# 1 模块 1

# Python 数据分析概述

当今世界对信息技术的依赖程度日渐加深，每天都会产生和存储海量的数据，如上传到手机中的图像和视频、用于高清电视的数字电影、ATM 中的银行数据、机场和重要活动的安全录像、滴滴专车的路线记录、通过移动网络传输的微信语音通话，以及用于日常沟通的短信等。如何管理和使用这些数据，逐渐成为数据科学领域中一个全新的研究课题。

本模块介绍数据分析的概念和主要流程以及数据分析工具的选择。后面会陆续介绍如何借助于 Python 库将学到的概念和流程转化为 Python 代码。

## 1.1 数据分析简介

### 1.1.1 什么是数据分析

数据不等同于信息。对于没有任何形式可言的字节流，除了数量、用词和发送的时间外，其他一无所知，很难理解其本质。数据分析的目的是把隐没在一大批看起来杂乱无章的数据中的信息集中和提炼出来，以找出所研究对象的内在规律。在实际应用中，数据分析可帮助人们做出判断，以便进行决策。

广义数据分析包括狭义数据分析和数据挖掘。狭义数据分析是指根据分析目的，采用对比分析、分组分析、交叉分析和回归分析等分析方法，对收集到的数据进行处理与分析，提取有价值的信息，发挥数据的作用，得到一个特征统计量结果的过程。数据挖掘则是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，通过应用聚类模型、分类模型、智能推荐和关联规则等技术，挖掘潜在价值的过程。

广义数据分析是指根据一定目标，通过统计分析、聚类、分类等方法发现大量数据中的目标隐含信息的过程，如图 1-1 所示。

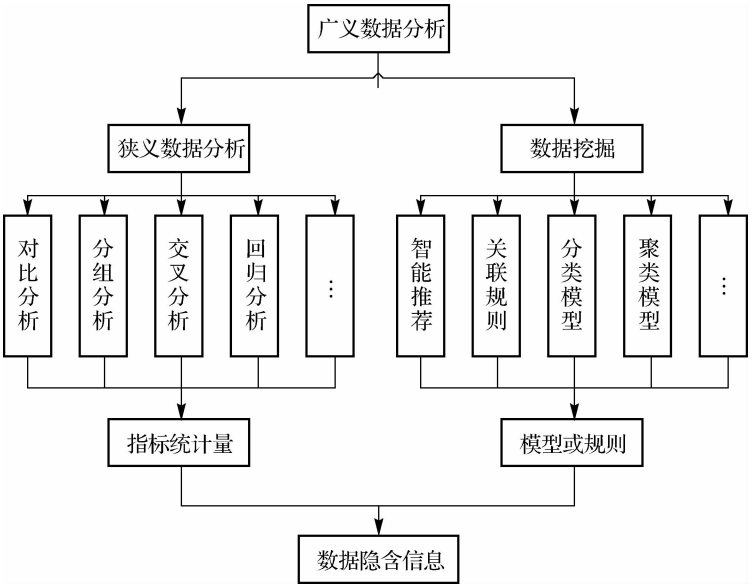


图 1-1　广义数据分析

数据分析最初用作数据保护，现在已发展成数据建模的方法论，从而完成了一门真正学科的蜕变。模型实际上是指将所研究的系统转化为数学形式。一旦建立数学或逻辑模型，对系统的响应能做出不同精度的预测，我们就可以预测在给定输入的情况下，系统会给出怎样的输出。

### 1.1.2　数据分析的范畴

数据分析学科研究的问题面很广。数据分析过程要用到很多工具和方法，它们对计算机、数学和统计学知识要求较高，如图 1-2 所示。

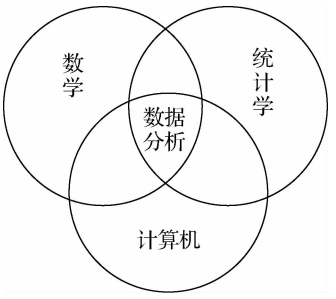


图 1-2　数据分析的范畴

数据分析涉及大量数学知识，因此具备扎实的数学功底很重要，至少要能理解正在做的事。熟悉常用的统计学概念也很有必要，因为所有对数据进行的分析和解释都以这些概念为基础。无论从事什么领域的数据分析工作，具备良好的计算机知识都是必要的。数据分析的各个步骤都离不开计算机技术，如用于计算的软件（FORTRAN、MATLAB 等）和编程

语言(Java、Python 等)。要高效地处理随信息技术迅猛发展而产生的海量数据,就必须用到特定的技能。数据研究和抽取,要求分析人员掌握各种常见格式数据的处理技巧。数据通常以某种结构形式组织在一起,存储于文件或数据库表中,格式多样。常见的数据存储格式有 XML、JSON、XLS、CSV 等。很多应用都能处理这些格式的数据文件。

除了以上这些知识,数据分析人员还应掌握相关应用领域的知识,这些知识可以帮助数据分析人员更好地理解研究对象及需要什么样的数据提供帮助。

## 1.2 数据分析的流程

通常很多问题看上去相当复杂难解,但是一个良好的流程能够帮助数据分析人员将复杂的问题分解成更容易处理的小步骤,这有助于实现全面且可重复实施的分析方法,使数据分析人员把精力放在可以掌握问题重点的步骤中。

数据分析可以用以下几步来描述:需求分析、数据抽取、数据预处理、数据分析与建模、模型评估、最终部署。

### 1.2.1 需求分析

需求分析一词来源于产品设计,主要是指从用户提出的需求出发,挖掘用户内心的真实意图,并转化为产品需求的过程。数据分析中的需求分析是数据分析环节的第一步,也是非常重要的一步,决定了后续的分析方向和方法。需求分析决定着数据分析的整体分析方向和分析内容。

### 1.2.2 数据抽取

数据抽取是数据分析工作的基础,是指根据需求分析的结果提取、收集数据。数据抽取一定要本着创建预测模型的目的,数据抽取对数据分析的成功起着至关重要的作用。所采集的样本数据必须尽可能地反映实际情况,即能够描述系统对来自现实刺激的反应。如果原始数据采集不当,即使数据量很大,这些数据描述的情境往往也与现实相左或存在偏差。

很多领域的应用需要从周边环境搜寻数据,往往依赖于外部的实验数据,甚至常通过采访或调查来收集数据。在这种情况下,寻找包含数据分析所需全部信息的数据源难度很大。这时往往需要从多种数据源收集信息,使数据尽可能地具有普遍性。

网络数据是指存储在互联网中的各类视频、图片、语音和文字等信息。但网络中的大多数数据获取起来具有一定的难度。不是所有的数据都是以文件或数据库的形式存在的,有些数据以这样或那样的格式存在于 HTML 页面中;有的内容明确,有的则不然。为了获取网页中的内容,人们研究出了 Web 抓取(Web scraping)方法,通过识别网页中特定的 HTML 标签采集数据。有些软件就是专门用来抓取网页的。它们找到符合条件的标签,从中抽取目标数据。查找、抽取完成后,就得到了用于数据分析的数据。

### 1.2.3 数据预处理

数据往往来自不同的数据源，有着不同的表现形式和格式。因此，在分析数据之前，所有这些不同的数据都要被处理成可用的形式。

数据预处理是指对获取的数据进行清洗和标准化处理，以及把数据变换为优化过的形式，以便处理这些数据的过程。数据中存在的很多问题都必须解决，如存在无效的、模棱两可的数据，值缺失，字段重复以及有些数据超出范围等。数据清洗可以去掉重复、缺失、异常、不一致的数据；数据标准化可以去除特征间的量纲差异；数据变换则可以通过离散化、哑变量处理等技术满足后期数据分析与建模需求。

### 1.2.4 数据分析与建模

数据分析与建模是指通过对比分析、分组分析、交叉分析、回归分析等分析方法，以及聚类模型、分类模型、关联规则、智能推荐等模型与算法，发现数据中的有价值信息，并得出结论的过程。

数据分析与建模主要分为以下两个方面的用途：一是预测系统所产生的数据的值，使用回归模型；二是为新数据分类，使用分类模型或聚类模型。事实上，根据输出结果的类型，模型可以分为以下三种。

(1)分类模型：模型输出结果为类别型。

(2)回归模型：模型输出结果为数值型。

(3)聚类模型：模型输出结果为描述性。

生成这些模型的简单方法包括线性回归、逻辑回归、分类、回归树和 K-近邻算法。分析方法有多种，且每一种都有自己的特点，擅长处理和分析特定类型的数据。每一种方法都能生成一种特定的模型，选取哪种方法与模型的自身特点有关。

### 1.2.5 模型评估

模型评估是指对于已经建立的一个或多个模型，根据其模型的类别，使用不同的指标评价其性能优劣的过程。常用的聚类模型评价方法有 ARI 评价法、AMI 评价法、V-measure 评分、FMI 评价法和轮廓系数等。常用的分类模型评价指标有准确率、精确率、召回率、F1 值、ROC 和 AUC 等。常见的回归模型评价指标有平均绝对误差、均方误差、中值绝对误差等。

在模型评估阶段，我们会验证用先前采集的数据创建的模型是否有效。该阶段之所以重要，是因为直接与真实系统数据比较，可评估模型所生成的数据的有效性。但其实该阶段我们是从整个数据分析过程所使用的初始数据集中取一部分用于验证。一般来说，用于建模的数据称为训练集，用于验证模型的数据称为验证集。

通过比较模型和实际系统的输出结果，可以评估错误率。使用不同的测试集，可以得出模型的有效性区间，预测结果只在一定范围内有效，或因预测值取值范围而异，预测值和有

效性之间存在不同层级的对应关系。通过模型评估过程，不仅可以得到模型的确切有效程度(其形式为数值)，还可以比较它与其他模型有什么不同。

### 1.2.6 最终部署

数据分析的最后一步是部署，旨在展示结果，就是给出数据分析的结论。若应用场景为商业，部署过程将分析结果转换为对购买数据分析服务的客户有益的方案。若应用场景为科技领域，则将成果转换为设计方案或科技出版物。也就是说，部署过程基本上就是把数据分析得到的结果应用到实践中去。

数据分析或挖掘的结果有多种部署方式。通常，数据分析师会在这个阶段为管理层或客户撰写报告，从概念上描述数据分析结果，以便他们做出相应决策，真正用分析结果指导实践。

## 1.3 Python 和数据分析

### 1.3.1 为什么选用 Python

目前主流的数据分析语言有 Python、R、MATLAB 这 3 种。其中，Python 具有丰富和强大的库。它常被称为胶水语言，能够把用其他语言制作的各种模块很轻松地连在一起，是一门简单易学的程序设计语言。R 语言是用于统计分析、图形表示和报告的编程语言与软件环境。R 语言在 GNU 通用公共许可证下免费提供，并为各种操作系统(如 Linux、Windows 和 Mac)提供预编译的二进制版本。MATLAB 主要用于矩阵运算、绘制函数与数据、实现算法、创建用户界面和连接其他编程语言的程序等，侧重于工程计算、信号处理、金融建模设计与分析等领域。

比起 R 和 MATLAB，Python 不仅提供数据处理平台，而且有其他语言和专业应用所没有的特点。Python 库一直在增加，算法的实现采用更具创新性的方法。Python 能做脚本语言，还能操作数据库，随着 Django 等框架的问世，还能开发 Web 应用。这些特点使得 Python 在所有可用于数据分析的语言中与众不同。

### 1.3.2 Python 数据分析常用类库

#### 1. IPython

IPython 是 Python 科学计算标准工具集的组成部分，它将其他所有的工具联系到了一起，为交互式和探索式计算提供了一个强健而高效的环境。同时，它是一个增强的 Python Shell，目的是提高编写、测试、调试 Python 代码的速度。IPython 主要用于交互式数据处理，是用于交互式并行和分布式计算的基础架构。

另外，IPython 还提供了一个类似于 Mathematica 的 HTML 笔记本、一个基于 Qt 框架的 GUI 控制台，具有绘图、多行编辑以及语法高亮显示等功能。

### 2. NumPy

NumPy 是 Numerical Python 的简称，是一个 Python 科学计算的基础包。NumPy 提供了以下内容。

(1)快速高效的多维数组对象 ndarray。

(2)用于对数组执行元素级计算以及直接对数组执行数学运算的函数。

(3)用于读/写硬盘上基于数组的数据集的工具。

(4)线性代数运算、傅里叶变换以及随机数生成。

(5)用于将 C、C++、FORTRAN 代码集成到 Python 的工具中。

对于数值型数据，NumPy 数组在存储和处理数据时要比内置的 Python 数据结构高效得多。此外，由低级语言(如 C 和 FORTRAN)编写的库可以直接操作 NumPy 数组中的数据，无须进行任何数据复制工作。

### 3. SciPy

SciPy 是世界上著名的 Python 开源科学计算库，建立在 Numpy 之上。它增加的功能包括数值积分、最优化、统计和一些专用函数。SciPy 函数库在 NumPy 库的基础上增加了众多的数学、科学以及工程计算中常用的库函数，如线性代数、常微分方程数值求解、信号处理、图像处理、稀疏矩阵等。SciPy 是基于 Numpy 构建的一个集成了多种数学算法和函数的 Python 模块。通过给用户提供一些高层的命令和类，SciPy 在 Python 交互式会话中，大大增加了操作和可视化数据的能力。SciPy 主要包括以下内容。

(1)scipy. integrate：数值积分列程和微分方程求解器。

(2)scipy. linalg：扩展了由 numpy. linalg 提供的线性代数例程和矩阵分解功能。

(3)scipy. optimize：函数优化器(最小化器)及根查找算法。

(4)scipy. signal：信号处理工具。

(5)scipy. sparse：稀疏矩阵和稀疏线性系统求解器。

### 4. pandas

pandas 是 Python 数据分析的核心库，最初作为金融数据分析工具而被开发出来。pandas 为时间序列分析提供了很好的支持。它提供了一系列能够快速、便捷地处理结构化数据的数据结构和函数。

pandas 兼具 NumPy 高性能的数组计算功能及电子表格和关系型数据库灵活的数据处理功能。它提供了复杂精细的索引功能，以便更为便捷地完成重塑、切片和切块、聚合以及选取数据子集等操作。

### 5. Matplotlib

Matplotlib 是最流行的用于绘制数据图表的 Python 库，是 Python 的 2D 绘图库。它非常适合创建出版物中的图表。Matplotlib 最初由 John D. Hunter(JDH)创建，目前由一个庞大的开发团队维护。Matplotlib 的操作比较简单，用户只需几行代码即可生成直方图、功率谱图、条形图、错误图和散点图等图形。Matplotlib 提供了 pylab 模块，其中包括了 NumPy

和 pyplot 中许多常用的函数，方便用户快速进行计算和绘图。Matplotlib 与 IPython 结合得很好，提供了一种非常友好的交互式数据绘图环境。绘制的图表也是交互式的，用户可以利用绘图窗口中工具栏中的相应工具放大图表中的某个区域，或对整个图表进行平移浏览。

#### 6. Seaborn

Seaborn 是 Python 基于 Matplotlib 的数据可视化工具。它提供了很多高层封装的函数，帮助数据分析人员快速绘制美观的数据图形，而避免了许多额外的参数配置问题。Seaborn 可以轻松绘制常见的图形，包括散点图、柱状图、饼图、直方图、盒图、概率密度图、小提琴图和点对图等。

#### 7. Scikit-learn

Scikit-learn 是一个简单有效的数据挖掘和数据分析工具，可以供用户在各种环境下重复使用。而且 Scikit-learn 建立在 NumPy、SciPy 和 Matplotlib 基础之上，对一些常用的算法进行了封装。目前，Scikit-learn 的基本模块主要有数据预处理、模型选择、分类、聚类、数据降维和回归 6 个。在数据量不大的情况下，Scikit-learn 可以解决大部分问题。对算法不精通的用户在执行建模任务时，并不需要自行编写所有的算法，只需简单地调用 Scikit-learn 库中的模块即可。

#### 8. Spyder

Spyder 是一个强大的交互式 Python 语言开发环境，提供高级的代码编辑、交互测试和调试等特性，支持 Windows、Linux 和 Mac 系统。Spyder 包含数值计算环境，得益于 IPython、NumPy、SciPy 和 Matplotlib 的支持。Spyder 可用于将调试控制台直接集成到图形用户界面的布局中。Spyder 的最大优点就是模仿 MATLAB 的“工作空间”，可以很方便地观察和修改数组的值。Spyder 的界面由许多窗格构成，用户可以根据自己的喜好调整它们的位置和大小。当多个窗格出现在同一个区域时，将使用标签页的形式显示。

## 1.4 案例：Python 数据分析集成开发环境的部署

Python 拥有 NumPy、SciPy、Pandas、Matplotlib 和 Scikit-learn 等功能齐全、接口统一的库，能为数据分析工作提供极大的便利。库的管理及版本问题使得数据分析人员并不能够专注于数据分析，而是将大量时间花费在与环境配置相关的问题上。基于上述原因，Python 的 Anaconda 发行版应运而生。

#### 1. 了解 Python 的 Anaconda 发行版

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux、Mac、Windows 系统，提供了包管理与环境管理的功能，可以很方便地解决多版本 Python 并存、切换以及各种第三方包安装问题。Anaconda 利用工具/命令 conda 来进行包和环境的管理，并且已经预装了 180 多个与 Python 相关的包，使得数据分析人员能够更加顺畅、专注地使用 Python 来解决数据分析的相关问题。

因此，推荐数据分析初学者安装此 Python 发行版。读者只需到 Anaconda 官方网站(http://continuum.io/downloads)下载合适的安装包即可。另外，读者也可以选择国内镜像网站(https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/)下载相应的安装包。

### 2. 在 Windows 系统中安装 Anaconda

进入 Anaconda 官方网站，下载 Windows 系统中的 Anaconda 安装包，选择最新的版本。安装 Anaconda 的具体步骤如下。

(1)单击如图 1-3 所示的“Next”按钮进入下一步。

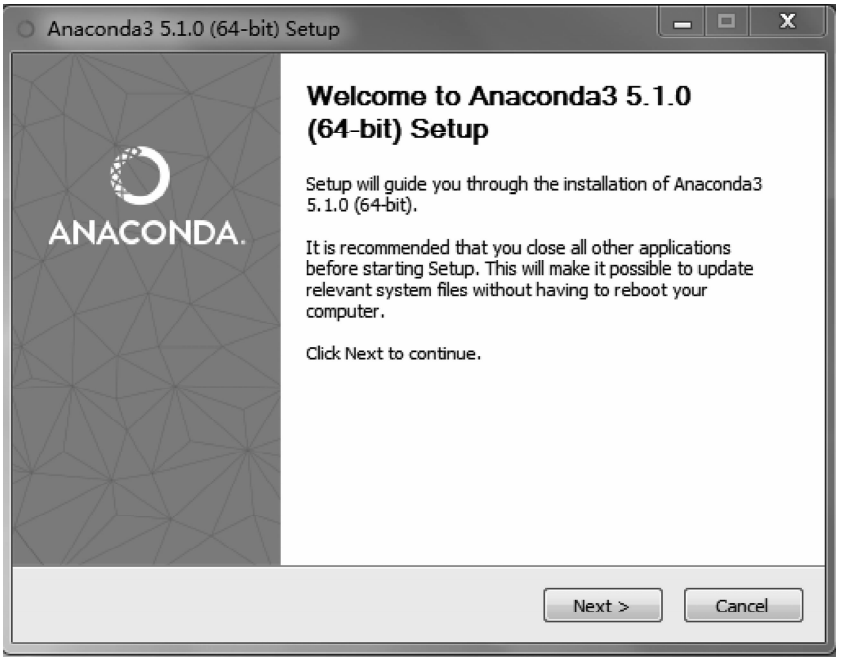


图 1-3　Windows 系统安装 Anaconda 步骤 1

(2)单击图 1-4 所示的“I Agree”按钮，同意协议并进入下一步。

(3)选中图 1-5 中的“All Users(requires admin privileges)”单选按钮，单击“Next”按钮进入下一步。

(4)在图 1-6 中单击“Browse”按钮，选择在指定的路径安装 Anaconda，选择完成后单击“Next”按钮进入下一步。

(5)图 1-7 中的两个复选框分别代表了允许将 Anaconda 添加到系统路径环境变量中、Anaconda 使用的 Python 版本为 3.6。选中后，单击“Install”按钮，等待安装结束。

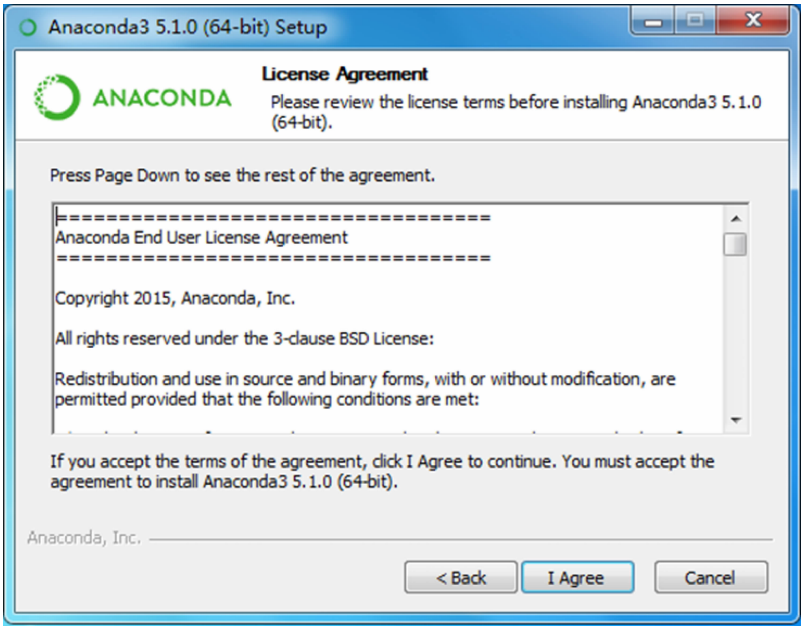


图 1-4　Windows 系统安装 Anaconda 步骤 2

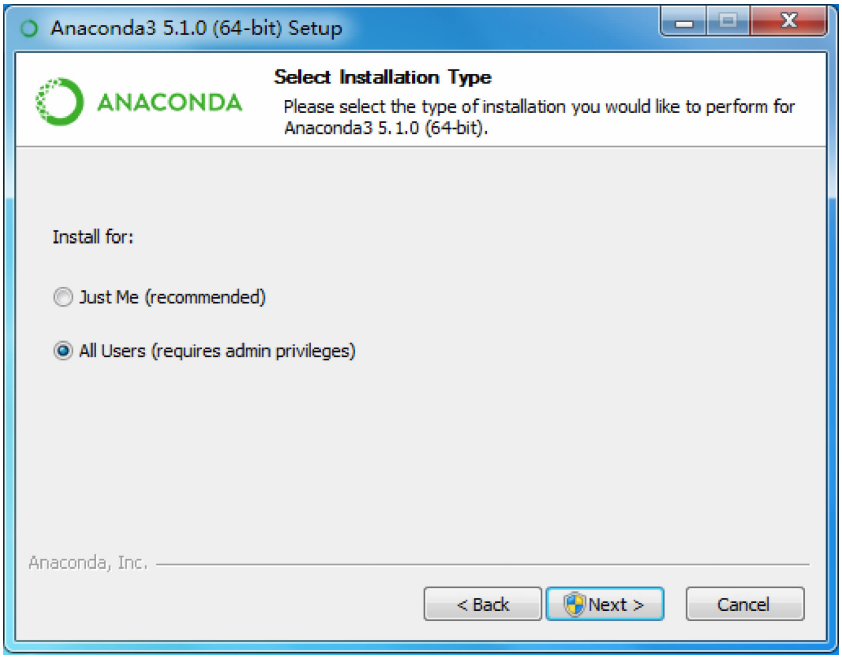


图 1-5　Windows 系统安装 Anaconda 步骤 3

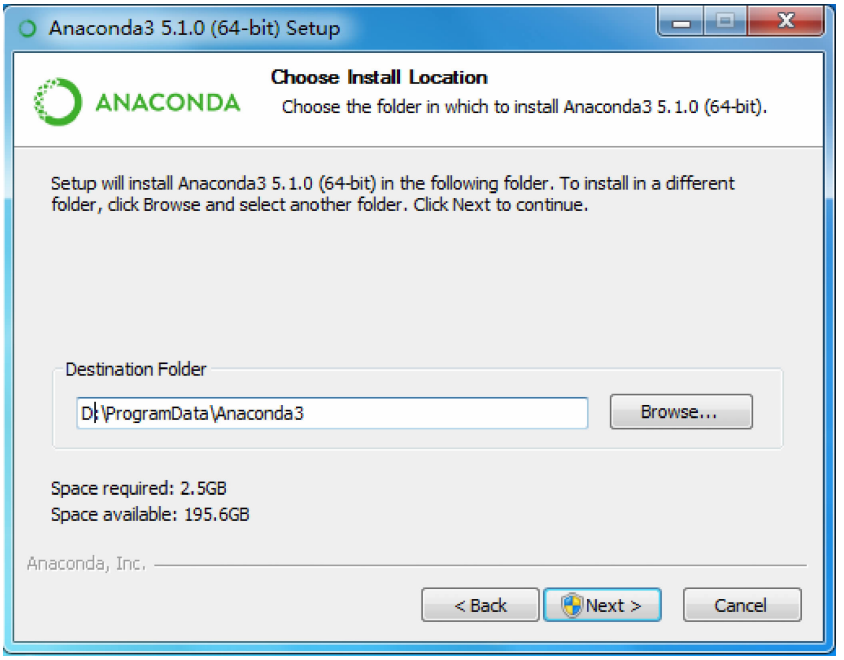


图 1-6　Windows 系统安装 Anaconda 步骤 4

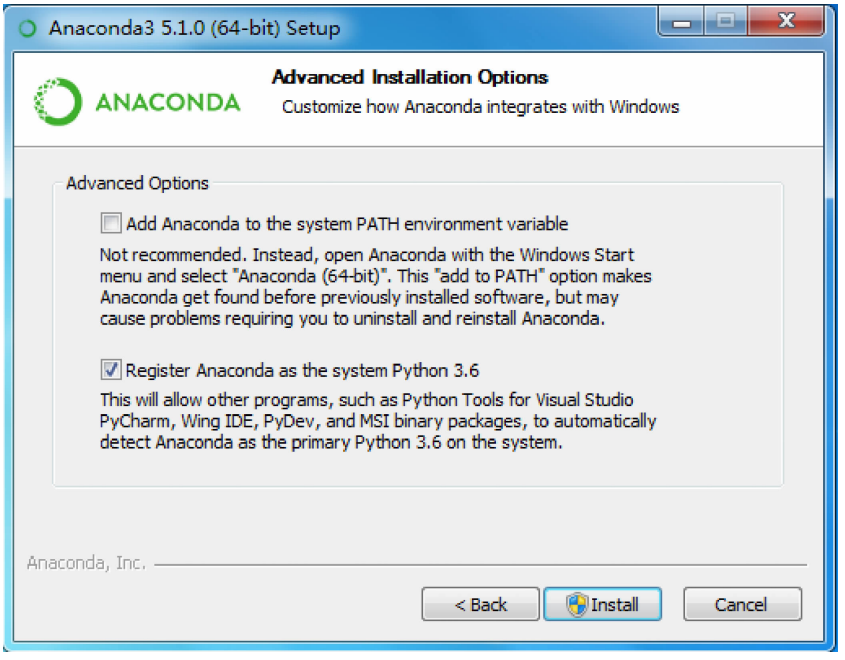


图 1-7　Windows 系统安装 Anaconda 步骤 5

(6)单击图 1-8 中的“Finish”按钮,完成 Anaconda 安装。

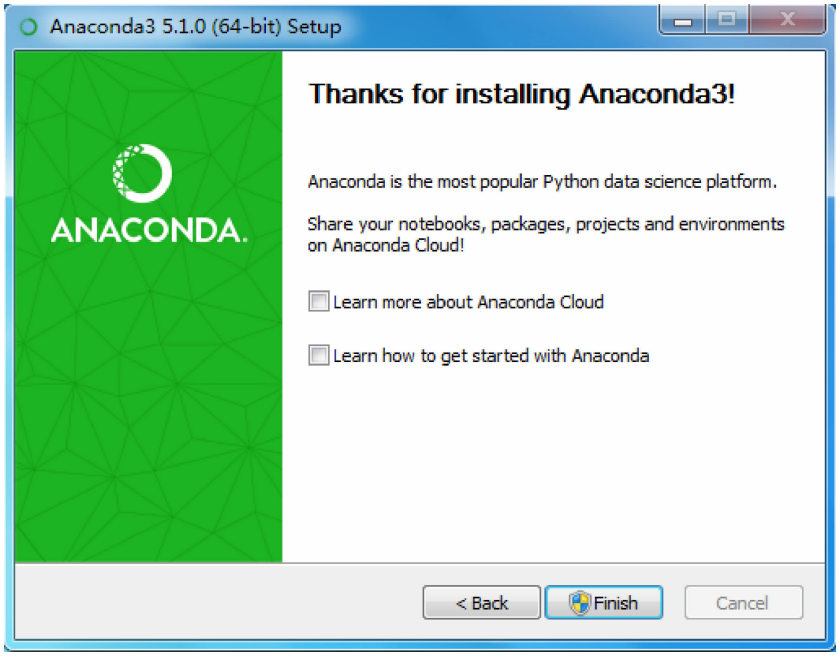


图 1-8 Windows 系统安装 Anaconda 步骤 6

### 3. 在 Linux 系统中安装 Anaconda

从 Anaconda 官方网站下载 Linux 系统的 Anaconda 安装包,选择最新的 Python 版本。Linux 系统中安装 Anaconda 的具体步骤如下。

(1)打开一个用户终端 Terminal。使用 cd 命令将当前路径切换为 Anaconda 安装包所在的文件路径,如图 1-9 所示。


图 1-9 Linux 系统安装 Anaconda 步骤 1

(2)输入命令“bash Anaconda3-5. 1. 0-Linux-x86_64. sh”,如图 1-10 所示。

(3)按 Enter 键后,出现软件协议相关内容,在阅读时连续按 Enter 键读取全文,在协议末尾会让用户确认是否同意以上协议,输入“yes”并按 Enter 键确认同意,如图 1-11 所示。


图 1-10 Linux 系统安装 Anaconda 步骤 2

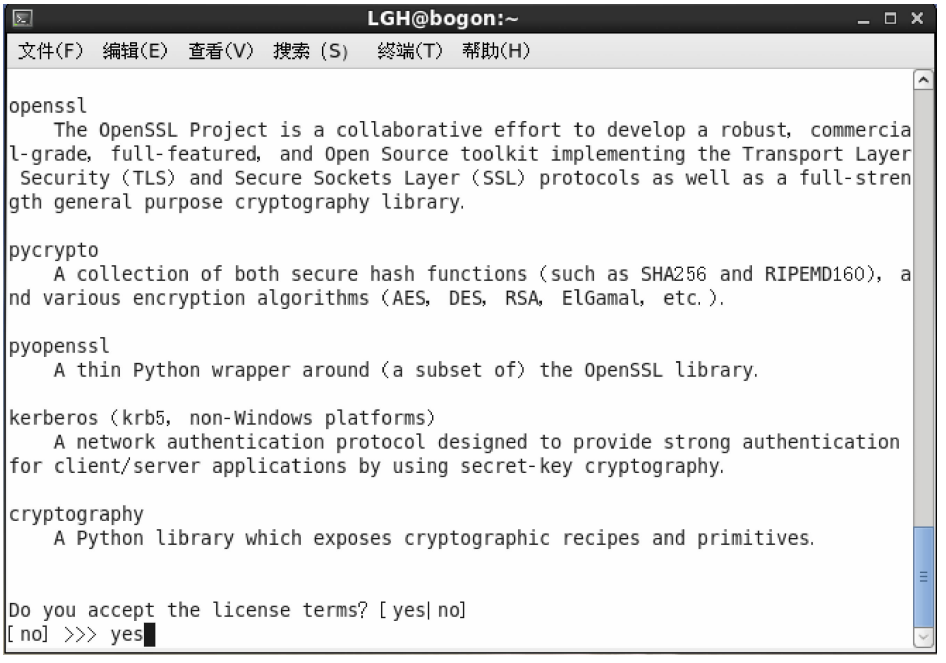


图 1-11 Linux 系统安装 Anaconda 步骤 3

(4)同意协议后,软件就会安装。在安装过程快结束时,将提示用户是否将 Anaconda 的安装路径加入系统当前用户的环境变量中,输入“yes”并按 Enter 键确认同意,如图 1-12 所示。

(5)等待安装完成,完成后使用 Linux 系统的文本编辑器 Vim 或者 gedit 查看当前用户环境变量。输入命令“vi /home/LGH/.bashrc”并按 Enter 键,出现图 1-13 所示界面,表示环境变量配置完成,说明 Anaconda 安装成功。

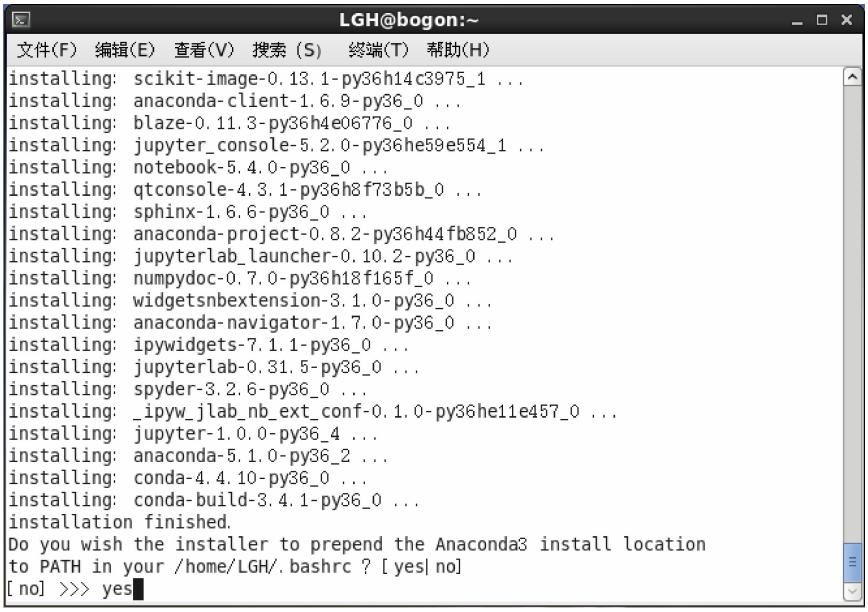


图 1-12 Linux 系统安装 Anaconda 步骤 4


图 1-13 Linux 系统安装 Anaconda 步骤 5

(6)如果未配置完成,在图 1-14 所示的界面末尾添加 Anaconda 安装目录的环境变量“export PATH="/home/LGH/anaconda3/bin:$PATH"”即可。


图 1-14 Linux 系统安装 Anaconda 步骤 6

### 4. 掌握 Jupyter Notebook 的基本功能

1)启动 Jupyter Notebook

在安装完 Python、配置好环境变量并安装 Jupyter Notebook 后，在 Windows 系统下单击“开始”菜单中的 Jupyter Notebook 命令即可启动 Jupyter Notebook，如图 1-15 所示。

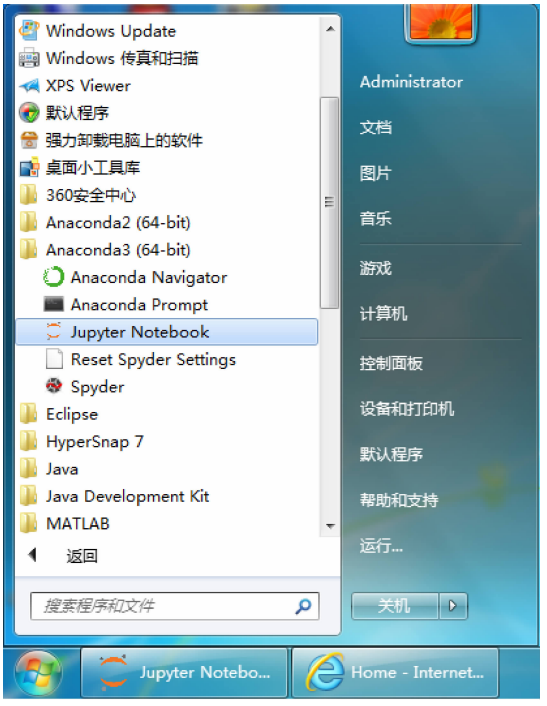


图 1-15　在 Windows 系统下启动 Jupyter Notebook

在 Linux 系统下的终端输入命令“jupyter notebook”并按 Enter 键，即可启动 Jupyter Notebook，如图 1-16 所示。


图 1-16　在 Linux 系统下启动 Jupyter Notebook

2)新建一个 Notebook

打开 Jupyter Notebook 后会在系统默认的浏览器中出现如图 1-17 所示的界面。

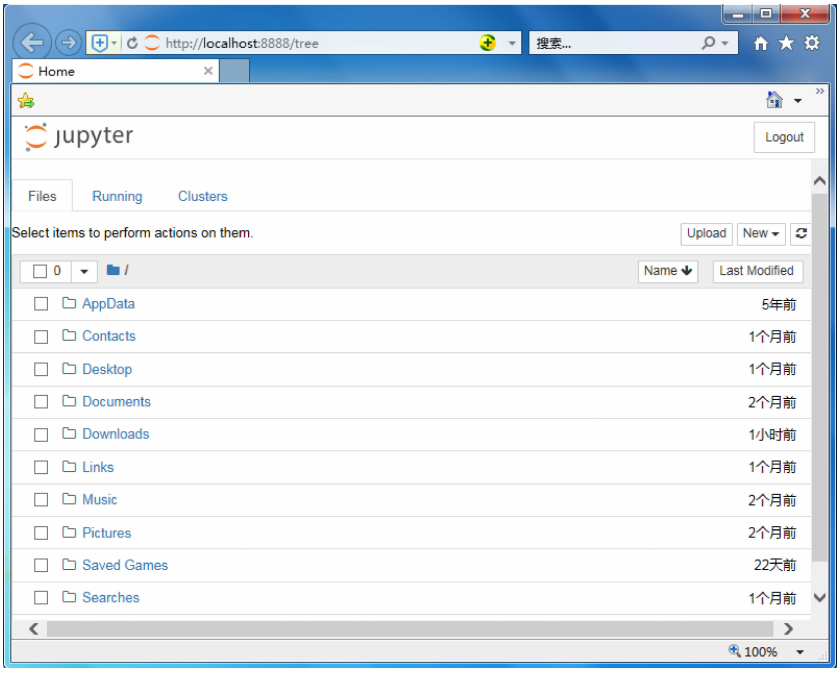


图 1-17 Jupyter Notebook 主页

单击右上方的“New”下拉按钮，弹出下拉列表，如图 1-18 所示。

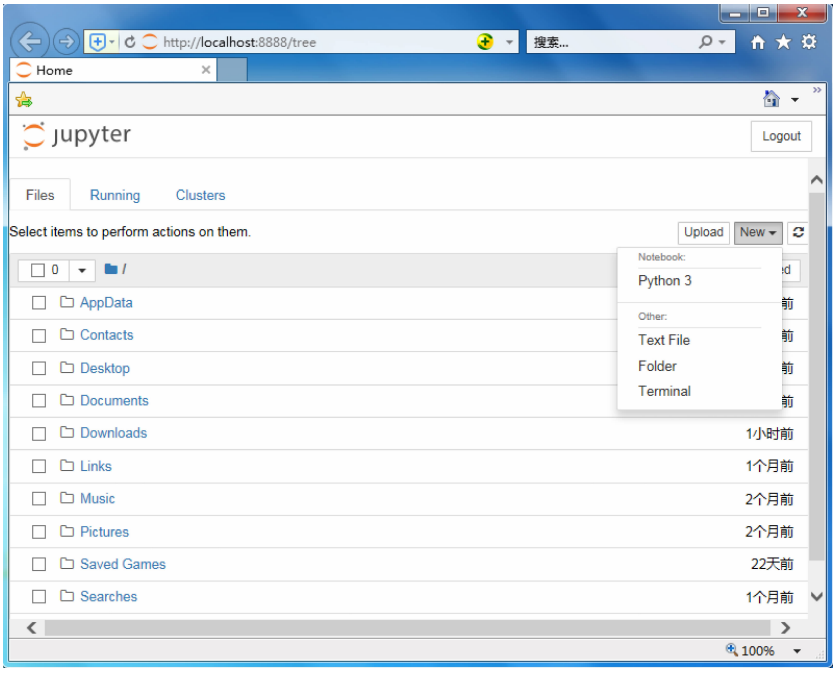


图 1-18 “New”下拉列表

在下拉列表中选择需要创建的 Notebook 类型。其中“Text File”为纯文本型，“Folder”为文件夹，“Python 3”表示 Python 运行脚本，灰色字体表示不可用项目。选择“Python 3”选项，进入 Python 3 脚本编辑界面，如图 1-19 所示。

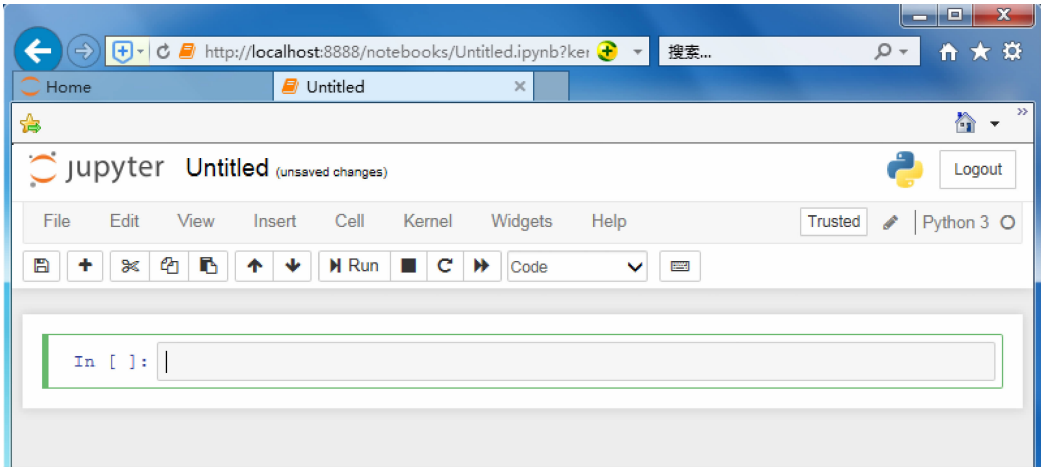


图 1-19　Jupyter Notebook Python 3 脚本编辑界面

3）Jupyter Notebook 的界面及其构成

Notebook 文档由一系列单元构成，主要有两种形式的单元，如图 1-20 所示。

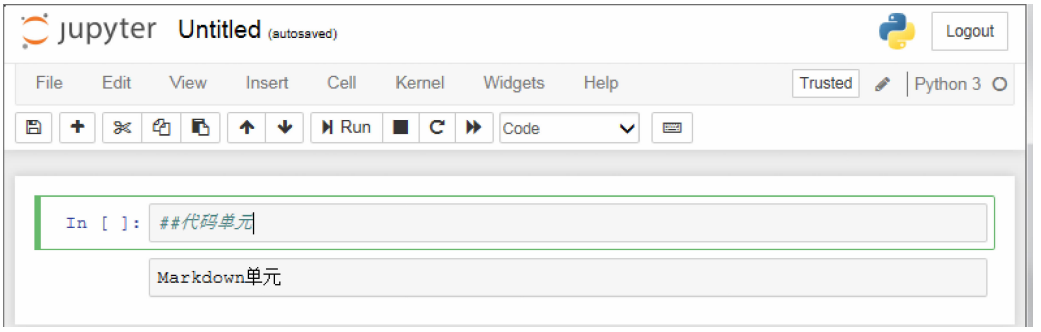


图 1-20　Jupyter Notebook 的两种单元

（1）代码单元。这里是用户编写代码的地方，通过按 Shift＋Enter 快捷键运行代码，其结果显示在本单元的下方。代码单元左边有“In[　]:”编号，方便使用者查看代码的执行次序。

（2）Markdown 单元。在这里可对文本进行编辑，采用 Markdown 的语法规范，可以设置文本格式，插入链接、图片甚至数学公式。同样按 Shift＋Enter 快捷键可以运行 Markdown 单元，显示格式化的文本。

### 5. Jupyter Notebook 的高级功能

1）更改 Jupyter Notebook 的工作空间

在 Anaconda Prompt 中输入“jupyter notebook --generate-config”，如果该配置文件不

存在，那么将会产生一个初始化配置文件；如果该配置文件已经存在，那么会出现图 1-21 所示信息，从中可以见到配置文件存在的位置，此时输入“N”，即不要覆写源文件。在 Anaconda Prompt 中输入“iPython profile create”并按 Enter 键，可以找到关于 Jupyter Notebook 的配置文件的位置。

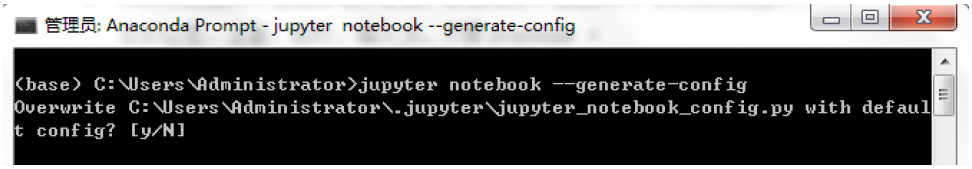


图 1-21 Jupyter Notebook 配置文件位置

在其配置文件 iPython_notebook_config. py 中，有如下语句。

# The directory to use for notebooks and kernels.

# c. NotebookApp. notebook_dir = u''

该句就是用来指定其工作空间的。例如，默认的工作空间是用户名文件夹。如果现在想要将工作空间变为“D:\Jupyter”，那么需要做如下更改。

# The directory to use for notebooks and kernels.

# c. NotebookApp. notebook_dir = u'D:\Jupyter'

注意：路径最后一级后面不要加“\”。

2）Markdown

Markdown 是一种可以使用普通文本编辑器编写的标记语言。通过简单的标记语法，它可以使普通文本内容具有一定的格式。Jupyter Notebook 的 Markdown 单元比基础的 Markdown 的功能更强大。

标题是标明文章和作品等内容的简短语句。读者写报告或者写论文时，标题是不可或缺的，尤其是论文的章节等需要使用不同级别的标题。Markdown 作为一款排版工具，一般使用类 Atx 形式，在首行前加一个“#”字符代表一级标题，加两个“#”字符代表二级标题，以此类推。图 1-22 和图 1-23 分别为 Jupyter Notebook 中的 Markdown 标题代码和展示。

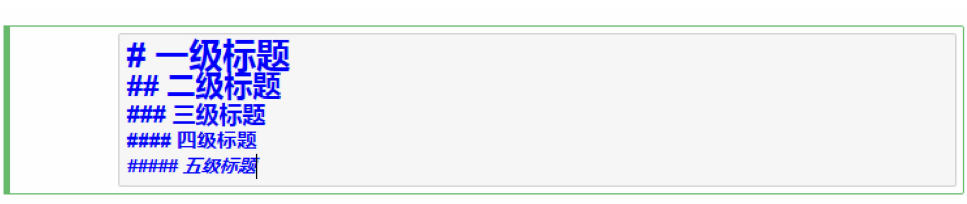


图 1-22 Jupyter Notebook 中的 Markdown 标题代码

# 一级标题

## 二级标题

### 三级标题

#### 四级标题

##### *五级标题*

图 1-23　Jupyter Notebook 中的 Markdown 标题展示

文档中为了凸显部分内容，一般对文字使用加粗或斜体格式，使得该部分内容变得更加醒目。对于 Markdown 排版工具而言，通常使用星号“*”和下划线“_”作为标记字词的符号。前后有两个星号或下划线表示加粗，前后有 3 个星号或下划线表示斜体。图 1-24 和图 1-25 分别为 Jupyter Notebook 中的 Markdown 的加粗/斜体的代码和展示。

```
数据分析概述
**数据分析概述**
***数据分析概述***
__数据分析概述__
___数据分析概述___
```

图 1-24　Jupyter Notebook 中的 Markdown 的加粗/斜体代码

数据分析概述
**数据分析概述**
***数据分析概述***
**数据分析概述**
***数据分析概述***

图 1-25　Jupyter Notebook 中的 Markdown 的加粗/斜体展示

在 Jupyter Notebook 的 Markdown 单元也可以使用 LaTeX 来插入数学公式。在文本行中插入数学公式，应使用两个“$”符号，如 $f(x) = 3x + 7$。如果要插入一个数学区块，则使用两个“$$”符号，如用“$$ \sqrt[2]{b^2-4ac} $$”表示下式。

$$\sqrt{b^2-4ac} \tag{1-1}$$

在输入上述公式的 LaTeX 表达式后，运行结果如图 1-26 所示。

```
In [ ]: $$ \sqrt[2]{b^2-4ac}$$
```

$$\sqrt[2]{b^2-4ac}$$

图 1-26　Jupyter Notebook 中的 Markdown 的 LaTeX 语法示例

## 小　　结

本模块根据目前的数据分析发展状况，将数据分析具象化，首先介绍了数据分析的概念、范畴、流程，阐述了使用 Python 进行数据分析的优势，列举了 Python 数据分析重要类库的功能。紧接着阐述了 Anaconda 的特点，实现了在 Windows 和 Linux 两个系统中安装 Anaconda 数据分析环境。最后展示了数据分析工具 Jupyter Notebook 的优异特性及使用方法。

## 习题 1

### 一、选择题

1. 下列关于数据和数据分析的说法正确的是(　　)。

A. 数据就是数据库中的表格

B. 文字、声音和图像都是数据

C. 数据分析只能是对过去发生事情的描述和分析

D. 数据分析的数据只能是结构化的

2. 下列分析方法属于狭义数据分析的是(　　)。

A. 智能推荐　　B. 关联规则

C. 交叉分析　　D. 文本分类

3. 下列关于数据分析流程的说法错误的是(　　)。

A. 需求分析是数据分析最重要的一部分

B. 数据预处理是能够建模的前提

C. 模型评估能评价模型的优劣

D. 声音和图像无法用数据分析

4. 下列关于数据分析工具的说法正确的是(　　)。

A. MATLAB 是最适合开发网络应用的语言

B. R 语言主要应用于工程计算、控制设计

C. MATLAB 拥有大量的第三方库，而且开源

D. Python 拥有大量的第三方库，能调用 C、Java 等其他程序语言

5. Jupyter Notebook 不具备的功能是(　　)。

A. 直接生成一份交互式文档

B. 安装 Python 库

C. 导出 HTML 文件

D. 将文件分享给他人

## 二、操作题

1. 在计算机上完成 Anaconda 环境的安装，并优化 Jupyter Notebook 的配置。

2. 用 Jupyter Notebook 创建一个 Markdown 文档，包含三级目录。