

1.2 为何利用 Python 进行数据分析

对很多人来说，Python 编程语言拥有强大的吸引力。从 1991 年面世以来，Python 与 Perl、Ruby 等语言成为最流行的解释型语言。Python 和 Ruby 从 2005 年之后格外流行，可以基于众多 web 框架，比如 Rails (Ruby) 和 Django (Python) 进行网站搭建。此类语言通常称为脚本语言，因为它们可以用于快速编写小型程序、脚本或对其他任务进行自动化。我并不喜欢“脚本语言”这个术语，因为它的言外之意似乎是“脚本语言”无法构建大型软件。在解释型语言中，由于历史和文化上的原因，Python 发展出了一个大型、活跃的科学计算及数据分析社区。在过去十年里，Python 已经从一个最前沿或者说“后果自负”的科学计算语言，成为数据科学、机器学习和学术 / 工业界通用软件开发等领域最为重要的语言之一。

在数据科学、交互式计算以及数据可视化等领域，Python 经常被拿来和其他开源或商业编程语言、工具进行对比，比如 R、MATLAB、SAS、Stata 等。近些年，Python 提高了对类库的支持（比如 pandas 和 scikit-learn），使得它成为数据分析任务的一个流行选择。再综合考虑 Python 在通用软件工程上的总体实力，它便成为搭建数据应用的首选语言。

1.2.1 Python 作为胶水

Python 在科学计算方面的成功部分是因为它很容易整合 C、C++ 和 FORTRAN 等语言的代码。大部分现代计算环境都拥有相似的存量程序集，这些程序集使用 FORTRAN 和 C 的库进行线性代数、调优、积分、快速傅里叶变换等算法运算。很多公司和国家实验室都使用 Python 将过去数十年产生的存量软件黏合在一起。

很多程序中包含了一小部分运行起来要花费大量时间的代码，而大量“胶水代码”却很少运行。很多情况下，胶水代码的执行时间可以忽略不计；精力应当更多地投入优化计算瓶颈的方面，有些时候需要将代码迁移到底层语言，比如 C。

1.2.2 解决“双语言”难题

在许多组织中，通常会使用一种更特殊的计算语言，比如用 SAS 或 R 针对想法进行研究、原型实现和测试，之后再将这些想法迁移到用 Java、C# 或 C++ 编写的大型生产环境上。逐渐地，人们发现 Python 不但更适用于研究和原型实现，也适合搭建生产系统。当一种语言可以满足需求时，为什么要维持两个开发环境呢？当使用相同程序工具集来兼顾研究人员和软件工程师的好处越发明显时，我认为会有越来越多的公司选择走使用一种语言的路线。