网易云课堂
五星好评课程

GROWING DATA HEROS
WITH PYTHON

# Python
# 全栈数据工程师养成攻略

**视频讲解版**

张宏伦 编著

门槛**低**、内容**全**、实例**多**、收获**大**
你需要的就是这样的系列**视频教程**

网易云课堂配套课程

扫一扫书中**二维码**，跟着视频**轻松**学
视频时长超过 **900** 分钟

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

网易云课堂
五星好评课程

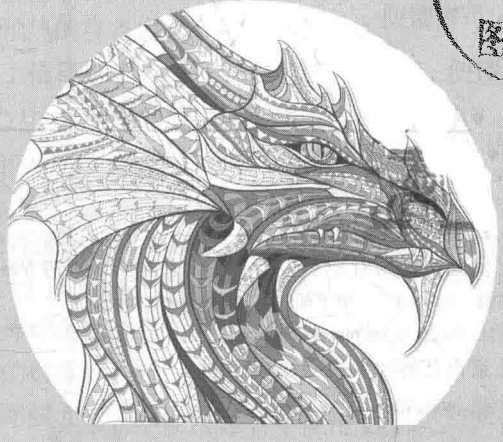
GROWING DATA HEROS
WITH PYTHON

# Python
# 全栈数据工程师养成攻略

视频讲解版

张宏伦 编著





人民邮电出版社
北京

**图书在版编目（CIP）数据**

Python全栈数据工程师养成攻略 : 视频讲解版 / 张宏伦编著. -- 北京 : 人民邮电出版社, 2017.11
ISBN 978-7-115-46869-7

Ⅰ. ①P… Ⅱ. ①张… Ⅲ. ①软件工具－程序设计
Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第224338号

## 内 容 提 要

本书以 Python 为主，结合其他多门编程语言，从数据的获取、存储、分析和可视化等方面，全面地介绍如何实现一些小而美的数据应用。本书共 12 章，第 1 章介绍编程之前的准备工作；第 2 章介绍 Python 中最为核心和常用的语法；第 3 章介绍如何使用 Python 编写爬虫并获取数据；第 4 章介绍如何使用 Python 操作 MySQL 数据库并存储数据；第 5 章介绍如何在 R 语言中使用 ggplot2 绘制静态可视化图形；第 6 章介绍自然语言理解的相关内容以及如何使用 Python 处理文本数据；第 7 章介绍 HTML、CSS、JavaScript 等前端基础；第 8 章介绍 JQuery、ThinkPHP、Flask 等进阶内容；第 9 章介绍 ECharts、D3、Processing 等动态数据可视化工具；第 10 章和第 11 章分别介绍 Python 在机器学习和深度学习中的应用；第 12 章介绍如何通过一个好的故事将自己的数据成果分享和展示给他人。

本书适用于有一定编程基础，希望了解数据分析、人工智能等知识领域，进一步提升个人技术能力的社会各界人士。

◆ 编　　著　张宏伦
责任编辑　刘　博
责任印制　陈　犇

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：17　　2017 年 11 月第 1 版
字数：453 千字　　2017 年 11 月北京第 1 次印刷

定价：59.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

# 前言

随着数据时代的到来，越来越多的人对如何使用数据和挖掘数据价值产生了浓厚兴趣和迫切需求。他们来自于互联网、公共服务、新闻、法律、医疗、设计等不同行业，正在不断地接触和使用日益增长的数据。掌握如何进行数据获取、存储、分析和可视化等技术，对他们当下的工作和未来的发展都能起到重要的作用。

我是一名数据爱好者，乐于不断学习，喜欢挑战自己。在参加了多项数据领域的大型赛事之后，萌生了将自己的经历和经验进行整理总结，并分享给其他广大数据爱好者的想法。我希望这份总结基于理论但不囿于理论，以数据为核心并涵盖尽可能多的领域，注重实战项目和编程能力，帮助对数据感兴趣却不知从何下手的读者概览各方面内容，快速理解掌握相关技能，有所收获并动手实践起来，先全面了解，再深入钻研。

序言 暖个场子

Python 是一门简单易学、功能强大的编程语言，在数据领域中也提供了丰富而完善的支持，可以非常方便地完成各种和数据相关的任务，如处理图片、文本和音频，以及实现经典的机器学习和深度学习模型等。因此本书将以 Python 为主，结合其他多门编程语言，从数据的获取、存储、分析和可视化等方面，全面讲解如何实现一些小而美的数据应用，让每个人都可以独立自主地达成一些数据成就。

本书首先介绍了替读者预热的准备内容。第 1 章讨论数据工程的概念和各种编程语言的特点，带领读者在个人计算机上搭建好 Python 编程环境，并概览了日常生活中数据的组织结构和常见类型。第 2 章介绍 Python 中最为核心和常用的语法，使得即便是之前没有接触过编程的新手，也能快速掌握 Python 的使用方法，通过 Python 完成一些简单的任务，如处理文本数据并进行词频统计。

第 3 章介绍网络爬虫的背景知识和实现原理，以及如何使用 Python 编写简单的爬虫，读者可以根据个人兴趣，从各大门户网站上获取需要的数据。第 4 章介绍如何在个人电脑上搭建 Web 环境，并以关系型数据库中的 MySQL 为例，讨论如何进行数据的存储，使读者可以更好、更方便地存储和管理获取到的数据。

第 5 章介绍另一门简单而强大的编程语言——R 语言，并讨论如何在 R 语言中使用 ggplot2 绘制条形图、折线图、散点图等静态可视化图形，从而更直观地展示从数据中得到的结论。第 6 章介绍日常生活中最为常见和重要的文本数据以及与自然语言理解相关的研究和应用，如何通过 Python 中的 jieba 分词完成中文文本的分词、关键词提取、词性标注等任务，还介绍了词嵌入的概念以及如何训练蕴含语义的词向量。

交互网站是数据可视化的一种重要形式，因此第 7 章介绍 HTML、CSS、JavaScript 等前端基础。第 8 章则介绍一些 Web 进阶内容，包括基于 JavaScript 的前端框架 JQuery，以及如何使用 ThinkPHP 和 Flask 等后端框架实现一个涉及数据库操作的简易个人博客，使读者可以根据个人需求独立设计和完成兼具前后端的网站。第 9 章介绍 ECharts、D3、Processing 等可视化工具的使用，通过交互网站和视频等形式实现数据的更加丰富多样的动态可视化，让读者更好地感受数据的魅力。

接下来的两章选取了机器学习和深度学习两大热门领域的核心内容，为读者进一步实现数据价值的深度分析和挖掘打下坚实基础。第 10 章介绍机器学习的基本概念、常用的经典模型及其实现，并讨论了 XGBoost 模型的训练和调参技巧。第 11 章介绍深度学习的基本概念、CNN 和 RNN 等神经网络的核心思想和应用场景，并以手写数字识别模型为例，介绍如何使用 Python 中的 Keras 实现深度学习模型的定义和训练。

第 12 章介绍如何通过一个好的故事，将自己的数据成果分享和展示给他人，以及如何制作有内容、有颜值的 PPT，使读者在提升自我各方面技术能力的同时，能够有意识地培养和锻炼自己的演讲和交流能力。

在编写本书时，我的妻子、亲人和好友给予了很多帮助，在此非常感谢你们的支持。

希望拿到这本书的每个人，都能感受数据之美，并通过挖掘数据价值，爱上数据。

由于作者水平有限，书中难免存在一些错误或不准确的地方，恳请各位读者不吝斧正。相关意见和建议可以通过知乎“张宏伦”、微信公众号“宏伦工作室”进行反馈，也可以向邮箱 zhanghonglun@sjtu.edu.cn 发送邮件，期待收到各位读者宝贵的意见和建议。书中全部视频也可通过网易云课堂观看。

扫描关注
微信公众号

扫描访问网易
云课堂配套课程

# 目 录

# 第1章 写在前面

## 1.1 数据工程和编程语言

近年来大数据（BigData）的概念火得不行，之前流行的互联网+，换成大数据+后又成就了一大批创业公司。政府部门对大数据战略部署同样重视，各种大数据产业园和科技区如雨后春笋般火热发展。很多不同行业的人言必称大数据，时常把大数据时代的 4 个 V 和 3 种思维[1]挂在嘴边，但他们心里所说的和实际所做的，大多只是大数据领域上层应用中的一个子集，即基于数据做一些统计、分析和展示，甚至很多时候数据并不满足“大”的特征。

数据工程和编程语言

当然，这本书的目的并不是探讨大数据的知识体系和技术架构，而是从个人角度出发，介绍如何在时间有限（可能你并不是大数据领域的专业从事人员）和资源有限（可能你只有一台笔记本电脑可以运行程序）的条件下，实现一些个人能力足以完成的、简单而有趣的数据工程和数据应用。这本书的读者可能已经具备一定的编程基础，也有可能之前未曾接触过任何代码，在经过恰当的学习和足够的练习之后，都可以拿出自己的笔记本电脑，独立实现让人惊艳的数据成果和作品。

### 1.1.1 如何玩转数据

在进行一项数据工程之前，首先需要考虑并解决一些问题，想清楚这些问题的答案比直接撸起袖子写代码更为重要。

#### 1. 获取

我们的数据从何而来？巧妇难为无米之炊，如果希望做出有价值、有意义的成果，所用数

1 参见《大数据时代》，[英]维克托·迈尔-舍恩伯格 肯尼思·库克耶◎著，盛杨燕 周涛◎译

据的数量和质量都应得到保证。理想情况下自然是别人准备好数据提供给我们，但现实情况往往是需要我们自己去获取。如果不具备大规模部署传感器和海量用户上传数据等采集数据的能力，那么通过爬虫从已有网站上获取结构化数据则是唯一的解决途径。因此需要考虑并解决的问题包括以下几点，我需要哪方面的数据？哪些网站已经具备了这些数据？我需要从这些网站分别采集哪些数据？多大的数据量才能满足我的需求？数据是一次获取即可，还是需要持续更新？如果需要持续更新，应当达到怎样的更新频率？

### 2. 存储

我们需要把获取的数据存储下来，以便进一步使用。不同的数据量和数据类型，可能适合于不同的存储方案。对于数据量较少、后续处理较简单的情况，可以将数据存储到静态文件中，如 txt、csv、json 等格式文件。这种方法读写都十分方便，并且易于数据的复制和共享。对于数据量较大、后续处理较复杂的情况，可以将数据存储到一些通用而且成熟的开源数据库中，如 MySQL、PostgreSQL 等关系型数据库，以及 MongoDB、Neo4j 等非关系型数据库（NoSQL）。这种方法更为稳定且易于维护，支持数据的 Create、Update、Read、Delete 等后续操作。如果有部署 Web 网站应用的需求，那么将数据库作为后端数据存储则是更好的选择。因此需要考虑并解决的问题包括：我有多大数据量需要存储？后续处理是否复杂？数据是否会持续更新？我应该选择哪种数据存储方案？

### 3. 分析

在经过必要的清洗工作之后，我们希望从数据中挖掘出感兴趣的价值和结论。一方面可以进行一些简单的计算汇总工作，从不同维度聚合出对应的结果；另一方面也可以从统计学或机器学习的角度出发，分析数据不同字段之间的关联，同时训练一些分类或聚类[2]的模型，用以解决实际应用问题。不同类型的数据，如文本、数值和类别值等，所涉及的数据分析方法可能完全不同。因此需要考虑并解决的问题包括：我的数据属于何种类型？我希望从数据中挖掘出哪些价值？我希望通过数据完成哪些任务？我应当选择哪些分析技术和算法模型？

### 4. 可视化

用数据可视化的方法表达和展示所得结论。正所谓一图胜千言，枯燥的数据和苍白的语言也许并不足以承载数据的价值，而借助图形、色彩、布局等视觉元素则能更生动、更丰富、更全面地诠释数据的灵魂。我们既可以使用散点图、折线图、柱状图等经典图形，也可以大开脑洞去尝试一些天马行空的表达形式，充分探索组织图形、色彩和布局等内容的可能性。因此需要考虑并解决的问题包括：我需要展示哪些数据和结论？哪种图形和表现形式最能满足我的需求？可视化

2 分类和聚类的概念参见第 10 章机器学习

是选择静态图片、交互网站，还是动态视频？

如果以上 4 个步骤的问题都已经想清楚，那么恭喜你，可以按照你的想法开始玩转数据了。通过获取、存储、分析和可视化，将原始数据逐步提升为信息、知识和价值，这便是玩转数据最大的魅力和乐趣所在。

### 1.1.2 关于编程语言

哪种编程语言最好，最适合做数据工程？如果真要讨论起来，这将是一个永远没有结论的哲学问题。既然无法给这个问题一个合适的答案，不如将单选题变为多选题，毕竟只学习一门语言可能远远不够。以全栈数据工程师为目标，我们应当各方面内容都有所涉足，同时具备自己最为擅长和习惯使用的一至两门语言。

C++和 Java 这两门语言最好熟悉其一，从而了解编程语法的基本内容和面向对象的编程思想。熟悉的要求是指不用完全掌握和精通，在需要用到的时候查一查，能够快速回想起相关内容即可。很多人会发现，掌握一门语言之后，再去学其他语言便能很快上手，因为不同语言之间的编程思想都是基本相通的。

Python 是一门简单好用而且功能强大的语言，也是笔者使用最多、最为熟悉的一门语言。Python 的强大之处在于其具备极为丰富的功能包，从前端到后端，从软件到硬件，从机器学习到自然语言理解，几乎无所不包、全栈通吃。同时对语法的约束和限制也没有 C++、Java 那样严格，因此非常适合新手学习。有一句经典的玩笑话，“Python 大法好，除了炒菜别的都可以干”。

R 是一门统计分析语言，和 Python 类似，具有数量众多且功能强大的包，以及庞大而活跃的用户社区。近年来 R 的学习门槛和成本都在不断降低，可以用于进行专业的统计分析和图形绘制，极力推荐同时掌握 Python 和 R。

除此之外，还有和 Web 网站开发相关的一些语言，如前端的 HTML、CSS 和 JavaScript，后端的 PHP、NodeJS 等。在这些语言的基础上还衍生出了丰富的封装和框架[3]，便于用户更快、更好地进行开发。就像 Python 的功能包难以全部掌握一样，和 Web 网站开发相关的封装和框架更是难以全部熟悉。

笔者个人习惯于使用 Python 获取数据并写入文件或数据库中，结合 Python 和 R 进行数据分析和挖掘。至于数据可视化部分，则使用 R 绘制静态图形，基于 Web 网站实现动态交互可视化。

本书的后续章节将以 Python 为主，完整地介绍如何进行数据的获取、存储、分析和可视化，以个人能力独立完成一些有趣的事情。

---

3 如果将原始语言理解成木材，那么封装和框架便是造好的轮子，可以大大节省开发时间

# 1.2 带好装备——Python 和 Sublime

带好装备 Python 和 Sublim

在正式撸起袖子开始写代码之前，需要做好一些准备工作。对我们而言，最为重要的两件装备，便是编程语言和编辑器。

## 1.2.1 Python

Python 是一门语法简单但功能强大的编程语言，也是笔者使用最多、最为熟悉的一门语言。Python 中有很多方便好用的功能包，使用这些包，可以用 Python 来做很多有意思的事情。

### 1. 下载和安装

在 Mac 和 Linux 操作系统上一般会默认自带 Python，Windows 上如果没有的话，可以访问地址（https://www.python.org/），下载并安装 Python。当然，除了手动安装 Python 之外，更推荐使用下面将会介绍到的 Anaconda。

Python 的主流版本有 2.7 和 3.5 两种，语法和内容上存在很多不同。虽然 3.5 更新一些，但 3.5 对 2.7 向下并不兼容，很多在 2.7 中可以使用的包，在 3.5 中无法正常运行，因此 2.7 完全过渡到 3.5 仍需要一段时间。现阶段推荐使用 2.7 版本，熟练掌握 2.7 的用法，即使若干年后再切换至成熟之后的 3.5 版本，也并非难事。

### 2. pip

pip 是 Python 的包管理工具，有了 pip 之后，安装或者删除某个 Python 包，如用于数值计算的 numpy，只需要在系统命令行[4]中输入 pip install numpy 或 pip uninstall numpy 即可，而不用费力去网上查找和下载。

以下文章链接提供了在 Windows 或 Mac 上安装 pip 的操作过程。当然，除了手动安装 pip 之外，更推荐使用下面将会介绍到的 Anaconda。

- Windows，（http://www.tuicool.com/articles/eiM3Er3/）
- Mac，（http://www.xuebuyuan.com/593678.html）

### 3. Anaconda

除了手动安装 Python 和 pip 之外，更好、更方便的选择是安装一个类似 Anaconda 这样的编程组合套餐。Anaconda 包含了 Python 和一些常用的包，以及用于管理包的 pip，这意味着只要安装了 Anaconda，我们所需的软件就一气呵成地全部装好了，类似肯德基的外带全家

4 Mac 中的命令行即终端，Windows 中的命令行即 CMD

桶。在浏览器中访问链接（https://www.continuum.io/downloads）下载和安装 Anaconda，安装完毕后，即可正常使用 Python。

### 1.2.2 Sublime

某些语言可能会有自己专用的编辑器和编程环境，如 Java 的 Eclipse 等。这类专用编辑器可以在编写该门语言的代码时，提供提示和快速补全等，或者具备一些语言对应的特殊功能。但笔者更习惯使用并推荐给读者的是一款通用、简单而且强大的文本编辑器——SublimeText。它可以打开任意类型的文本文件，可以用它编写任何语言的代码，如 Python 和 R，甚至用 Latex 写论文也是没问题的。

#### 1. 下载和安装

Sublime Text 有 2 和 3 两个版本，自然是对应 Python 的 2.7 和 3.5。同样推荐大家使用 Sublime Text 2 即可，因为其不需要激活或注册，可直接使用，功能也完全可以满足需求。虽然定期会出现激活提醒的弹窗，但直接关闭即可，并不影响使用。在浏览器中访问链接（http://www.sublimetext.com/2），并根据你的操作系统选择相应版本，下载并安装即可。

安装完成后即可使用 Sublime Text，它主要有以下两点好处。

（1）支持非常多的扩展插件，每个插件都可以让 Sublime Text 的功能变得更加强大。

（2）代码中的不同地方会用不同的颜色高亮显示，增强可读性和编程体验。

#### 2. 安装插件

SublimeText 之所以功能强大，是因为其提供了相当多的功能插件。在 SublimeText 中安装插件之前，需要做一些准备工作。打开 Sublime Text 之后，按 Ctrl+`组合键调出 Sublime Text 的命令行，其中`在键盘上 1、2、3 等数字键的左边，Sublime Text 的软件界面底部将出现一行灰色的输入框。访问链接（https://packagecontrol.io/installation#st2），复制 Sublime Text 2 标签页中的代码，将其粘贴至刚才出现的命令行中并按回车键，Sublime Text 将运行一些安装工作。运行完毕后重启 Sublime Text，如果在 Preferences 中能看到 Package Control 一项，则说明准备工作已经完成了。

接下来，按 Ctrl+Shift+P 组合键调出 Package Control，如果是 Mac，则使用 Command+Shift+P[5]，或者直接在 Preferences 中单击 Package Control。输入 install，在提示选项中单击 Install Package，然后在列表中查找需要的插件并单击安装即可。如果是卸载插件，则在刚才的 Package Control 中，输入 remove 并单击 Remove Package，然后选择需要删除的插

5 本书中涉及 Ctrl 键的大部分地方，如果是 Mac 则对应 Command 键

件即可。

### 3. 使用和操作

打开 Sublime Text 之后，可以直接将文件夹拖入 Sublime Text 的软件界面中，文件夹会自动加入 Sublime Text 左半部分的 FOLDERS，在这里可以方便地查看文件夹的目录结构和内容。

在文件夹上右击鼠标，会弹出“新建文件”“重命名文件夹”“新建文件夹”“删除文件夹”“在文件夹中查找”“将文件夹从项目中移除”6 个子菜单。其中“删除文件夹”会在系统目录中同时删除该文件夹，而“将文件夹从项目中移除”只是将该文件夹从 SublimeText 的 FOLDERS 中移除，系统目录中的文件夹依旧保留。

## 1.2.3 运行 Python 代码的方法

一般来说，运行 Python 代码的方法主要有以下 3 种。

（1）在系统命令行中输入 Python，进入 Python 提供的交互编程环境，如图 1-1 所示。其优点是可以交互式执行代码，每敲一行代码后按回车键即可运行，已经生成的变量和函数[6]也保存在编程环境中。缺点是无法修改历史代码并重新运行，代码编辑上也存在诸多不变。因此，这一方法多用于探索和尝试，例如，忘记了某个函数的用法，可以在交互编程环境中运行代码进行尝试。

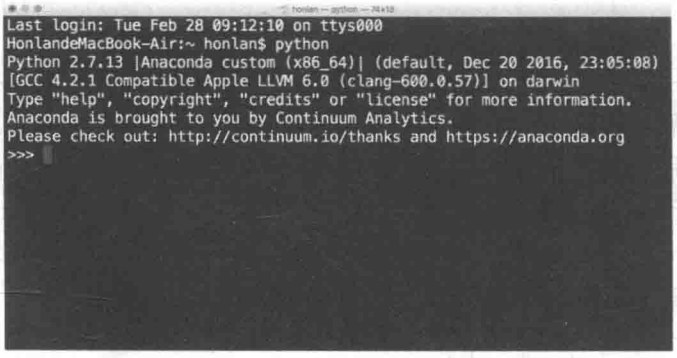



图 1-1 Python 交互编程环境

（2）使用 IPython Notebook 等交互编程工具。IPython Notebook[7]对 Python 内核进行了一层 Web 封装，从而提供了基于 Web 界面的友好交互编程环境，操作上更灵活、更方便，

6 变量和函数的概念参见第 2 章

7 IPython Notebook 的更多内容参见 10.3 节

功能上更自由、更强大，可以便捷地管理文件和项目、交互式进行编程、分块编辑代码并多次运行、轻松实现代码与他人的分享，因此对于编程新手而言也是非常理想的选择。

（3）在 Sublime Text 等编辑器中编写代码，编写完毕后直接在编辑器中运行。例如，按 Ctrl+B 组合键可在 Sublime Text 中运行代码。也可以在编写完毕后打开命令行，切换到代码所在目录，输入 python code.py，其中 code.py 是需要运行的代码。

笔者个人更习惯和青睐第三种方法，因为可以享受一气呵成写完所有代码并运行的畅快，当然更主要的原因是 Sublime Text 提供了非常舒适和便捷的编程体验。

### 1.2.4 Hello World

程序员之间有个不成文的规定，即但凡学点什么新东西都得先来一个 Hello World。在 Sublime Text 中按 Ctrl+N 组合键新建一个文件，输入以下代码之后，按下 Ctrl+S 组合键保存。文件名任意，后缀名为 py，如 test.py，保存时记得选择保存的路径。

```
print 'Hello World'
```

保存完毕后，按下 Ctrl+B 组合键运行写好的代码，即可看到打印出来的文本内容，如图 1-2 所示。

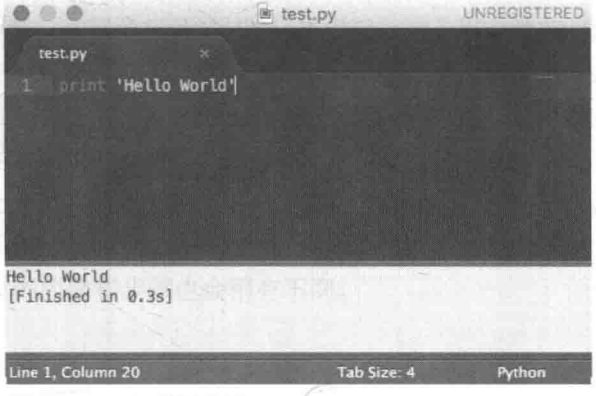



图 1-2 在 Sublime Text 中打印 Hello World

## 1.3 数据结构和常见类型

解读数据结构和类型

在正式开始编程之前，为了进一步加强对数据的理解和把握，先了解在日常生活中，数据大多呈现的结构，以及常见的数据类型。

### 1.3.1 数据的结构

在数据技术（Data Technology，DT）时代，我们的日常生活中随时随地都会产生、接触和使用到各式各样的数据，它们的结构和形式具备很多共性。以地铁数据为例，可以将其分为静态数据和动态数据两大类。

- 静态数据：包括线路信息和站点信息等，例如，一共有哪几条地铁线路，每条线路包含哪些站点，各个站点的名称、首末班车时间等信息。这类数据一般不包含时间戳，更新频率较低，数据量整体较小。
- 动态数据：主要是地铁的刷卡记录，乘客在进站和出站时的刷卡操作都会产生一条刷卡数据，里面记录了乘客的地铁卡 ID、刷卡站点、刷卡时间、刷卡费用等信息。这类数据一般包含时间戳，用于表明数据产生的时间，并且不断产生、不断积累，往往蕴含巨大的潜在价值。

以上提到的时间戳是怎样的一个概念呢？时间戳是指从 1970 年 1 月 1 日 0 时 0 分 0 秒到某一时刻之间所经历的秒数，可以为整数或浮点数。对于同一个时刻，不同的人会有不同的表述方式，例如“2017 年 1 月 1 日 15 时”和“17 年元旦下午 3 点”，即不同格式的时间文本，因此无法统一和计算。通过时间戳的概念，可以使用整数或浮点数来表示任意一个时刻，从而便于代码运算和比较两个时刻之间的时间差。

日常生活中的大多数据都可以使用行和列的结构来表示。每一行表示一条记录，或者称为一项观测。例如，在地铁线路数据里，每一行代表一条地铁线路的记录；每一列表示一个字段，或者称为一项属性。例如，在地铁线路数据里，每行记录可能包含线路名称、运营时间、线路颜色等字段。这种数据结构称为关系型数据，通常会包含一行表头，用于说明每列字段的意义和数值类型，可以用二维数组或二维表的概念来表示。Excel 中的表格、关系型数据库，如 MySQL 中的数据表、R 中的数据框、Python 中 pandas 包提供的 Dataframe 等，都属于这种数据结构。

### 1.3.2 数据的类型

#### 1. txt

txt 是最常见的文本数据类型，或许也是我们大多数人第一次使用电脑所接触的文件类型。txt 中存放的是纯文本，可以记录任意文本内容，每行的长度是可变的，文件的总行数也是任意的，因此读写起来非常自由，但也给进一步使用代码处理带来了不便，毕竟机器更擅长处理结构化数据，而不是非结构化数据。

#### 2. csv

csv（Comma Separated Values）即逗号分隔值，里面存放的依旧是文本内容，但是以一