

第3章 大数据可视化方法

本章学习目标

- 了解可视化图的分类。
- 了解统计图表的分类。
- 了解文本可视化的概念和流程。
- 掌握词云的实现。
- 了解网络可视化的概念。
- 掌握社交网络图的实现。
- 了解空间信息可视化的概念。

本章先向读者介绍大数据可视化图的分类,再介绍文本可视化的概念和流程,接着介绍词云的制作及社交网络图的实现,最后介绍空间信息可视化的概念。

3.1 可视化图表介绍

3.1.1 统计图表介绍

图是表达数据最直观、最强大的方式之一,通过图的展示能够对数据进行变换,从而让枯燥的数字能吸引人们的注意力。在实现数据可视化图时应当考虑有什么数据,需要用图做什么,该如何展示数据。

在统计图表的每一种类型的图表中都可包含不同的数据可视化图,例如柱状图、K线图、饼图、折线图、散点图、气泡图、雷达图、面积图、漏斗图、和弦图、环形图、直方图、热力图、仪表盘图以及密度图等。

1. 柱状图

柱状图使用垂直或水平的柱子显示类别之间的数值比较。在柱状图中一个轴表示需要对比的分类维度,另一个轴代表相应的数值。柱状图又可分为纵向柱状图和横向(或水平)柱状图。图 3-1 显示了纵向柱状图;图 3-2 显示了横向柱状图。

2. K线图

K线图又称阴阳图、棒线、红黑线或蜡烛线,常用于展示股票交易数据。K线就是指将各种股票每日、每周、每月的开盘价、收盘价、最高价、最低价等涨跌变化状况用图形的方式表现出来。图 3-3 显示了 K 线图。

扫一扫



视频讲解

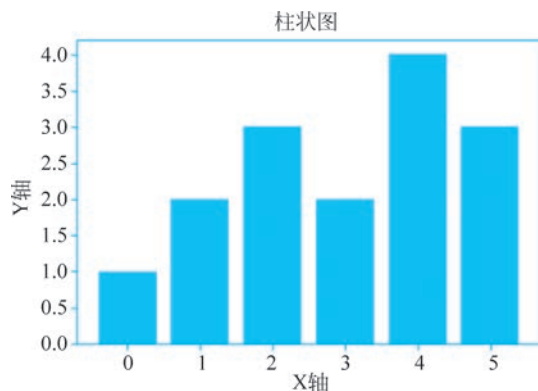


图 3-1 纵向柱状图

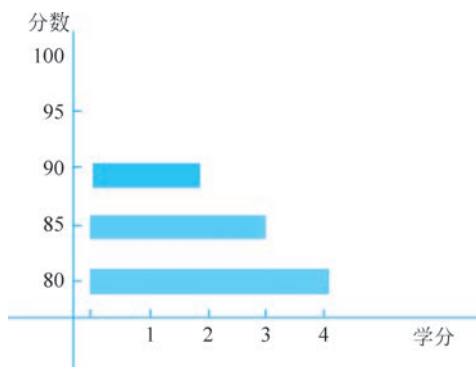


图 3-2 横向柱状图



图 3-3 K线图

3. 饼图

饼图用于表示不同分类的占比情况，通过弧度大小来对比各种分类。饼图通过将一个圆饼按照分类的占比划分成多个区块，整个圆饼代表数据的总量，每个区块（圆弧）表示该分类占总体的比例大小，所有区块（圆弧）的和等于 100%。图 3-4 显示了饼图。

4. 折线图

折线图用于显示数据在一个连续的时间间隔或者时间跨度上的变化，它的特点是反映事物随时间或有序类别变化的趋势。在折线图中，数据是递增还是递减、增减的速率、增减的规律（周期性、螺旋性等）、峰值等特征都可以清晰地反映出来。图 3-5 显示了折线图。

5. 散点图

散点图是指在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变化的大致趋势，据此可以选择合适的函数对数据点进行拟合。图 3-6 显示了散点图。

6. 气泡图

气泡图是一种多变量图表，是散点图的变体，也可以认为是散点图和百分比区域图的组合。图 3-7 显示了气泡图。

收入行业占比

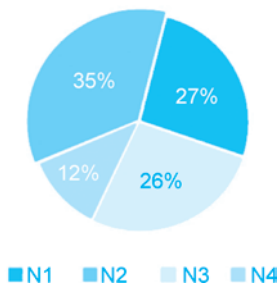
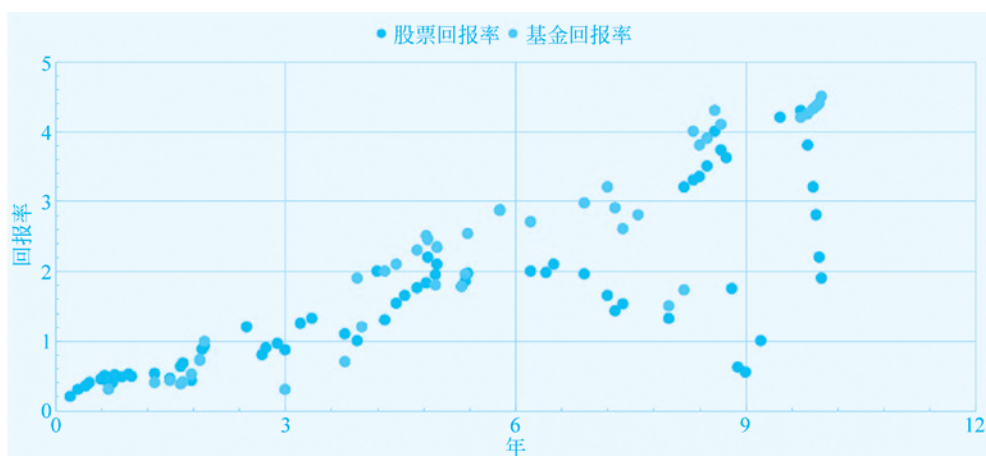
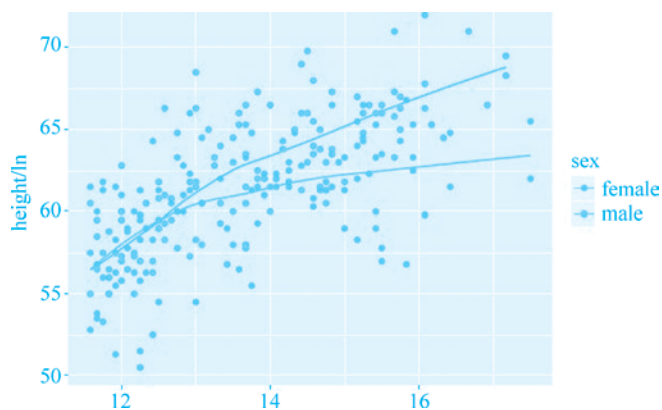
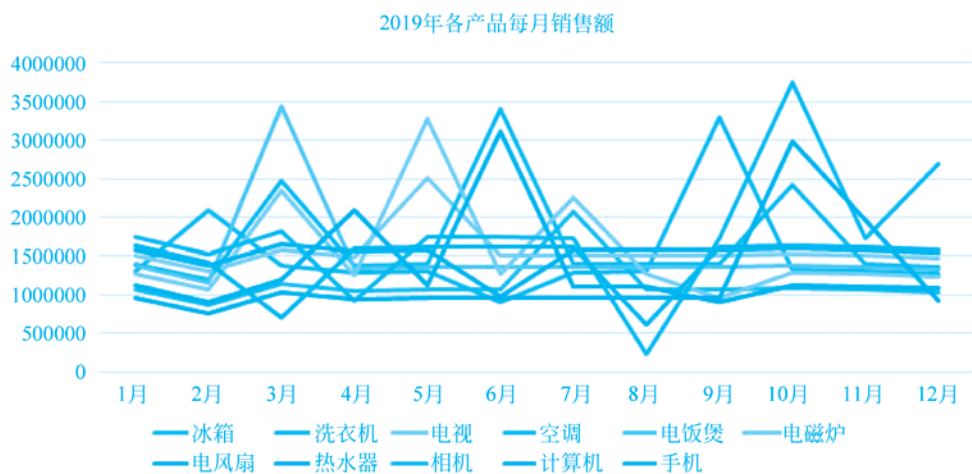


图 3-4 饼图



7. 雷达图

雷达图又叫戴布拉图、蜘蛛网图。传统的雷达图被认为是一种表现多维(4 维及以上)

数据的图表。它将多个维度的数据量映射到坐标轴上,这些坐标轴起始于同一个圆心点,通常结束于圆周边缘,将同一组的点使用线连接起来就成为雷达图。图 3-8 显示了雷达图。

8. 面积图

面积图又叫区域图。它是在折线图的基础之上形成的,它将折线图中折线与自变量坐标轴之间的区域使用颜色或者纹理填充,填充区域称为面积,颜色的填充可以更好地突出趋势信息。图 3-9 显示了面积图。

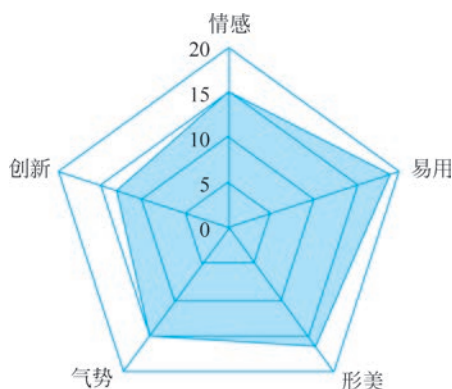


图 3-8 雷达图

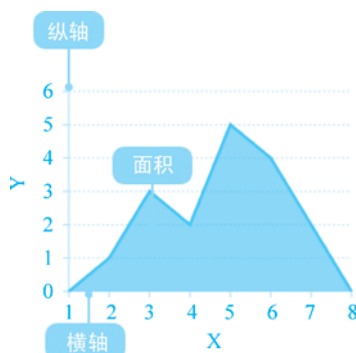


图 3-9 面积图

9. 漏斗图

漏斗图适用于业务流程比较规范、周期长、环节多的单流程单向分析,通过漏斗各环节业务数据的比较能够直观地发现和说明问题所在的环节,进而做出决策。漏斗图从上到下有逻辑上的顺序关系,表现了随着业务流程的推进业务目标完成的情况。图 3-10 显示了漏斗图。

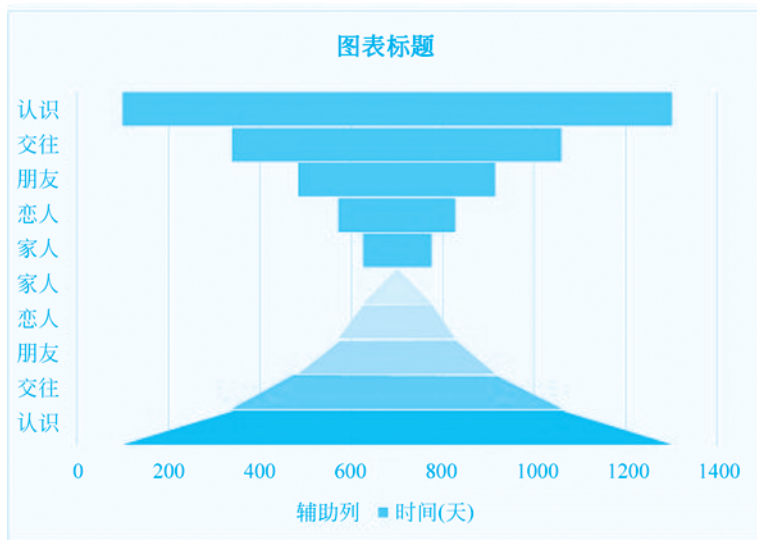


图 3-10 漏斗图

10. 和弦图

和弦图是一种显示矩阵中数据间相互关系的可视化方法,在图中节点数据沿圆周径向排列,节点之间使用带权重(有宽度)的弧线连接。图 3-11 显示了和弦图。

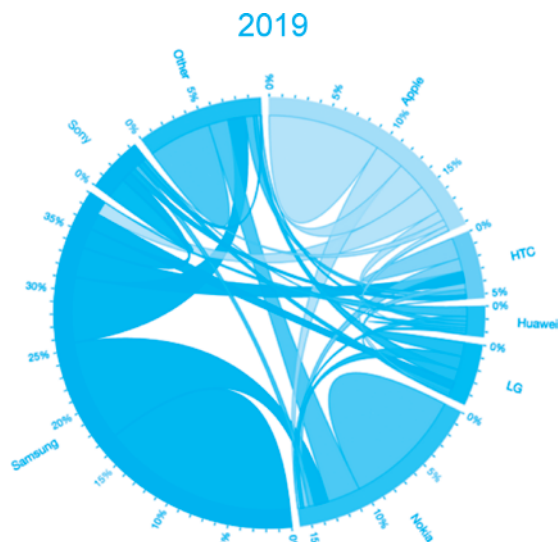


图 3-11 和弦图

11. 环形图

环形图是由两个及两个以上大小不一的饼图叠在一起,挖去中间部分所构成的图形,主要用于区分或表明某种关系。图 3-12 显示了环形图。

12. 直方图

直方图的形状类似于柱状图,却有着与柱状图完全不同的含义。直方图涉及统计学的概念,首先要对数据进行分组,然后统计每个分组内数据元的数量。图 3-13 显示了直方图。

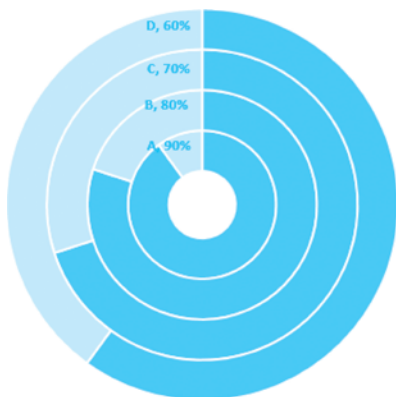


图 3-12 环形图

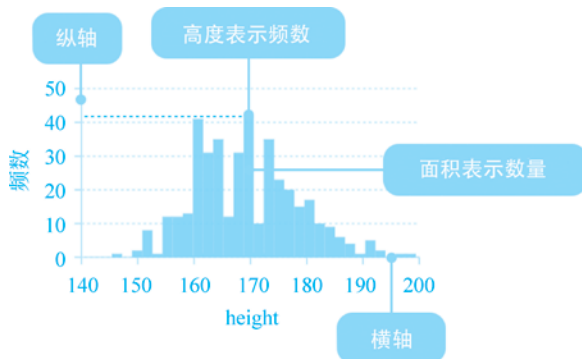


图 3-13 直方图

13. 热力图

热力图以特殊高亮的形式显示访客热衷的页面区域和访客所在的地理区域的图示,热力图可以显示不可点击区域发生的事情。图 3-14 显示了热力图。

14. 仪表盘图

仪表盘图是一种拟物化的图表,刻度表示度量,指针表示维度,指针角度表示数值。仪表盘图就像汽车的速度表一样,有一个圆形的表盘及相应的刻度,有一个指针指向当前数

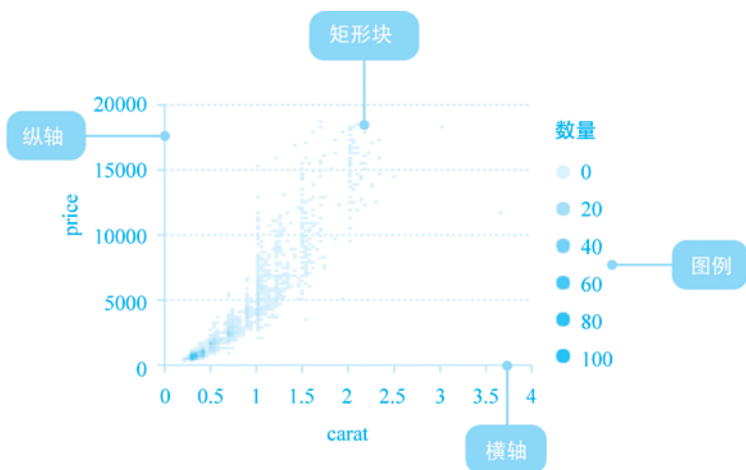


图 3-14 热力图

值。目前很多的管理报表或报告上都使用这种图表,以直观地表现出某个指标的进度或实际情况。图 3-15 显示了仪表盘图。

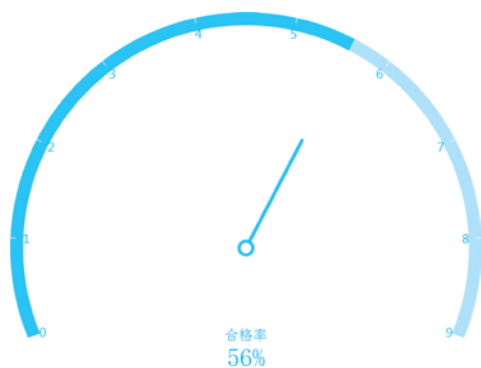


图 3-15 仪表盘图

15. 密度图

密度图用于显示数据在连续时间段内的分布状况。这种图表是直方图的变种,使用平滑曲线来绘制数值水平,从而得出更平滑的分布。密度图的峰值显示数值在该时间段内高度集中的位置。图 3-16 显示了密度图。

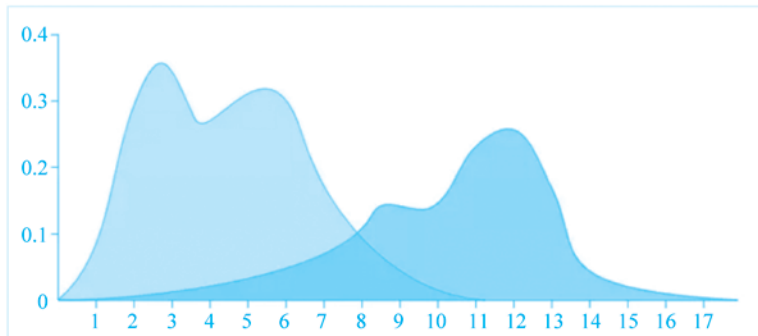


图 3-16 密度图

【例 3-1】 使用 Python 绘制折线图。

该例的代码如下：

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']      # 设置字体
game = ['1月', '2月', '3月', '4月', '5月', '6月']
scores = [23, 10, 38, 30, 36, 20]
plt.plot(game, scores)
plt.title("游戏得分")
plt.show()
```

该例的运行结果如图 3-17 所示。

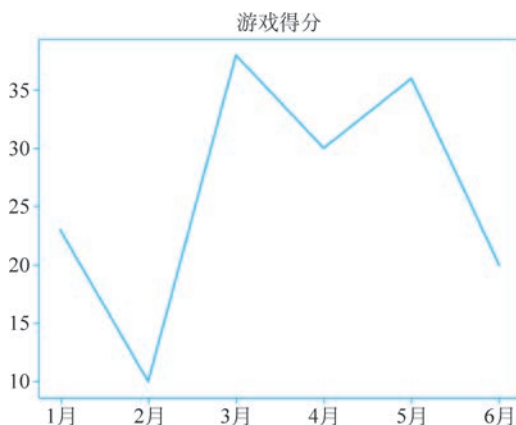


图 3-17 折线图

3.1.2 数据功能图表介绍

在大数据的可视化图中,按照数据的作用和功能可以把图分为比较类图、分布类图、流程类图、地图类图、占比类图、区间类图、关联类图、时间类图和趋势类图等。

1. 比较类图

比较类图可视化的方法通常是显示值与值之间的不同和相似之处,使用图形的长度、宽度、位置、面积、角度和颜色来比较数值的大小,用于展示不同分类间的数值对比以及不同时间点的数据对比。

常见的比较类图主要有柱状图、双向柱状图、气泡图、子弹图、色块图、漏斗图以及直方图等。图 3-18 显示了色块图。

2. 分布类图

分布类图可视化的方法通常是显示频率,将数据分散在一个区间或分组,并使用图形的位置、大小、颜色的渐变程度来表现数据的分布。分布类图通常用于展示连续数据上数值的分布情况。

常见的分布类图主要有箱形图、等高线图、热力图、散点图、分布曲线图、色块图以及直方图等。图 3-19 显示了箱形图。

3. 流程类图

流程类图可视化的方法通常是显示流程流转和流程流量。一般流程都会呈现出多个环节,每个环节之间会有相应的流量关系,因此这类图形可以很好地表示这些流量关系。

133	114	80	66	60	53	35	26	21
193	165	115	96	77	64	51	38	30
205	176	123	102	82	68	54	41	32
216	185	129	121	86	72	57	43	34
250	214	150	125	100	83	66	50	40
256	219	153	128	102	85	68	51	40
259	222	155	129	103	86	69	51	41
290	249	174	145	116	96	77	58	46
322	276	193	161	129	107	86	64	51

图 3-18 色块图

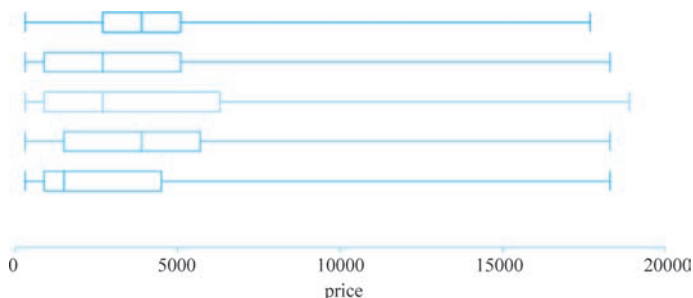


图 3-19 箱形图

常见的流程类图主要有漏斗图以及桑基图等。图 3-20 显示了桑基图。

4. 地图类图

地图类图可视化的方法是显示地理区域上的数据,并在显示时使用地图作为背景,通过图形的位置来表现数据的地理位置。地图类图通常用来展示数据在不同地理区域上的分布情况。

常见的地图类图主要有带气泡的地图以及分级统计地图等。图 3-21 显示了带气泡的地图。

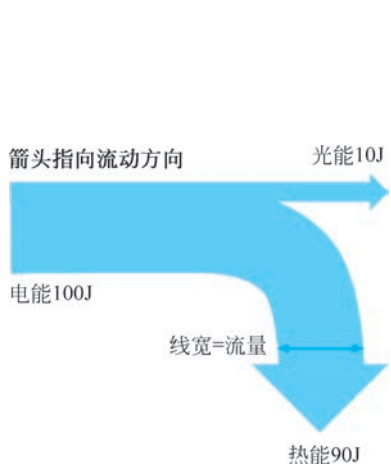


图 3-20 桑基图

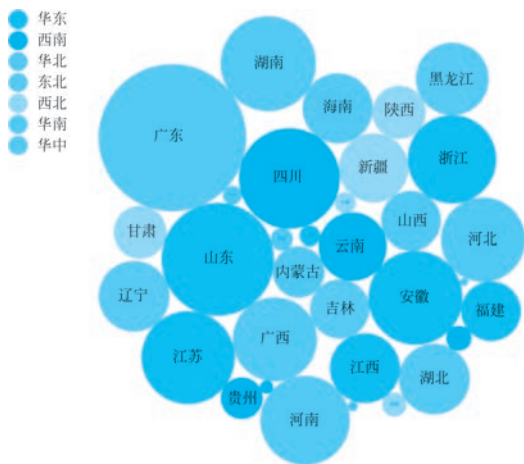


图 3-21 带气泡的地图

5. 占比类图

占比类图可视化的方法是显示同一维度上的占比关系。

常见的占比类图主要有环图、马赛克图、饼图、堆叠柱状图以及矩形树图等。图 3-22 显

示了堆叠柱状图。

6. 区间类图

区间类图可视化的方法是显示同一维度上值的上限和下限之间的差异。区间类图使用图形的大小和位置表示数值的上限和下限,通常用于表示数据在某一个分类(时间点)上的最大值和最小值。

常见的区间类图主要有仪表盘图以及堆叠面积图等。图 3-23 显示了堆叠面积图。

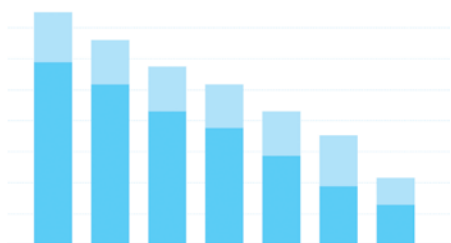


图 3-22 堆叠柱状图

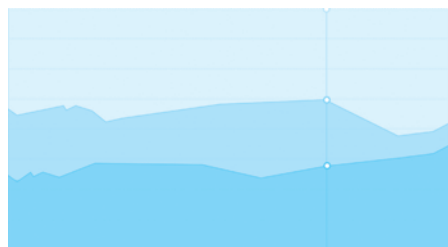


图 3-23 堆叠面积图

7. 关联类图

关联类图可视化的方法显示数据之间的相互关系。关联类图使用图形的嵌套和位置表示数据之间的关系,通常用于表示数据之间的前后顺序、父子关系以及相关性。

常见的关联类图主要有和弦图、桑基图、矩形树图、树状图以及韦恩图等。图 3-24 显示了韦恩图。

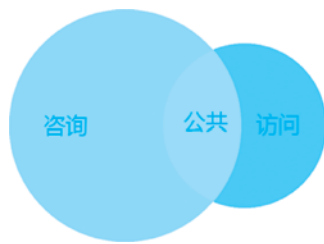


图 3-24 韦恩图

8. 时间类图

时间类图可视化的方法显示以时间为特定维度的数据。时间类图使用图形的位置表现出数据在时间上的分布,通常用于表现数据在时间维度上的趋势和变化。

常见的时间类图主要有面积图、K线图、折线图、卡吉图以及螺旋图等。图 3-25 显示了卡吉图。

9. 趋势类图

趋势类图可视化的方法分析数据的变化趋势。趋势类图使用图形的位置表现出数据在连续区域上的分布,通常展示数据在连续区域上的大小变化的规律。

常见的趋势类图主要有面积图、K线图、折线图以及回归曲线图等。图 3-26 显示了回归曲线图。

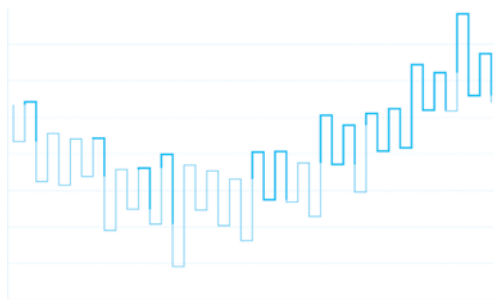


图 3-25 卡吉图

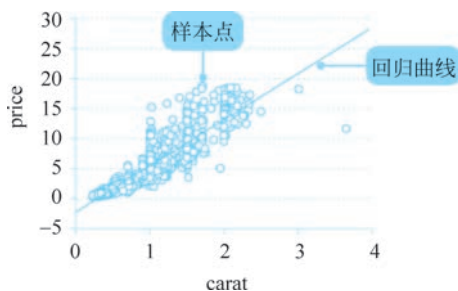


图 3-26 回归曲线图

扫一扫



视频讲解

3.1.3 可视化图表的选择与使用技巧

1. 可视化图表的选择

在进行数据可视化时要选择合适的图表,这样才能精确地传达数据信息。下面以常见图表为例介绍可视化图表的基本类型和选用原则。

1) 柱状图

柱状图利用柱子的高度比较清晰地反映数据的差异,通常用于不同时期或不同类别数据之间的比较,也可以用来反映不同时期和不同数据的差异。柱状图的局限在于它仅适用于中小规模的数据集,当数据较多时不易分辨。

例如,如果大家想研究网站中文章阅读量与互动率的趋势,则可以采用柱状图。

2) 条形图

条形图用来反映分类项目之间的比较,适用于跨类别比较数据。在进行数据可视化时如果需要比较项类的大小、高低,则适合使用条形图。

3) 折线图

折线图是数据随着时间推移发生变化的一种图表,可以预测未来的发展趋势,相对于柱状图,折线图能反映大数据集的走势,还适合多个数据集走势的比较。描述事物随时间维度变化时常需要使用该图形。

4) 饼图

饼图主要用来分析公司内部各个组成部分对事件的影响,其各部分的占比之和必须是100%。在需要描述某一部分占总体的百分比时适合使用饼图,例如占据公司全部资金一半的两个渠道、某公司员工的男女比例等。当需要比较数据时,尤其是比较两个以上整体的成分时,请务必使用条形图或柱形图,切勿要求看图人将扇形转换成数据在饼图间相互比较,因为人的肉眼对面积大小不敏感,会导致对数据的误读。

此外,为了使饼图发挥最大作用,在使用中一般不宜超过6个部分;如果要表达6个以上的部分,需要使用条形图。

5) 散点图

散点图使用两组数据构成多个坐标点,分析坐标点的分布情况,判断两个变量之间的关联或分布趋势。如果需要表达数据之间的关联关系,则可以使用散点图或气泡图。

6) 漏斗图

使用漏斗图可以清晰明了地看出每个层级的转化,如果想查看具体到每天的日期与实施转化数据的关系,则可以使用漏斗图。

表3-1显示了图表的应用场景及适用的类型;表3-2显示了图表较适合的数据维度。

表 3-1 图表的应用场景及适用的类型

数据关系	应用场景	图表类型
分类比较	地区产品销售量	柱状图、条形图
时间序列	近五年水质检测质量	折线图、柱状图
总体构成	网站推广渠道份额占比	饼图、条形图
频次分布	网站不同消费能力的用户占比	柱状图
关联关系	女性体重随年龄的分布	散点图、气泡图
转换关系	日期与工作转换率的关系	漏斗图

表 3-2 图表较适合的数据维度

图表类型	较适合的数据维度	图表类型	较适合的数据维度
柱状图	二维数据	饼图	一维数据
条形图	二维数据	散点图	三维数据
折线图	二维数据		

2. 可视化图表的使用技巧

1) 折线图的使用技巧

折线图的使用技巧如下:

(1) 折线图中各点的连接可以使用曲线和直线,这样可以使曲线较为美观,直线数据展示更为清晰。

(2) 折线的颜色要清晰,尽量不要与背景色和坐标轴线的颜色相近。

(3) 折线图线条尽量不要超过 4 条,过多的线会导致界面混乱,从而无法阅读。

2) 柱状图的使用技巧

柱状图的使用技巧如下:

(1) 柱状图中的颜色尽量不要超过 3 种。

(2) 柱状图中柱子间的宽度和间隙要适当。当柱子太窄时,用户的视觉可能会集中在两个柱子之间的负空间。

(3) 在对多个数据系列排序时,如果不涉及日期等特定数据,最好能符合一定的逻辑,用直观的方式引导用户更好地查看数据。此时可以通过升序或降序排列,例如按照数量从多到少对数据进行排列,也可以按照字母顺序等来排列。

3) 饼图的使用技巧

饼图的使用技巧如下:

(1) 饼图适合用来展示单一维度数据的占比,要求其数值中没有零或负值,并确保各分块占比之和为 100%。

(2) 饼图不适合用于精确数据的比较,因此当各类别数据占比较接近时,人们很难识别出每个类别占比的大小。

(3) 大多数人的视觉习惯是按照顺时针和自上而下的顺序去观察,因此在绘制饼图时建议从 12 点开始沿顺时针右边的第一个分块绘制饼图最大的数据分块,这样可以有效地强调其重要性。

4) 散点图的使用技巧

散点图的使用技巧如下:

(1) 如果一个散点图没有显示变量之间的任何关系,那么或许该图表类型不是此数据的最佳选择。

(2) 散点图只有在有足够多的数据点,并且数据之间有相关性时才能呈现很好的结果。如果一份数据只有极少的信息或者数据间没有相关性,那么绘制一个很空的散点图和不相关的散点图都是没有意义的。

(3) 如果数据包含不同系列,可以给不同系列使用不同的颜色。

3.2 文本可视化

3.2.1 文本可视化概述

文字是传递信息最常用的载体,文本信息是互联网中最主要的信息类型。与图形、语音和视频信息相比,文本信息的体积更小、传输更快,并且更容易生成。

将互联网中广泛存在的文本信息用可视化的方式表示能够更加生动地表达蕴含在文本中的语义特征,例如逻辑结构、词频、动态演化规律等。因此,针对一篇文章,文本可视化能更快地告诉读者文章在讲什么;针对社交网络上的发言,文本可视化可以帮读者将所有信息归类;针对一个大新闻,文本可视化可以帮读者捋顺事情发展的脉络;针对一本长篇小说,文本可视化能够帮读者厘清每个人物的关系;针对一系列的文档,读者可以通过文本可视化找到它们之间的联系;等等。图 3-27 显示了文本可视化的实例。

1. 文本可视化的流程

文本可视化涵盖了信息收集、文本信息挖掘(文本预处理、文本特征抽取、文本特征度量)、视图绘制和交互设计等过程。其中,文本信息挖掘等技术充分发挥计算机的自动处理能力,将无结构的文本信息自动转换为可视的有结构信息,而视图绘制使人类视觉认知、关联、推理的能力得到充分的发挥。因此文本可视化有效地结合了机器智能和人工智能,为人们更好地理解文本和发现知识提供了新的有效途径。

文本可视化的流程如图 3-28 所示。



图 3-27 文本可视化的实例

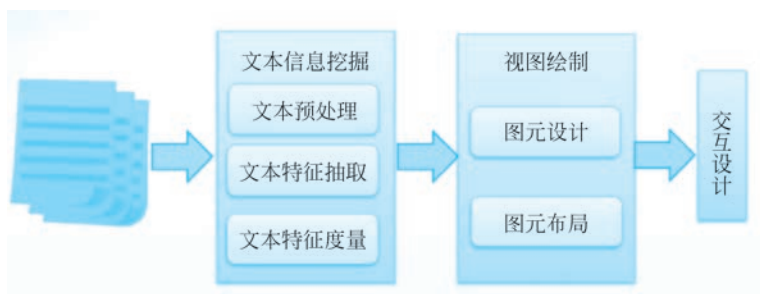


图 3-28 文本可视化的流程

从图 3-17 可以看出,文本可视化的流程主要包含以下 3 个方面。

(1) 文本信息挖掘:文本信息挖掘依赖于自然语言处理,因此词袋模型、命名实体识别、关键词抽取、主题分析、情感分析等是较常用的文本分析技术。文本信息挖掘主要是通过分词、抽取、归一化等操作提取出文本词汇及相关的內容。

(2) 视图绘制:可视化呈现是将文本分析后的数据用视觉编码的形式来处理,其中涉

及的内容有尺寸、颜色、形状、方位、纹理等,并使用各种图表来描述。

(3) 交互设计: 为了使用户能够通过可视化有效地发现文本信息的特征和规律, 通常在可视化设计中根据使用的场景为系统设置一定程度的交互功能。

2. 文本可视化的实现

文本可视化的实现是文本可视化的重要步骤, 大致要经历以下几步。

(1) 在文本中进行分词计算, 提取关键词, 并去掉冗余的文字。

(2) 为提取出来的关键词计算权重, 即决定哪些词着重显示。一般来说, 权重较高的词会显示在较引人注目的地方。

(3) 为可视化显示布局, 在布局中要计算出每个词或英文单词的摆放位置, 并最终呈现在用户面前。常见的文本可视化呈现的结果有标签云、文本地图、网络图以及叠式图等。图 3-29 显示了文本可视化分析的显示结果。

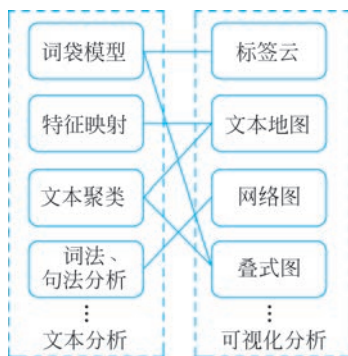


图 3-29 文本可视化分析的显示结果

3. 文本可视化的类型

文本可视化的类型, 除了包含常规的图表类(例如柱状图、饼图、折线图等形式)外, 在文本领域使用较多的可视化类型主要有以下 3 种。

(1) 基于文本内容的可视化: 基于文本内容的可视化包括基于词频的可视化和基于词汇分布的可视化, 常用的可视化形式有词云、分布图和 Document Cards 等。

(2) 基于文本关系的可视化: 基于文本关系的可视化研究文本内外的关系, 帮助人们理解文本内容和发现规律, 常用的可视化形式有树状图、节点连接的网络图、力导向图、叠式图和 Word Tree 等。

(3) 基于多层面信息的可视化: 基于多层面信息的可视化主要研究如何结合信息的多个方面帮助用户从更深层次理解文本数据, 发现其内在规律。其中, 包含时间信息和地理坐标的文本可视化近年来受到人们越来越多的关注。其常用的可视化形式有地理热力图、ThemeRiver、SparkClouds、TextFlow 和基于矩阵视图的情感分析可视化等。

扫一扫



视频讲解

3.2.2 词云概述及实现方法

1. 词云概述

词云也称为标签云或文字云, 它是一种典型的文本可视化技术。词云对文本中出现频率较高的“关键词”予以视觉上的突出, 从而形成“关键词云层”或“关键词渲染”。在词云中会过滤掉大量的文本信息, 使网页浏览者只要一眼扫过文本就可以领略文本的主题。图 3-30 和图 3-31 所示为词云的显示效果。

从图 3-30 和图 3-31 中可以看出, 在词云中一般用字号大小、字体颜色等图形属性对文本关键词进行可视化。其中字号大小常用于表示该关键词的重要性, 字号越大表示该关键词越重要。



图 3-30 词云的显示效果 1



图 3-31 词云的显示效果 2

2. 词云的实现

1) 在线词云的制作

登录网址“<http://yciyun.com/>”,选择“线上作品”,单击“基本”按钮,并选择不同的形状,即可直接生成词云,操作界面如图 3-32 所示。

每当用户使用鼠标单击并选择不同形状时都会生成不同的词云,如图 3-33 和图 3-34 所示。



图 3-32 生成词云的界面



图 3-33 生成的词云 1

用户在登录后还可以在“数据”面板中选择不同的文字、文字大小以及文字颜色,操作界面如图 3-35 所示。

2) 使用 Python 3 制作词云

使用 Python 3 制作词云需要导入 WordCloud 库,该库是 Python 中的一个非常优秀的词云展示第三方库。



图 3-34 生成的词云 2



图 3-35 选择不同的数据

从网上下载并安装 WordCloud 库,然后在 Windows 7 命令提示符窗口中输入以下命令:

```
import WordCloud
```

如果运行没有报错,则表示已经成功安装 WordCloud 库,如图 3-36 所示。



图 3-36 安装并导入 WordCloud 库

此外,为了能够在 Python 3 中显示中文字符,还需要安装另外一个库——jieba。该库也是一个 Python 第三方库,用于中文分词。在下载并安装 jieba 库后,在 Windows 7 命令提示符窗口中输入以下命令:

```
import jieba
```

如果运行没有报错,则表示已经成功安装 jieba 库,如图 3-37 所示。

```
>>> import jieba
>>>
```

图 3-37 安装并导入 jieba 库

jieba 涉及的算法如下。

(1) 基于 Trie 树结构实现高效的词图扫描,生成句子中汉字所有可能成词的情况所构成的有向无环图(DAG)。

(2) 采用了动态规划查找最大概率路径,找出基于词频的最大切分组合。

(3) 对于未登录词,采用了基于汉字成词能力的 HMM 模型,使用了 Viterbi 算法。

jieba 支持的 3 种分词模式如下。

(1) 精确模式:试图将句子最精确地切开,适合文本分析。

(2) 全模式:把句子中所有可以成词的词语都扫描出来,速度非常快,但是不能解决歧义问题。

(3) 搜索引擎模式:在精确模式的基础上对长词再次切分,提高召回率,适用于搜索引擎分词。

下面是几个制作词云的例子。

【例 3-2】 制作英文词云。

代码如下:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
with open('1.txt','r') as f:
    mytext = f.read()
wordcloud = WordCloud().generate(mytext)
plt.imshow(wordcloud, interpolation = "bilinear")
plt.axis("off")
plt.show()
```

语句的含义如下。

- from wordcloud import WordCloud: 导入 WordCloud 库。
- import matplotlib.pyplot as plt: 导入 matplotlib 库,用于显示图形。
- with open('1.txt','r') as f: 将 1.txt 文本中的数据导入程序中。
- mytext=f.read(): 读取该文本中的数据。
- wordcloud=WordCloud().generate(mytext): 利用 mytext 中存储的文本内容制作词云。
- plt.imshow(wordcloud,interpolation="bilinear") plt.axis("off"): 设置显示的词云图中无坐标轴。
- plt.show(): 可视化显示。

1) txt 中的内容如下:

The reason for translating afresh Beccarias Dei Delitti e delle Pene (Crimes and Punishments) is, that it is a classical work of its kind, and that the interest which belongs to it is still far from being merely historical.

程序的运行结果如图 3-38 所示。

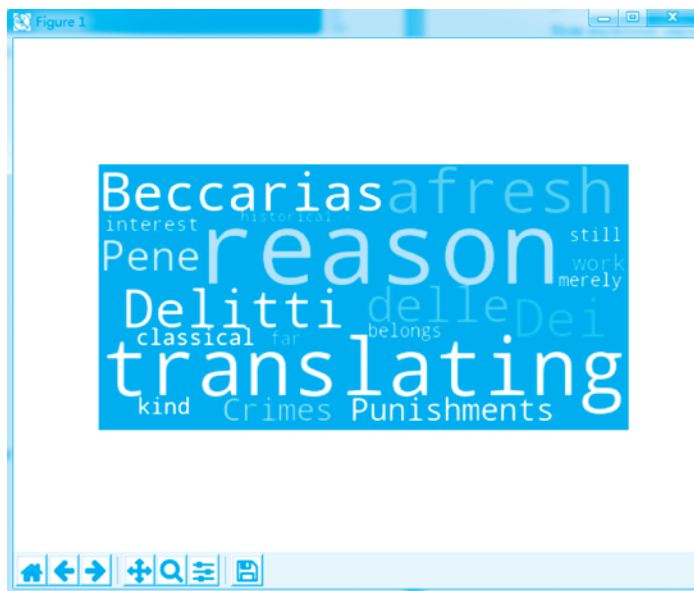


图 3-38 制作英文词云

【例 3-3】制作中文词云。

代码如下:

```
from wordcloud import WordCloud
import jieba
with open('2.txt', 'r', encoding = 'utf-8') as f:
    text = f.read()
cut_text = " ".join(jieba.cut(text))
cloud = WordCloud(
    # 设置字体,不指定就会出现乱码
    font_path = "C:\\Windows\\Fonts\\stxingka.ttf",
    # font_path = path.join(d, 'simsun.ttc'),
    # 设置背景色
    background_color = 'white',
    # 词云形状
    max_words = 4000,
    # 最大号字体
    max_font_size = 60
)
wCloud = cloud.generate(cut_text)
wCloud.to_file('cloud.jpg')
import matplotlib.pyplot as plt
plt.imshow(wCloud, interpolation = 'bilinear')
plt.axis('off')
plt.show()
```

- `import jieba`: 导入 jieba 库。
- `with open('2.txt','r',encoding='utf-8') as f`: 设置输入为中文字符。
- `cut_text = " ".join(jieba.cut(text))`: 使用全模式进行中文分词。

程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系,它按照特定规则组织计算机指令,使计算机能够自动进行各种运算处理。按照程序设计语言规则组织起来的一组计算机指令称为计算机程序。程序设计语言也叫编程语言。

图 3-39 制作中文词云

3.3.1 网络可视化概述

扫一扫



视频讲解

从图 3-40 和图 3-41 可以看出,社交网络图侧重于显示网络内部的实体关系,它将实体作为节点,一张社交网络图可以由无数多个节点组成,并用边连接所有的节点。通过分析社交网络图可以直观地看出每个人或每个组织的相互关系。



图 3-40 网络关联图

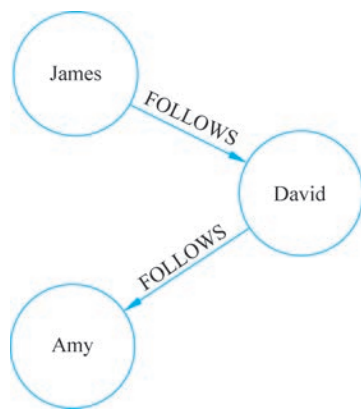


图 3-41 知乎粉丝关系图

社交网络是一种复杂网络,单纯地研究网络中的节点或计算网络中的统计信息并不能完全揭示网络中的潜在关系,因此对于社交网络来说最直观的可视化方式是网络结构。图 3-42 显示了家庭中的社会关系,该图一共有 10 个节点和 20 条边。

树状图是社交网络图的常见表现形式,也是一种流行的利用包含关系表达层次化数据的可视化方法。由于其呈现数据时高效的空间利用率和良好的交互性,受到人们众多的关注,得到深入的研究,并在科学、社会学、工程、商业等领域得到了广泛的应用。图 3-43 显示了树状图可视化。

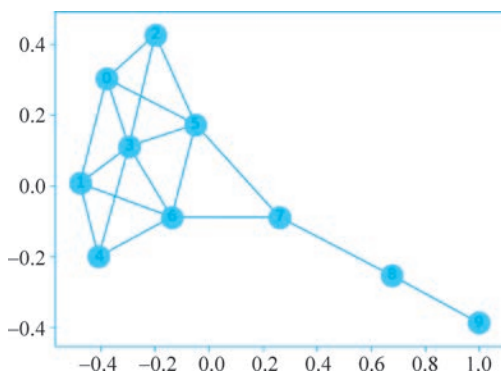


图 3-42 家庭中的社会关系网络图



图 3-43 树状图可视化

值得注意的是,对于具有海量节点和边的大规模网络,例如节点规模达到上百万个的网络,如何在有限的空间中进行可视化,是网络可视化面临的一个难点。

3.3.2 使用 Python 3 制作社交网络图

在 Python 3 中可以制作社交网络图,在制作时需要先导入 networkx 库,该库是一个用 Python 语言开发的图论与复杂网络建模工具,内置了常用的图与复杂网络分析算法,可以方便地进行复杂网络数据分析、仿真建模等工作。

在使用 networkx 库绘制网络图时,常用 node 表示节点、cycle 表示环(通常环是封闭

扫一扫



视频讲解

的)、edges 表示边。

【例 3-4】 使用 networkx 库绘制无向网络图。

代码如下：

```
from matplotlib import pyplot as plt
import networkx as nx
G = nx.Graph()
G.add_nodes_from([1,2,3])
G.add_edges_from([(1,2),(1,3)])
nx.draw_networkx(G)
plt.show()
```

语句的含义如下。

- from matplotlib import pyplot as plt: 导入 matplotlib 库。
- import networkx as nx: 导入 networkx 库。
- G=nx.Graph(): 建立无向图。
- G.add_nodes_from([1,2,3]): 创建节点 1、2、3。
- G.add_edges_from([(1,2),(1,3)]): 加边集合 1,2 和 1,3。
- nx.draw_networkx(G): 绘制图形。
- plt.show(): 显示图形。

程序的运行结果如图 3-44 所示。

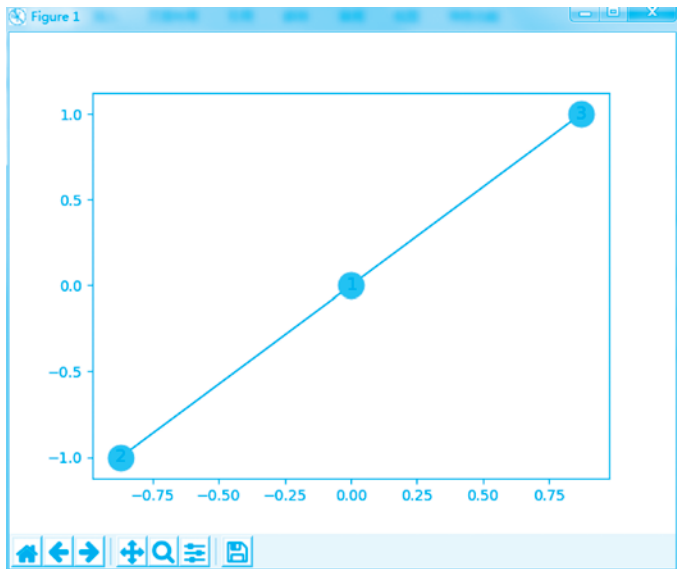


图 3-44 无向网络图

【例 3-5】 使用 networkx 库绘制有向网络图。

代码如下：

```
import networkx as nx
import matplotlib.pyplot as plt
```

```
G = nx.DiGraph()
G.add_nodes_from([0,1,2])
G.add_edges_from([(0,1),(1,2),(2,0)])
nx.draw_networkx(G,)
plt.show()
```

语句的含义如下。

- `G = nx.DiGraph()`: 建立有向图。
- `G.add_nodes_from([0,1,2])`: 添加节点。
- `G.add_edges_from([(0,1),(1,2),(2,0)])`: 添加边。

程序的运行结果如图 3-45 所示。

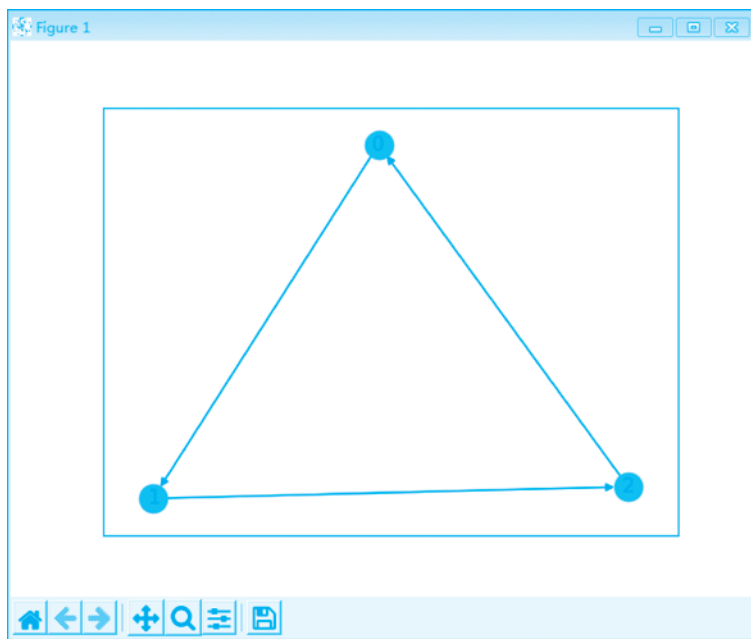


图 3-45 有向网络图

3.4 空间信息可视化

3.4.1 空间信息可视化概述

空间信息可视化是指运用计算机图形图像处理技术将复杂的科学现象和自然景观及一些抽象概念图形化的过程。空间信息可视化常用地图学,计算机图形图像技术将地学信息输入、查询、分析、处理,采用图形、图像,结合图表、文字、报表,以可视化形式实现交互处理和显示。

空间信息可视化是以可视化的方式显示输出空间信息,通过视觉传输和空间认知活动去探索空间事物的分布及其相互关系,以获取有用的知识,并进而发现规律。

空间信息可视化的主要表现形式有地图、多媒体信息、动态地图、三维仿真地图以及虚拟现实等。

(1) 地图: 空间信息可视化的最主要形式,也是最古老的形式。

(2) 多媒体信息：使用文本、图形、图像、声音、录像、音频、视频等各种形式综合、形象地表现空间信息，是空间信息可视化的重要形式。

(3) 动态地图：一种处于运动状态的数字地图，借助于计算机综合处理多种媒体信息的功能，将文字、图形、图像、声音、动画及视频技术相结合，使多种信息逻辑地连接并集成为一个有机的具有人性化操作界面的空间信息传输系统。

(4) 三维仿真地图：利用地图动画技术直观而又逼真地显示地理实体运动变化的规律和特点。

(5) 虚拟现实：以视觉为主，也结合听、触、嗅甚至味觉来感知环境，使人们犹如进入真实的地理空间环境之中并与之发生交互作用。它除了对三维空间和一维时间仿真外，还包含对自然交互方式的仿真。

3.4.2 空间信息可视化建模

空间信息可视化建模与传统可视化建模的最大区别是用户可以自己在地理空间中交互，获取不同层面的信息。在空间信息可视化的实现中经常要使用到 3D 图形，3D 图形可以让空间信息的展现变得真实。

在 Python 3 中可以通过导入 Axes3D 库来绘制 3D 图形。图 3-46 绘制了 3D 螺旋图；图 3-47 绘制了 3D 直方图；图 3-48 绘制了 3D 轮廓图。

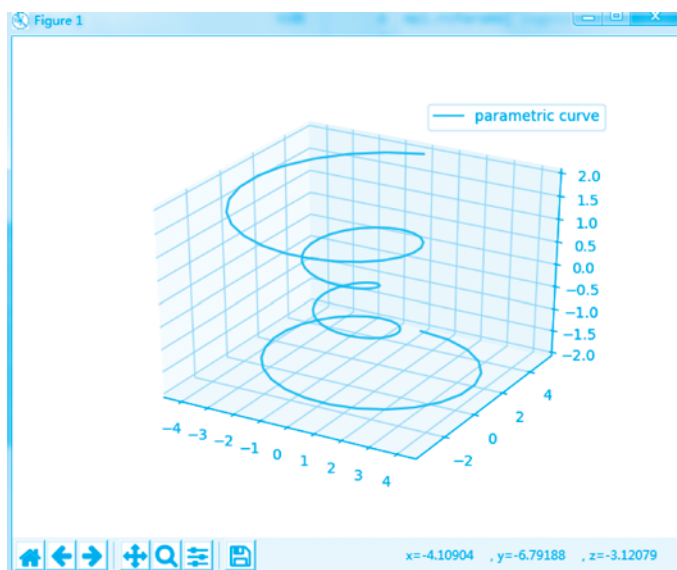


图 3-46 3D 螺旋图

【例 3-6】 使用 Python 绘制 3D 螺旋图。

该例的代码如下：

```
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']      # 设置字体
plt.rcParams['axes.unicode_minus'] = False       # 设置负号
```

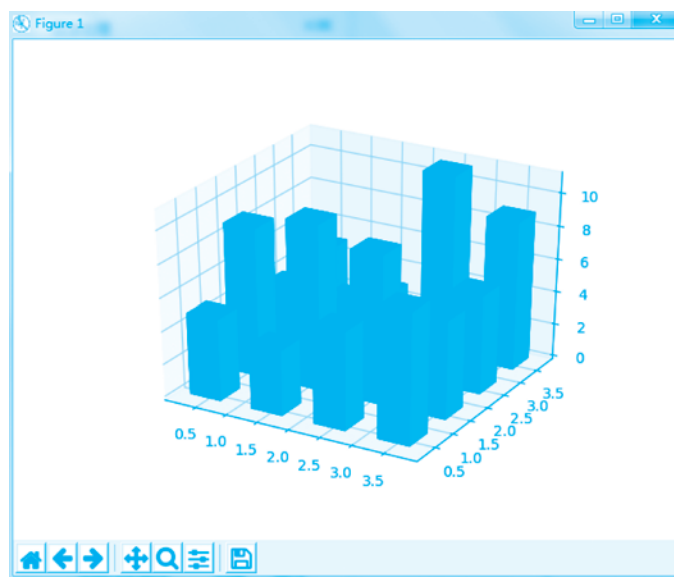


图 3-47 3D 直方图

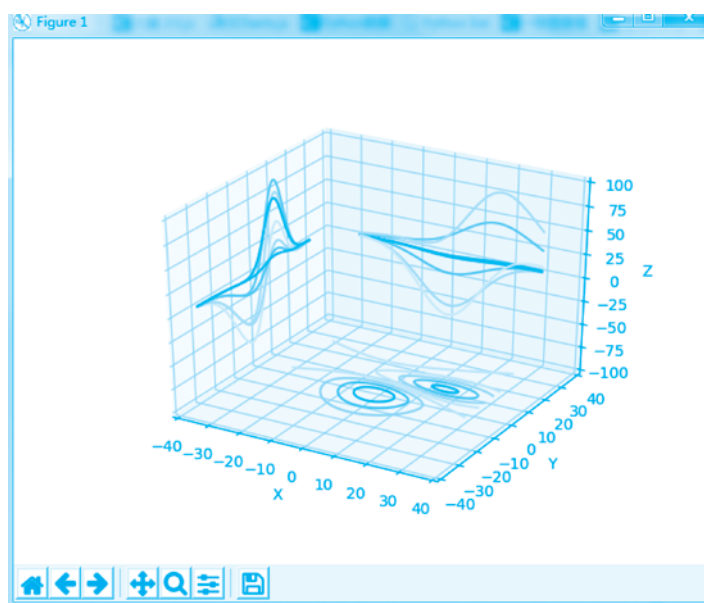


图 3-48 3D 轮廓图

```
fig = plt.figure()
ax = fig.add_subplot(projection = '3d')
theta = np.linspace(-4 * np.pi, 4 * np.pi, 100)
z = np.linspace(-2, 2, 100)
r = z**2 + 1
x = r * np.sin(theta)
y = r * np.cos(theta)
ax.plot(x, y, z, label = 'parametric curve')
ax.legend()
```

```
plt.title("3D 螺旋图")  
plt.show()
```

该例的运行结果如图 3-49 所示。

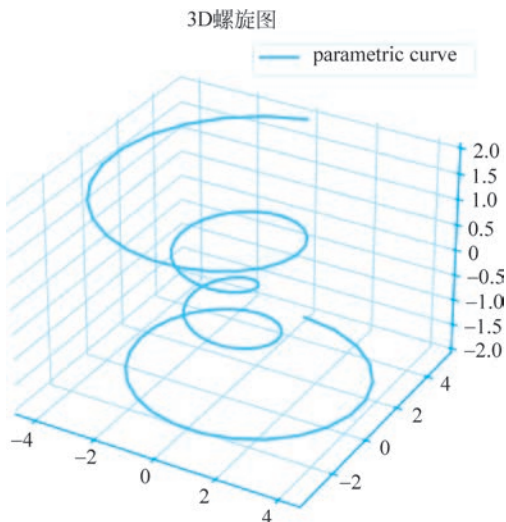


图 3-49 3D 螺旋图

3.4.3 空间信息可视化的应用

空间信息可视化的应用十分广泛,已经涉及大多数国民经济行业。图 3-50 显示了多媒体信息;图 3-51 显示了动态地图;图 3-52 显示了三维仿真地图;图 3-53 显示了虚拟现实(VR)图。



图 3-50 多媒体信息



图 3-51 动态地图



图 3-52 三维仿真地图



图 3-53 虚拟现实图

3.5 本章小结

(1) 图是表达数据最直观、最强大的方式之一,通过图的展示能够对数据进行变换,从而让枯燥的数字能吸引人们的注意力。

(2) 在统计图表的每一种类型的图表中都可包含不同的数据可视化图,例如柱状图、K线图、饼图、折线图、散点图、气泡图、雷达图、面积图、漏斗图、和弦图、环形图、直方图、热力图、仪表盘图以及密度图等。

(3) 文本可视化涵盖了信息收集、文本信息挖掘(文本预处理、文本特征抽取、文本特征度量)、视图绘制和交互设计等过程。

(4) 网络可视化通常展示数据在网络中的关联关系,一般用于描绘互相连接的实体。

(5) 空间信息可视化是指运用计算机图形图像处理技术将复杂的科学现象和自然景观以及一些抽象概念图形化的过程。

3.6 实训

1. 实训目的

通过本章实训了解大数据可视化图表的特点,能进行简单的与大数据可视化有关的操作,并掌握数据可视化图的绘制。

2. 实训内容

(1) 中文词云的制作,代码如下:

```
from wordcloud import WordCloud
import jieba
# 读取标点符号库
with open('18.txt', 'r', encoding = 'utf-8') as f:
    text = f.read()
```

```
cut_text = " ".join(jieba.cut(text))
cloud = WordCloud(
    background_color = "white",           # 背景颜色
    max_words = 100,                      # 显示最大词数
    font_path = "C:\\Windows\\Fonts\\stxinka.ttf", # 使用字体
    min_font_size = 40,
    max_font_size = 100,
    width = 800,                          # 图幅宽度
    height = 400
)
wCloud = cloud.generate(cut_text)
import matplotlib.pyplot as plt
plt.imshow(wCloud, interpolation = 'bilinear')
plt.axis('off')
plt.show()
```

18. txt 中的内容如下:

1982年《万历十五年》中文版在国内问世以来，这本书就一直畅销，经久不衰，无论是历史学者还是业余爱好者，无论是企业老板还是普通白领，都能从中获得一些启发和想法。也难怪，《人民的名义》中的高玉良都拿着《万历十五年》刻苦钻研。这是因为在这本书中，黄仁宇采用了一种新颖的写法，让任何人都可以几乎无障碍地接触正史，在他的笔下，历史不再是枯燥无味的归总和罗列。这种新颖的写法被黄仁宇本人称为“大历史观”，它既不像编年体史书那样依照时间的脉络叙述一个王朝的兴衰，也不像纪传体史书那样单独介绍每个人物的一生，而是把历史剖开一个横截面，像纪录片导演穿越到过去一样，把当时的一位皇帝和五位著名的大臣以一种群像式的叙事手法缓缓展开，从中折射出16世纪中国社会的完整面貌，并且，如同管中窥豹一样，从各种细节中观察公元1587年表面一派太平的大明朝为何会一步一步走向衰落和灭亡。那位皇帝自然就是万历皇帝，五位大臣分别是大学士张居正和申时行，南京都察院都御史海瑞，蓟州总兵戚继光，以及前云南姚安知府李贽。用黄仁宇的话来说，他们或是身败，或是名裂，没有一个能够功德圆满。从他们身上可以看出，那些隐藏在历史深处的症结已经开始慢慢积累，直至无可逆转。

运行该程序,如图 3-54 所示。



图 3-54 中文词云

(2) 绘制社交网络图,代码如下:

```
import networkx as nx
import matplotlib.pyplot as plt
G = nx.DiGraph()
G.add_nodes_from([1,2,3,4,5,6,7])
nx.add_cycle(G,[1,2,3])
G.add_edge(1,4)
G.add_edges_from([(3,5),(3,6),(6,7)])
nx.draw_networkx(G, )
plt.show()
```

语句 `nx.add_cycle` 表示添加封闭路径。

该程序的运行结果如图 3-55 所示。

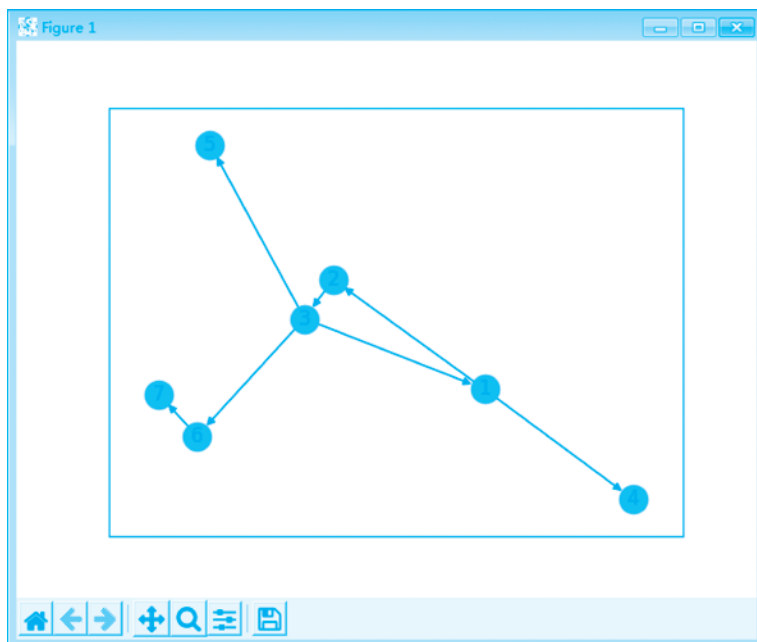


图 3-55 社交网络图

(3) 为图 3-56 所示的网络图至少添加 3 个新的网络节点,运行如图 3-57 所示。

(4) 流程图也叫“工作流程图”,主要用于显示流程中的顺序步骤,这种图表使用一系列相互连接的符号绘制出整个过程,使得过程易于理解,并有助于与其他人沟通。流程图可用于解释复杂/抽象的过程、系统、概念或算法的运作模式。绘制流程图还可以帮助人们规划和发展流程,或改进现有的流程图。在流程图中,不同符号代表不同意思,每种符号都具有各自的特定形状。图 3-58 显示了流程图。

请根据该流程图自行设计流程图。

(5) 树状图是一种常见的通过树状结构表示层次结构的图,该图通常从没有上级/父级成员的元素开始(根节点),然后加入节点,也称为分支,再用线连接根节点和分支节点表示成员之间的关系和连接。在树状图中处于最后的是枝叶节点(或称为末端节点),该节点是没有子节点的成员。图 3-59 显示了树状图。

请根据图 3-59 画出公司结构的树状图。

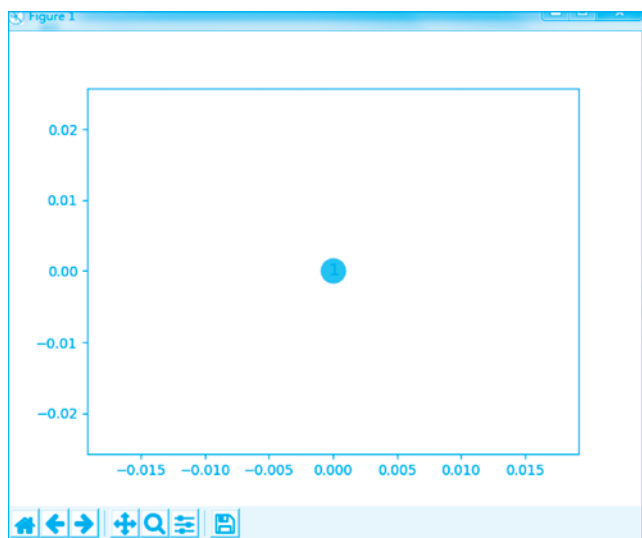


图 3-56 一个节点的网络图

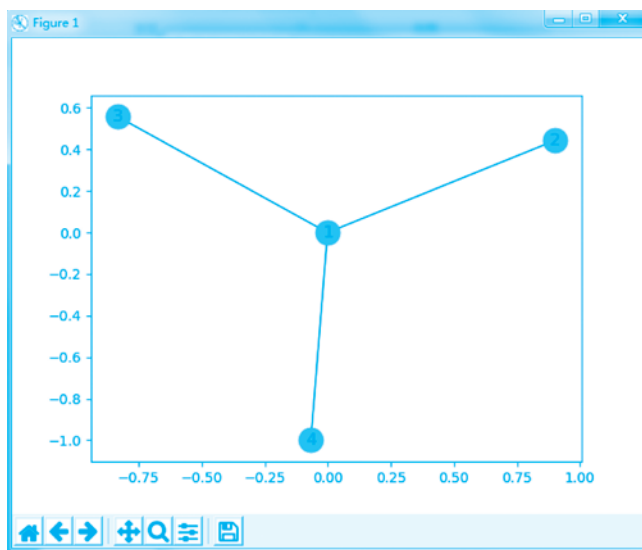


图 3-57 4 个节点的网络图

(6) 甘特图通常用作项目管理的组织工具,显示活动(或任务)列表和持续时间,也显示每项活动何时开始和结束。甘特图适合用来规划和估计整个项目所需的时间,也可显示相互重叠的活动。在甘特图中常用水平行代表活动,垂直列代表时间刻度。每项活动的持续时间由沿着时间刻度绘制的条形长度来表示。条形的开始位置代表活动开始,条形的结束位置代表活动结束。图 3-60 显示了甘特图。

请根据图 3-60 画出完成某项任务的甘特图。

(7) 登录网址“<http://yciyun.com/>”,选取自己感兴趣的文字内容制作一个词云,并下载。

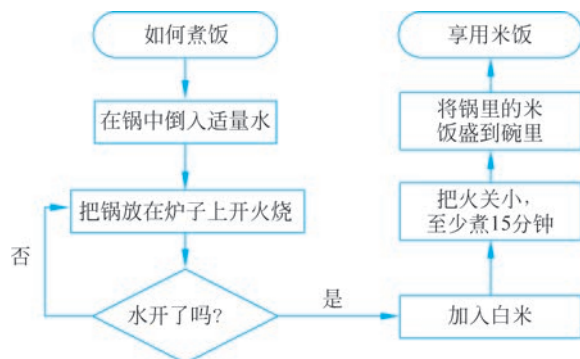


图 3-58 流程图

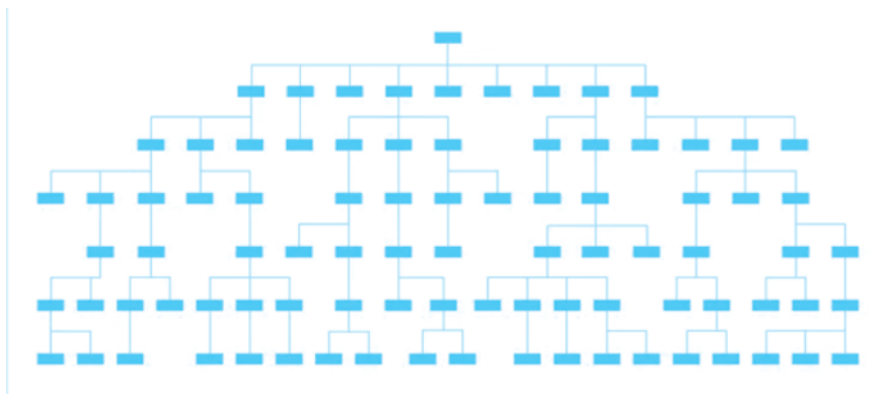


图 3-59 树状图

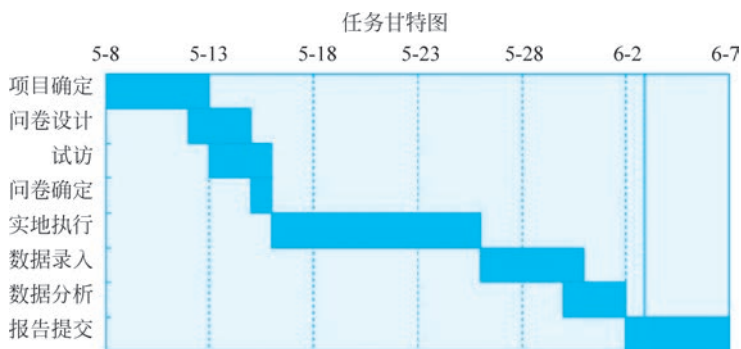


图 3-60 甘特图

习题 3

1. 数据可视化图按照数据功能可分为哪几类?
2. 在统计图表中包含哪些不同的数据可视化图?
3. 什么是文本可视化?
4. 如何使用 Python 3 实现文本可视化?
5. 什么是网络可视化?
6. 如何使用 Python 3 制作社交网络图?