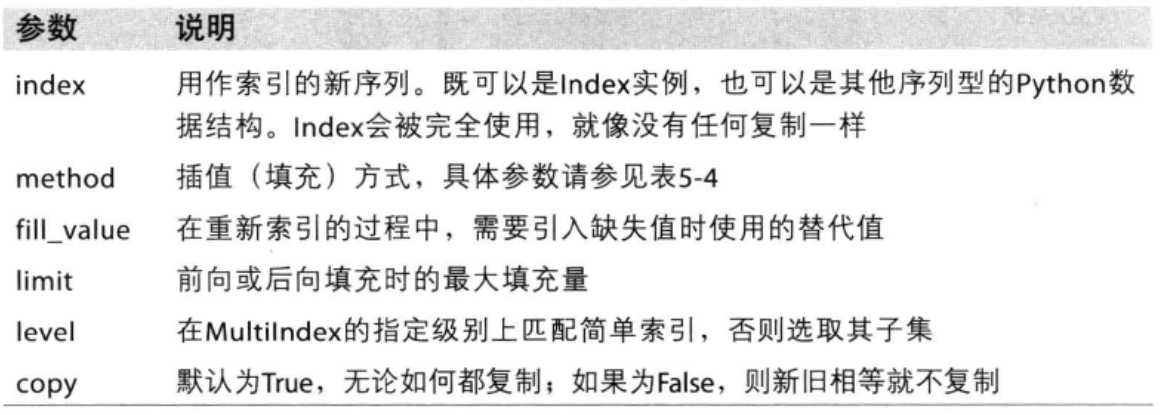
Obj.reindex([‘a’,’b’,’c’,’d’,’e’],fill\_value=0)

Obj.reindex([‘a’,’b’,’c’,’d’,’e’],method =’ffill’)



Data.reindex=(index=[‘a’,’b’,’c’,’d’,’e’],method=’ffill’,columns=[‘Texas’,’Utah’,’california’])

Data.ix[[‘a’,’b’,’c’,’d’,’e’], [‘Texas’,’Utah’,’california’]]

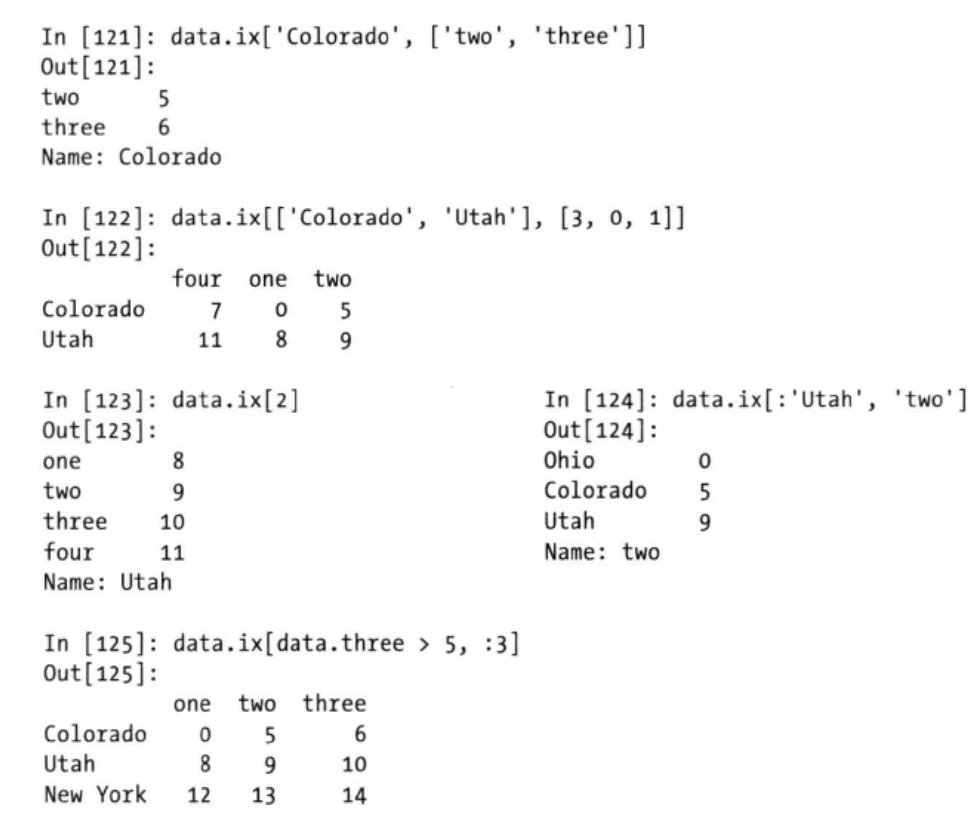


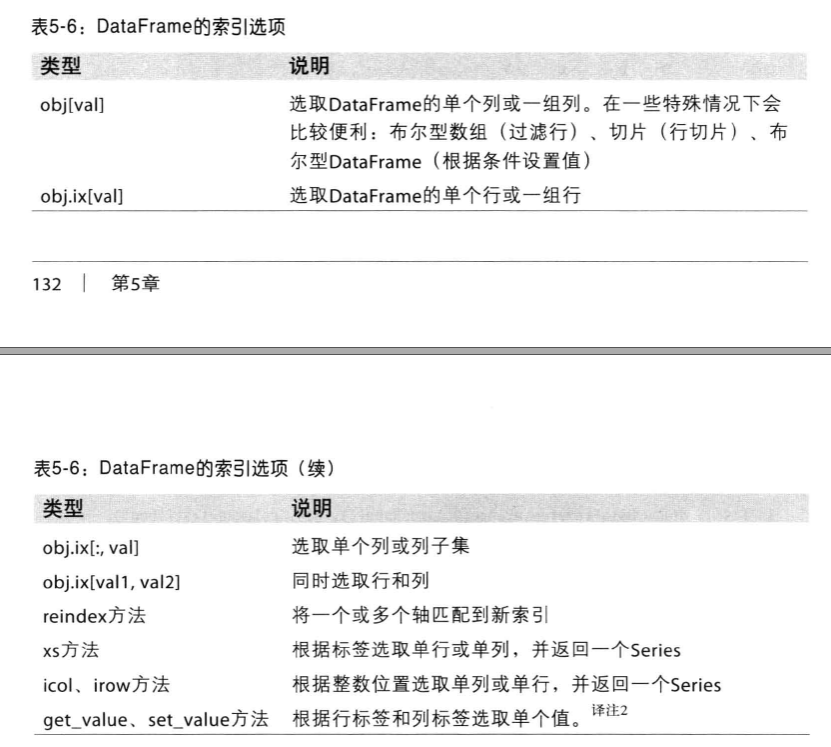
## 31、drop

Data.drop([列名、列名])

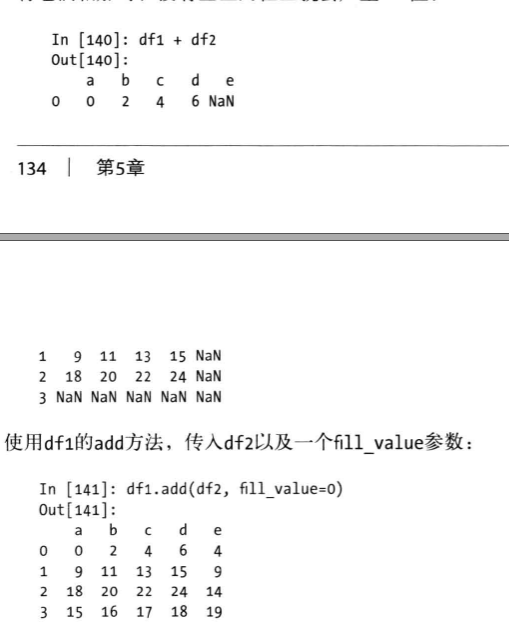
Data.drop([行名，行名],axis=1)

## 32、行索引ix



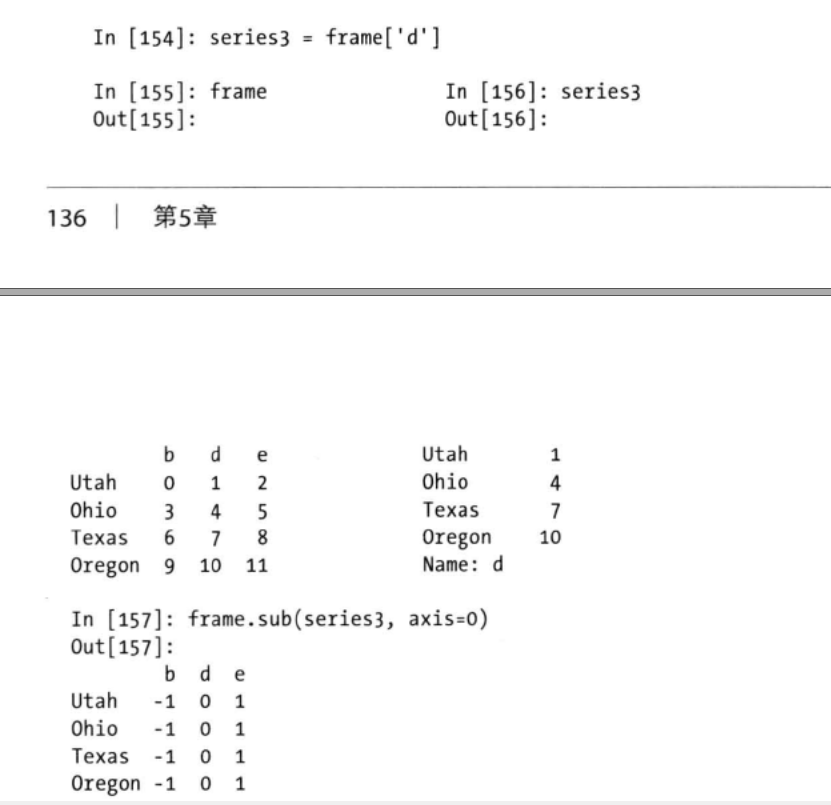


## 33、算数运算值填充





## 34、广播



## 35、函数及映射

### Apply 作用数组上

F=lambda x:x.max()-x.min()

Data.apply(f) data.apply(f,axis=1)

### Applymap 作用于元素上

Format=lambda x: ‘%.2f’ %x

Data.applymap(format)

### Serises作用于元素的map方法

Data[‘e’].map(format)

36、排序

Sort\_index() 按索引排序

Data.sort\_index() data.sort\_index(axis=1,ascending=True)

Series排序用order Na值会放到最后

Dataframe排序用by指定列

Data.sort\_index(by=[‘a’,’b’],ascending=False)

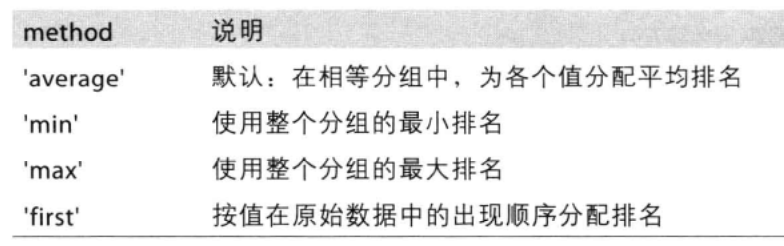
## 37、排名rank

Series.rank() 重复值会产生小数

Series.rank(method=’first’)按出现的顺序排序，整数排序

Series.rank(method=’first’,ascending=False) 降序排列

Data.rank(axis=1) dataframe可以在行或者列上排序



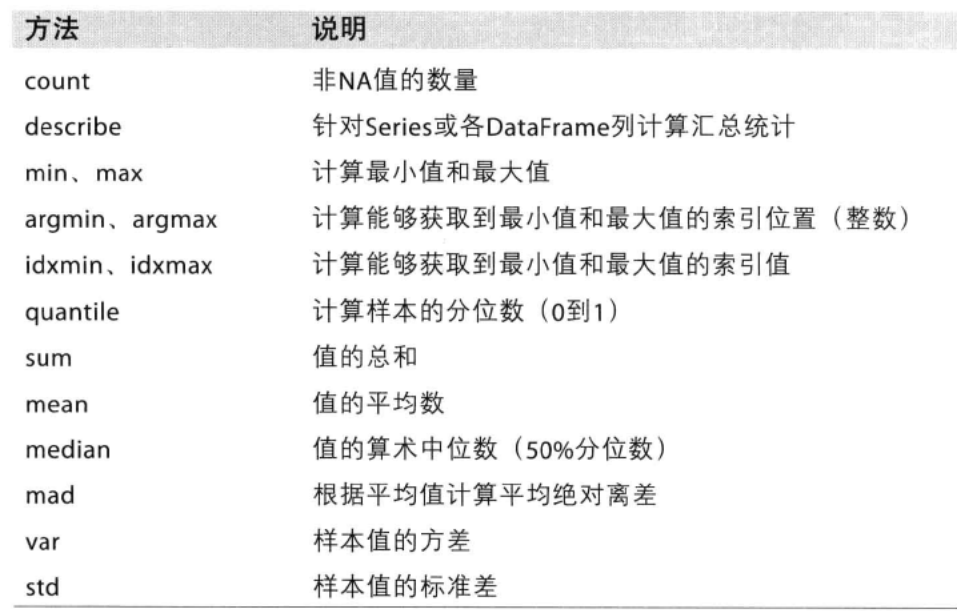
## 38、汇总及统计

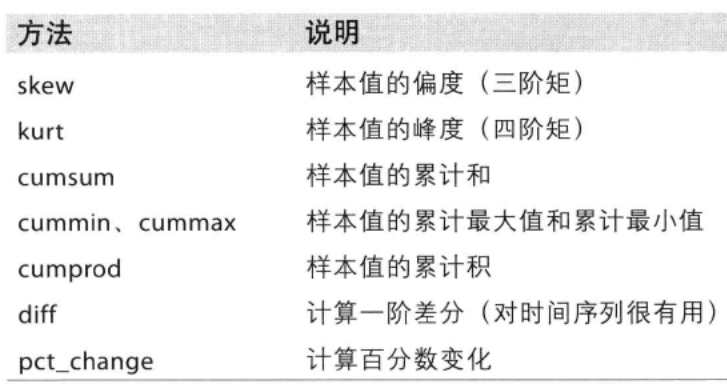
Data.sum() 列汇总 data.sum(axis=1) 行汇总，默认会跳过Na值，可通过skipna设置不跳过Na

Idmax，idmin最大值的索引

Cumsum累加

Data.describe()最大最小，均值方差，分位数 非数值型，返回稍有不同





## 39、相关系数、协方差

Corr 相关系数、cov协方差

Corrwith计算指定列与其他列的相关系数



## 40、唯一值，值计数、成员资格

Series.unique() 唯一值

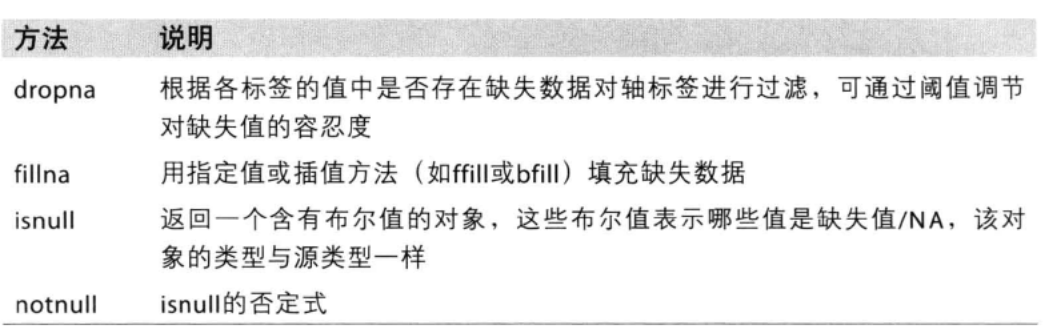
Value\_counts() 值计数

Isin(成员资格)

## 41、缺失值处理

Series处理

Data.isnull()判断是否缺失值



Data.dropna() 删除缺失值

Data[data.notnull()] 通过否定式过滤

Dataframe处理

Data.dropna() 删除所有含有Nan的行，可以通过axis=1指定删除含有Na的列。

全部删除通过传入参数

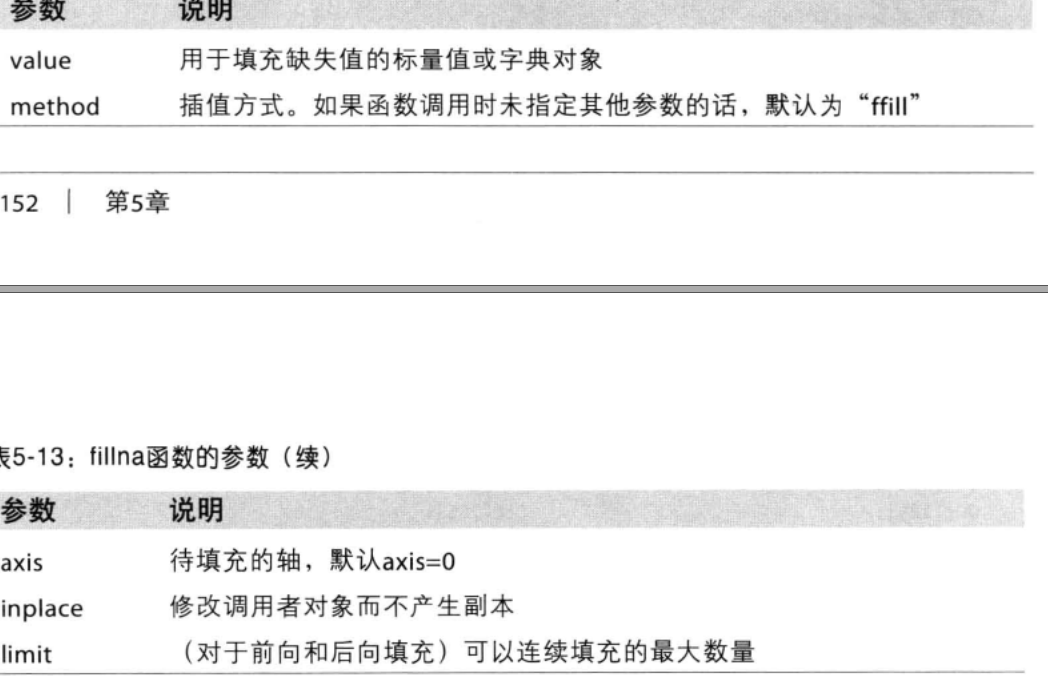
Data.dropna(how=’all’,axis=1)

保留部分可以通过thresh参数 data.dropna(thresh=3)

## 42、缺失值填充

Data.fillna(0,inplace=True) 缺失值填充

Data.fillna(method=’ffill’,limit=2)



## 43、根据级别汇总统计

Data.sum(level=’key2’)

Data.sum(level=’color’,axis=1)

Set\_index(),reset\_index()

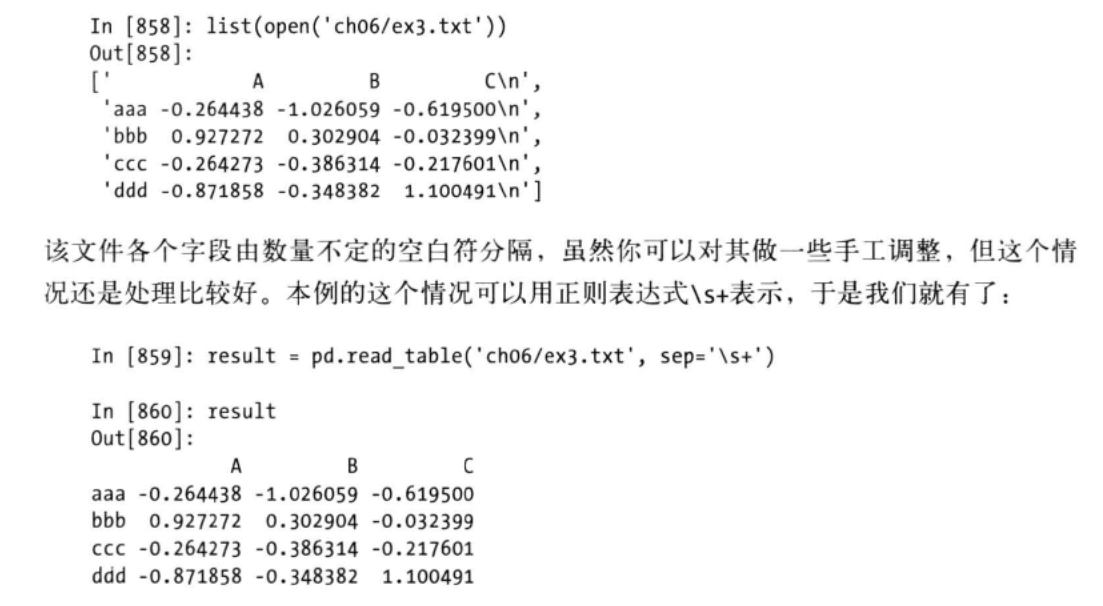
## 44、数据读取

Read\_csv读取文件、url带分隔符的数据，默认以逗号分隔

Read\_table 读取文件，url带分隔符的数据，默认以\t分隔

Read\_csv(‘path’,sep=’,’,header=None,columns=[],index\_column=,encoding=)

Read\_csv(‘path’,sep=’\s+’) 以正则表达式分隔

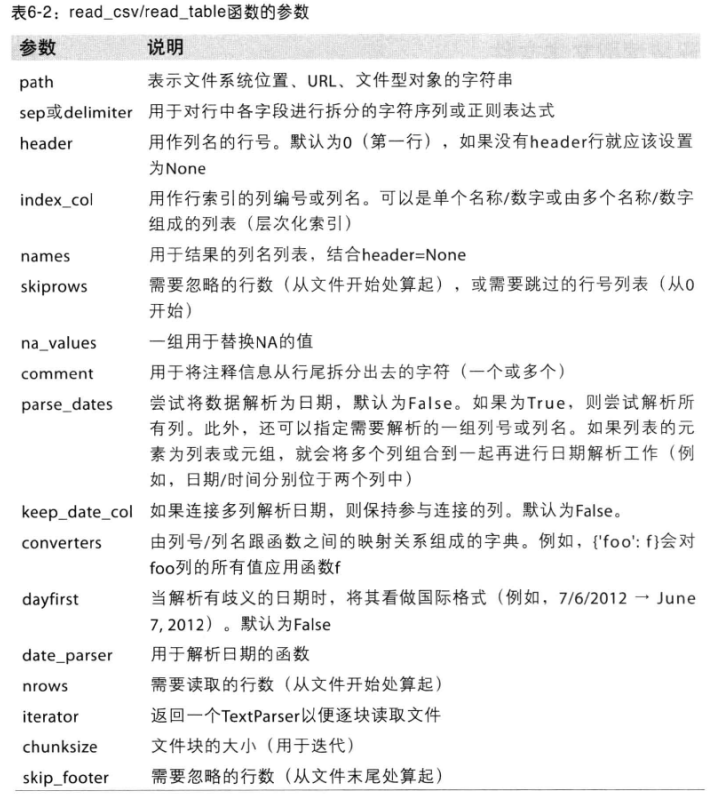


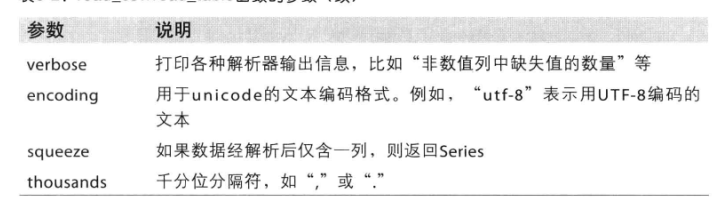
### 跳过某些行

Read\_csv(‘path’,skiprows=[0,2,3])

指定缺失值

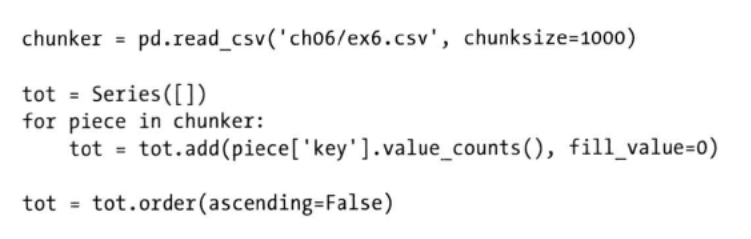
Read\_csv(‘path’,na\_values=[‘ ‘,’Null’,’Na’])





指定读取几行

Read\_csv(‘path’,nrows=5) 读取5行

Read\_csv(‘path’,chunksize=1000)

### 写数据到文件

Data.to\_csv(‘path’,sep=’|’,na\_rep=’NULL’,index=False,header=False,cols=[‘a’,’a’,’c’])

Series 读取

Series.from\_csv(‘path’,parse\_dates=True)

## 45、Json格式

Json.loads()

Json.dumps()

## 46、二进制文件读写pickle

With open(‘path/a.pickle’,’wb’) as fs:

Pickle.dump(data,fs,0)

With open(‘path/a.pickle’,’rb’) as fs:

Data=pickle.load(fs,0)

## 47、hdf5格式(二进制)待研究

H5py pytable

## 48、html解析 待研究

<https://blog.csdn.net/qq_42787271/article/details/81612428>

## 49、读取excel文件 待研究

Read\_excel

Xlrd

Openpyxl

## 50、读取数据库数据 待研究

## 51、数据合并

### Merge

Pd.merge(df1,df2,on=’key’)

Pd.merge(df1,df2,left\_on=’lkey1’,right\_on=’rkey’)

Pd.merge(df1,df2,on=’key’,how=’inner’)

Pd.merge(df1,df2,on=[‘key1’,’key2’],how=’outer’) 多列关联





Pd.merge(df1,df2,left\_on=’key’,right\_index=True)

### Concatenate

Np.concatenate([arr1,arr2],axis=1)

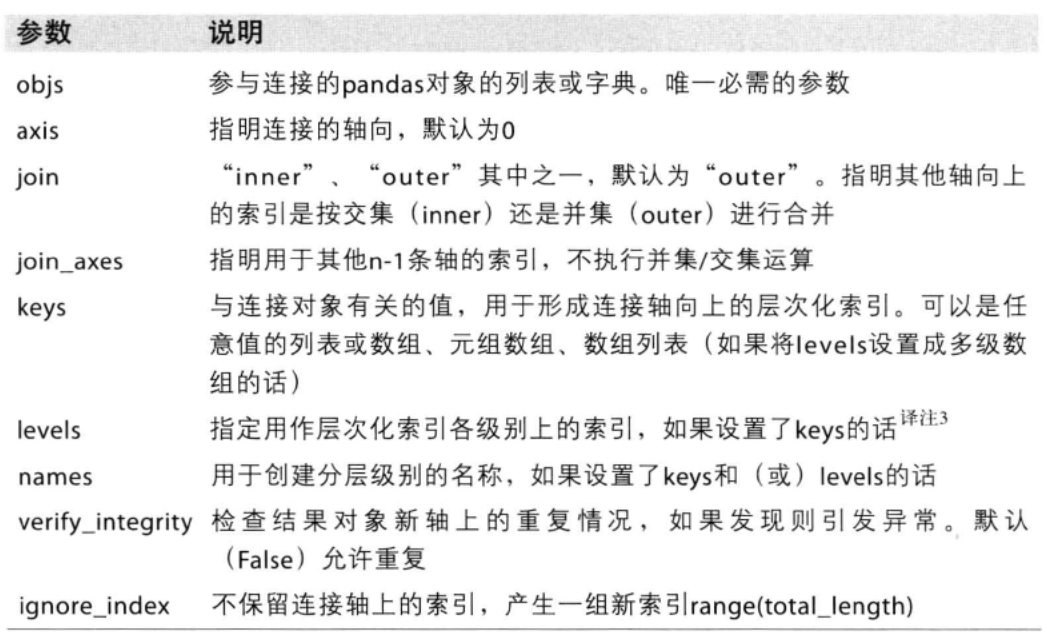
### Concat

Pd.concat([df1,df2,df3])

pd.concat([df1,df2],axis=1)

pd.concat([df1,df2],axis=1,join=’inner’)

pd.concat([df1,df2],axis=1,ignore\_index=True)



Np.where(data.isnull(a),b,c)

## 52、重塑及旋转

Reshape、pivot

Stack unstack

Data.stack() 行列转换

层次化索引 data.unstack()转换为dataframe

## 53、移除重复数据

Data.duplicated() 检查行是否重复

Data.drop\_duplicates()

## 54、数据转换

Data[‘foog’].map(str.lower).map(meat\_to\_animal)

Data[‘food’].map(lambda x:meat\_to\_animal[x.lower()])

55、替换值

Data.replace(-9999,np.nan)

## 56、重命名轴索引

Data.rename(index=[],columns=[])

## 57、离散化-分箱

Pd.cut(data[‘age’],bin=,labels=) 指定距离分箱

开区间可以通过参数right=False来修改

Pd.cut(data[‘age’],4) 等距分箱

Pd.qcut(data[‘age’],5) 等频分箱，可以设定指定频率分箱

Pd.qcut(data[‘age’],[0,0.1,0.4,0.9,1]) 指定分位数分箱

58、过滤异常值

Data[np.abs(data)>3].any(1)

Data[np.abs(data)>3]=np.sign(data)\*3 将绝对值大于3的，限制到-3或者3

## 58、随机采样

Data.take(np.random.permutation(5))

Data.take(np.random.permutation(len(data)[:4])

## 59、哑变量

Pd.get\_dummies(df[‘key’])

Pd.get\_dummies(df[‘key’],prefix=’key’)

## 60、字符串操作

Val.split(‘,’) 分列

Val.strip() 去掉前后空格

‘::’.join(pieces) 双冒号把字符串连接起来

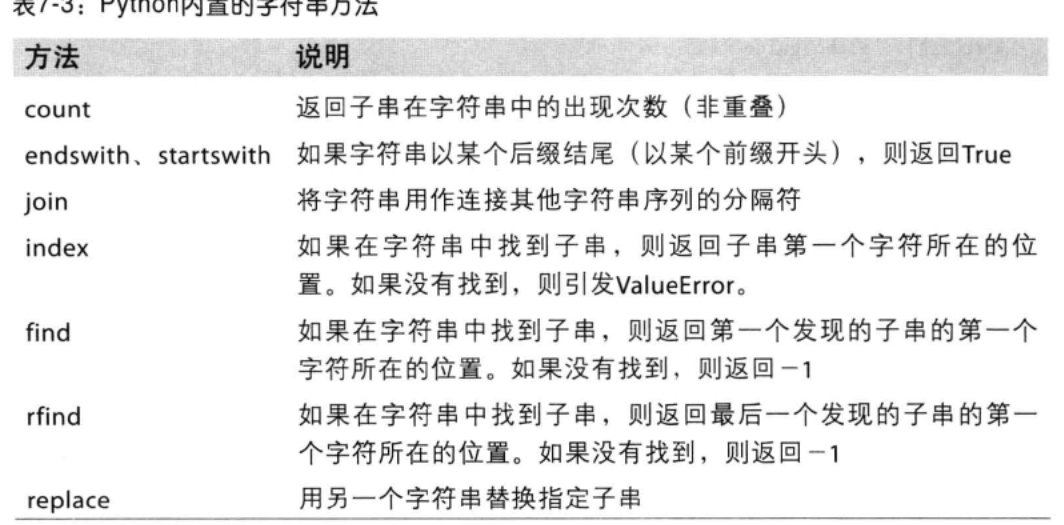
字符串定位 in ‘guido’ in var

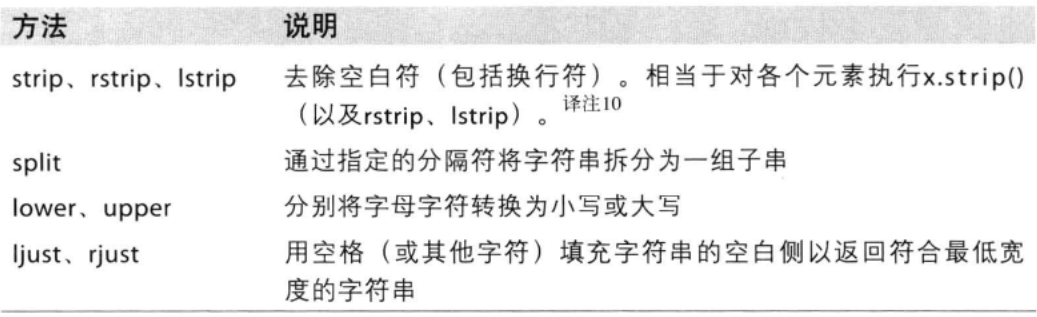
Var.index(‘a’) 找不到a会引发异常

Var.find(‘a’) 找不到a会返回-1

Val.count(‘a’) a出现的次数

Val.replace(‘,’,’’) 将val中，替换为空





## 61、正则表达式re模块 待研究

<http://regex.learncodethhearway.org/book/>

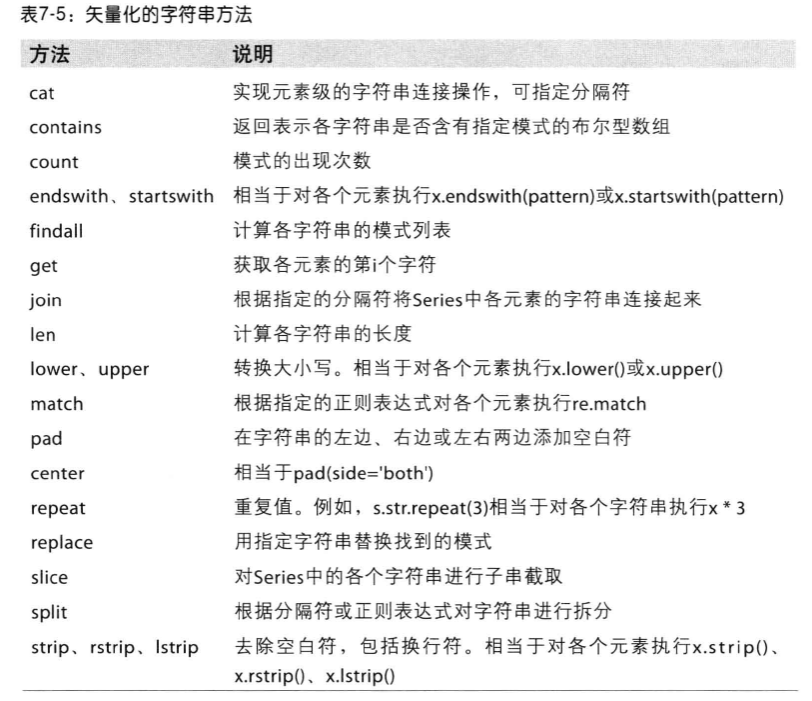
re 模式匹配、替换、拆分

match/search/findall



## 62、padans其他str方法

Str.contains(‘gmail’)



63、sort\_index,sort\_value对index排序和值排序

64、data.loc通过标签选择，data.iloc通过索引排序；