

# 第 3 章



## 时间数据可视化

每一个数据都是带有时间的,只不过在特定的情况下会把时间忽略掉,只关注扁平的数据。在大数据时代,随着数据处理能力的增强和处理方法的增多,时序大数据越来越受到重视。本章主要介绍时间数据在大数据中的应用以及对应的图形表示方法。

本章主要分为两部分:连续型时间数据可视化,主要有阶梯图、折线图、拟合曲线;以及离散时间数据的处理,主要有散点图、柱形图、堆叠柱形图。



第 3 章

### 3.1 时间数据在大数据中的应用

对于数据来说,时间是一个非常重要的维度和属性。历史数据的积累是大数据“大”的一个重要原因。时间序列数据存在于各个领域,如金融和商业交易记录、社会经济指标记录、气象观测数据、动植物种群数据等。金融和商业记录包括股票交易价格以及交易量,各种商品的销售价格和销售量;社会经济指标包括 GDP(国内生产总值)、CPI(消费者物价指数)等指数。这些带时间维度的数据中蕴含着大量的信息,是指导国家制定政策、企业调整战略的重要依据。

时间数据有离散和连续两种,无论哪种数据的可视化,最重要的目的都是从中发现数据随时间变化的趋势。具体表现在:什么保持不变?什么发生改变?改变的数据是上升还是下降?改变的原因是什么?不同数据随时间变化的方向是否一致?它们变化的幅度是否有关联?是否存在周期性的循环?这些变化中存在的模式超脱于某个时刻,蕴含着丰富的信息,只有依靠在时间维度的观察分析才能被发现。

## 3.2 连续型时间数据可视化

连续型时间数据在任意两个时间点之间可以细分出无限多个数值,它是连续不断变化现象的记录。温度是人们最常接触的时间数据,一天内任意一个时刻的温度都可以被测量到。另外,股票的实时价格也可以近似看作连续型时间数据。下面给出几个连续型时间数据的可视化示例。

### 3.2.1 阶梯图

阶梯图是 X-Y 图的一种,通常用于 Y 值发生离散的改变,且在某个特定的 X 值位置发生了一个突然的变化。阶梯图可以用无规律、间歇阶跃的方式表达数值随时间的变化。如银行利率就可以用阶梯图表示:银行利率一般在较长时间内保持不变,由银行选择在特定时间节点进行调整。阶梯图的基本框架如图 3-1 所示。

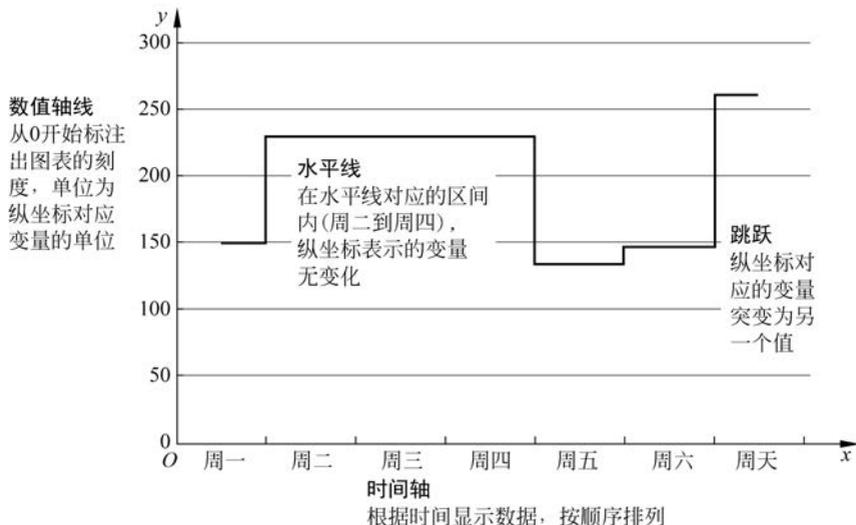


图 3-1 阶梯图的基本框架

### 3.2.2 折线图

折线图是用直线段将各数据点连接起来而组成的图形,以折线方式显示数据的变化趋势。在折线图中,沿水平轴均匀分布的是时间,沿垂直轴均匀分布的是数值。折线图比较适用于表现趋势,常用于展现如人口增长趋势、书籍销售量、粉丝增长进度等时间数据。这种图表类型的基本框架如图 3-2 所示。

从图 3-2 可以看出数据变化的整体趋势。注意,横轴长度会影响展现的曲线趋势,若图中的横轴过长,点与点之间分隔的间距比较大,则会使得整个曲线非常夸张;若横轴过短,则用户又有可能看不出数据的变化趋势。所以合理地设置横轴的长度十分重要。

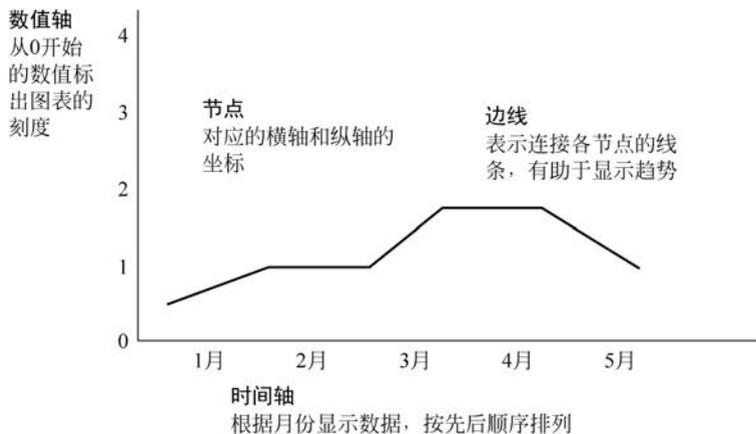


图 3-2 折线图的基本框架

### 3.2.3 螺旋图

也称为时间系列螺旋图。螺旋图沿阿基米德螺旋线画上基于时间的数据；图表从螺旋形的中心点开始往外发展；十分多变,可使用条形、线条或数据点；沿着螺旋路径显示。适合用来显示大型数据集,通常显示长时间段内的数据趋势,因此能有效显示周期性的模式。螺旋图的基本框架如图 3-3 所示。



图 3-3 螺旋图的基本框架

### 3.2.4 热图

热图通过色彩变化来显示数据,当应用于表格时,热图适合用来交叉检查多变量的数据。热图不局限于时间数据的可视化,适用于显示多个变量之间的差异,显示是否有彼此相似的变量以及彼此之间是否有相关性。由于热图依赖颜色来表达数值,难以提取特定数据点或准确指出色块间的差异。如图 3-4 所示是热图的一个示例。

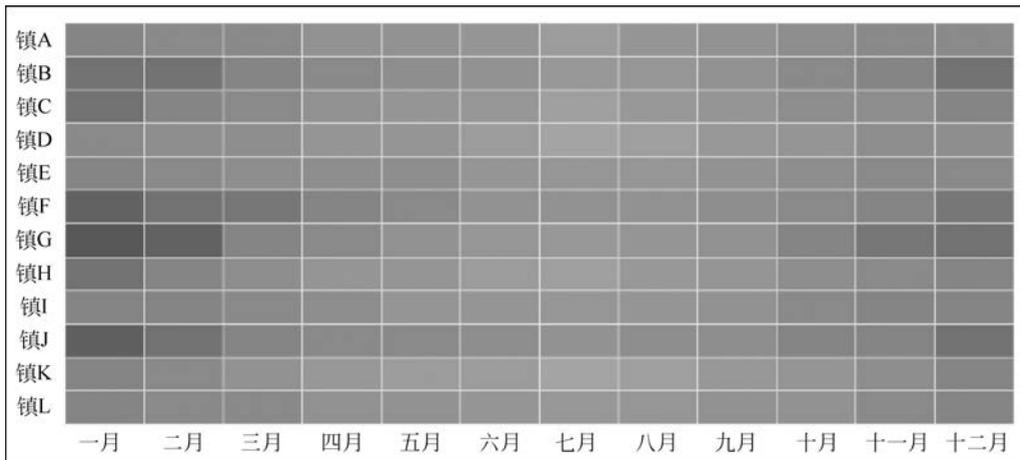


图 3-4 热图示例：镇 A~镇 L 十二个月温度数据

### 3.3 离散型时间数据可视化

离散型时间数据又称为不连续性时间数据，这类数据在任何两个时间点之间的个数是有限的。在离散型时间数据中，数据来自某个具体的时间点或者时段，可能的数值也是有限的。例如，每届奥运会奖牌的总数或者是各个国家的金牌数就是离散数据，各资格考试每年的通过率也是离散型数据。类似的生活实例有很多，下面将介绍如何对这些离散型时间数据进行可视化处理。

#### 3.3.1 散点图

散点图是指在数理统计回归分析中，数据点在直角坐标系平面上的分布图。散点图表示因变量随自变量而变化的趋势，由此趋势可以选择合适的函数进行经验分布的拟合，进而找到变量之间的函数关系。对于离散时间数据，水平轴表示时间，垂直轴表示对应的数值。散点图的基本框架如图 3-5 所示。

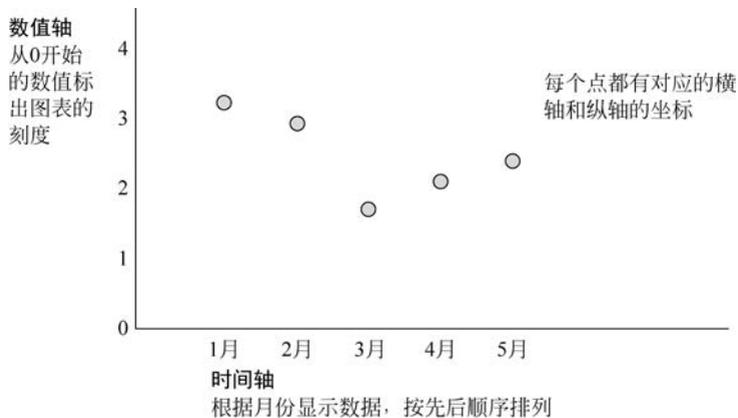


图 3-5 散点图的基本框架

### 3.3.2 柱形图

柱形图又称条形图、直方图,是以高度或长度的差异来显示统计指标数值的一种图形。柱形图简明、醒目,是一种常用的统计图形。如图 3-6 所示为其基本框架。

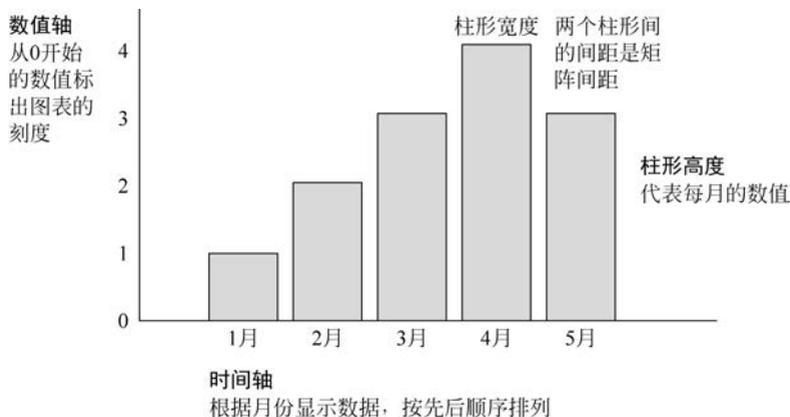


图 3-6 柱形图的基本框架

柱形图一般用于显示一段时间内的数据变化或显示各项之间的比较情况。另外,数值的体现就是柱形的高度。柱形越矮则数值越小,柱形越高则数值越大。另外需要注意的是,柱形的高度与相邻柱形间的间距决定了整个柱形图的视觉效果的美观程度。如果柱形的宽度小于间距,则会使读者的注意力集中在空白处而忽略了数据,所以合理地选择宽度很重要。

### 3.3.3 堆叠柱形图

堆叠柱形图是普通柱形图的变体,堆叠柱形图会在一个柱形上叠加一个或多个其他柱形,一般它们具有不同的颜色。若数据存在子分类,并且这些子分类相加有意义的话,则可以使用堆叠柱形图来表示。堆叠柱形图的基本框架如图 3-7 所示。

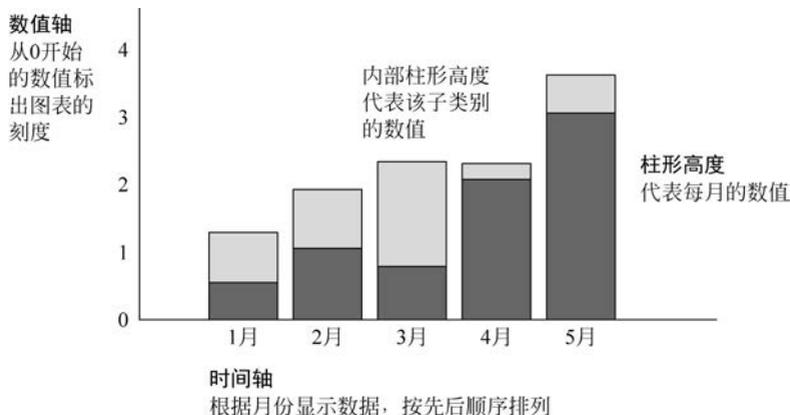


图 3-7 堆叠柱形图的基本框架

### 3.3.4 点线图

点线图是离散型数据可视化的一种形式。可以说点线图是柱形图的一种变形,但更令人聚焦到端点。如图 3-8 所示是点线图的一个一般示例。

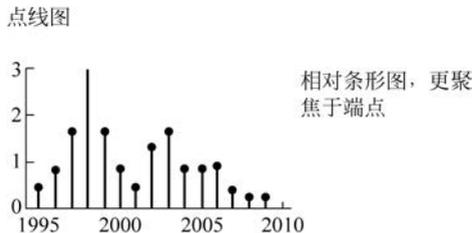


图 3-8 点线图的一般示例

股市中有一种特殊的点线图。一条线表示一个交易时段,一个点表示收市价,线高低点表示最高价及最低价,如图 3-9 所示。可以让投资者了解市价与当时交易时段高低价的关系,代表市场气氛倾向乐观或悲观。

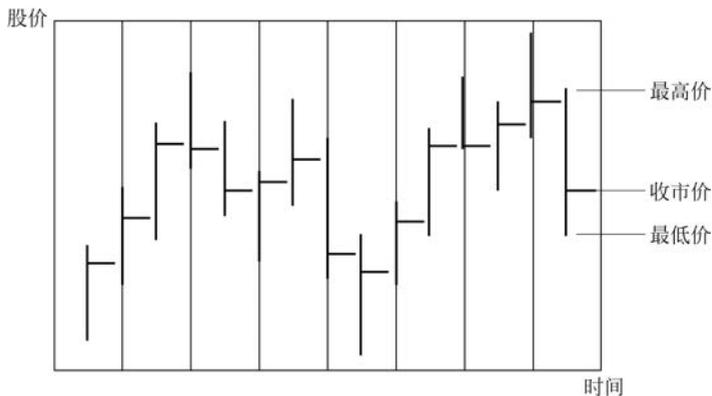


图 3-9 股市中的点线图

## 3.4 可视化图表的选择

经过前几节的学习,可以总结出时间数据的可视化涉及三个维度:表达、比例和布局。

### 1. 表达维度

表达维度有以下几种。

(1) 线性的、以典型的阅读方式呈现时间内容;将时间数据作为二维的线图显示;X轴表示时间、Y轴表示其他的变量。

(2) 径向将时间序列编码为弧形;沿圆周排列;适合呈现周期性的时变型数据,如螺旋图。

- (3) 网格和日历相对应；一般采用表格映射的方式。
- (4) 螺旋表达可用条形、线条或数据点，沿着螺旋路径显示。
- (5) 随机两类：一是基于排版形式的随机；二是时间曲线的随机，使相似的时间点彼此接近。

这些维度的直观表示如图 3-10 所示。

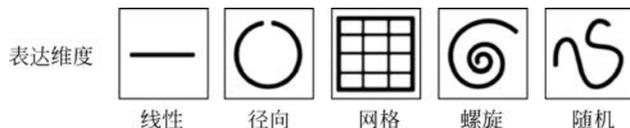


图 3-10 表达维度的直观表示

## 2. 比例维度

- (1) 比例维度(按时间顺序)可以被用来表示事件的顺序,或者事件的持续时间。
- (2) 相对顺序是指存在一个基线事件在时间零点,可以被用在多时间线的对比。
- (3) 对数的比例从按时间的前后顺序排列的比例转换而来,强调了最早的或最近的事件,对数比例适用于长范围或不均匀的事件布局。
- (4) 次序比例中连续事件之间的距离是相等的,只表达事件的顺序。
- (5) 次序+中间时长可以用来表示长时间和不均匀分布的事件。

比例维度的直观表示如图 3-11 所示。

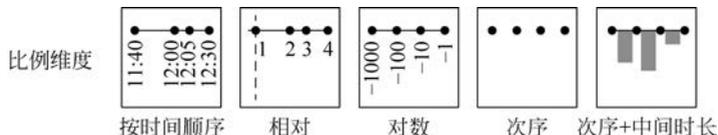


图 3-11 比例维度的直观表示

## 3. 布局维度

- (1) 单一时间线。
- (2) 多个时间线。
- (3) 分段时间线,在这种形式中,一个时间线被有意义地进行划分,进行另一种形式的比较。
- (4) 多个时间线加上分段时间线,指不同属性时间线加上分割的时间段,可以进行多种形式的比较。

布局维度的直观表示如图 3-12 所示。

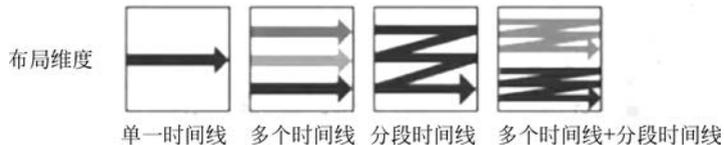


图 3-12 布局维度的直观表示

要根据数据的特点来确定合适的维度。例如,时间数据有周期性:一天中的时间,一周中的每一天以及一年中的每个月都在周而复始,对齐这些时间段通常是有好处的。有时需要看到坡度或者点之间的变化率,而用连续的线时,会更容易看到坡度;用散点图,数据和坐标轴一样,但视觉暗示与连续的线不同。散点图的重点在每个数值上,趋势不是那么明显。

时间数据还可能有循环性。很多事情都是在规律性地重复着。因为数据在重复,所以比较每周同一天的数据就有了意义。例如,比较每一个星期一的情况。把时间可视化成连续的线或循环有些困难,但是可以把日子按每周分成段,这样就能直接比较循环情况了。循环中的异常数据也可以被发现,如图 3-13 所示。

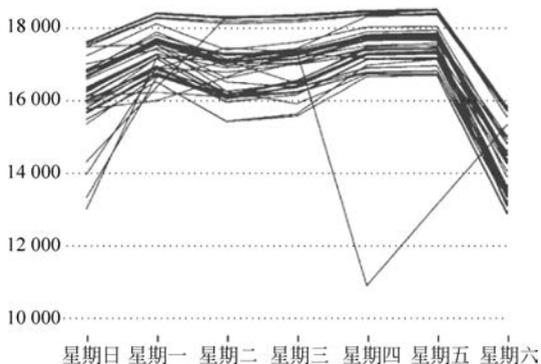


图 3-13 循环中的异常数据

有时也可以将不同的图结合起来进行创新,如图 3-14 所示,将热图和螺旋图结合起来显示一年的气象变化,直观、美观,而且体现出循环性。

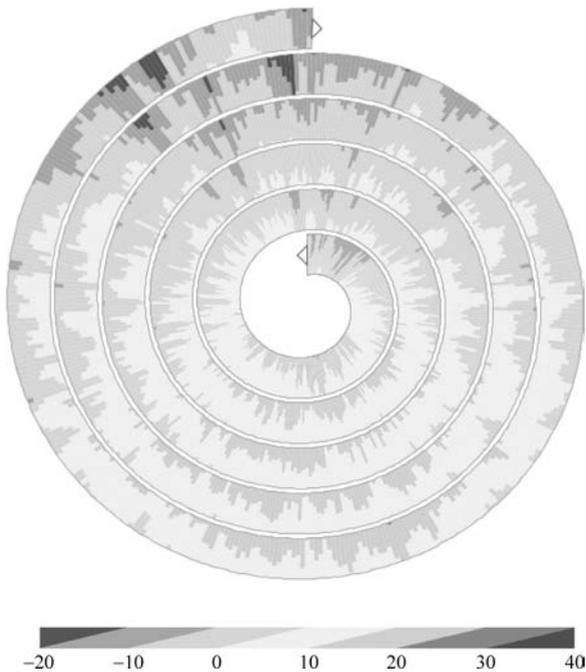


图 3-14 热图与螺旋图结合展示气象数据

## 习 题

1. 举例说明学习或生活中时间数据的应用。
2. 连续型时间数据和离散型时间数据有何区别？
3. 除了书中提到的,还有哪些时间数据可视化的例子？