# [【原】十分钟搞定pandas](http://www.cnblogs.com/chaosimple/p/4153083.html)

本文是对pandas官方网站上《10 Minutes to pandas》的一个简单的翻译，原文在[这里](http://pandas.pydata.org/pandas-docs/stable/10min.html)。这篇文章是对pandas的一个简单的介绍，详细的介绍请参考：[Cookbook](http://pandas.pydata.org/pandas-docs/stable/cookbook.html#cookbook) 。习惯上，我们会按下面格式引入所需要的包：



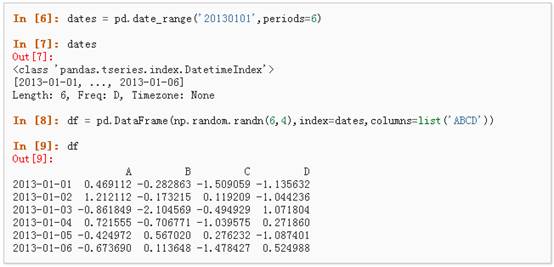
# 一、            创建对象

可以通过 [Data Structure Intro Setion](http://pandas.pydata.org/pandas-docs/stable/dsintro.html#dsintro) 来查看有关该节内容的详细信息。

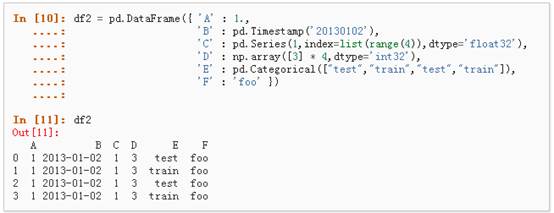
1、可以通过传递一个list对象来创建一个Series，pandas会默认创建整型索引：



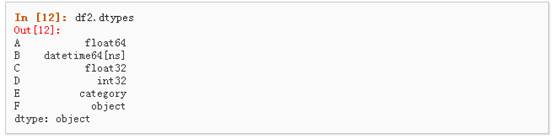
2、通过传递一个numpy array，时间索引以及列标签来创建一个DataFrame：



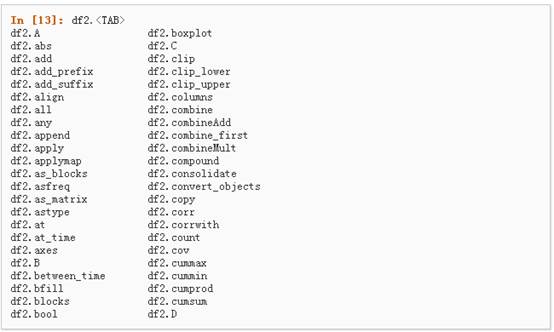
3、通过传递一个能够被转换成类似序列结构的字典对象来创建一个DataFrame：



4、查看不同列的数据类型：



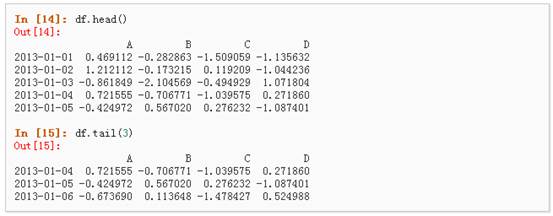
5、如果你使用的是IPython，使用Tab自动补全功能会自动识别所有的属性以及自定义的列，下图中是所有能够被自动识别的属性的一个子集：



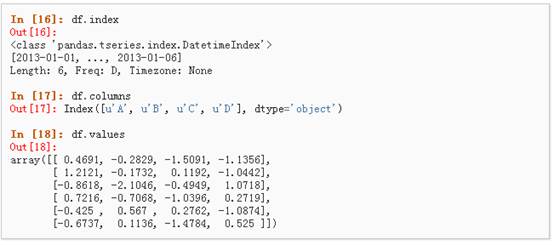
# 二、            查看数据

详情请参阅：[Basics Section](http://pandas.pydata.org/pandas-docs/stable/basics.html#basics)

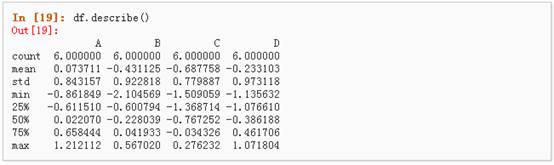
1、  查看frame中头部和尾部的行：



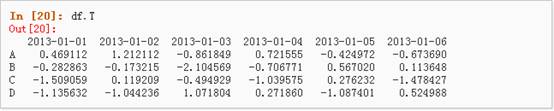
2、  显示索引、列和底层的numpy数据：



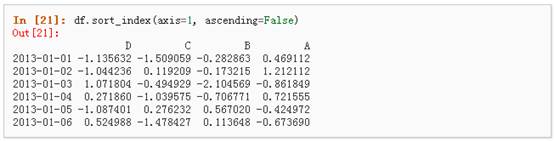
3、  describe()函数对于数据的快速统计汇总：



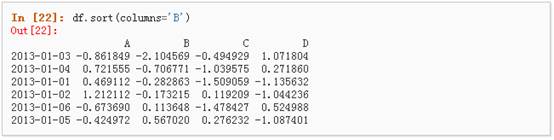
4、  对数据的转置：



5、  按轴进行排序



6、  按值进行排序

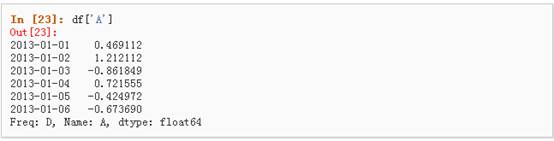


# 三、            选择

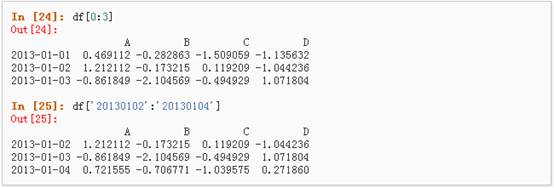
虽然标准的Python/Numpy的选择和设置表达式都能够直接派上用场，但是作为工程使用的代码，我们推荐使用经过优化的pandas数据访问方式： .at, .iat, .loc, .iloc 和 .ix详情请参阅[Indexing and Selecing Data](http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing) 和 [MultiIndex / Advanced Indexing](http://pandas.pydata.org/pandas-docs/stable/advanced.html#advanced)。

  获取

1、 选择一个单独的列，这将会返回一个Series，等同于df.A：



2、 通过[]进行选择，这将会对行进行切片

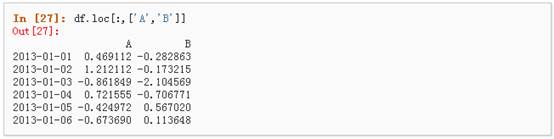


  通过标签选择

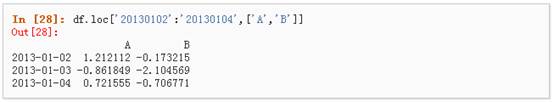
1、 使用标签来获取一个交叉的区域



2、 通过标签来在多个轴上进行选择



3、 标签切片



4、 对于返回的对象进行维度缩减



5、 获取一个标量

http://images.cnitblog.com/blog/407700/201412/091345152121400.jpg

6、 快速访问一个标量（与上一个方法等价）

http://images.cnitblog.com/blog/407700/201412/091345156038112.jpg

  通过位置选择

1、 通过传递数值进行位置选择（选择的是行）



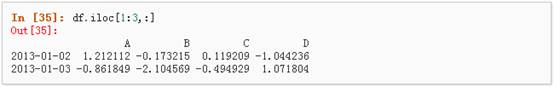
2、 通过数值进行切片，与numpy/python中的情况类似



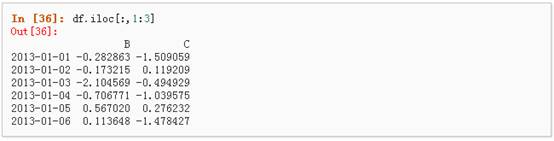
3、 通过指定一个位置的列表，与numpy/python中的情况类似



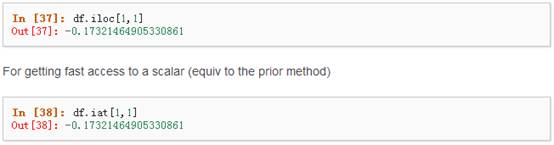
4、 对行进行切片



5、 对列进行切片

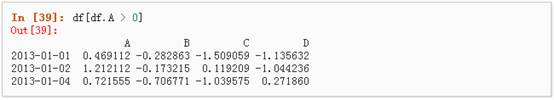


6、 获取特定的值

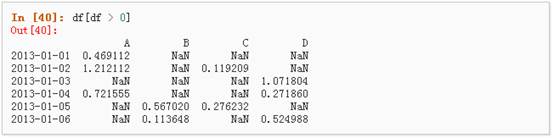


  布尔索引

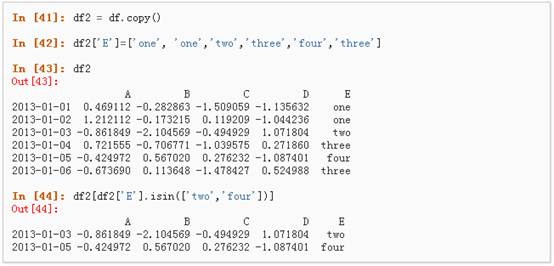
1、 使用一个单独列的值来选择数据：



2、 使用where操作来选择数据：

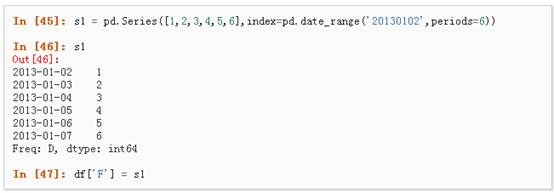


3、 使用isin()方法来过滤：



  设置

1、 设置一个新的列：



2、 通过标签设置新的值：

http://images.cnitblog.com/blog/407700/201412/091345207902196.jpg

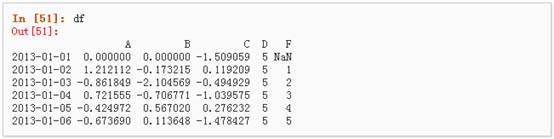
3、 通过位置设置新的值：

http://images.cnitblog.com/blog/407700/201412/091345215409566.jpg

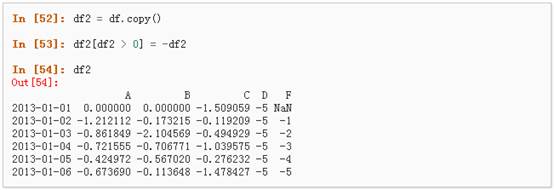
4、 通过一个numpy数组设置一组新值：

http://images.cnitblog.com/blog/407700/201412/091345224314895.jpg

上述操作结果如下：



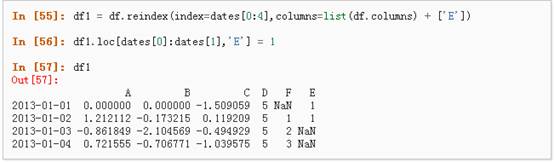
5、 通过where操作来设置新的值：



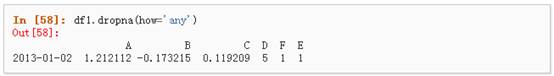
# 四、            缺失值处理

在pandas中，使用np.nan来代替缺失值，这些值将默认不会包含在计算中，详情请参阅：[Missing Data Section](http://pandas.pydata.org/pandas-docs/stable/missing_data.html#missing-data)。

1、  reindex()方法可以对指定轴上的索引进行改变/增加/删除操作，这将返回原始数据的一个拷贝：、



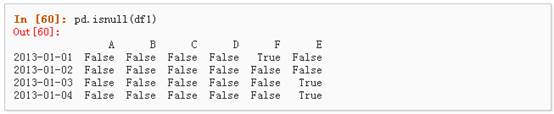
2、  去掉包含缺失值的行：



3、  对缺失值进行填充：



4、  对数据进行布尔填充：

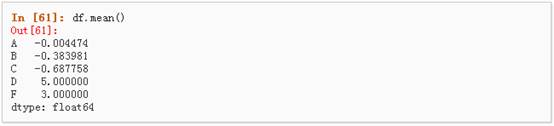


# 五、            相关操作

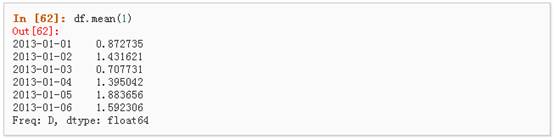
详情请参与 [Basic Section On Binary Ops](http://pandas.pydata.org/pandas-docs/stable/basics.html#basics-binop)

  统计（相关操作通常情况下不包括缺失值）

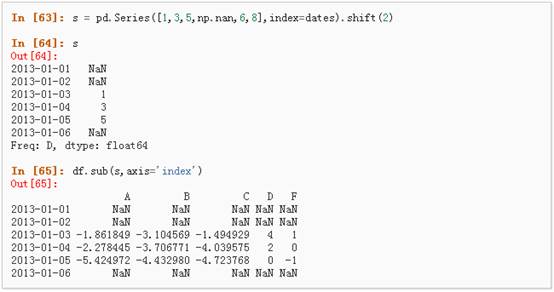
1、  执行描述性统计：



2、  在其他轴上进行相同的操作：

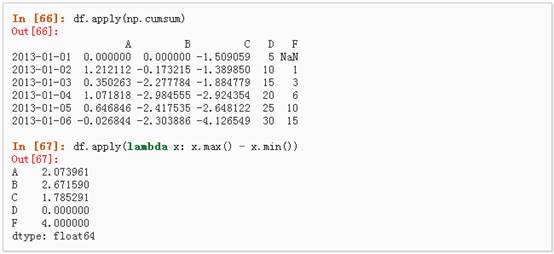


3、  对于拥有不同维度，需要对齐的对象进行操作。Pandas会自动的沿着指定的维度进行广播：



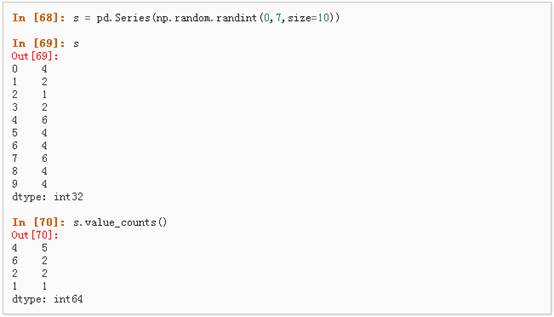
  Apply

1、  对数据应用函数：



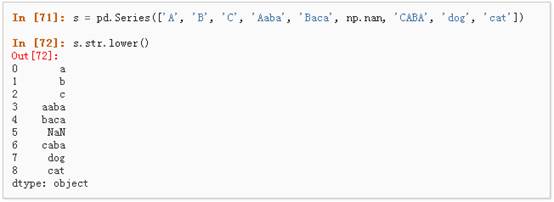
  直方图

具体请参照：[Histogramming and Discretization](http://pandas.pydata.org/pandas-docs/stable/basics.html#basics-discretization)



  字符串方法

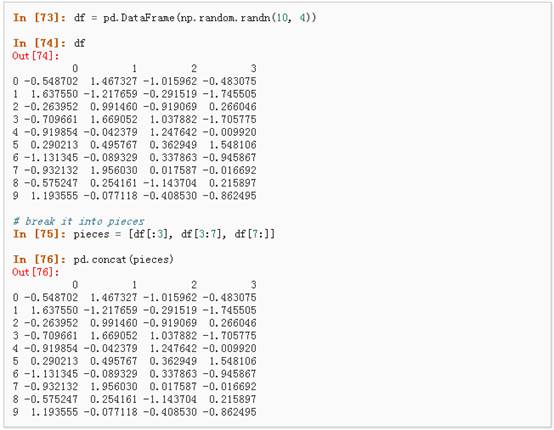
Series对象在其str属性中配备了一组字符串处理方法，可以很容易的应用到数组中的每个元素，如下段代码所示。更多详情请参考：[Vectorized String Methods](http://pandas.pydata.org/pandas-docs/stable/text.html#text-string-methods).



# 六、            合并

Pandas提供了大量的方法能够轻松的对Series，DataFrame和Panel对象进行各种符合各种逻辑关系的合并操作。具体请参阅：[Merging section](http://pandas.pydata.org/pandas-docs/stable/merging.html#merging)

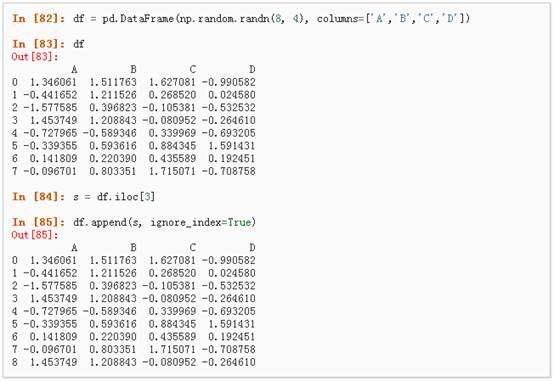
  Concat



  Join 类似于SQL类型的合并，具体请参阅：[Database style joining](http://pandas.pydata.org/pandas-docs/stable/merging.html#merging-join)



  Append 将一行连接到一个DataFrame上，具体请参阅[Appending](http://pandas.pydata.org/pandas-docs/stable/merging.html#merging-concatenation)：



# 七、            分组

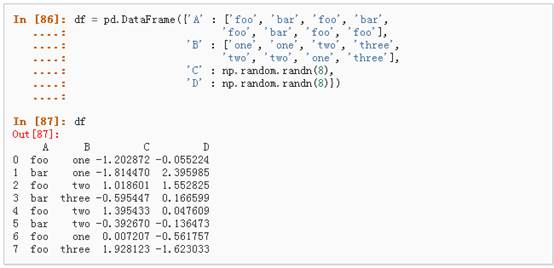
对于”group by”操作，我们通常是指以下一个或多个操作步骤：

  （Splitting）按照一些规则将数据分为不同的组；

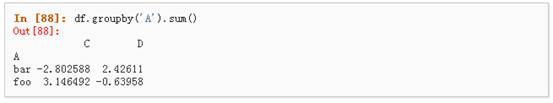
  （Applying）对于每组数据分别执行一个函数；

  （Combining）将结果组合到一个数据结构中；

详情请参阅：[Grouping section](http://pandas.pydata.org/pandas-docs/stable/groupby.html#groupby)



1、  分组并对每个分组执行sum函数：



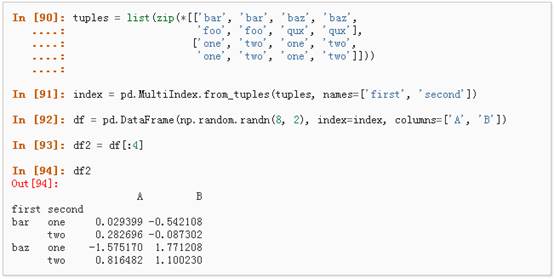
2、  通过多个列进行分组形成一个层次索引，然后执行函数：

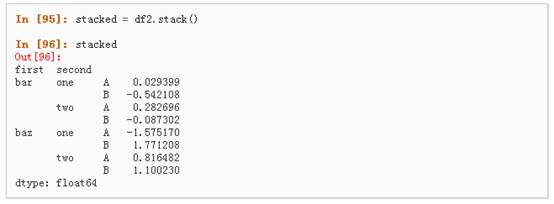


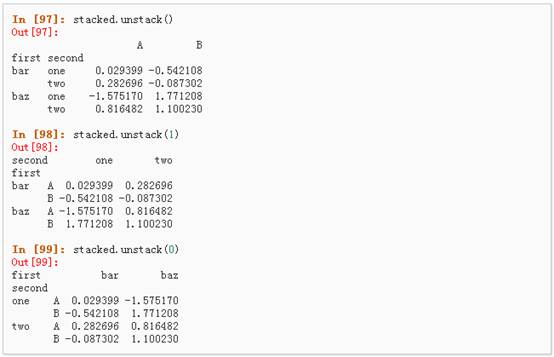
# 八、            Reshaping

详情请参阅 [Hierarchical Indexing](http://pandas.pydata.org/pandas-docs/stable/advanced.html#advanced-hierarchical) 和 [Reshaping](http://pandas.pydata.org/pandas-docs/stable/reshaping.html#reshaping-stacking)。

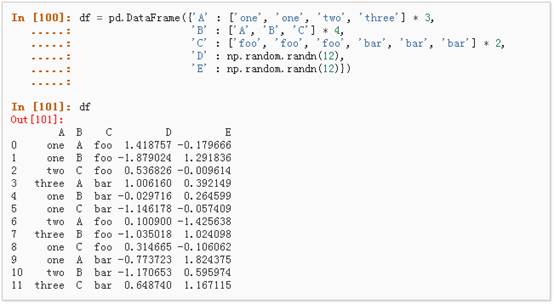
  Stack



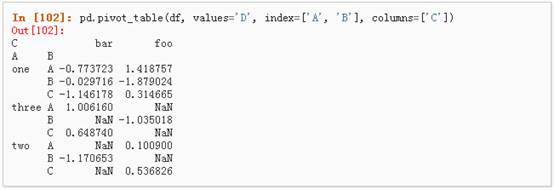




  数据透视表，详情请参阅：[Pivot Tables](http://pandas.pydata.org/pandas-docs/stable/reshaping.html#reshaping-pivot).

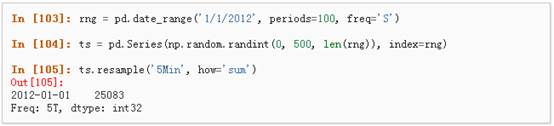


可以从这个数据中轻松的生成数据透视表：

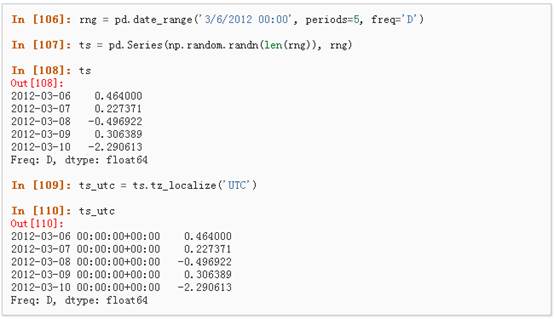


# 九、            时间序列

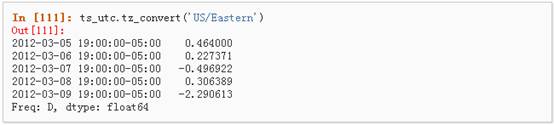
Pandas在对频率转换进行重新采样时拥有简单、强大且高效的功能（如将按秒采样的数据转换为按5分钟为单位进行采样的数据）。这种操作在金融领域非常常见。具体参考：[Time Series section](http://pandas.pydata.org/pandas-docs/stable/timeseries.html#timeseries)。



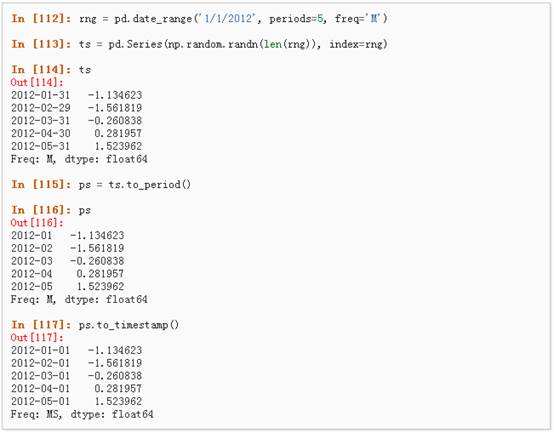
1、  时区表示：



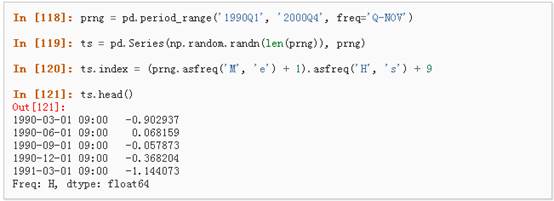
2、  时区转换：



3、  时间跨度转换：



4、  时期和时间戳之间的转换使得可以使用一些方便的算术函数。



# 十、            Categorical

从0.15版本开始，pandas可以在DataFrame中支持Categorical类型的数据，详细 介绍参看：[categorical introduction](http://pandas.pydata.org/pandas-docs/stable/categorical.html#categorical)和[API documentation](http://pandas.pydata.org/pandas-docs/stable/api.html#api-categorical)。

http://images.cnitblog.com/blog/407700/201412/091345345093572.jpg

1、  将原始的grade转换为Categorical数据类型：



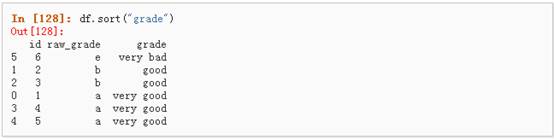
2、  将Categorical类型数据重命名为更有意义的名称：

http://images.cnitblog.com/blog/407700/201412/091345356033669.jpg

3、  对类别进行重新排序，增加缺失的类别：



4、  排序是按照Categorical的顺序进行的而不是按照字典顺序进行：

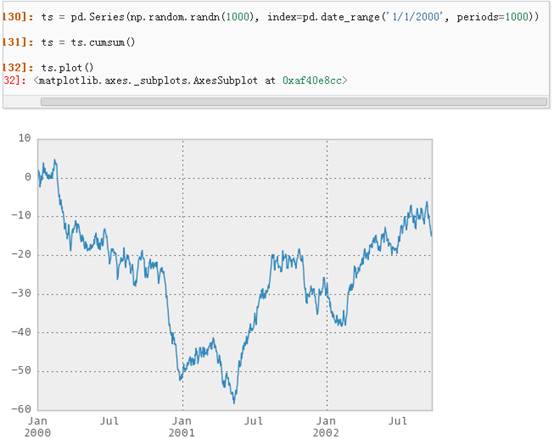


5、  对Categorical列进行排序时存在空的类别：

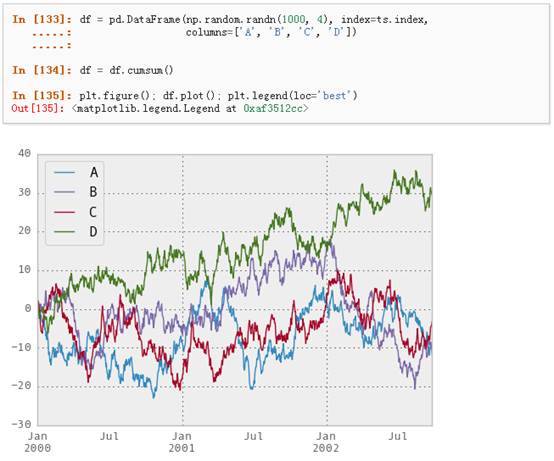


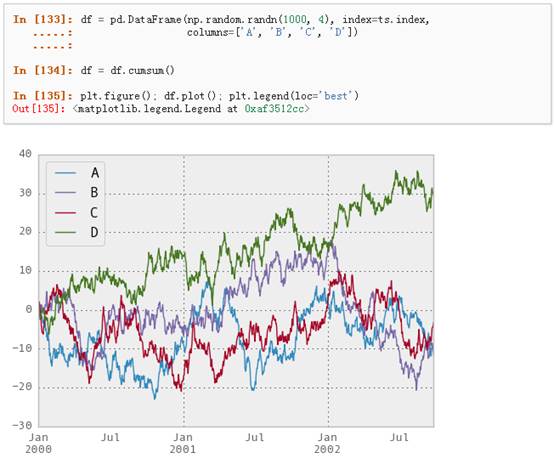
# 十一、           画图

具体文档参看：[Plotting](http://pandas.pydata.org/pandas-docs/stable/visualization.html#visualization) docs



对于DataFrame来说，plot是一种将所有列及其标签进行绘制的简便方法：





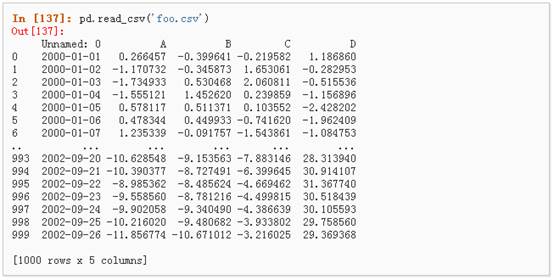
# 十二、           导入和保存数据

  CSV，参考：[Writing to a csv file](http://pandas.pydata.org/pandas-docs/stable/io.html#io-store-in-csv)

1、  写入csv文件：

http://images.cnitblog.com/blog/407700/201412/091345399933395.jpg

2、  从csv文件中读取：

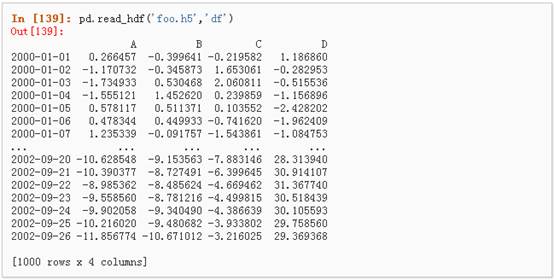


  HDF5，参考：[HDFStores](http://pandas.pydata.org/pandas-docs/stable/io.html#io-hdf5)

1、  写入HDF5存储：

http://images.cnitblog.com/blog/407700/201412/091345410091453.jpg

2、  从HDF5存储中读取：

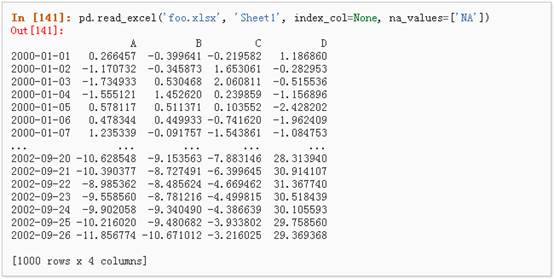


  Excel，参考：[MS Excel](http://pandas.pydata.org/pandas-docs/stable/io.html#io-excel)

1、  写入excel文件：

http://images.cnitblog.com/blog/407700/201412/091345417757350.jpg

2、  从excel文件中读取：



[来自为知笔记(Wiz)](http://www.wiz.cn/i/4158d356)