

第一部分

Python 编程基础

第 1 章

引 言

学习目标

- 了解 Python 编程语言，包括其语法特点、生态系统和开源社区等。
- 了解 Python 语言的起源以及当前的发展趋势。
- 掌握 Python 数据分析的基本概念与步骤。
- 了解 Python 在数据分析领域中的重要性和应用场景。

随着数据科学的迅猛发展，Python 作为一种强大的编程语言，已成为数据分析领域的首选工具。本章将引导你了解 Python 数据分析的基础概念，探索其强大功能，并揭示数据分析在实际应用中的巨大潜力。

1.1 为什么学习 Python

在当今科技飞速发展的时代，学习编程语言已经成为许多人的必然选择。而 Python 作为一门流行的编程语言，其具有许多吸引人的特点。

1.1.1 简单易学的语法

学习 Python 的一个主要原因是其简单易学的语法。Python 的语法设计旨在使初学者能够迅速入门，同时对有经验的开发者也很友好。以下是一些强调 Python 语法易学性的关键特点。

(1) 清晰简洁。Python 以其清晰简洁的语法而著称，相较于其他编程语言，它减少了冗余的符号和结构。这种简洁性使得 Python 代码更易于理解，有助于降低学习曲线，同时提高代码的可维护性。此外，清晰简洁的代码结构也有助于减少错误和提高代码的可靠性，为开发者提供更愉悦的编程体验。这种特性使 Python 成为初学者和专业开发者的首选，推动着其在不同领域的广泛应用。

(2) 强调可读性。Python 的设计哲学之一是强调代码的可读性。这意味着在 Python 中编写的代码更接近自然语言，更易于理解。这种特性使得团队成员能够更容易地理解、维护和扩展彼此的代码，促进了团队协作，提高了团队效率。

(3) 缩进代替大括号。Python 使用缩进而不是大括号来表示代码块。这种缩进的方式

使得代码的结构更加直观，也促使程序员编写更一致的代码。Python 的这一特点也使新手更容易理解代码块的层次关系，同时鼓励更规范的代码风格，提高了整体代码的可读性和可维护性。

(4) 少量关键字。Python 拥有相对较少的关键字，这降低了学习的难度，且大多数关键字都具有直观的含义，这有助于开发者快速理解和使用。例如，Python 中的关键字如 `if`、`for`、`while`、`def` 等，都是直接对应于编程逻辑中的基本概念，如条件判断、循环和函数定义。

(5) 解释型语言。解释型语言意味着 Python 代码在运行时会被解释器逐行解释执行，而不是像编译型语言那样先编译成机器代码再执行。这种特性使得 Python 的开发过程更加快速和灵活，因为开发者可以立即看到代码的执行结果，而不需要等待编译过程。

(6) 动态类型系统。Python 是一种动态类型语言，无需显式声明变量的类型。这简化了代码，使得变量的使用更加自由灵活。这种灵活性不仅提高了编码效率，同时也使代码更容易调试和修改，适应了快速迭代和开发的需求。

1.1.2 强大的可视化功能

Python 还以其丰富的可视化功能而闻名（图 1-1，图 1-2）。具体来说，在数据分析和展示领域，Python 以其丰富的可视化功能和强大的可视化工具库，为数据分析和展示领域提供了强有力的支持。用户可以利用这些工具将数据转化为直观、易于理解的图表形式，从而更好地理解 and 传达数据中的信息。

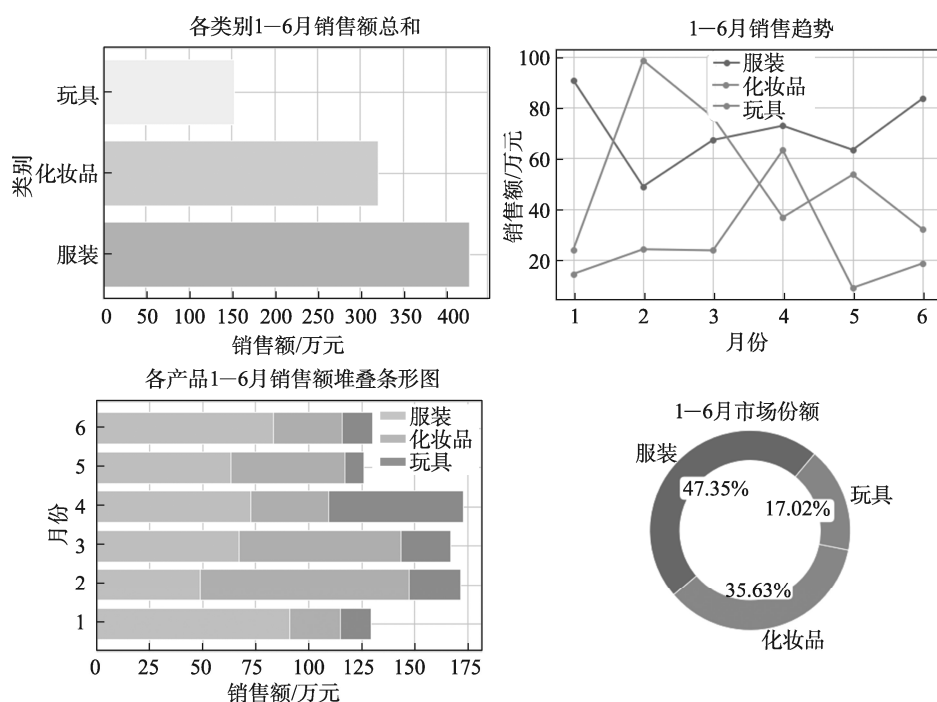


图 1-1 数据可视化（一）

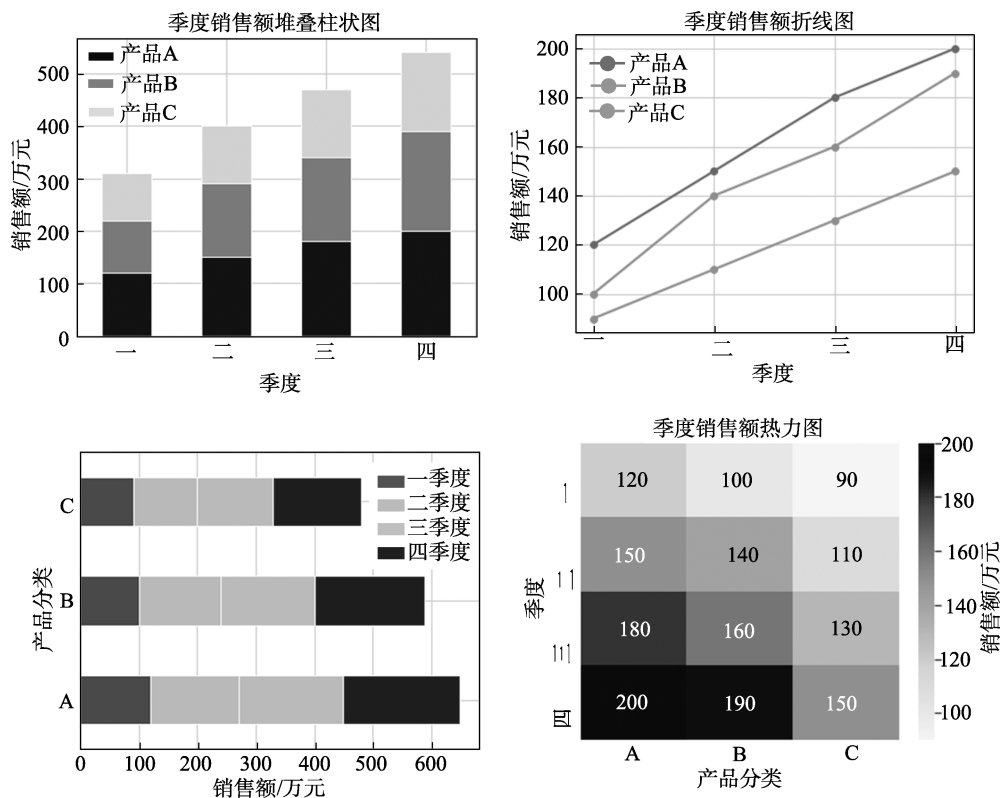


图 1-2 数据可视化（二）

1.1.3 多用途的编程语言

Python 是一种多用途的编程语言，其简洁易读的语法使初学者能够轻松入门。具体来讲，在 Web 开发上，Python 有很多强大的 Web 开发框架，这些框架极大地简化了 Web 应用的开发流程，使开发者能够快速搭建起功能丰富的网站和应用程序；在数据科学和人工智能领域，Python 是最常用的语言之一，提供了众多优秀的机器学习和深度学习框架；Python 还常用于编写自动化脚本，无论是系统管理、文件处理，还是网络爬虫，Python 都能提供简洁的代码实现。

总体而言，无论是构建高效的 Web 应用、进行复杂的数据分析，还是编写简单的自动化脚本等，Python 都展现了其灵活性和适应性，为不同领域的开发任务提供了一种统一的、强大的解决方案。

1.1.4 生态系统与开源社区

在 Python 的生态系统中，有许多优秀的工具和库，如 SciPy、Matplotlib、TensorFlow、PyTorch 等，涵盖了科学计算、机器学习、人工智能等多个领域。这使 Python 成为一个全面且强大的编程工具，适用于各种不同的应用场景。通过参与这个充满活力的生态系统，开发者可以更轻松地掌握新技术、解决问题，并与同行进行交流与合作。

同时, Python 的开源本质使其成为一个由全球开发者社区支持的语言。这个社区提供了大量的学习资源、文档和教程, 帮助初学者快速入门, 并为专业开发者提供深入的技术洞察。开源的特性还意味着 Python 会不断更新和改进, 通过社区的合作, 不断推动语言的发展, 确保其保持在技术创新的前沿。

总体而言, Python 的强大之处不仅在于语言本身的设计和简洁性, 也在于多用途的应用以及其庞大、活跃的开源社区, 这使得学习者和开发者能够更轻松地获取并应用最新的技术和工具。

1.2 Python 的历史和发展

1.2.1 Python 的起源

Python 的起源可以追溯到 1989 年, 由吉多·范罗苏姆 (Guido van Rossum) 在荷兰国家数字与计算机科学研究中心 (Centrum Wiskunde & Informatica, CWI) 工作时开始开发。Python 的名字受到英国喜剧团体 Monty Python 的影响, 强调了一种幽默和可读性的设计理念。吉多的目标是创建一种易读且功能强大的编程语言, 以促使程序员更加愉快地工作。

Python 的第一个公开发布版本在 1991 年发布, 它是一个解释型、动态类型的编程语言。从一开始, Python 就具备了函数、异常处理, 包括列表和字典在内的核心数据类型, 以及基于模块的扩展系统。

此后, Python 的发展逐渐得到了更多开发者的参与和支持, 形成了一个庞大的开源社区。在过去的几十年里, Python 经历了多个版本的更新和改进, 不断增加新的特性和不断优化, 成为当今最受欢迎的编程语言之一。



扩展阅读 1.1 Python 的重要版本更新

1.2.2 Python 的发展趋势

Python 的使用在过去几年中经历了显著增长, 在当前的 IT 就业市场上依然是首屈一指、备受追捧的技术之一。如图 1-3 所示, 根据 2024 年编程语言排行榜的最新数据, Python 以 16.4% 的市场份额稳居第一, 保持着其在编程语言中的领先地位。

同时, Python 也在多个领域取得了广泛应用, 以下是 Python 发展趋势的一些关键方面。

(1) 数据科学和机器学习。Python 在数据科学和机器学习领域的应用呈现显著增长。Python 拥有众多强大的库和框架, 如 NumPy、Pandas、Matplotlib、Scikit-learn、TensorFlow 和 PyTorch 等, 使其成为数据科学家和机器学习工程师的首选语言。这些工具提供了丰富的功能, 包括数据处理、统计分析、机器学习模型的开发和部署。

(2) 人工智能。人工智能领域中, Python 已经确立了其主导地位, 成为首选编程语言之一。其通用性和易用性使研究人员和工程师能够专注于算法和模型的开发, 而无需过多关注编程细节。TensorFlow 和 PyTorch 等流行的深度学习框架都是用 Python 编写的, 为构建、训练和部署复杂的神经网络模型提供了便利, 这使得 Python 在自然语言处理、计算机视觉和语音识别等关键领域也取得了广泛的成功和应用。

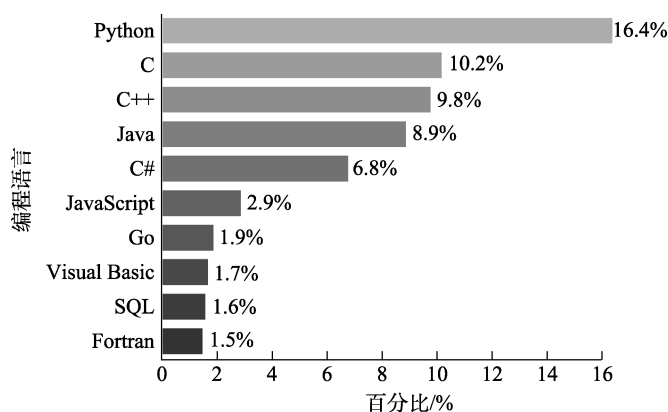


图 1-3 Python 市场份额

(3) Web 开发。在 Web 开发中, Python 广泛应用于构建高效、可扩展的 Web 应用程序。强大的 Web 框架使得开发者能够快速搭建各种规模的 Web 应用。例如, 使用 Django 框架可以迅速构建功能强大的 Web 应用, 而 Flask 则提供了更轻量级的框架, 适用于小型到中型规模的项目。这些框架不仅提供了简化开发流程的工具, 还有助于维护良好的代码结构和提高开发效率。

(4) 自动化和脚本。Python 以其简洁的语法和强大的标准库, 被广泛应用于自动化脚本开发。无论是文件管理、数据处理, 还是批量任务执行, Python 都能提供高效的解决方案。例如, 系统管理员可以使用 Python 编写脚本, 实现对重复性任务(如日志分析、备份管理、用户权限分配等)的自动化处理, 大幅度提高工作效率。

(5) 大数据和科学计算。Python 在大数据领域的应用不断增加, 引入了相关的工具和库, 如 PySpark 和 Dask 等, 使 Python 成为处理大规模数据集和进行科学计算的强大工具。PySpark 为 Python 提供了与 Apache Spark 的无缝集成, 为分布式数据处理提供了便利。同时, Dask 则使 Python 能够进行并行计算和任务调度, 使其适用于处理大规模数据集的计算密集型任务。这些工具的引入进一步拓展了 Python 的应用领域, 使其成为大数据处理和科学计算领域的首选语言之一。

1.2.3 Python 的在线编程

随着技术的发展, 出现了许多在线平台和工具, 开发者能够直接在浏览器中进行 Python 编程。这些平台通常提供了交互式的开发环境, 无需用户在本地安装 Python 解释器。Python 在线编程被广泛用于教育领域, 以下是其中的一些基本信息。

(1) 教育目的。Python 的在线编程平台在学校和大学等教育机构中得到广泛应用, 用于教授编程基础、数据科学、机器学习等课程。这些平台为学生提供了一个互动的学习环境, 使他们能够直接在浏览器中进行编码、实时运行代码、调试错误, 并即时查看代码执行结果。这种实时的反馈机制有助于加深学生对编程概念的理解, 提高编程技能, 并培养解决问题的能力。此外, 在线编程平台通常集成了丰富的学习资源和教学工具, 为教育者和学生提供了便捷而有效的教学环境。

(2) 在线编程环境的便利性。在学校中, 学生可能会面临使用不同设备的情况。在线

编程平台通过浏览器提供了无需本地安装 Python 的灵活性，学生可以在任何地方、使用任何设备访问并学习编程。这种便利性消除了对特定操作系统或硬件的依赖，为学生提供了更大的灵活性，使他们能够在不同场合和设备上轻松学习和实践编程技能。同时，这也减轻了学校和教育机构维护学生机器环境的负担，简化了教学过程，使学习编程更加无障碍和普及。

(3) 实时协作和分享。一些在线编程平台支持实时协作，使学生能够在项目上共同协作，提高团队协作能力。通过在线编程环境，多个用户可以同时编辑和查看相同的代码文件，实时同步变更，从而实现实时的协作效果。这种功能不仅有助于学生学习团队合作与编程实践，还提升了解决问题的协同能力。此外，学生可以轻松分享他们的项目和代码。通过简单的链接或共享选项，他们能够向同学、老师或整个社区展示自己的成果。这种分享机制促进了知识的传播和交流，使学习过程更具社交性，也为他人提供了学习和借鉴的机会。这样的实时协作和分享功能使在线编程平台成为学生之间、学生与教师之间互动的强大工具。

1.3 利用 Python 进行数据分析

1.3.1 数据分析的基本概念

数据分析是指通过采用适当的统计分析方法，对大量收集到的数据进行处理和研究，从中提取有用的信息、揭示潜在的模式，并最终形成结论。这一过程有助于支持决策、发现趋势、解决问题，为业务和研究提供深刻的洞察。通过数据分析，组织和个人能够更好地理解他们所面临的挑战和机遇，以便更明智地制定战略和采取行动。

在 Python 中，数据分析得到了强大的支持，有许多丰富的数据科学库可供利用。其中包括但不限于 Pandas, NumPy, Matplotlib, Statsmodels 等。这些库提供了广泛的功能，涵盖了数据的处理、统计分析、可视化和机器学习等方面。Pandas 用于处理和操作数据表格，NumPy 提供了高性能的数学运算，Matplotlib 用于绘制图表，而 Statsmodels 支持统计建模。这样数据分析人员能够在 Python 环境中轻松地进行数据探索和建模工作，从而提高了效率和便捷性。

1.3.2 数据分析的基本步骤

如图 1-4 所示，数据分析通常涉及一系列基本步骤，这些步骤帮助从原始数据中提取信息，得出结论，支持决策制定。这些基本步骤通常包括问题定义、制定统计假设、数据收集、数据预处理、描述性统计分析、统计建模以及验证与解释。通过系统性地执行这些步骤，数据分析人员能够深入理解数据集，发现潜在的模式和趋势，从而为业务决策提供可靠的基础。

1. 问题定义

问题定义是进行统计分析的关键步骤。通过明确研究问题，确定需要关注的变量和相应的假设，以便能够更有针对性地选择适当的统计技术，并规划有效的数据收集方法，从

而确保统计分析能够有效地解决研究问题。



图 1-4 数据分析基本步骤

2. 制定统计假设

在进行统计分析时，制定明确的统计假设是关键步骤。原假设（ H_0 ）是研究者提出的关于数据总体的陈述，通常表达为没有效应或差异。备择假设（ H_1 ）则陈述相对于原假设的期望变化，可能是效应存在、关系存在或差异存在。通过明确定义这两个假设，为后续统计测试提供了明确的方向，使其能够在数据分析中检验假设，从而得出对所研究问题的有力结论。

3. 数据收集

在数据分析过程中，数据收集是至关重要的步骤。它涉及通过合适的方法获取与研究问题相关的数据，并确保收集的数据在方法和样本选择方面符合要求。有效的数据收集需要仔细规划和执行，包括明确定义变量、选择适当的抽样方法、确保数据的可靠性和有效性，并在实际收集过程中保持一致性，以确保所得数据对后续的统计分析具有可信度和代表性。

4. 数据预处理

数据预处理是数据分析过程中至关重要的步骤，如图 1-5 所示。其中数据清洗发挥着

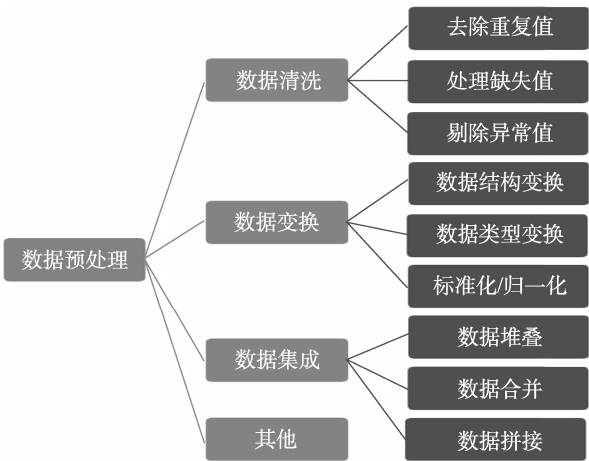


图 1-5 数据预处理

重要的作用，数据清洗旨在消除数据中的缺失值、异常值和重复值，以确保数据质量和一致性。另外，我们也对数据进行变换和集成，使其适用于统计分析，为建模和决策提供可靠的基础。整个过程通常需要深入了解数据的特点，并运用适当的工具和技术，如 Pandas 和 NumPy 库，以便高效地清理和准备数据，为进一步的统计分析打下坚实的基础。

5. 描述性统计分析

描述性统计分析是数据分析的关键步骤，通过计算一些基本的统计指标，我们能够对数据的整体情况有一个初步了解。这些统计指标包括均值、中位数、标准差等，它们提供了数据集的集中趋势、离散程度等重要信息。例如均值反映了数据的平均水平，中位数描述了数据的中间位置，而标准差则反映了数据的离散程度。通过这些指标的计算和分析，我们能够对数据集的特征有更深入的认识，为后续的统计建模和决策提供基础。

6. 统计建模

在数据分析的进阶阶段，统计建模成为一种有力的工具，特别是当我们需要理解变量之间的关系、进行预测或者探索潜在的模式时。统计建模通过应用各种统计方法，使我们能够量化变量之间的相互影响，并据此构建预测性模型或探索性模型。这一过程涉及选择适当的模型、拟合模型参数、评估模型性能等步骤，进而从数据中提取有关现象和关联的深刻见解。

例如，线性回归用于建模因变量与一个或多个自变量之间的线性关系。通过拟合一条直线，线性回归可以帮助我们了解变量之间的趋势，并用于预测新的观测结果。这种模型适用于连续型因变量的情况，如预测销售额、温度等。

又如，逻辑回归主要用于描述和推断二分类或多分类因变量与一组自变量的关系。比如在市场营销领域中，我们探讨消费者购买汽车的影响因素，这里的因变量为是否购买汽车，即“是”或“否”，为二分类变量。通过二分类逻辑回归（binary logistic regression）分析，我们就可以大致了解到底哪些因素是购买汽车的影响因素。线性回归和逻辑回归都是统计建模中常用的工具，根据问题的性质选择合适的模型能够更准确地理解数据和做出预测。

7. 验证与解释

在统计建模之后，验证模型的适用性以及对统计结果进行解释是至关重要的步骤。通过采用如残差分析等方法，验证模型的准确性和稳健性，以确保其对业务或研究问题提供有意义的洞察。同时，对统计结果进行解释则有助于深入理解模型的影响因素，为决策提供可靠的依据。这两个步骤相互补充，确保了模型不仅是合适的，而且能够为业务或研究问题提供有意义的洞察。

1.3.3 数据分析的应用领域

Python 被越来越多的用于数据分析当中，其重要性体现在许多方面。

1. 决策支持

数据分析为决策制定提供了有力的支持。通过对大量数据的分析，决策者能够更好地了解现状、趋势和潜在的影响，从而做出更明智、基于事实的决策。

假设一个电商公司想要优化其产品推广策略，决策者希望通过数据分析了解不同产品特征之间的关系，以及它们与销售表现的关系，从而制定更有效的推广策略。

决策者可以利用散点矩阵图来观察不同因素之间的关系，以及它们与销售表现的关系，如图 1-6 所示。通过观察图中的分布情况，可以得到一些初步的洞察：

- （1）如果发现某个因素与销售额之间存在强烈的正相关关系，那么公司可以考虑更加关注这个因素；
- （2）通过观察不同类别的数据点分布，可以了解不同类别产品的差异，为差异化推广提供支持。

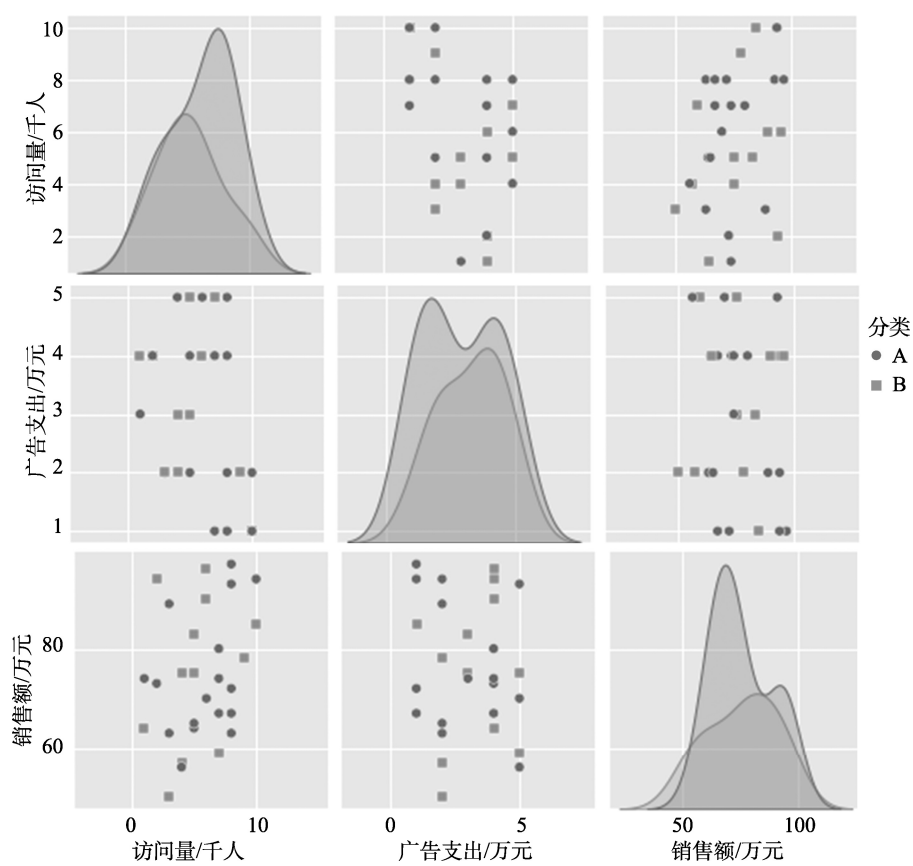


图 1-6 决策支持——散点矩阵图

2. 业务优化

通过对业务数据的深入分析，组织能发现效率低下、资源浪费的环节或优化机会。这有助于提高生产力、减少成本，并改善业务流程。

假设有一家全球性的电子零售公司，销售各种电子产品，其产品分布在不同的地区和仓库。通过绘制太阳爆炸图（图 1-7）来展示销售情况，并更深入地了解不同地区、仓库和产品的销售贡献，具体如下：

- （1）在每个地区，每个仓库销售了哪些产品；

- (2) 不同产品在销售中的贡献；
- (3) 在整个电商公司范围内，哪个地区、仓库和产品的销售额最高，哪些相对较低。

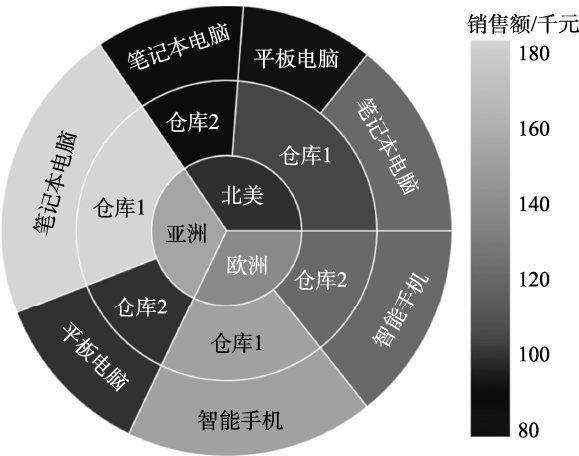


图 1-7 业务优化——太阳爆炸图

3. 市场竞争

在竞争激烈的市场中，数据分析可以帮助企业更好地了解客户需求、市场趋势和竞争对手动态。这使得企业能够更灵活地调整战略，提高市场竞争力。

假设一家电子消费品公司想要了解其智能手机在市场上的表现，并以此制定未来策略。通过过去几个季度的销售数据（图 1-8），公司进行了市场趋势分析，并利用折线图来可视化数据，具体包含以下内容。

- (1) 公司销售额折线：显示了公司每个季度的智能手机销售额的变化。
- (2) 竞争对手销售额折线：每个主要竞争对手都有一条线，展示了它们在相同季度的销售额变化。

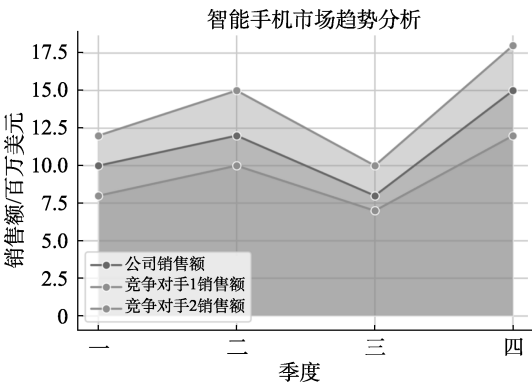


图 1-8 市场竞争——折线图

4. 客户洞察

数据分析有助于深入了解客户行为、偏好和需求。通过分析客户数据，企业可以个性

化服务、改进产品，提高客户满意度，并增加客户忠诚度。

假设一家电子商务公司希望通过数据分析深入了解其在线平台上不同产品类别的客户购买行为，以便更好地满足客户需求。公司收集了每位客户购物车中电子产品、服装以及家用电器的购买数量，通过小提琴图（图 1-9）来可视化不同产品类别的购买分布。

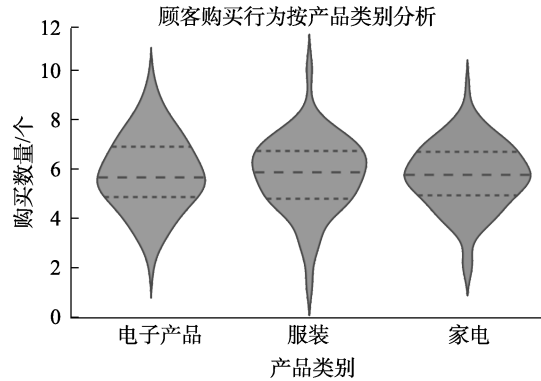


图 1-9 客户洞察——小提琴图

5. 统计建模

统计建模在数据分析中扮演着关键的角色，它通过数学和统计学的方法帮助我们理解数据之间的关系，并进行分析和预测。Python 作为一门强大的编程语言，提供了丰富的库和工具，使得统计建模变得更加灵活和可行。

考虑一个实际场景：我们拥有一个包含房屋面积和价格的数据集，希望建立一个能够准确预测房屋价格的模型。在这个背景下，线性回归模型是一个常用的选择，它通过拟合一条直线来描述房屋面积与价格之间的关系。

使用 Python 进行这样的建模过程十分便捷：首先，我们可以利用 Pandas 库加载和处理数据。接着，通过 Statsmodels 库中的线性回归模型，我们能够构建一个数学模型，其中房屋面积充当自变量，价格为因变量，最终结果如图 1-10 所示。

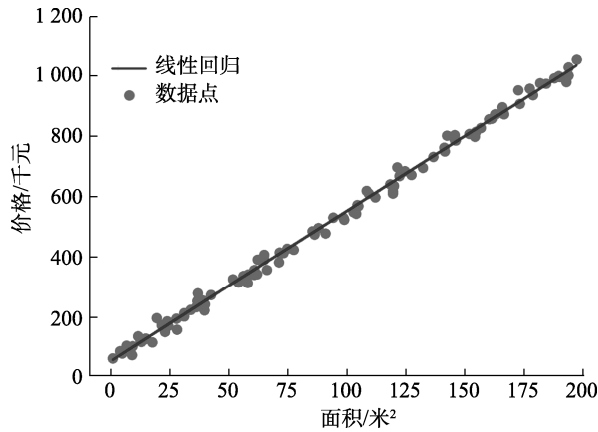


图 1-10 统计建模——回归分析图

本章小结

本章为读者提供了 Python 数据分析的入门知识。以下是本章的主要内容和要点：

（1）Python 的简洁语法：介绍了 Python 语法的易学性，包括其清晰简洁的代码结构、强调可读性、使用缩进代替大括号等特点。

（2）Python 的可视化功能：讨论了 Python 在数据可视化方面的强大功能，以及它是如何支持数据分析和展示的。

（3）Python 的多用途性：概述了 Python 作为编程语言的多样性，包括 Web 开发、数据科学、人工智能、自动化脚本等。

（4）Python 的生态系统与开源社区：强调了 Python 生态系统中的工具和库，如 Scipy、Matplotlib、TensorFlow 等，以及开源社区对 Python 发展的贡献。

（5）Python 的历史和发展：回顾了 Python 的起源以及 Python 的发展趋势。

（6）数据分析的基本概念：定义了数据分析的概念，即通过统计分析方法处理数据、提取信息，并形成结论。

（7）数据分析的基本步骤：概述了数据分析的基本步骤，包括问题定义、制定统计假设、数据收集、数据预处理、描述性统计分析、统计建模以及验证与解释。

（8）数据分析的应用领域：讨论了数据分析在决策支持、业务优化、市场竞争、客户洞察和统计建模方面的应用。