教案 1　数据可视化导论

教学目标

知识目标

- 了解什么是数据可视化
- 熟悉数据可视化的方式，可以选择正确的数据可视化图表
- 了解常见的数据可视化工具
- 认识 matplotlib，能够在 Python 环境中安装 matplotlib
- 了解数据可视化常用工具的基本用法

能力目标

- 培养学生的实际操作能力
- 帮助学生理解数据可视化工具之间的相通性，形成解决实际应用问题的方法能力
- 提高学生对新兴的数据可视化技术有较高的敏锐性

素质目标

- 具有良好的团队意识，优秀的合作、协调、沟通能力
- 性格开朗外向，善于和工作伙伴和睦相处
- 有强烈责任心，肯吃苦耐劳，办事麻利，做事认真仔细、负责
- 责任心强、认真度高、吃苦耐劳
- 为人诚实，工作勤奋

任务 1 认识数据可视化技术

一、数据可视化的定义

数据可视化，顾名思义，就是将数据转换成图或表等，以一种更直观的方式展现和呈现数据，让读者能“一眼看懂”你想表达的信息。 通过“可视化”的方式，复杂的数据通过图形化的手段进行有效表达，准确高效、简洁全面地传递某种信息，甚至我们帮助发现某种规律和特征，挖掘数据背后的价值。

数据可视化是数据分析工作中重要的一环，对数据潜在价值的挖掘有着深远的影响。为了让读者直观地看出文字数据与图形数据之间的差异。

二、数据可视化的作用

在表达简洁信息上素来有“文不如表，表不如图”或“一图胜千言”的说法、数据可视化改变了传统的文字描述信息的模式，转而使用视觉语言更高效地传送重要信息和描述重要细节。数据可视化主要有以下几方面的作用。

1. 化繁为简，实现可视化
2. 更快发现新趋势、新机遇
3. 有效增强数据交互性
4. 复杂信息易理解
5. 数据多维度显示
6. 直观展示图
7. 突破记忆限制

任务 2 了解数据可视化的理论基础

一、数据可视化的流程

1. 数据采集

数据采集是数据分析和可视化的第一步，在可视化设计过程中一定要先了解数据的来源、采集方法和数据属性，这样才有助于更好地解决后续的问题。

2. 数据处理和变换

数据处理和变换，是进行数据可视化的前提条件，主要包括数据预处理和数据挖掘两个过程。

进行数据预处理的原因是，通常前期的数据采集得到的数据，不可避免地会含有噪声和误差，数据质量较低，所以需要做数据预处理前期采集到的数据往往包含了噪声和误差，数据的质量较低。

数据挖掘则是因为数据的特征、模式往往隐藏在海量的数据中，需要进行更深一步的数据挖掘才能获取到。

3.可视化映射

将数据进行清洗、去噪，并按照业务目的进行数据处理之后，就可以进行可视化映射环节了。可视化映射是整个数据可视化流程的核心，是指将处理后的数据信息映射成可视化元素的过程。其主要目的是让用户通过可视化结果理解数据信息以及数据背后隐含的规律。

4.用户感知

在数据可视化的过程中要进行组织和筛选。可视化结果不仅可以通过可视化图表让用户被动感知信息，也可以提供交互方式让用户主动获取信息，交互是通过可视化的手段辅助分析决策的直接推动力。只有当用户感知到数据传达出的信息时，才能说明数据可视化有一定的效果，但如何让用户更好地感知信息，是一个复杂的问题，需要不断总结。

二、数据可视化的基础图表

图表是数据可视化最基础的应用，它代表图形化的数据通常以所用的图形符号命名，例如使用圆形符号的饼图、使用线条符号的折线图等。

1. 折线图

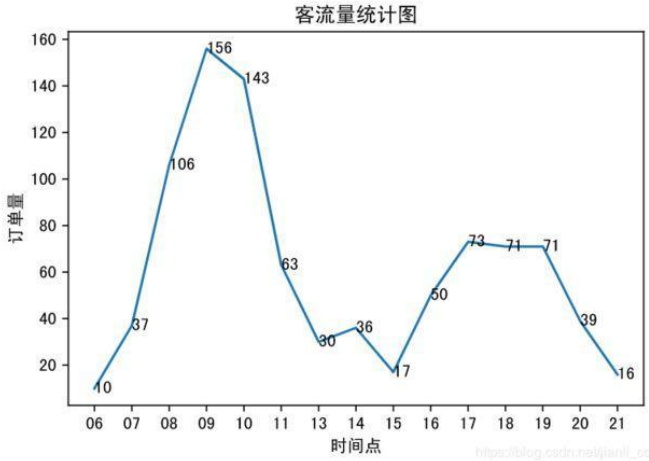

图 1-15 客流量统计图

2. 柱形图

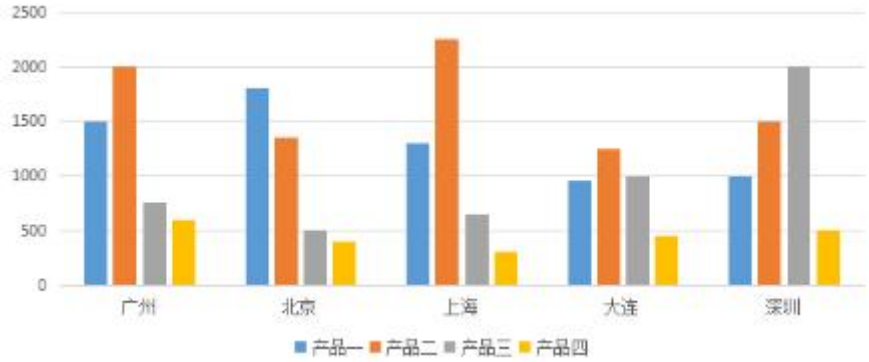

图 1-16 各类产品在不同城市的受欢迎程度

3. 条形图

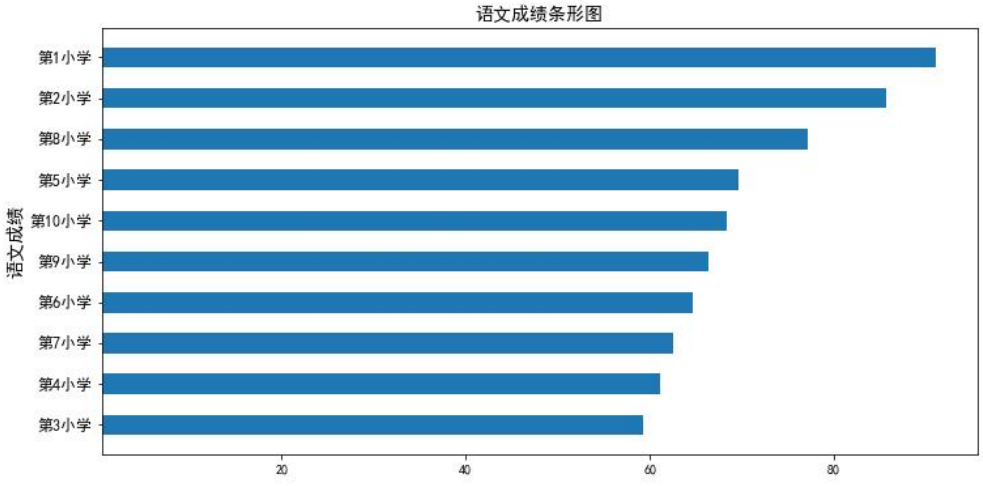

图 1-17 成绩统计图

4. 堆积图

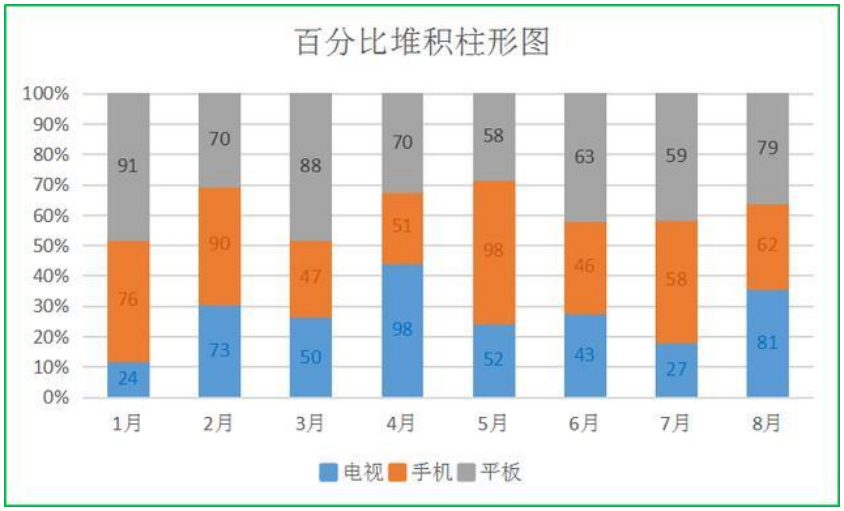

图 1-18 各电子设备销售百分比

5. 直方图

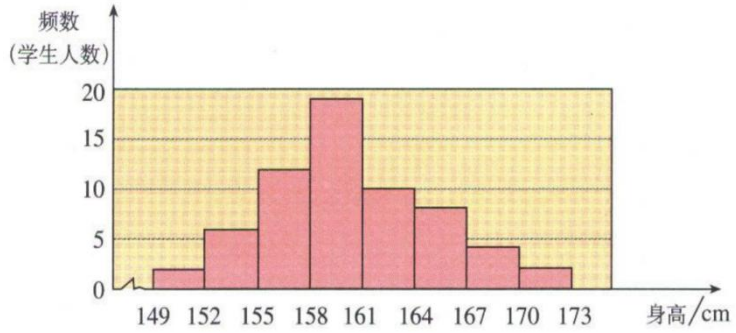

图 1-19 学生身高统计图

6. 箱形图

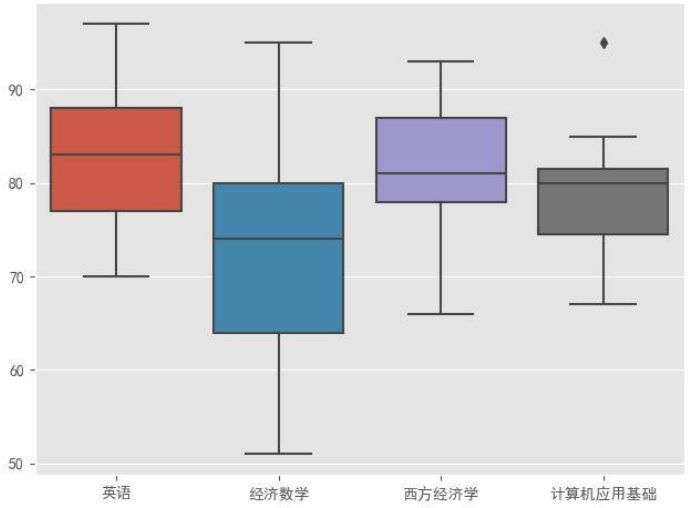

图 1-20 学生成绩分布情况

7. 饼图

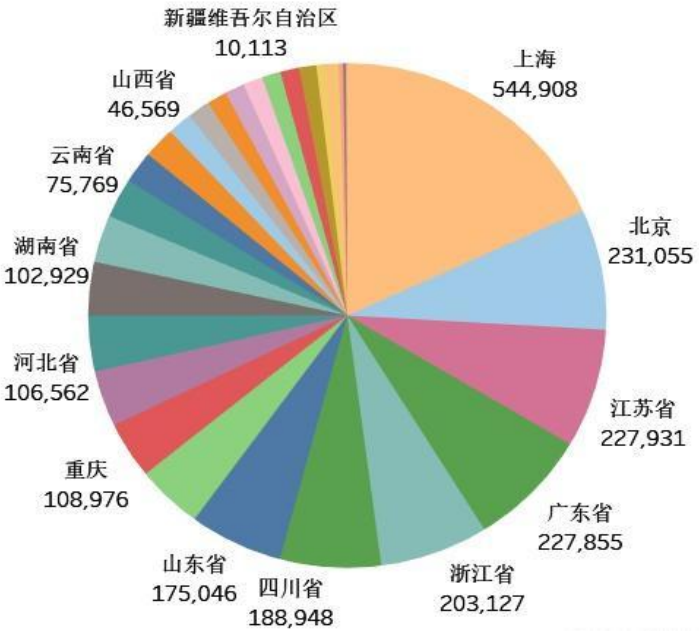

图 1-21 全国各省居民的人均收入情况

8. 散点图

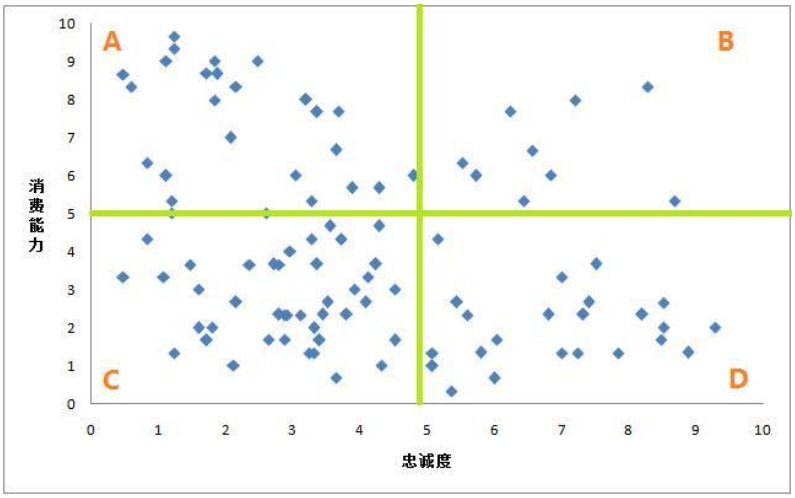

图 1-22 顾客消费能力和忠诚度关系分析图

9. 气泡图

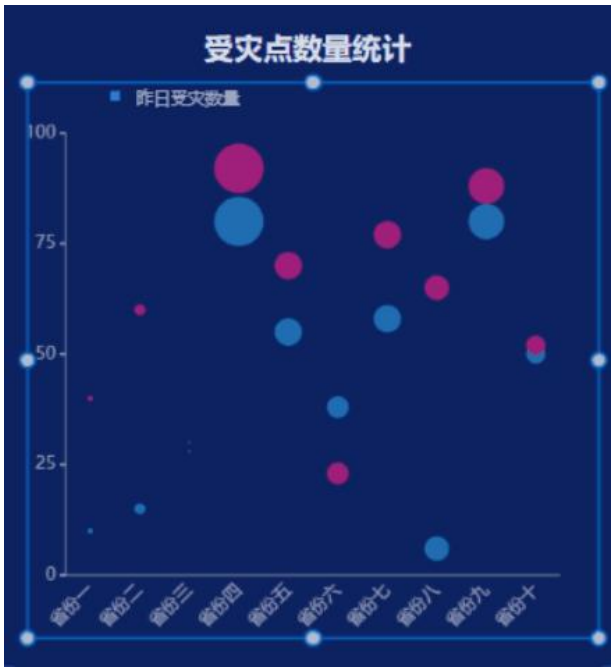

图 1-23 受灾点分布图

10. 雷达图

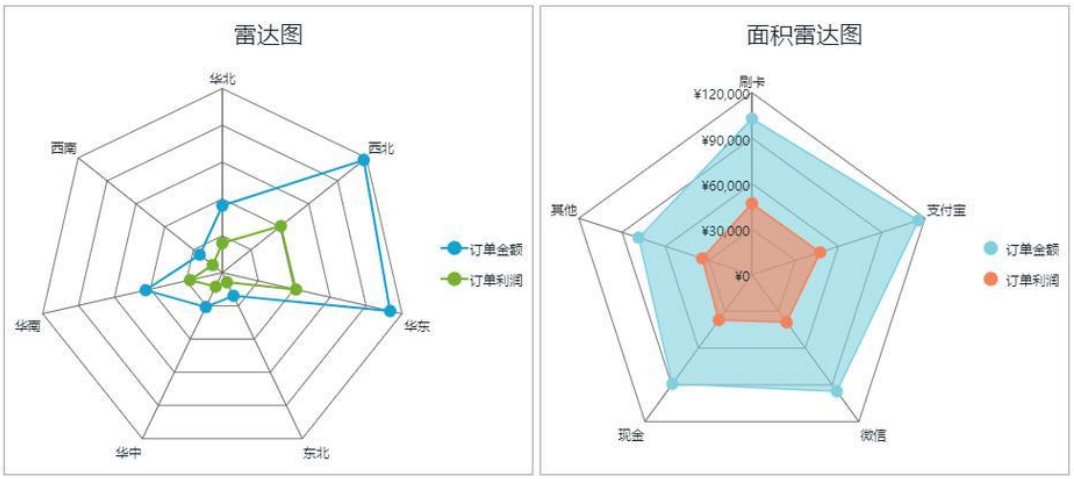

图 1-24 支付方式统计图

11. 3D 图表

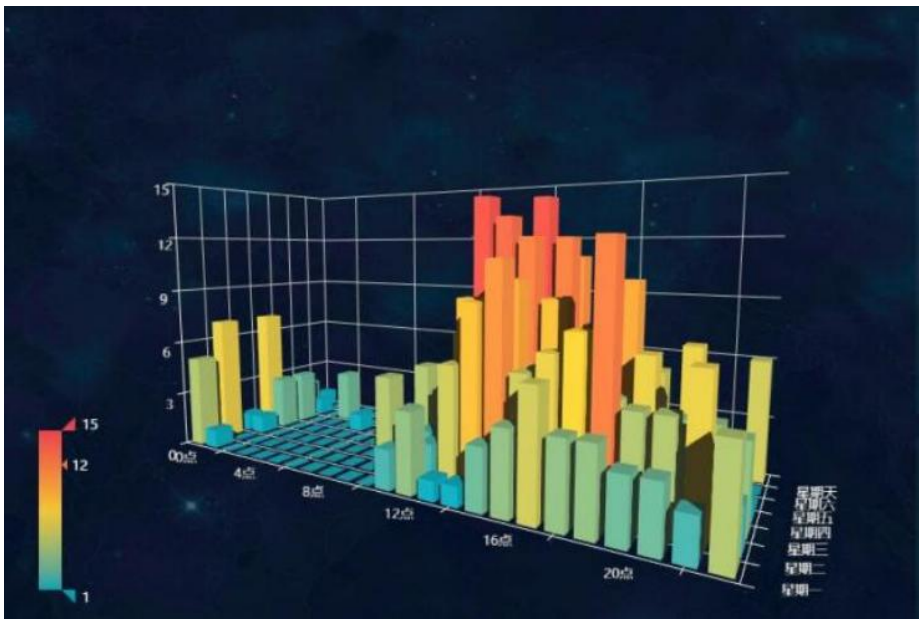

图 1-25 睡眠时间波动三维图

任务 3 数据读取和处理

在 Python 中，使用 pandas 库可以方便地从多种外部文件格式（如 CSV、Excel 和数据库）导入数据，并对其进行必要的预处理，包括校验数据的完整性和一致性、清洗重复和异常值，以及合并多个数据集。这些步骤对于确保数据质量并为后续的数据分析和建模奠定坚实基础至关重要。

一、读取数据

数据的读取是进行数据预处理、数据建模和分析的基础。对于不同的数据文件，pandas 提供了不同函数进行读取。常见的数据文件格式有 3 种，分别是 CSV 文件、Excel 文件和数据库。

1. 读取 CSV 文件数据

在 Python 中读取 CSV 文件可以使用多种方法，包括内置的 csv 模块和第三方库如 pandas。

read_csv 函数的方法定义：

```
pandas.read_csv(filepath_or_buffer, sep='\t' , delimiter=None, header='infer', names=None, index_col=None,usecols=None, dtype=None)
```

参数说明

- filepath_or_buffer: 文件路径或 URL。
- sep: 指定分隔符，默认为逗号。
- header: 指定哪一行作为列名，默认为 0（第一行）。
- names: 如果文件中没有标题行，可以使用这个参数指定列名。
- index_col: 指定哪一列作为索引。
- usecols: 指定要读取的列。
- dtype: 指定列的数据类型。

2. 写入 CSV 文件

使用 pandas 的 to_csv 方法可以方便地将 DataFrame 写入 CSV 文件：

```
import pandas as pd
```

```
def write_to_csv_with_pandas(filename, data):
    """
    将数据写入 CSV 文件。
    参数:
    filename (str): 要写入的 CSV 文件名。
    data (dict or list of dict): 要写入的数据，可以是字典或字典列表。
    """
    df = pd.DataFrame(data)
    df.to_csv(filename, index=False, encoding='utf-8')
# 示例数据
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [30, 25, 35],
    'City': ['New York', 'Los Angeles', 'Chicago']
}
# 调用函数写入数据到 CSV 文件
write_to_csv_with_pandas('output_pandas.csv', data)
```

我们可以看到使用 csv 模块和 pandas 库都可以方便地读写 CSV 文件。csv 模块适用于简单的读写操作，而 pandas 库则提供了更强大的数据处理功能，适用于复杂的数据分析任务。根据具体需求选择合适的方法，可以大大简化数据处理流程。

二、处理数据

数据处理是数据分析和可视化过程中至关重要的一步。它涉及多个步骤，包括数据校验、清洗、转换和集成等，旨在提高数据的质量和可靠性，从而为后续的分析和可视化提供坚实的基础。

任务 4 数据可视化案例初体验

一、matplotlib 绘图库

matplotlib 作为 Python 及其科学计算库 NumPy 的第三方绘图软件包，具有设计与数字化品质高、适合科学出版等优点。matplotlib 不止是一个数学绘图库，它也是可视化和分析工具中最流行之一。开发者仅需简单的代码就可以生成折线图、直方图、散点图、条形图、饼图等。在使用 matplotlib 绘制图形的过程中，一般还会用到 NumPy、pandas 等第三方工具包。

二、seaborn 绘图库

seaborn 是一个基于 matplotib 旦数据结构与 pandas 统一的统计图制作库。该库提前已经定义好了一套自己的风格，然后也封装了一系列的方便的绘图函数。

Seabn 是一个在 Python 中制作有吸引力和丰富信息的统计图形的库。它构建在 MatPultLB 的顶部，与 PyDATA 栈紧密集成，包括对 SIMPY 和 BANDA 数据结构的支持以及 SISPY 和 STATSMODEL 的统计例程。

Seaborn 其实是在 matplotib 的基础上进行了更高级的 API 封装，从而使得作图更加容易。在大多数情况下使用 seaborn 就能做出很具有吸引力的图，而使用 matplotib 就能制作具有更多特色的图，可以把 Seaborn 视为 matplotib 的补充。Seabn 是基于 MatPultlLB 的 Python

可视化库。它为绘制有吸引力的统计图形提供了一个高级接口。

三、 使用 pyecharts 绘图库

ECharts 是一个纯 Javascript 的图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器，底层依赖轻量级的 Canvas 类库 ZRender，提供直观、生动、可交互、可高度个性化定制的数据可视化图表。ECharts 提供了常规的折线图、柱状图、散点图、饼图、K 线图，用于统计的盒形图，用于地理数据可视化的地图、热力图、线图，用于关系数据可视化的关系图、treemap，多维数据可视化的平行坐标，还有用于 BI 的漏斗图、仪表盘，并且支持图与图之间的混搭。

pyecharts 是一个用于生成 Echarts 图表的类库。实际上就是 Echarts 与 Python 的对接。使用 pyecharts 可以生成独立的网页，也可以在 flask，Django 中集成使用。

Echarts 是一个由百度开源的数据可视化，凭借着良好的交互性，精巧的图表设计，得到了众多开发者的认可。Python 是一门富有表达力的语言，很适合用于数据处理。为了方便开发者使用 python 语言对数据进行可视化分析，便开发了 pyecharts 库。