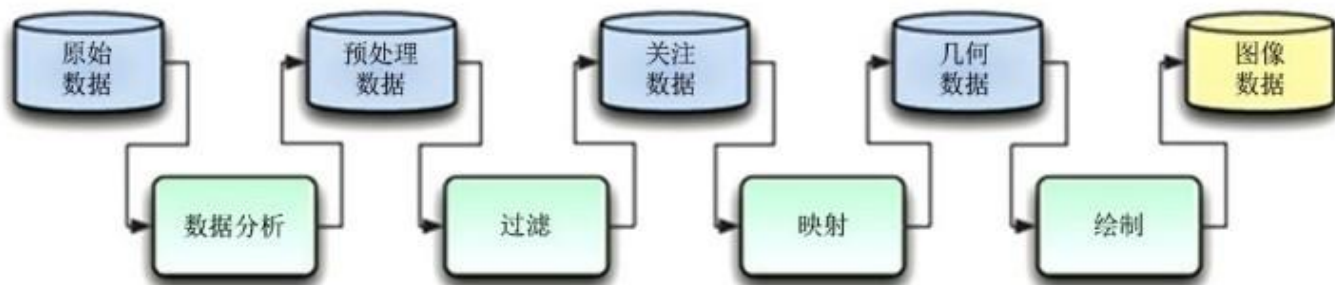


第4章 数据可视化基础

1 数据可视化基本框架

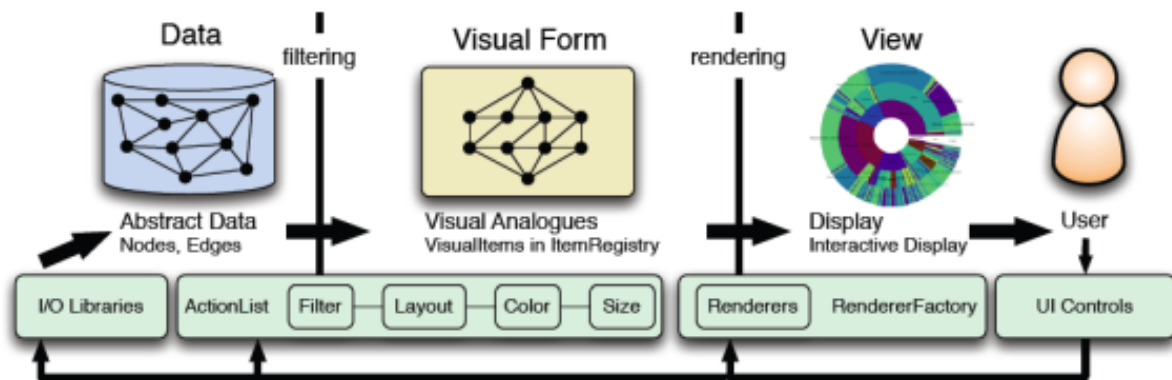
流水线模型

- 科学可视化和信息可视化分别设计了可视化流程的参考体系结构模型，并被广泛应用于数据可视化系统中
- 早期的科学可视化流水线 [Haber1990]
 - 描述了从数据空间到可视空间的映射，包含串行处理数据的各个阶段
 - 数据分析、数据滤波、数据的可视映射和绘制
 - 该流水线为数据处理和图形绘制的嵌套组合



信息可视化参考流程

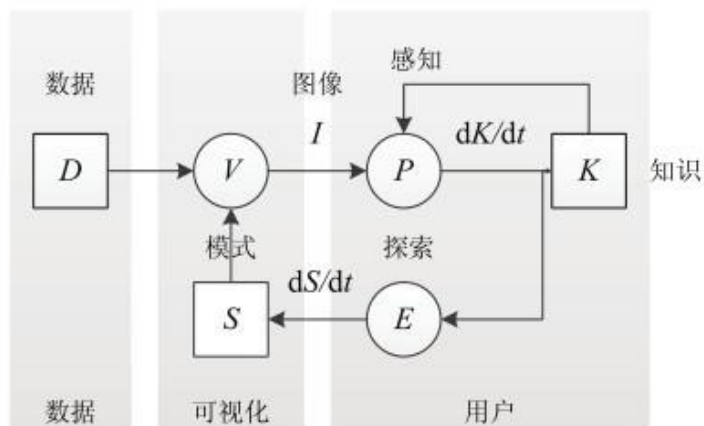
- Card, Mackinlay和Shneiderman描述的信息可视化流程模型：将流水线改进成回路且用户的交互可以出现在流程的任何阶段。后继几乎所有著名的信息可视化系统和工具包都支持这个模型，而且绝大多数系统在基础层都兼容，只存在细微的实现差异



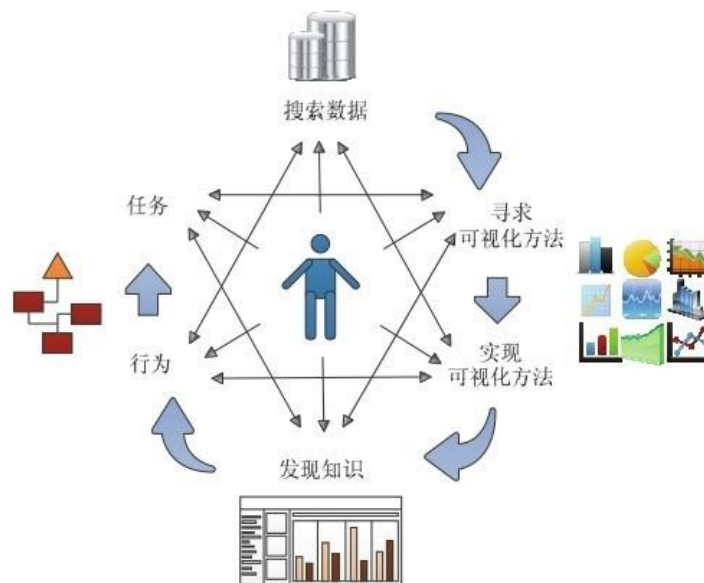
Card, S. K., Mackinlay, J. D., and Shneiderman, B., editors.
Readings in Information Visualization: Using Vision to Think, 1999.

可视化参考流程

- Jark Van Wijk等人提出的可视化循环模型 [Wijk2008]

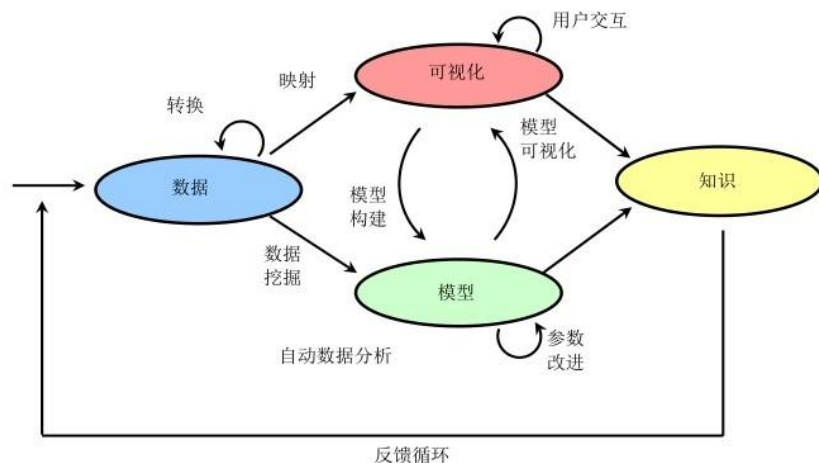


- C.Stolte等人提出的可视化循环模型 [Stolte2002]



可视分析模型

- 可视分析学的基本流程，通过人机交互将自分析动和可视方法紧密结合
 - 流水线的起点是输入的数据，终点是提炼的知识
- 从数据到知识有两个途径
 - 交互的可视化方法
 - 自动的数据挖掘方法
- 两个途径的中间结果分别是对数据的交互可视化结果和从数据中提炼的数据模型。用户既可以对可视化结果进行交互的修正，也可以调节参数以修正模型



数据可视化流程中的核心要素

- 数据可视化流程中的核心要素包括三个方面
 - 数据表示与变换
 - 数据的可视化呈现
 - 用户交互

数据表示与变换

- 数据可视化的基础是数据表示和变换
- 为了允许有效的可视化、分析和记录，输入数据必须从原始状态变换到一种便于计算机处理的结构化数据表示形式
 - 通常这些结构存在于数据本身，需要研究有效的数据提炼或简化方法以最大程度地保持信息和知识的内涵及相应的上下文
- 有效表示海量数据的主要挑战在于采用具有可伸缩性和扩展性的方法，以便忠实地保持数据的特性和内容
 - 此外，将不同类型、不同来源的信息合成为一个统一的表示，使得数据分析人员能及时聚焦于数据的本质也是研究重点

数据的可视化呈现

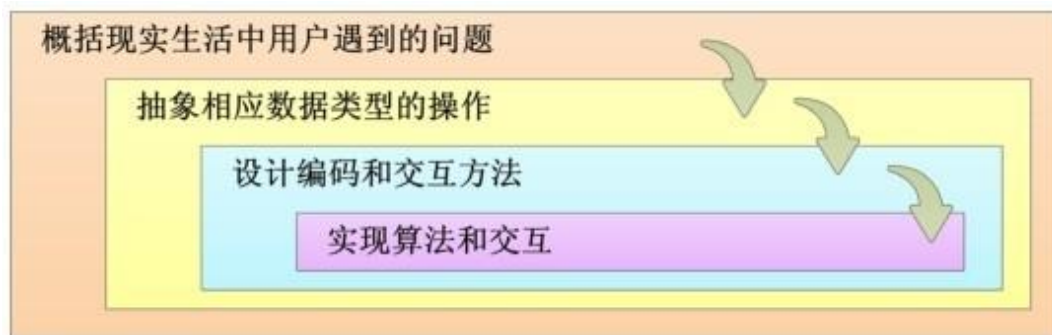
- 大量的数据采集通常是以流的形式实时获取的，针对静态数据发展起来的可视化显示方法不能直接拓展到动态数据。这不仅要求可视化结果有一定的时间连贯性，还要求可视化方法达到高效以便给出实时反馈
- 因此
 - 需要研究新的软件算法
 - 更强大的计算平台(如分布式计算或云计算)
 - 显示平台(如一亿像素显示器或大屏幕拼接)
 - 交互模式(如体感交互、可穿戴式交互)

用户交互

- 对数据进行可视化和分析的目的在于解决目标任务。有些任务可明确定义，有些任务则更广泛或者一般化
- 通用的目标任务可分成三类：生成假设、验证假设和视觉呈现
 - 数据可视化可以用于从数据中探索新的假设，也可以证实相关假设与数据是否吻合，还可以帮助数据专家向公众展示其中的信息
 - 交互是通过可视的手段辅助分析决策的直接推动力
- 有关人机交互的探索已经持续很长时间，但智能、适用于海量数据可视化的交互技术，如任务导向的、基于假设的方法还是一个未解难题
 - 其核心挑战是新型的可支持用户分析决策的交互方法
 - 这些交互方法涵盖底层的交互方式与硬件、复杂的交互理念与流程，更需要克服不同类型的显示环境和不同任务带来的可扩充性难点

数据可视化设计

- 数据可视化的设计简化为四个级联的层次 [Munzner2009]
 - 最外层(第一层)问题刻画层，刻画真实用户的问题
 - 第二层抽象层，将特定领域的任务和数据映射到抽象且通用的任务及数据类型
 - 第三层编码层，设计与数据类型相关的视觉编码及交互方法
 - 最内层(第四层)，创建正确完成系统设计的算法



层次嵌套模型优点

➤ 优点：

- 无论各层次以何种顺序执行，都可以独立地分析每个层次是否已正确处理
- 三个内层同属设计问题，但每一层又有所分工
- 四个层次极少按严格的时序过程执行，而往往是迭代式的逐步求精过程：某个层次有了更深入的理解之后，将用于指导精化其他层次
- 在第一层中，可视化设计人员采用以人为本的设计方法，与目标用户群相处大量时间，了解目标受众的需求
- 第二层将第一层确定的任务和数据从采用特定领域的专有名词的描述转化为更抽象、更通用的信息可视化术语的描述
- 第三层是可视化研究的核心内容：设计可视编码和交互方法。视觉编码和交互这两个层面通常相互依赖
- 第四层设计与前三个层次匹配的具体算法，相当于一个细节描述的过程

2 可视化中的数据

数据认知

- 人们对数据的认知，一般都经过从数据模型到概念模型的过程，最后得到数据在实际中的具体语义
 - 在处理数据时，最初接触的是数据模型
 - 概念模型是对数据的高层次描述，对应于人们对数据的具体认知。对数据进行进一步处理之前，需要定义数据的概念和它们之间的联系，同时定义数据的语义和它们所代表的含义
 - 概念模型的建立跟实际应用紧密相关

数据类型

- 根据数据分析要求，不同的应用可以采用不同的数据分类方法
 - 根据数据模型，可以分为
 - 浮点数、整数、字符等
 - 根据概念模型，可以定义
 - 数据所对应的实际意义或者对象，如汽车、摩托车、自行车等分类数据
- 在科学计算中，通常根据测量标度，将数据分为四类
 - 类别型数据、有序型数据、区间型数据和比值型数据

排 名	姓 名	时 间	性 别
1	小赵	3 分 10 秒	男
2	小钱	3 分 12 秒	男
3	小孙	3 分 18 秒	男
4	小李	3 分 40 秒	女
5	小周	3 分 52 秒	男
6	小吴	4 分 10 秒	女

科学计算数据类型

- 类别型数据：用于区分物体类别
 - 这些类别可以用于区分一组对象，但是无法提供对象的定量数据
- 有序型数据：用来表示对象间的顺序关系
 - 只能得到对象间的顺序关系，而无法根据序数间的数值差别，进行定量比较
- 区间型数据：用于得到对象间的定量比较。相对于有序型数据，区间型数据提供了详细的定量信息
 - 因为区间型数据基于任意的起始点，所以只能得到对象间的相对差别，并不能定义对象的绝对值
- 比值型数据：用于比较数值间的比例关系
 - 比值型数据基于真正意义上的0点，可以用来精确地定义比例

数据认知

- 不同的数据类型适用不同的操作算子
 - 区分度算子： $\neq$ ；序别算子： $>$ 、 $<$ ；加减算子： $+$ 、 $-$ ；乘除算子： $*$ 、 $/$ 。类别型数据适用于区分度算子，可以判断不同数据之间是否相等
- 不同的数据类型同时也对应不同的集合操作和统计计算
- 在数据可视化中，我们通常并不区分区间型数据和比值型数据，将数据类型进一步精简为三种
 - 类别型数据、有序型数据和数值型数据

类 型	基本操作 / 用途	集合操作	允许的统计计算
类别型	判断是否相等 $=$ 、 $\neq$	允许互换元素间位置	类别、模式、列联相关
有序型	判断大小 $=$ 、 $\neq$ 、 $>$ 、 $<$	计算元素单调递增（减）关系	中值、百分位数
区间型	判断差别 $=$ 、 $\neq$ 、 $>$ 、 $<$ 、 $+$ 、 $-$	允许元素间线性加减操作	平均值、标准方差、等级相关、积差相关
比值型	判断比例 $=$ 、 $\neq$ 、 $>$ 、 $<$ 、 $+$ 、 $-$ 、 $\times$ 、 $\div$	能判断元素间的相似度	变异系数

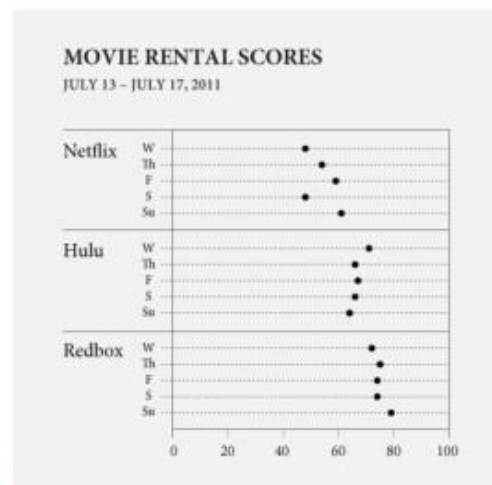
3 可视化的基本图表

原始数据绘图

- 原始数据绘图用于可视化原始数据的属性值，直观呈现数据特征
- 数据轨迹
- 柱状图(Bar Chart)
- 饼图(Pie Chart)
- 等值线图(Contour Map)
- 走势图(Sparkline)
- 散点图(Scatter Plot)和散点图矩阵(Scatter Plot Matrix)
- 维恩图(Venn Diagram)
- 热力图(Heat Map)

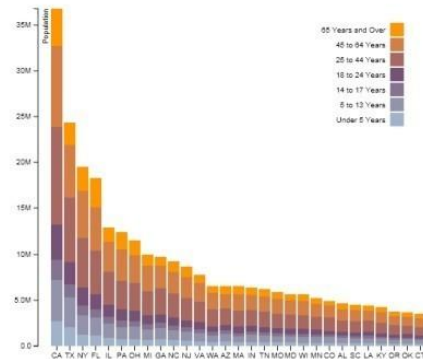
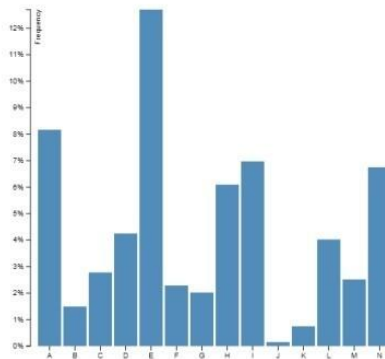
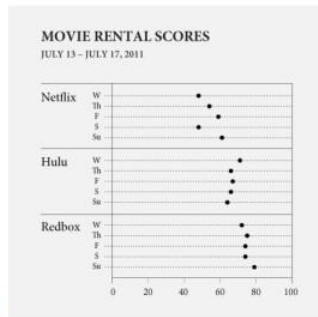
数据轨迹

- 数据轨迹是一种标准的单变量数据呈现方法：x轴显示自变量；y轴显示因变量



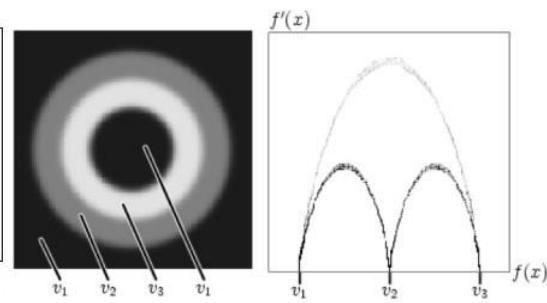
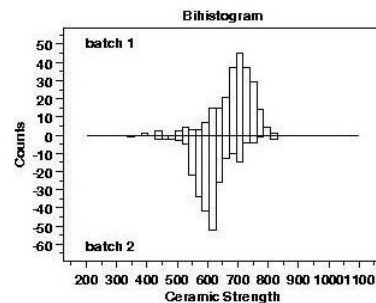
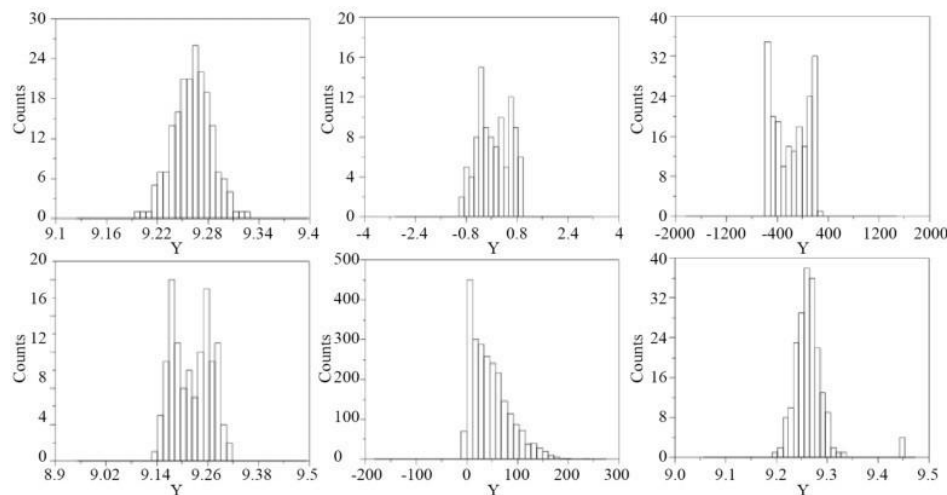
柱状图(Bar Chart)

- 柱状图采用长方形的形状和颜色编码数据的属性
- 柱状图的每根直柱内部也可用像素图(Pixel Chart)方式编码，也称为堆叠图(Stacked Graph)



直方图(Histogram)

- 直方图是对数据集的某个数据属性的频率统计
- 直方图可以呈现数据的分布、离群值和数据分布的模态。直方图的各个部分之和等于单位整体，而柱状图的各个部分之和没有限制



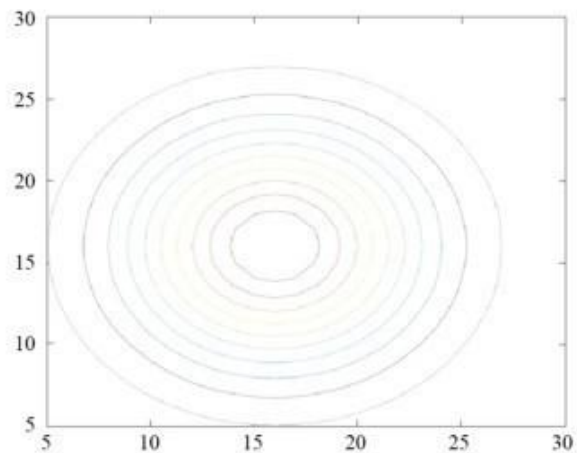
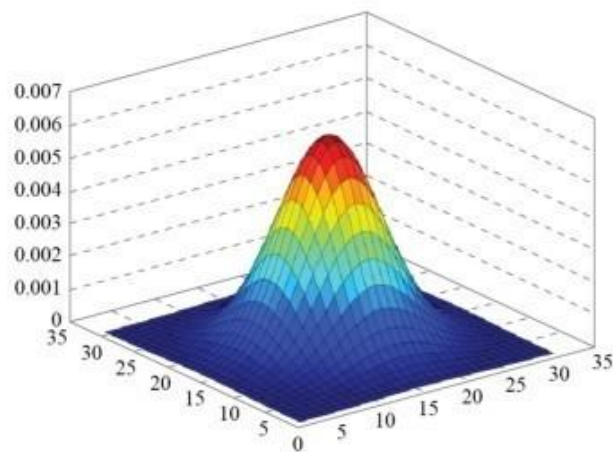
饼图(Pie Chart)

- ▶ 饼图采用了饼干的隐喻，用环状方式呈现各分量在整体中的比例。这种分块方式是环状树图等可视表达的基础



等值线图(Contour Map)

- ▶ 等值线图使用相等数值的数据点连线来表示数据的连续分布和变化规律



走势图(Sparkline)

- ▶ 走势图使用高度密集的折线图表达方式来展示数据随某一变量(时间、空间)的变化趋势。通常以折线图为基础，大小与文本相仿，往往直接嵌入在文本或表格中

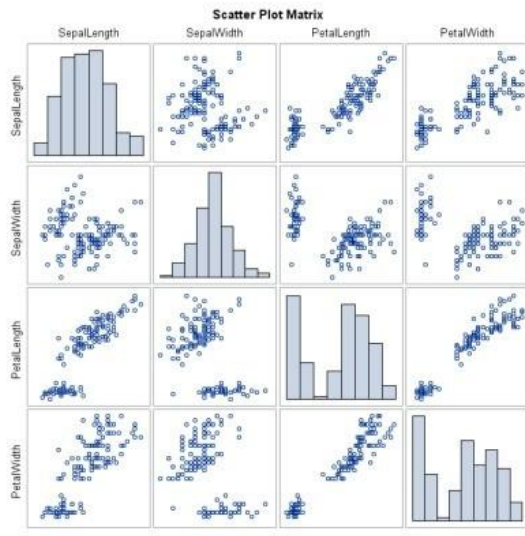
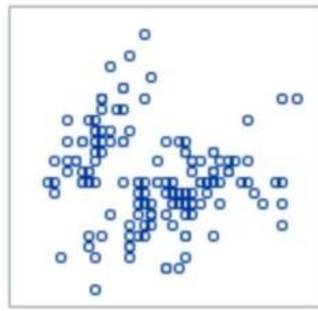
side of the flow of the text, i.e.:

 temperature **22.4° C**,

or be set inside it, , to visualise evidence on the spot. In a stacked or a table form such

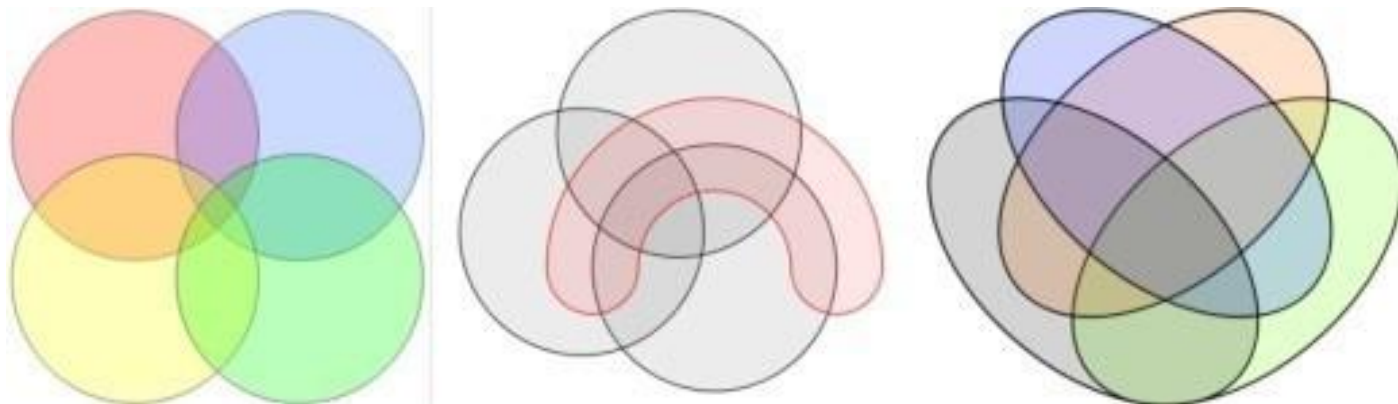
散点图(Scatter Plot)和散点图矩阵(Scatter Plot Matrix)

- 散点图中所有数据以点的形式出现在笛卡尔坐标系中，每个点所对应的横纵坐标即代表该数据在坐标轴所表示维度上的属性值大小
- 散点图矩阵是散点图的高维扩展，用来展现高维(大于二维)数据属性分布。可以通过采用尺寸、形状和颜色等来编码数据点的其他信息



维恩图(Venn Diagram)

- 维恩图使用平面上的封闭图形来表示数据集合间的关系。每个封闭图形代表一个数据集合，图形之间的交叠部分代表集合间的交集，图形之外的部分代表不属于该集合的数据部分



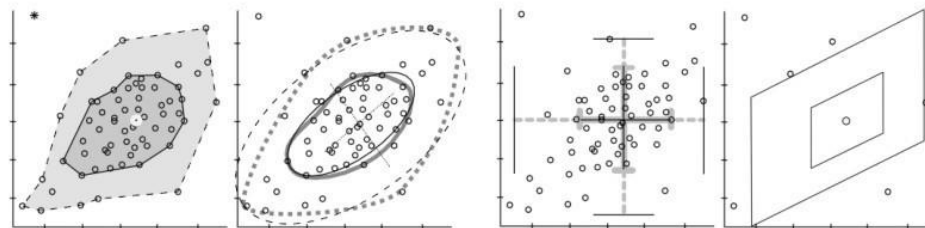
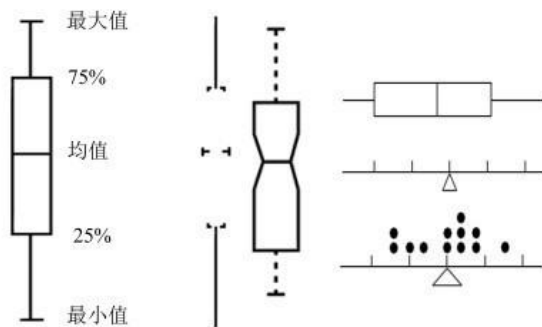
热力图(Heat Map)

- 热力图使用颜色来表达位置相关的二维数值数据大小



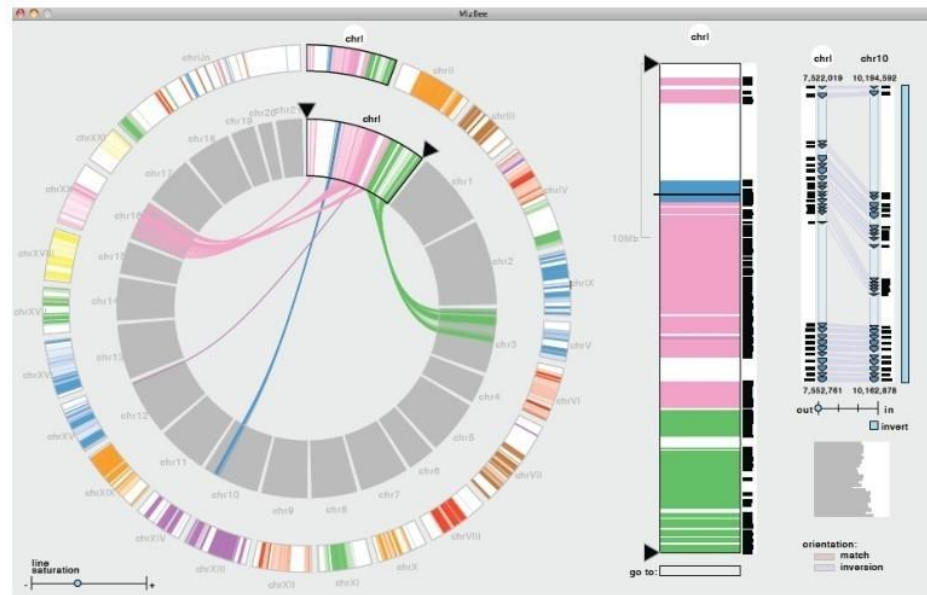
简单统计值标绘

- 盒须图：John Tukey发明，通过标绘简单的统计值来呈现一维和二维数据分布
 - 用一个长方形盒子表示数据的大致范围(数据值范围的25%~75%)，并在盒子中用横线标明均值的位置。同时，在盒子上部和下部分别用两根横线标注最大值和最小值



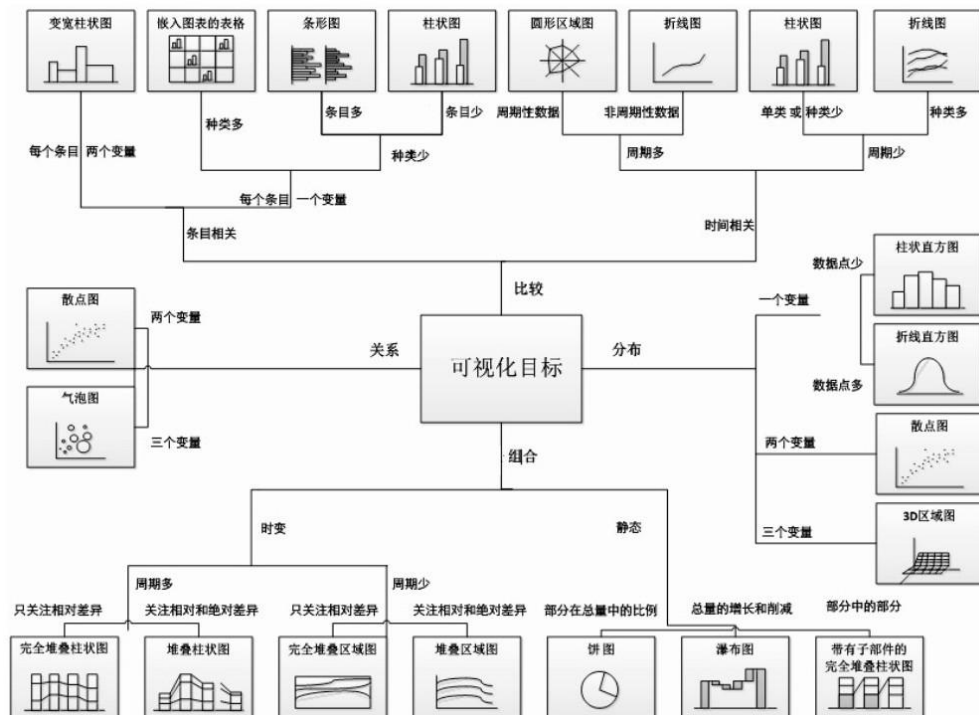
多视图协调关联

- 多视图协调关联(Multiple Coordinated Views)
将不同种类的绘图组合起来
 - 每个绘图单元可以展现数据某个方面的属性，并且通常允许用户进行交互分析，提升用户对数据的模式识别能力
- 多视图协调关联应用中，“选择”操作作为一种基本的探索方法
 - 对某个对象和属性进行“取消选择”
 - 选择属性的子集或对象的子集，以查看每个部分之间的关系的过程



统计可视化方法

► 根据分析需求可采用的统计可视化方法



4 可视化设计原则

可视化设计原则

➤ 挑战

- 过于复杂的可视化可能给用户带来理解上的麻烦，甚至可能引起误解和对原始数据信息的误读
- 缺少直观交互控制的可视化可能会阻碍用户以主观上更直观的方式获得可视化所包含的信息
- 美学因素影响用户对可视化设计的喜好或厌恶情绪

➤ 设计制作一个可视化视图的主要步骤

- 数据到图形元素(即标记)和视觉通道的映射
- 视图的选择，用户交互控制的设计
- 数据的筛选，即确定在有限的可视化视图空间中选择适当容量的信息进行编码，以避免在数据量过大情况下产生的视觉混乱

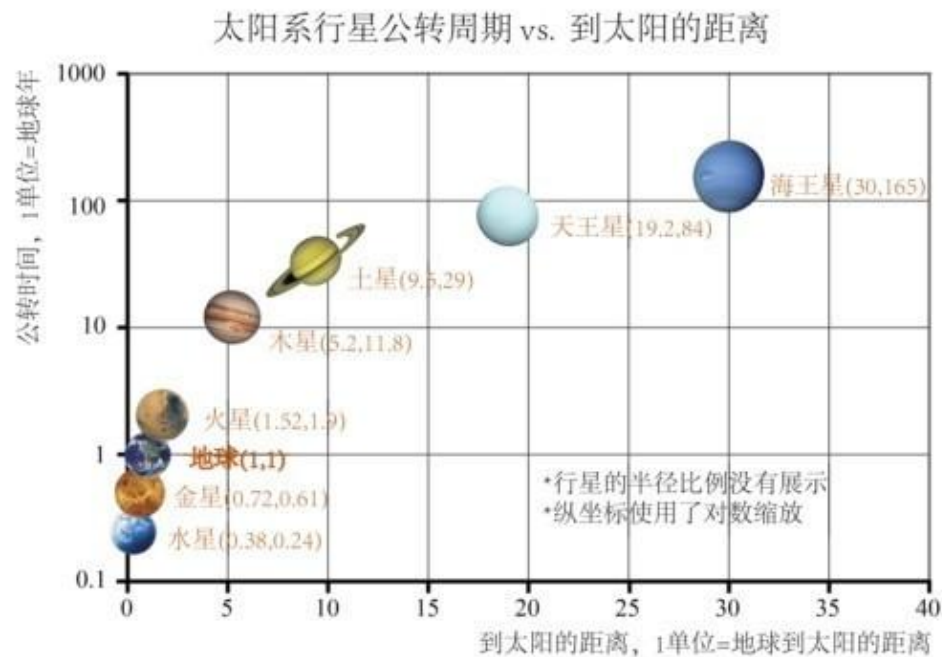
数据到可视化的直观映射

➤ 基本数据类型与视觉编码通道的关系



数据到可视化的直观映射

➤ 数据通常是基础数据类型组合，其可视化方法基于不同视觉编码通道的组合



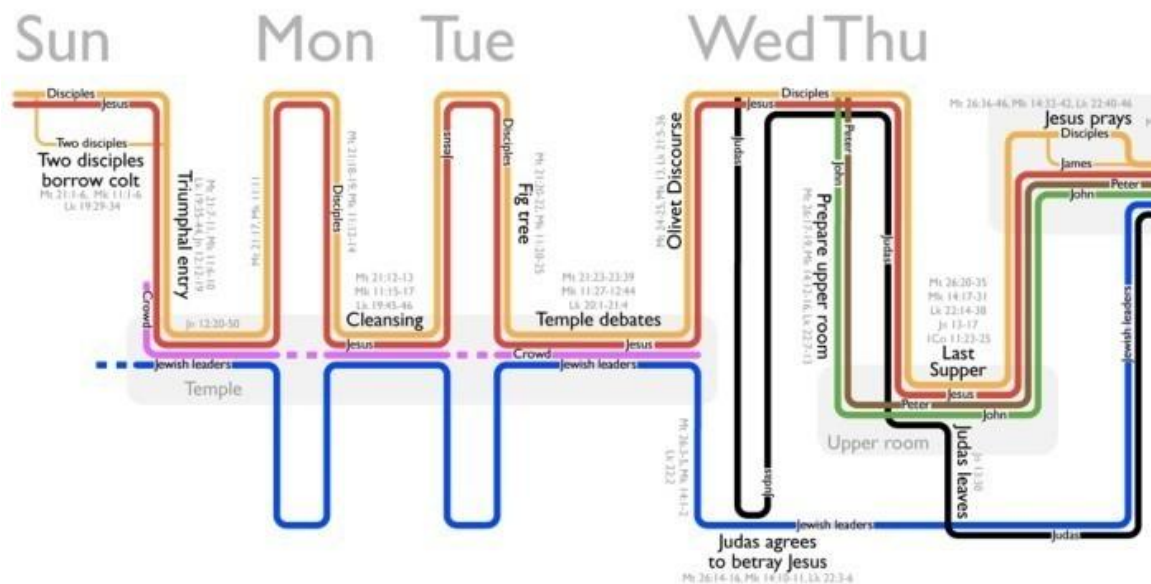
数据到可视化的直观映射

空间属性（如纬度和经度），映射到空间位置

如果两种数据属性存在时间上的关联，可以使用动画对其进行可视化

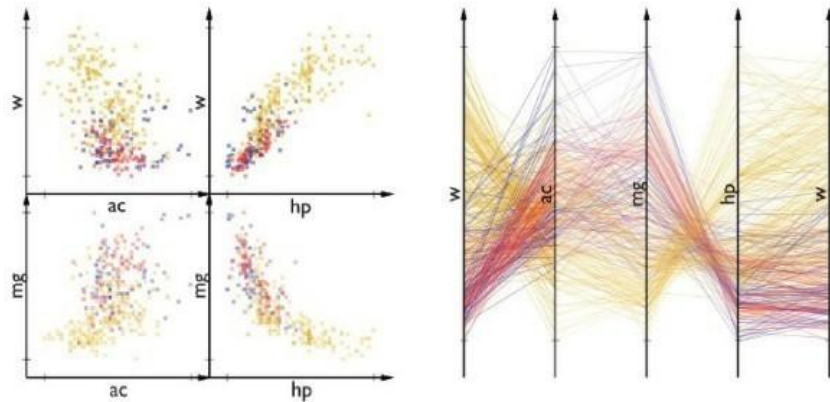
由于在许多文化中存在冷暖色调的传统，温度或密度映射为颜色直观易懂

数据到可视化的映射还要求设计者使用正确的视觉通道去编码数据信息



视图选择与交互设计

- 多视图时，视图选择使用简单而有效的方式进行命名和归类
 - 如通过标题标注
- 视图的交互主要包括
 - 滚动与缩放
 - 颜色映射的控制：调色盘
 - 数据映射方式的控制：保留用户对数据映射方式的控制交互
 - 数据缩放和裁剪工具
 - 细节层次(Level-Of-Detail, LOD)，隐藏或者突出数据的细节部分

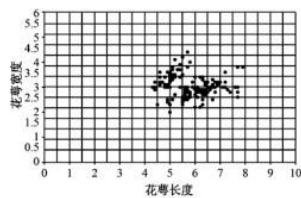


信息密度——数据的筛选

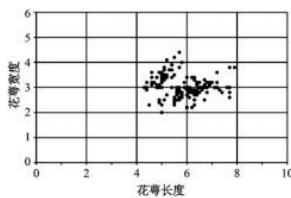
- 设计者必须决定可视化视图所需要包含的信息量
 - 一个好的可视化应当展示合适的信息，而不是越多越好
 - 失败的可视化案例主要存在两种极端情况：过少或过多地展示了数据的信息

美学因素

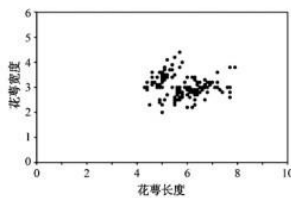
➤ 设计者需要认真仔细地对待可视化设计中的网格及其标注



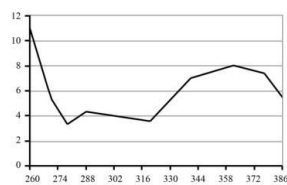
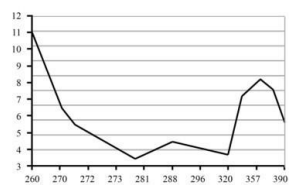
(a)



(b)



(c)

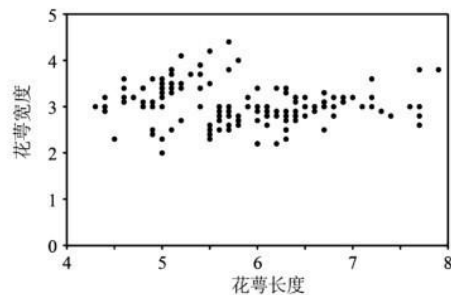
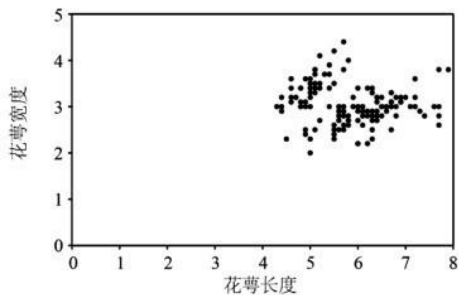


三 可视化设计的方法学中，相关因素

➤ 聚焦

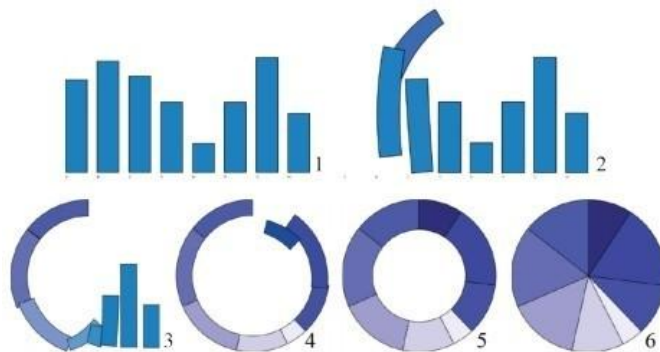
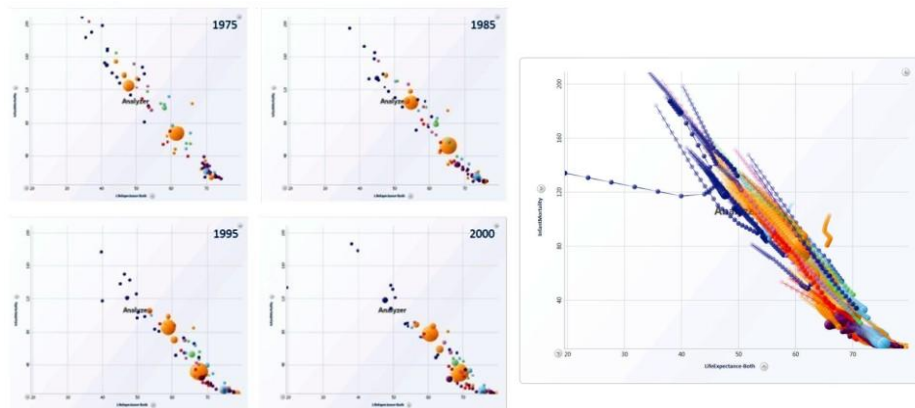
➤ 平衡

➤ 简单



动画与过渡

- 动画与过渡效果是通常被用于增加视图的丰富性与可理解性，或增加用户交互的反馈效果
 - 用时间换取空间，在有限的屏幕空间中展示更多的数据
 - 辅助不同可视化视图之间的转换与跟踪，或者辅助不同可视化视觉通道的变换
 - 增加用户在可视化系统中交互的反馈效果
 - 引起观察者注意力



可视化隐喻

- 在解释或者介绍人们不熟悉的事物和概念的时候，常常将其与一个人们所熟悉的事物进行比较来帮助理解，这样的手法称为隐喻 (Metaphor)
- 隐喻的设计包含三个层面：隐喻本体、隐喻喻体和可视化变量
- 本体和喻体之间存在某种关联或相似性
 - 如果本体和喻体具有不同的模态(语言、视觉、步态等)，隐喻也称为多模态隐喻
- 广告业、卡通和电影常用隐喻的手法表达其主题
- 时间隐喻和空间隐喻



颜色与透明度

- 颜色通常用于编码数据的分类或定序属性
- 当颜色的两种数据编码规则在用户所见的视图空间中存在相互遮挡时，可视化的设计者必须从中选择一种予以显示
 - 增加一个表示不透明度的 α 通道，用于表示离观察者更近的颜色对背景颜色的透过程度
- 颜色混合效果提供了数据可视化的上下文内容信息，方便对数据全局的把握

5 可视化理论发展

图形符号学

- 1983年Bertin使用符号学来描述图形，提出了信息的可视化编码原则，并严格地定义了二维图形及其对信息的表达过程
- 将图形系统严格区分为内容(所要表达的信息和数据)和载体(图形符号)。
 - 图形系统的定义建立在对图形符号的不同属性的理解和定义的基础之上

图形符号	点、线和面
位置变量	二维平面上的位置
视网膜变量	尺寸、数值、纹理、颜色、方向和形状

层次变量	关联性	选择性	有序性	定量性
平面	Y	Y	Y	Y
大小		Y	Y	Y
数值		Y	Y	
纹理	Y	Y	Y	
颜色	Y	Y		
方向	Y	Y		
形状	Y			

关系数据的图形表示

➤ 1986年Mackinlay提出一种可从数据库中自动提取信息并用图形方式显示的技术

图形符号	点、线和面
位置变量	一维、二维和三维
时间变量	动画
视网膜变量	颜色、形状、尺寸、饱和度、纹理和方向

编码技术	基本图形语言
视网膜变量	颜色、形状、尺寸、饱和度、纹理和方向
单个位置	横轴、竖轴
拼接位置	曲线图、柱状图、散点图
地图	道路图、地形图
链接	树、图、网络
杂项	饼图、维恩图

图形语法

- Wilkinson通过语法构造生成复杂的图形，以自底向上的方式组织最基本的元素形成更高级的元素
- 图形的构造过程分三个阶段：规范定义、组装和显示
 - 其中，规范定义是整个语法的基础，描述了不同图形对象间的转变和最终图形显示映射

数据	从数据集中生成变量的数据操作
转换	数据变量间的转换
框架	变量空间，包括变量间的操作
标度	标度转换
坐标	坐标系统
图形	图形及其美学属性
参考	用于图形对象间的对齐、分类和比较等

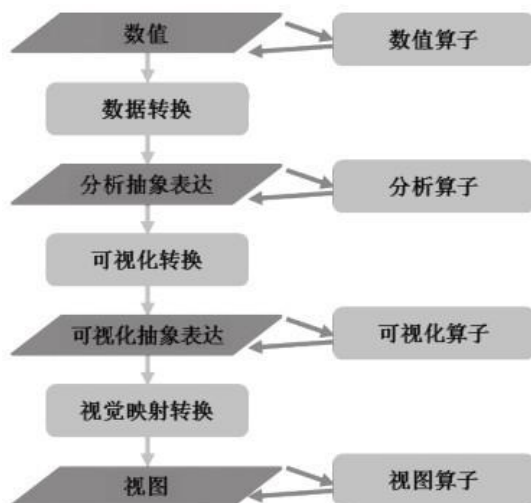
形 式	表 面	运 动	声 音	文 字
位置	颜色			
堆叠	色相			
躲避	亮度			
扰动	饱和度			
尺寸	纹理	方向	音调	
形状	图案	速度	声响	标签
多边形	粒度	加速	节奏	
符号	方向		语音	
图片	模糊			
旋转	透明度			

基于数据类型的研究

- Shneiderman从数据类型出发研究信息可视化过程，将数据分为7类
 - 一维
 - 二维
 - 三维
 - 时间
 - 多维
 - 树
 - 网络

基于数据状态模型的研究

- Chi[Chi2000]从数据状态模型(Data State Model)出发, 将可视化技术分解为四个数据转换阶段和三种数据转换操作。不同阶段分别对应不同的算子
- 整个可视化流程, 被分成四个不同的数据阶段: 数值、分析抽象表达、可视化抽象表达和视图
- 三种数据转换操作为: 数据转换、可视化转换和视觉映射转换



阶 段	描 述
数值	原始数据
分析抽象表达	关于数据（信息）的数据，又称元数据
可视化抽象表达	使用可视化技术，在屏幕上显示的可视信息
视图	可视化映射的最终产品，用户可通过其看到和解释所展示的图片

处理步骤	描 述
数据转换	从值中生成一些分析抽象表达（通常通过提取）
可视化转换	从分析抽象中获取可视化抽象形式，即为可视化内容
视觉映射转换	将信息转换为可视化形式，并显示为图形视图

多维关系数据库可视化分析系统

- Polaris系统支持大规模多维关系型数据库的查询、分析和可视化，完成关系型数据库中的主要挑战：发现结构和模式，获得因果关系
- Polaris提供了针对关系型数据库的接口，可快速递增式地生成图表式可视化

性 质	标识符号	有序型和数值型	数 值 型
形状	图元	     	
尺寸	矩形、圆、图元、文本	   	      
方向	矩形、线、文本	     	       
颜色	矩形、圆、线、图元、y轴条、x轴条、文本、甘特图等	          ... 	min  max