**20180524**

**第1章 准备工作**

Python有不断改良的库(主要是pandas)，使其成为数据处理任务的一大替代方案。

作为科学计算平台，Python的成功源于能够轻松集成C、C++及Fortran代码。大部分现代计算环境都利用一些Fortran和C库实现线性代数、优选、积分、快速傅里叶变换等算法。许多企业和国家实验室也利用Python粘合已经用了30多年的遗留软件系统。

大多数软件都是由少量需要占大部分执行时间的代码，和大量不经常执行的粘合剂代码组成。开发人员的精力几乎都花在优化计算瓶颈上，或直接转用低级语言(如C语言)。

最近几年，[Cython](http://cython.org/)项目已经成为Python领域创建编译型扩展和对接C/C++代码的一大途径。

因为Python有GIL(Global Interpreter Lock, 全局解释器锁，防止解释器同时执行多条Python字节码指令)的存在，对于高并发、多线程的程序(尤其是计算密集型)，Python并不十分理想。

* 重要的科学计算Python库

① NumPy

NumPy (Numerical Python)是Python科学计算的基础包，主要功能：

|  |
| --- |
| 1) 快速高效的多维数组对象ndarray  2) 对数组执行元素级计算和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、傅里叶变换、生成随机数  5) 将C、C++、Fortran代码集成到Python的工具 |

NumPy除了为Python提供快速数组处理能力，还能作为在算法之间传递数据的容器。对于数值型数据，NumPy数组在存储和处理数据时比内置的Python数据结构高效得多。此外，低级语言编写的库可以直接操作NumPy数组中的数据。

② pandas