# IPython

***主要用于交互式数据处理和利用matplotlib对数据进行可视化处理。***

1. ***类似Mathematica的HTML笔记本***
2. ***基于QT框架的GUI控制台，其中含有绘图、多行编辑以及语法高亮显示等***

***->%run 命令 所有文件通过%run命令当做python程序运行。***

* ***->基于Qt的GUI控制台***

***ipython qtconsole --pylab=inline***

* ***内省***

***在变量的前面或后面加上一个问号（？）可以将有关该对象的一些通用信息显示出来。***

* ***目录书签系统-%bookmark***

***保存常用目录的别名以便实现快速跳转。***

***定义书签：%bookmark db /home/wesm/Dropbox/***

***执行%cd实现跳转***

* ***交互式调试器 %debug***

***u(或up)和d（或down）即可在栈跟踪***

***run -d <filename.py>***

* ***->测试代码的执行时间 %time 和 %timeit***

***%time一次执行一条语句，报告总体执行时间。***

***%timeit 任意语句自动执行多次，生成一个非常精确的平均执行时间***

* ***->基本性能分析：%prun和%run -p***

***cProfile在执行一个程序或代码快时，记录各函数所耗费的时间。***

***~在命令行使用，它将执行整个程序然后输出各函数的执行时间。***

***python -m cProfile cprof\_example.py***

***%prun 和 %run -p 编程的方式分析任意代码块的性能。***

***eg:%prun -l 7 -s cumulative run\_experiment()***

# NumPy

* ***Numberical Python :是Python科学计算的基础包。***

***特点：***

1. ***多维数组对象ndarray；***
2. ***对数据执行元素级计算以及直接对数组执行数学运算的函数；***
3. ***读写硬盘上基于数组的数据集的工具；***
4. ***线性代数运算、傅立叶变换、以及随机数生成；***
5. ***用于将C、C++、Fortran代码集成到Python的工具；***
6. ***算法之间传递数据的容器；***

* ***理解NumPy数组以及面向数组的计算。***

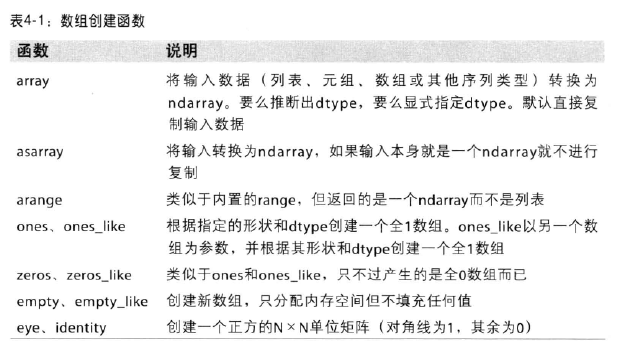
***对于大部分数据分析，主要关注功能集中在：***

* 1. ***用于数据整理和清理、子集构造和过滤、转换等快速的矢量化数组运算；***
  2. ***常用的数组算法、如排序、唯一化、集合运算等。***
  3. ***高效描述统计和数据聚合、摘要运算***
  4. ***用于异构数据集的合并、连接运算的数据对齐和关系型数据运算；***
  5. ***将条件逻辑表述为数组表达式***
  6. ***数据的分组运算（聚合、转换、函数应用等）***
* ***NumPy的ndarray：一种多维数组对象***

***Numpy最重要的一个特点是其N维数组对象（即ndarray），该对象是一个快速灵活的大数据容器。***

***Ndarray是一个通用的数据多维容器，即其中的所有元素必须是相同类型的；每个数组都有一个shape（一个表示各维度大小的元组）和一个dtype（一个用于说明数组数据类型的对象）***

* ***数组创建函数：***



* ***ndarray的数据类型***

***dtype（数据类型）是一个特殊的对象，它含有ndarray将一块内存解释为特定数据类型所需的信息。***

***通过ndarray的astype方法显式转换为其dtype***

***如果将浮点数转换成整数，则小数部分将会被截断。***

***如果某字符串数组表示的全是数字，可以用astype将其转换为数值形式。***

***大小相等的数组之间的任何算是运算都会将运算运用到元素级。***

***不同大小的数组之间的运算叫广播（broadcasting）***

* ***基本的索引和切片***

***与列表最重要的区别在于，数组切片是原始数组的视图，意味着数据不会被复制。***

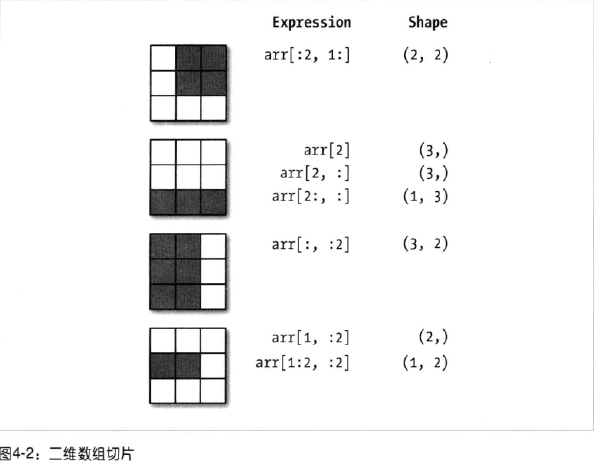
***np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])***

***括号外面的“维度”是一维、二维、三维、四维之类，而括号里面的可理解为“轴”，即“返回的低维数组含有原始高维数组某条轴上的所有数据”。***

* ***切片索引***

***可以在一个或多个轴上进行切片，也可以跟整数索引混合使用。***

***“只有冒号”表示选取整个轴。例如：arr2d[2,:1]***



* ***布尔型索引***

***布尔型数组的长度必须跟被索引的轴长度一致。***

***将布尔型数组跟切片、整数（或整数序列）混合使用。例如：data[names==’Bob’,2:]***

***可以使用不等于符号(!=),也可以通过负号（-）对条件进行否定。***

***选取三个条件中的两个组合运用多个布尔条件，使用&(和)、|（或）***

* ***花式索引***

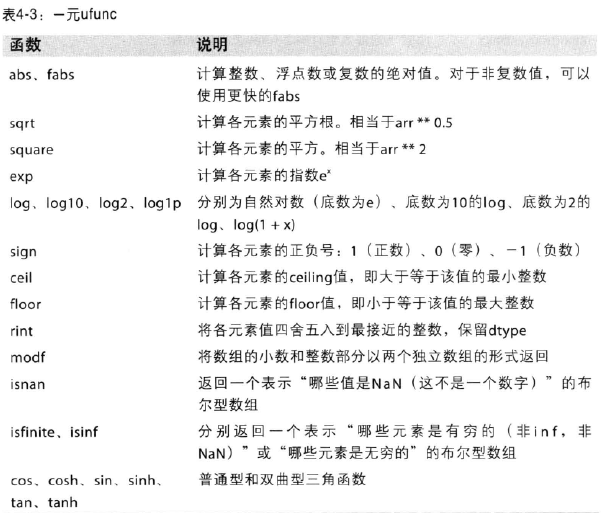
***花式索引（Fancy indexing）是一个Numpy术语，它指的是利用整数数组进行索引。***

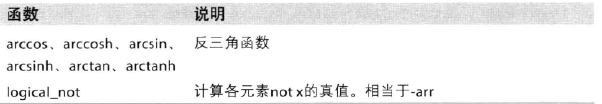
* ***数组转置和轴对换***

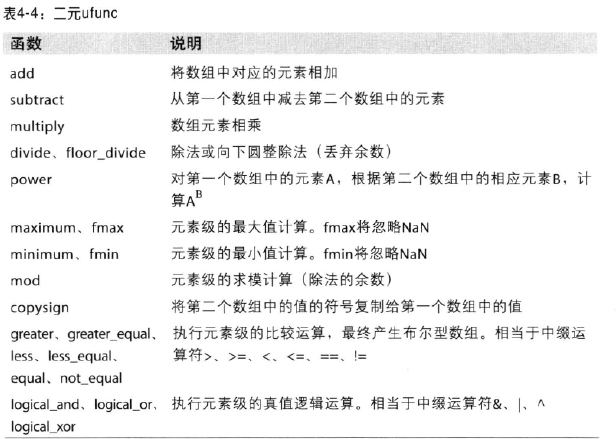
***转置（transpose）是重塑的一种特殊形式，它返回的是源数据的视图。***

* ***通用函数：快速的元素级数组函数***

***通用函数（即ufunc）是一种对ndarray中的数据执行元素级运算的函数。***





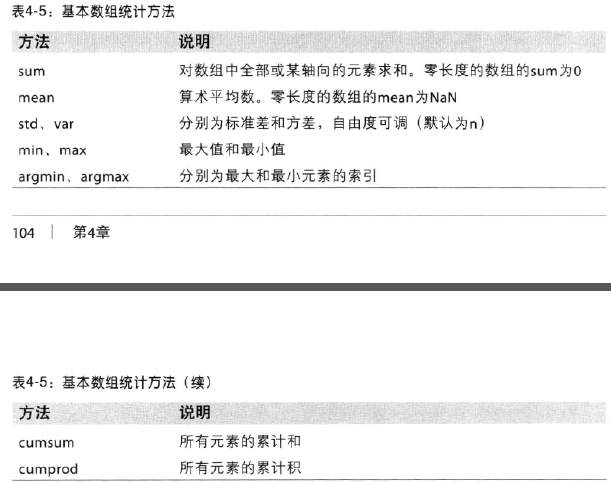


* ***利用数组进行数据处理***

***用数组表达式代替循环的做法，被称为矢量化。（一般来说，矢量化数组运算比等价的纯Python的方式快上一两个数量级）***

* ***数学和统计方法***

***通过数组上的一组数学函数对整个数组或某个轴向的数据进行统计计算。***

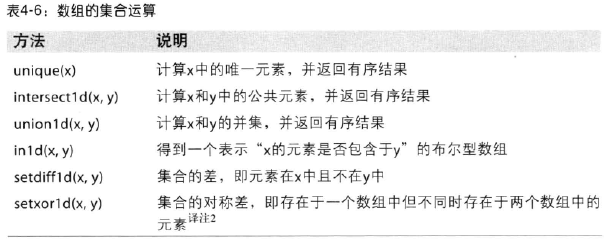


* ***排序***

***Numpy数组通过sort方法进行排序***

* ***唯一化以及其他的集合逻辑***

***NumPy提供了一些针对一维ndarray的基本集合运算。最常用的np.unique***



* ***将数据以二进制格式保存到磁盘***

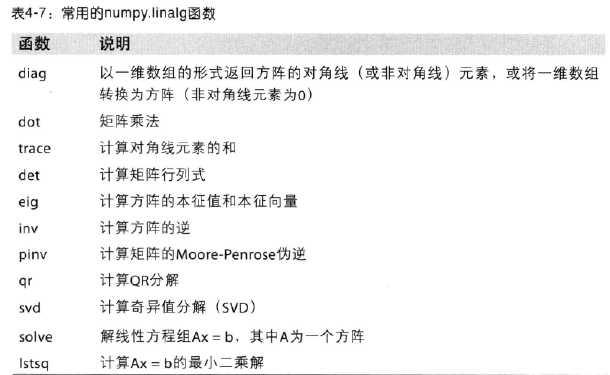
***np.save和np.load是读写磁盘数组数据的两个主要函数。默认情况下，数组是以未压缩的原始二进制格式保存在扩展名为.npy的文件中。***

***np.savez可以将多个数组保存到一个压缩文件中，将数组以关键字参数的形式传入。***

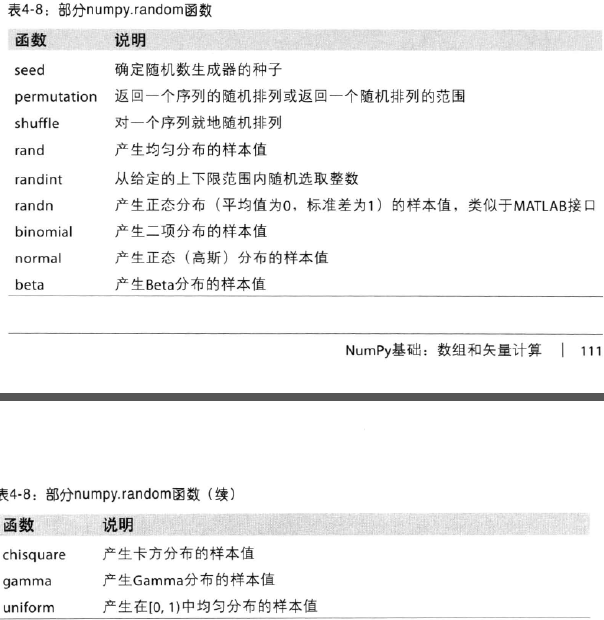
* ***存取文本文件***

***Loadtxt***

* ***线性代数***



* ***随机数生成***



# Pandas

***#Pandas其能做什么；#在数据分析中运用场景***

***提供快速处理结构化数据的大量数据结构和函数。***

* ***主要数据结构：Series 和 DataFrame***
* ***Series***

***一种类似于一维数组的对象，它由一组数据以及一组与之相关的数据标签（即索引）组成。***

***\*在算术运算中自动对齐不同索引的数据。***

***Series对象本身及其索引都有一个name属性。***

***Series的索引可以通过赋值的方式进行修改。***

* ***DataFrame***

***一个表格型的数据结构，它含有一组有序的列，每列可以是不同的值类型（数值、字符串、布尔值等）***

***DataFrame既有行索引也有列索引，***

***构建DataFrame，最常用的一种是直接传入一个由等长列表或NumPy数组组成的字典。***

***data = {‘state’:[‘Ohio’,’Ohio’,’Ohio’,’Nevada’,Nevada’],***

***‘year’:[2000,2001,2002,2001,2002],***

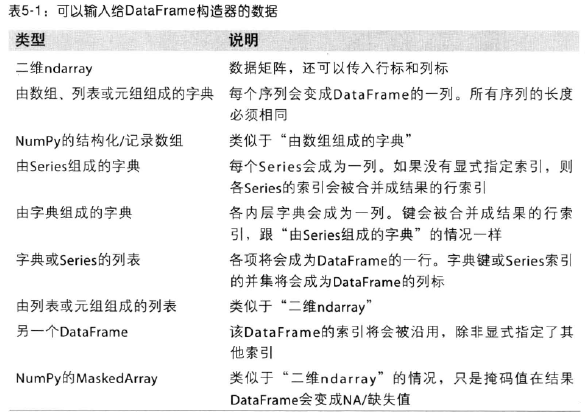
***‘pop’:[1.5,1.7,3.6,2.4,2.9]}***

***Frame = DataFrame(data)***

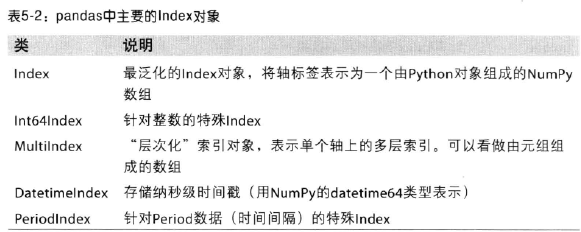
* ***关键字del用于删除列***

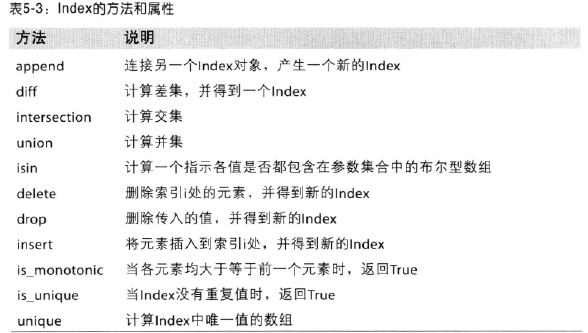
***通过索引方式返回的列只是想要数据的视图，并不是副本，因此，对返回的Series所做的任何修改都会反映在源DataFrame上。***

* ***另一种常见的数据形式是嵌套字典。即：外层字典的键作为列，内层键作为行索引。***



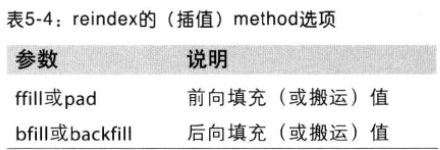
* ***索引对象***

***Pandas的索引对象负责管理轴标签和其它元数据（比如轴名称等）。*** 

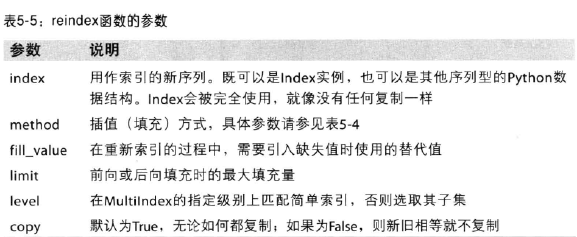


* ***重新索引***

***对于时间序列，重新索引时需要做一些插值处理。***



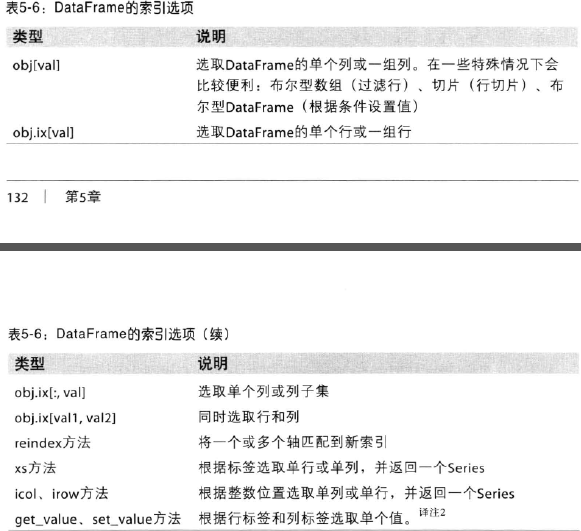
***对于DataFrame, reindex可以修改（行）索引、列，或两个都修改。***



* ***丢弃指定轴上的项***

***Drop方法返回的是一个在指定轴上删除了指定值的新对象。***

* ***索引、选取和过滤***

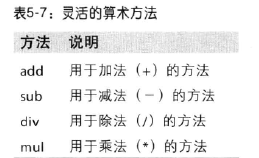


* ***算术运算和数据对齐***

***Pandas最重要的一个功能是，可以对不同索引的对象进行算术运算。在将对象相加时，如果存在不同的索引时，则结果的索引就是该索引对的并集。***

***自动的数据对齐操作在不重叠的索引出引入了NA值***

***在算术方法中充值***



* ***DataFrame 和Series之间的运算***

***默认情况下，DataFrame和Series之间的算术运算会将Series的索引匹配到DataFrame的列，然后沿着行一直向下广播。***

***如果某个索引值在dataframe的列或Series的索引中找不到，则参与运算的两个对象就会被重新索引以形成并集。***

* ***函数应用和映射***

***NumPy的ufuncs（元素级数组方法）可用于操作pandas对象。***

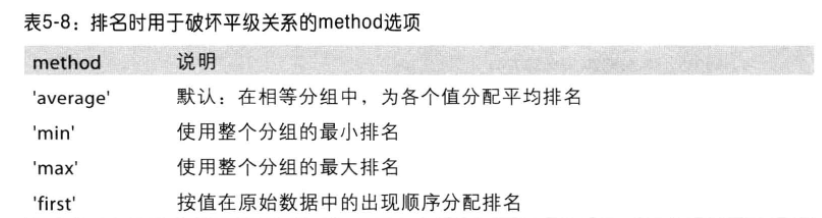
***Datafrmae的apply方法，将函数应用到由各列或行所形成的一维数组。***

* ***排序和排名***

***使用sort\_index方法对行或列索引进行排序（按字典顺序），它返回一个已排序的新对象。***

***数据默认是按升序排序。***

***若要按值对Series进行排序，可使用其order方法。***

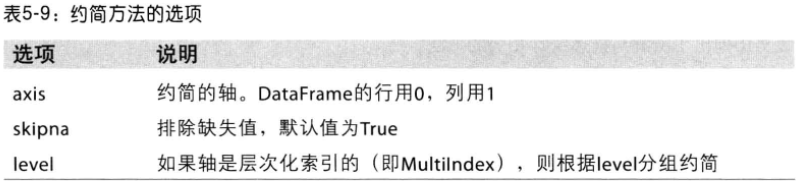


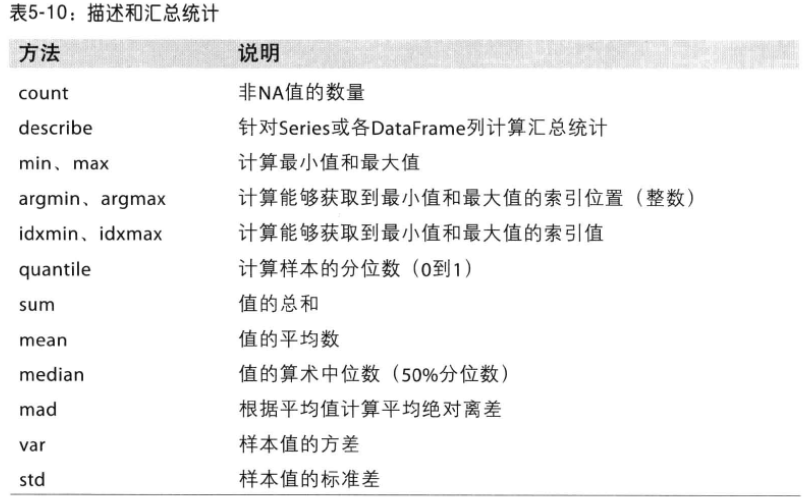
* ***带有重复值得轴索引***

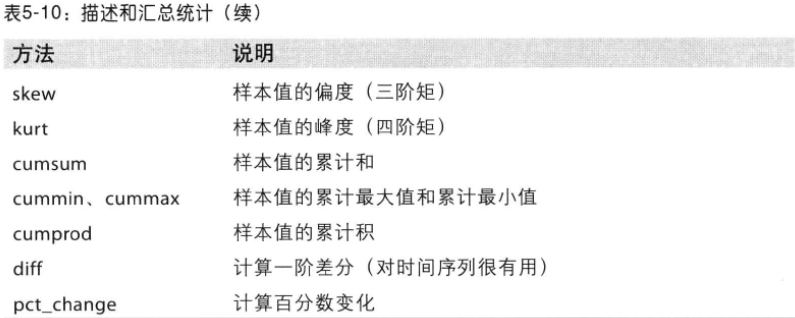
***索引的is\_unique属性判断索引值是否是唯一的。***

***对于带有重复值得索引，数据选取的行为将会有些不同。如果某个索引对应多个值，则返回一个Series，对应单个值，则返回一个标量值。***

* ***汇总和计算描述统计***







* ***相关系统与协方差***

***Series的corr方法用于计算两个Series中重叠的、非NA的、按索引对齐的值的相关系数。***

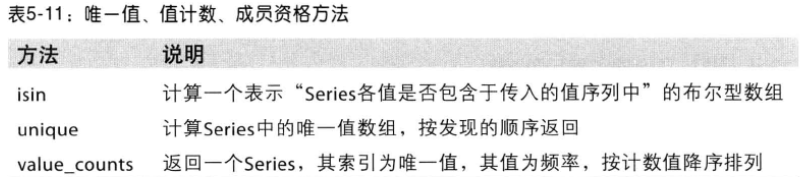
***Dataframe的corr和cov方法将以DataFrame的形式返回完整的相关系数或协方差矩阵。***

***DataFrame的corrwith方法，可以计算其列或行跟另一个Series或DataFrame之间的相关系数。***

***-传入一个Series将会返回一个相关系数值Series。***

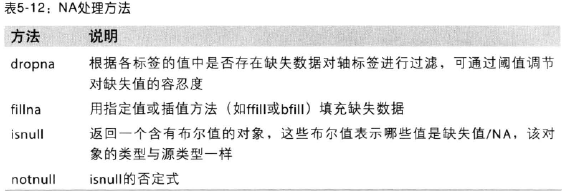
***-传入一个DataFrame则会计算按列名配对的相关系数。***

* ***唯一值、值计数以及成员资格***



* ***处理缺失数据***

***Pandas使用浮点值NaN（Not a Number）表示浮点和非浮点数组中的缺失数据。***



* ***滤除缺失数据***

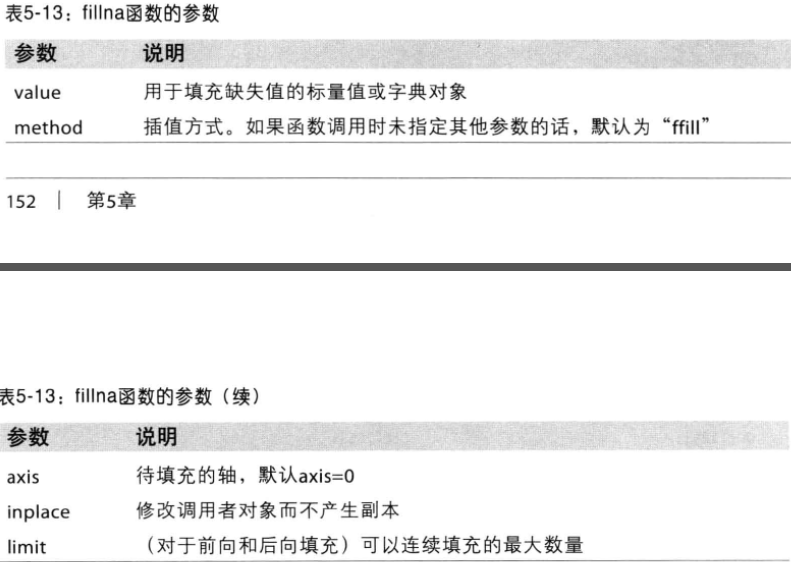
***对于一个Series，dropna返回一个仅含非空数据和索引值的Series。***

***对于Dataframe对象，默认丢弃任何含有缺失值的行。***

* ***填充缺失数据***

***通过一个字典调用fillna，实现对不同的列填充不同的值。***

***对reindex插值方法可使用fillna。***



* ***层次化索引：以低维度形式处理高纬度数据。（在一个轴上拥有多个（两个以上）索引级别。）***

***层次化索引在数据重塑和基于分组的操作（如透视表生成）中非常重要。***

* ***重排分级排序***

***Swaplevel接受两个级别编号或名称，返回一个互换了级别的新对象。***

* ***面板数据***

视其为三维版的Dataframe。

通过一个Dataframe对象组成的字典或一个三维ndarray创建Panel对象。

Panel中的每一项（类似于Dataframe的列）都是一个Dataframe。

# 数据加载、存储与文件格式



索引：将一个或多个列当做返回的Dataframe处理，以及是否从文件、用户获取列名；

类型推断和数据转换：包括用户定义值得转换、缺失值标记列表等；

日期解析：包括组合功能，比如将分散在多个列中的日期时间信息组合成结果中的单个列；

迭代：支持对大文件进行逐块迭代；

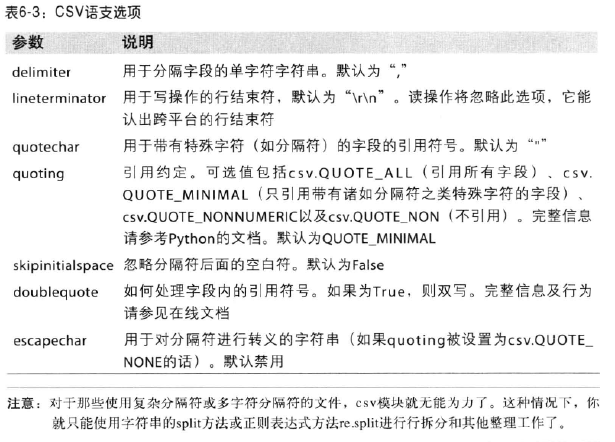
不规整数据问题：跳过一些行、页脚、注释或其他一些不重要的东西。

1. 定义”表头”信息
2. 定义“索引”信息
3. 多个列作为一个层次化索引
4. 使用正值表达式作为字段间分隔符





* 逐块读取文本文件
* 将数据写出到文本格式
* 手工处理分隔符格式



* JSON数据
* XML和HTML：web信息收集

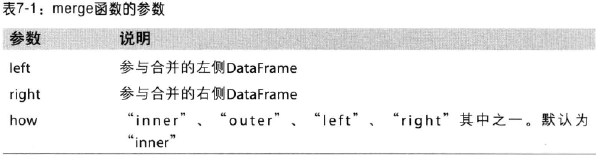
lxml.html处理Html；lxml.objectify处理XML

* 利用lxml.objectify解析XML
* 使用HDF5格式
* 读取Microsoft Excel文件
* 使用数据库
* 存取MongoDB中的数据

# 数据规整化：清理、转换、合并、重塑

* 合并数据集
* 数据库风格的Dataframe合并

数据集的合并（merge）或连接（join）运算时通过一个或多个键将行链接起来。



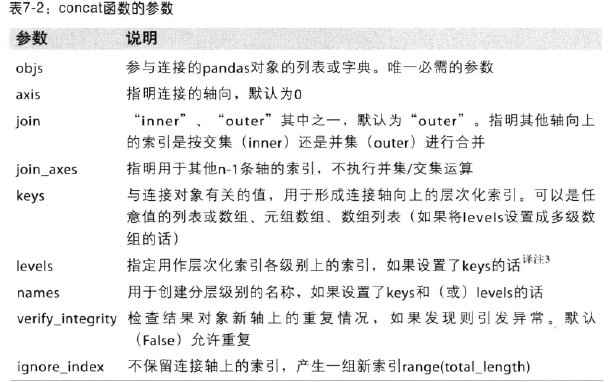


* 索引上的合并

Dataframe中的链接位与其索引中，通过传入left\_index=true或right\_index=true，说明引用该被用作链接链。

* 轴向链接

另一种数据合并运算也被称作连接（concatenation）、绑定（binding）或堆叠（stacking）



* 合并重叠数据

可能有索引全部或部分重叠的两个数据集。

* 重塑和轴向旋转-用于重新排列表格型数据。
* 重塑层次化索引

Stack：将数据的列“旋转”为行；

UnStack：将数据的行“旋转”为列；

* 将“长格式”旋转为“宽格式”

时间序列数据通常称为“长格式”或“堆叠格式”存储在数据和csv中。

DataFrame的pivot方法实现

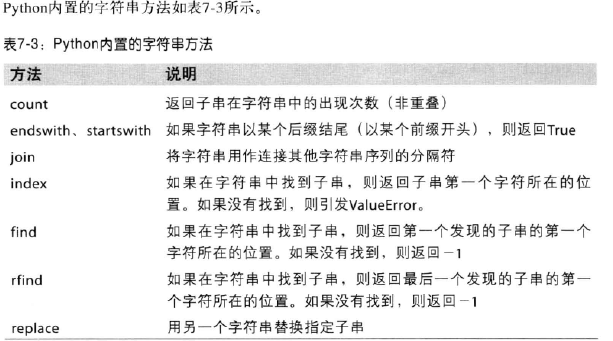
* 数据转换-过滤、清理以及其他的转换工作
* 移除重复数据：duplicated
* 利用函数或映射进行数据转换：map实现元素级转换以及其他数据清理工作。
* 替换值：fillna方法填充缺失数据；map用于修改对象的数据子集；

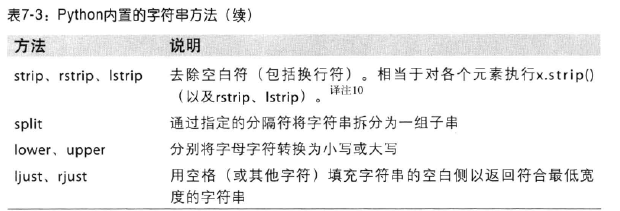
Replace实现以上两函数功能

* 重命名索引：rename
* 离散化和面元划分：为了便于分析，连续数据将被离散化或拆分为“面元”。

Pandas的cut，返回categorical对象类型。

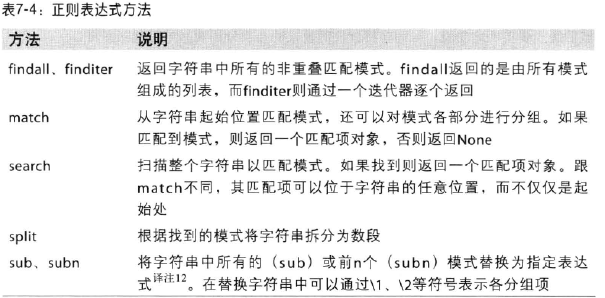
* 检测和过滤异常值
* 排列和随机采样：numpy.random.permutaion
* 计算指标、哑变量：将分类变量（categorical variable）转换为“哑变量矩阵”或“指标矩阵”
* 字符串操作



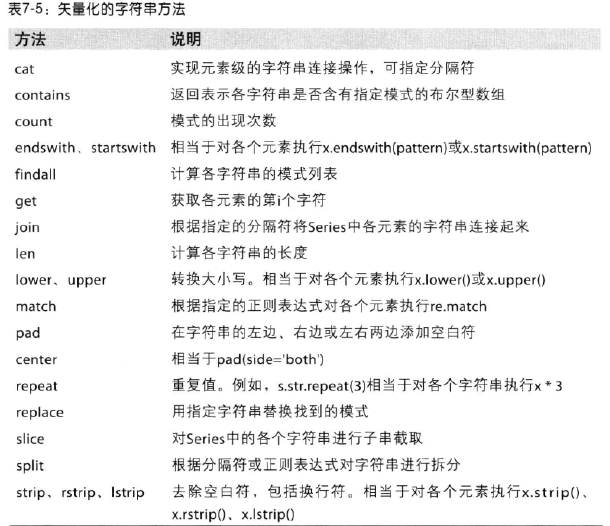


* 正则表达式

Python内置的re模块负责对字符串应用正则表达式。Re模块的函数分为：模式匹配、替换以及拆分。



* Pandas中矢量化的字符串函数



# 绘图和可视化

# 数据聚合与分组运算

目标：

1. 根据一个或多个键（可以是函数、数组或DataFrame列名）拆分对象；
2. 计算分组摘要统计，如计数、平均值、标准差、或用户自定义函数；
3. 对DataFrame的列应用各种各样的函数；
4. 应用组内转换或其他运算，如规格化、线性回归、排名或选取子集等；
5. 计算透视表或交叉表；
6. 执行分位数以及其它分组分析。

分组技术

对分组进行迭代

选取一个或一组列

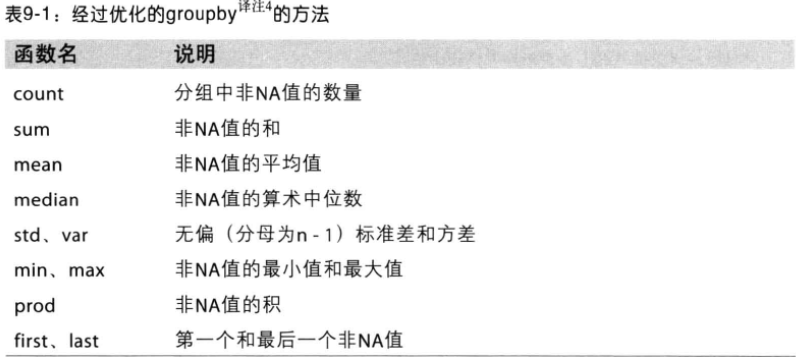
通过字典或Series进行分组

通过函数进行分组

根据索引级别分组

数据聚合

聚合：指任何能够从数组产生标量值的数据转换过程。



面向列的多函数应用

对不同的列使用不同的聚合函数，或一次应用多个函数。

以“无索引”的形式返回聚合数据

分组级运算和转换

介绍transform和apply 方法。

Apply:一般性的“拆分-应用-合并”

分位数和桶分析

透视表和交叉表

透视表（pivot talbe）根据一个或多个键对数据进行聚合，并根据行和列上的分组键将数据分配到各个矩形区域中。

交叉表：crosstab：一种用于计算分组频率的特殊透视表。

# 时间序列

时间序列数据主要的应用场景：

1. 时间戳(timestamp)，特定的时刻。
2. 固定时期（period），如2007年1月或2010年全年
3. 时间间隔（interval），由起始和结束时间戳表示。
4. 实验或过程时间，每个时间点都是相对于特定起始时间的一个量度。

* 日期和时间数据类型及工具

字符串和datetime的相互转换

* 时间序列基础

Pandas基本的时间序列类型是时间戳为索引的Series。

* 索引、选取、子集构造
  + 日期的范围、频率及移动

Pandas有一套标准时间序列频率以及用于重采样、频率推断、生成固定频率日期范围的工具。

* 生成日期范围

Pandas.date\_range

* 频率和日期偏移量

Pandas中的频率是由一个基础频率（base frequency）和一个乘数组成。基础频率通常以一个字符串别名表示，比如‘M’表示每月，‘H’表示每小时。

对于每个基础频率，都有一个被称为日期偏移量的对象与之对应。

锚点偏移量（anchored offset）

* 移动（超前和滞后）数据

移动（shifting）指的是沿着时间轴将数据前移或后移。

Shift通常用于计算一个时间序列或多个时间序列中的百分比变化。

* 通过偏移量对日期进行位移

在datetime或Timestamp对象上偏移。

* + 时区处理

本地化和转换

操作时区意识型Timestamp对象

不同时区之间的运算

* + 时期及其算术运算

时期（period）表示的是时间区间，比如数日、数月、数季、数年等。

* 时期的频率转换
* 按季度计算的时期频率
* 将Timestamp转换为Period
* 通过数组创建PeriodIndex
  + 重采用及频率转换

重采样（resampling）指将时间序列从一个频率转换到另一个频率的处理过程。

降采样（downsampling）将高频率数季聚合到低频率；

升采样（upsampling）将低频率数据转换到高频率

OHLC重采样

通过groupby进行重采样

升采样和插值

通过时期进行重采样

* + 时间序列绘图
  + 移动窗口函数

移动窗口-可以带有指数衰减权数

* 指数加权函数
* 二元移动窗口函数

有些统计运算(如相关系数和协方差)需要在两个时间序列上执行.

* 用户定义的移动窗口函数

Rolling\_apply在移动窗口上应用自己设计的数组函数。唯一要求：该函数要能从数组的各个片段中产生单个值（即约简）

* + 性能和内存使用方面的注意事项

# 金融和经济数据应用

截面（cross-section）表示某个时间点的数据。

多个数据项在多个时间点的截面数据构成一个面板（panel）。

面板数据即可以表示为层次化索引的DataFrame，也可以表示为三维的Panel pandas对象。

* 数据规整化方面的话题
  + 时间序列以及截面对齐
  + 频率不同的时间序列的运算

频率转换he重对齐主要方法：resample、reindex

Resample：用于将数据转换到固定频率；

Reindex：用于使数据符合一个新索引。

* + 使用Period
  + 时间和“最当前”数据选取
  + 拼接多个数据源

场景：

* + 1. 在一个特定的时间点shanghai，从一个数据源切换到另一个数据源；
    2. 用另一个时间序列对当前时间序列中的缺失值“打补丁”；
    3. 将数据中的符号（国家、资产代码的呢过）替换为实际数据。
  + 收益指数和累计收益
* 分组变换和分析
  + 分组因子暴露
  + 十分位和四分位分析
* 更多示例应用
  + 信号前沿分析
  + 期货合约转仓
  + 移动相关系数与线性回归

# Scipy

***Scipy是一组专门解决科学计算中各种标准问题域的包的集合：***

***Scipy.integrate:数据积分例程和微分方程求解器；***

***Scipy.linalg:扩展了由numpy.linalg提供的线性代数例程和矩阵分解功能；***

***Scipy.optimize：函数优化器（最小化器）以及根查找算法；***

***Scipy.signal：信号处理工具；***

***Scipy.sparse：稀疏矩阵和稀疏线性系统求解器；***

***Scipy.special：SPECFUM的包装器；***

***Scipy.stats：标准连接和离散概率分布、各种统计检验方法、以及更好的描述统计法；***

***Numpy与Scipy有机结合可替代Matlab的计算功能***

# 数据分析步骤

1. ***与外界进行交互：读取各种各样的文件格式和数据库***
2. ***准备：对数据进行清理、修整、整合、规范化、重塑、切片切块、变形等处理以便分析；***
3. ***转换：对苏聚集做一些数学和统计运算以产生新的数据集；***
4. ***建模和计算：将数据更统计模型、机器学习算法或其他计算工具联系起来；***
5. ***展示：创建交互式的或静态的图片或文字摘要***