

第一讲：信息可视化简介

信息可视化

曹楠（教授），石洋（研究员）

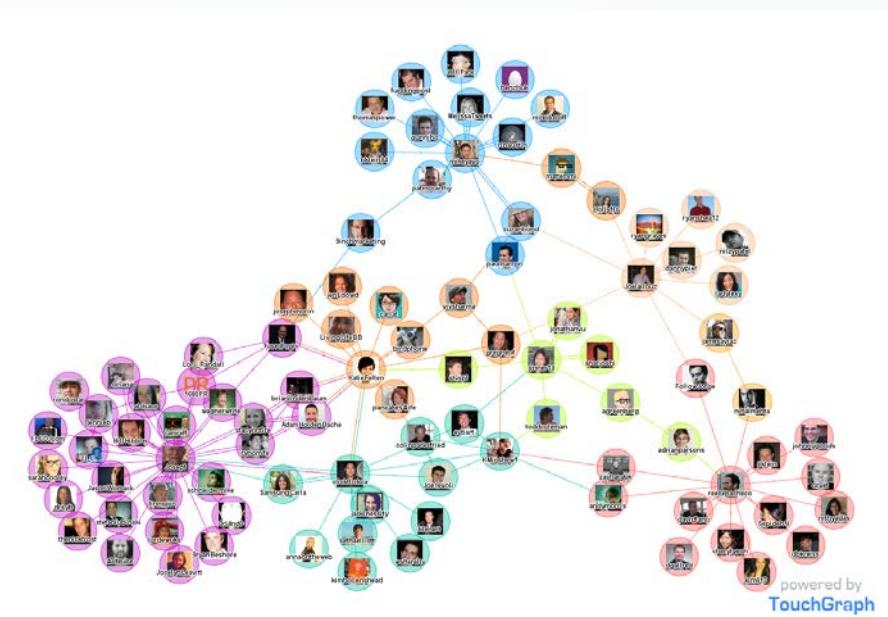
<https://idvxlab.com>

同济大学

课程大纲

- 什么是可视化？
- 为什么要可视化？
- 怎样对数据进行可视化？
- 如何对可视化进行评估？
- 怎样学习可视化？

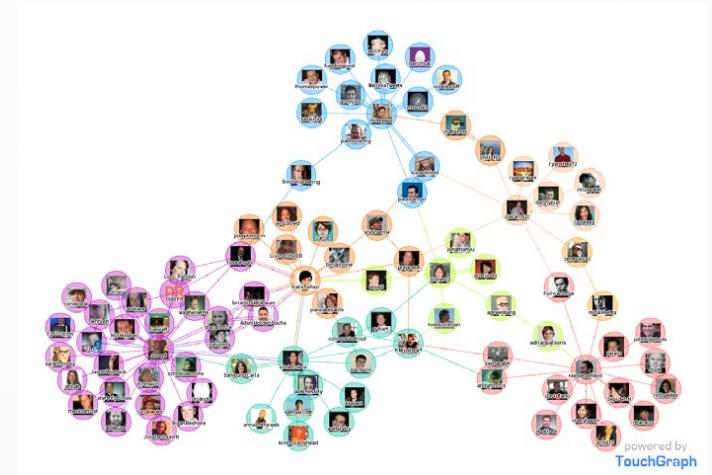
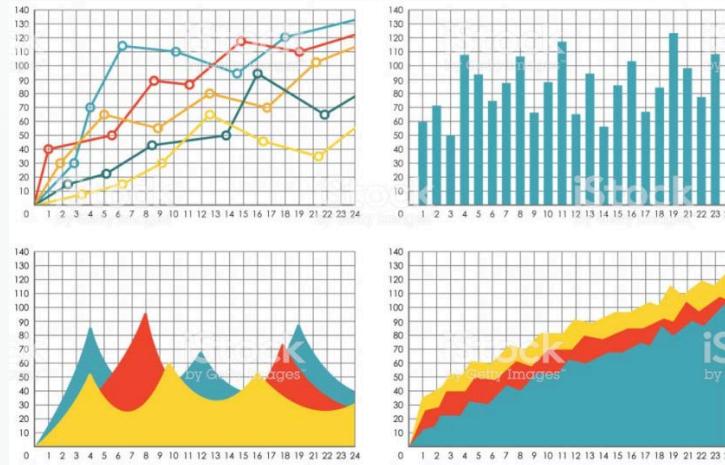
什么是可视化？



“任何通过创建 图像、图标、或动画 进行信息沟通的技术都叫做可视化”

Visualization is any **technique for creating images, diagrams, or animations to communicate a message** – Wikipedia ([https://en.wikipedia.org/wiki/Visualization_\(graphics\)](https://en.wikipedia.org/wiki/Visualization_(graphics)))

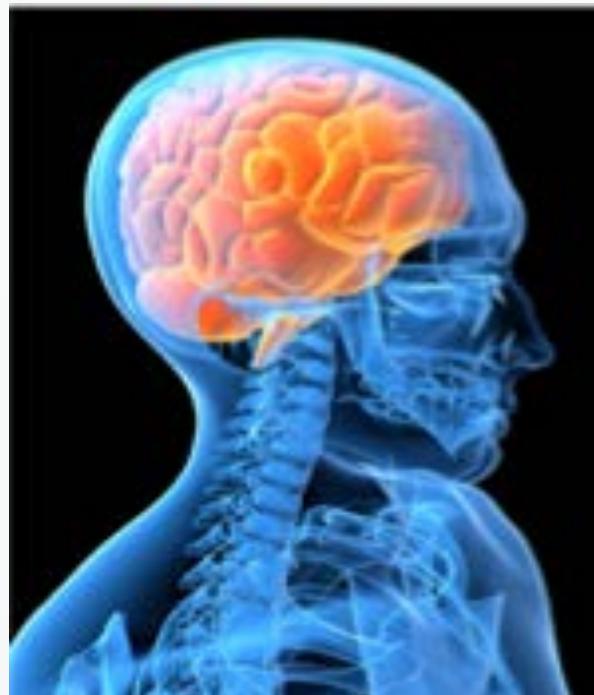
什么是数据可视化？



Anything that converts **data** into a visual representation (like charts, graphs, maps, sometimes even just tables)

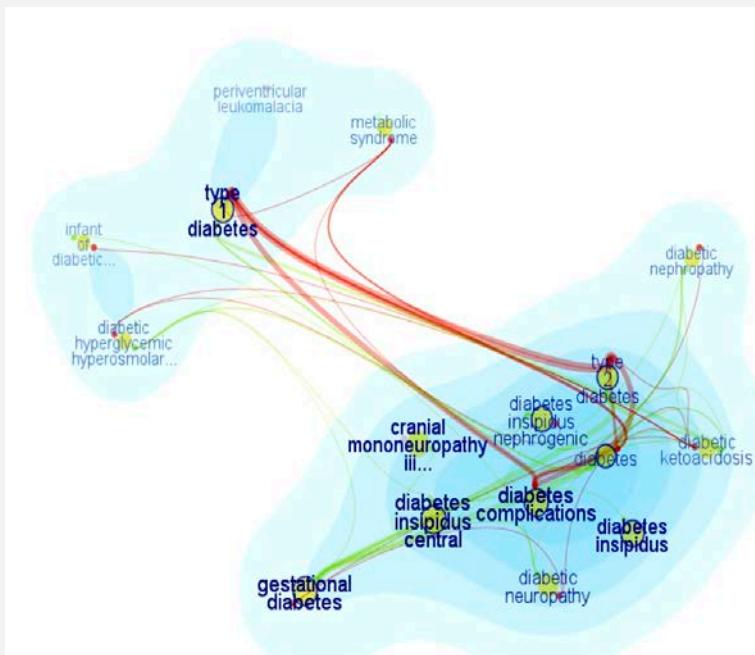
数据可视化领域的分支

科学可视化



对科学实验数据的直观展现

信息可视化



对抽象数据的直观展现

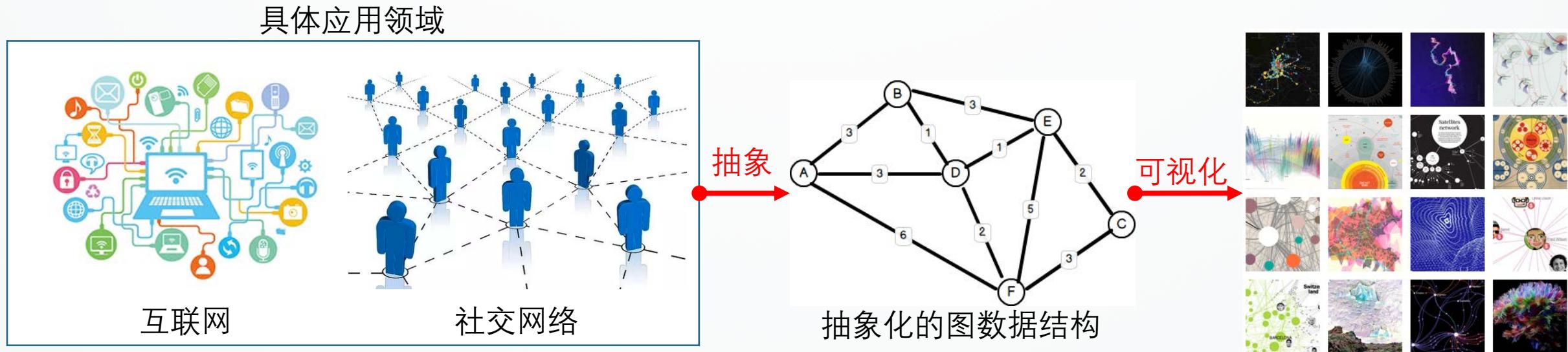
可视分析



对分析结果的直观展现，及交互式反馈，是一个跨领域的方向

什么是信息可视化？

- **信息可视化** - 是对**抽象数据** 直观的图形化展现
- **抽象数据** - 是对具体应用数据的高度概括与抽象，往往存储在数据结构当中

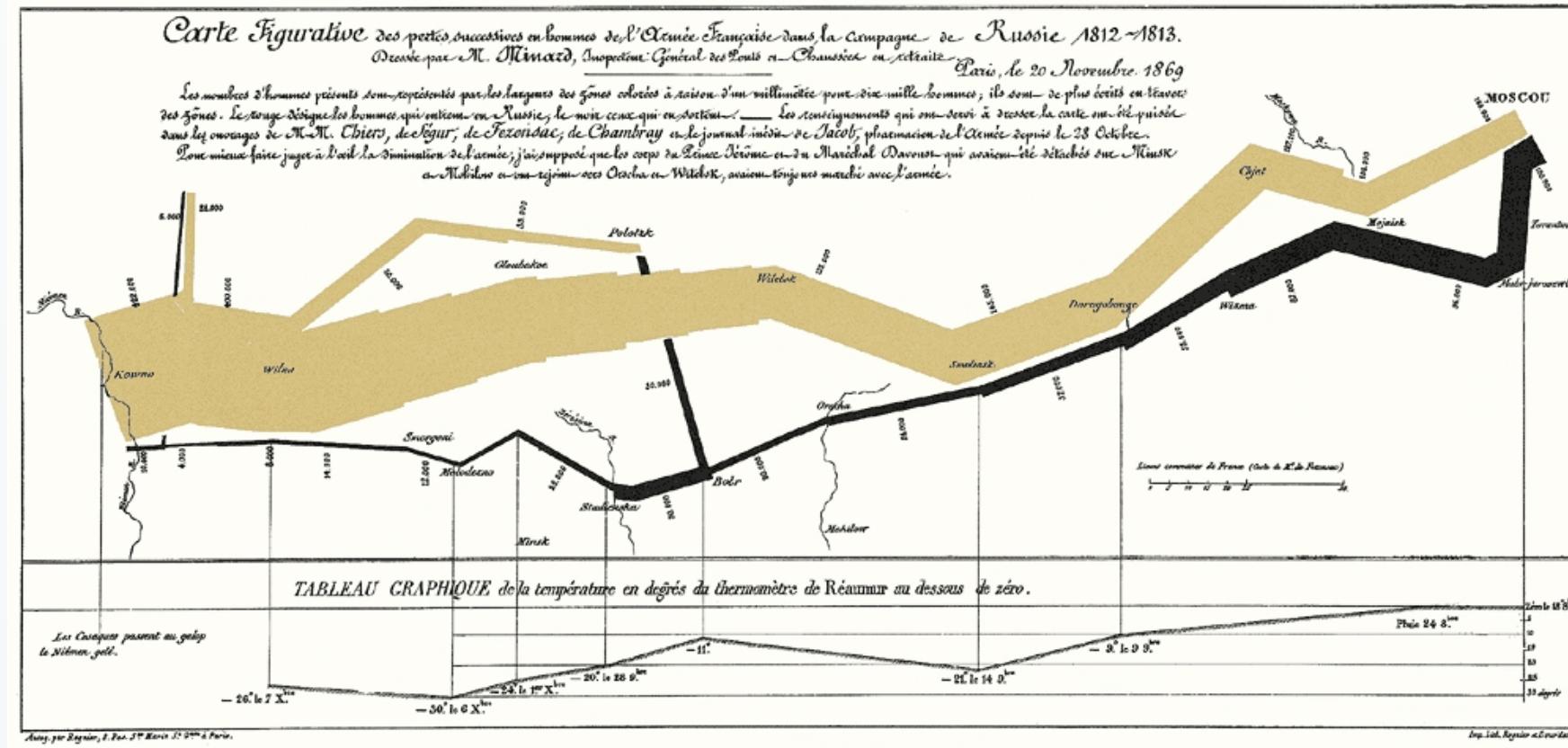


课程大纲

- 什么是可视化？
- **为什么要可视化？**
- 怎样对数据进行可视化？
- 如何对可视化进行评估？
- 怎样学习可视化？

为什么要可视化？

- 直观的展示信息



C.J. Minard, 1869

E.Tufte,Writings,Artworks,News

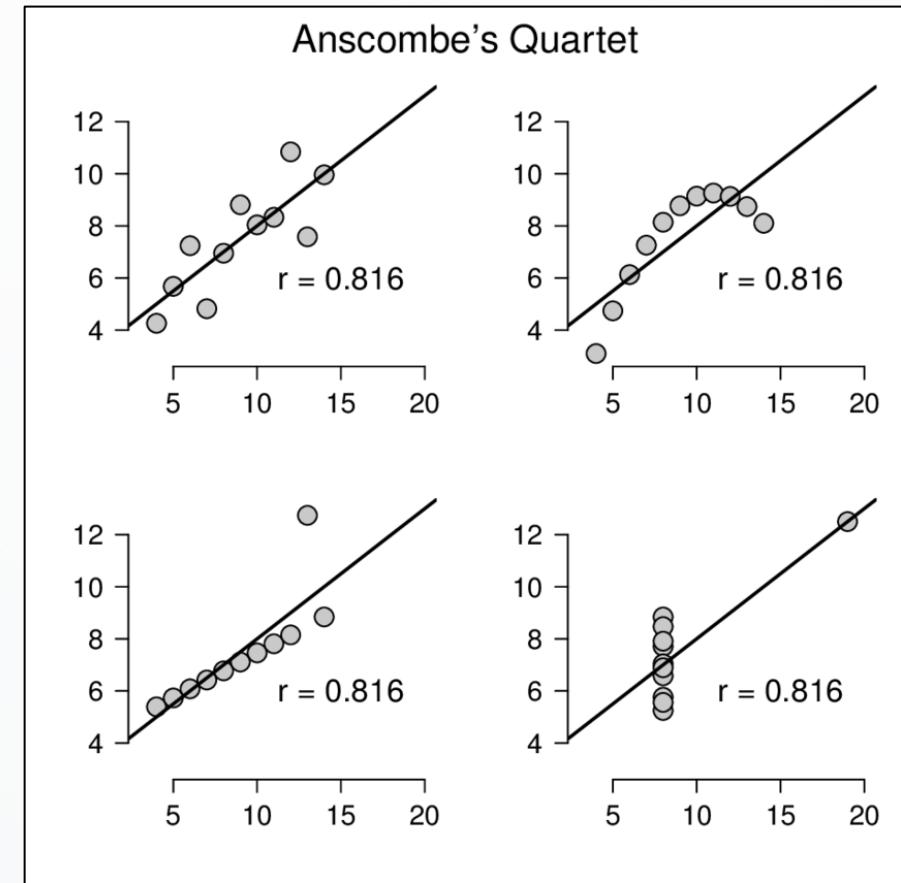
拿破仑进军莫斯科的路线图及败退路线，简单的可视化利用二维平面展现了包括空间、时间、气温、军队的人数等多个维度的信息

为什么要可视化？

- 直观的展示信息
- 帮助数据分析与推理

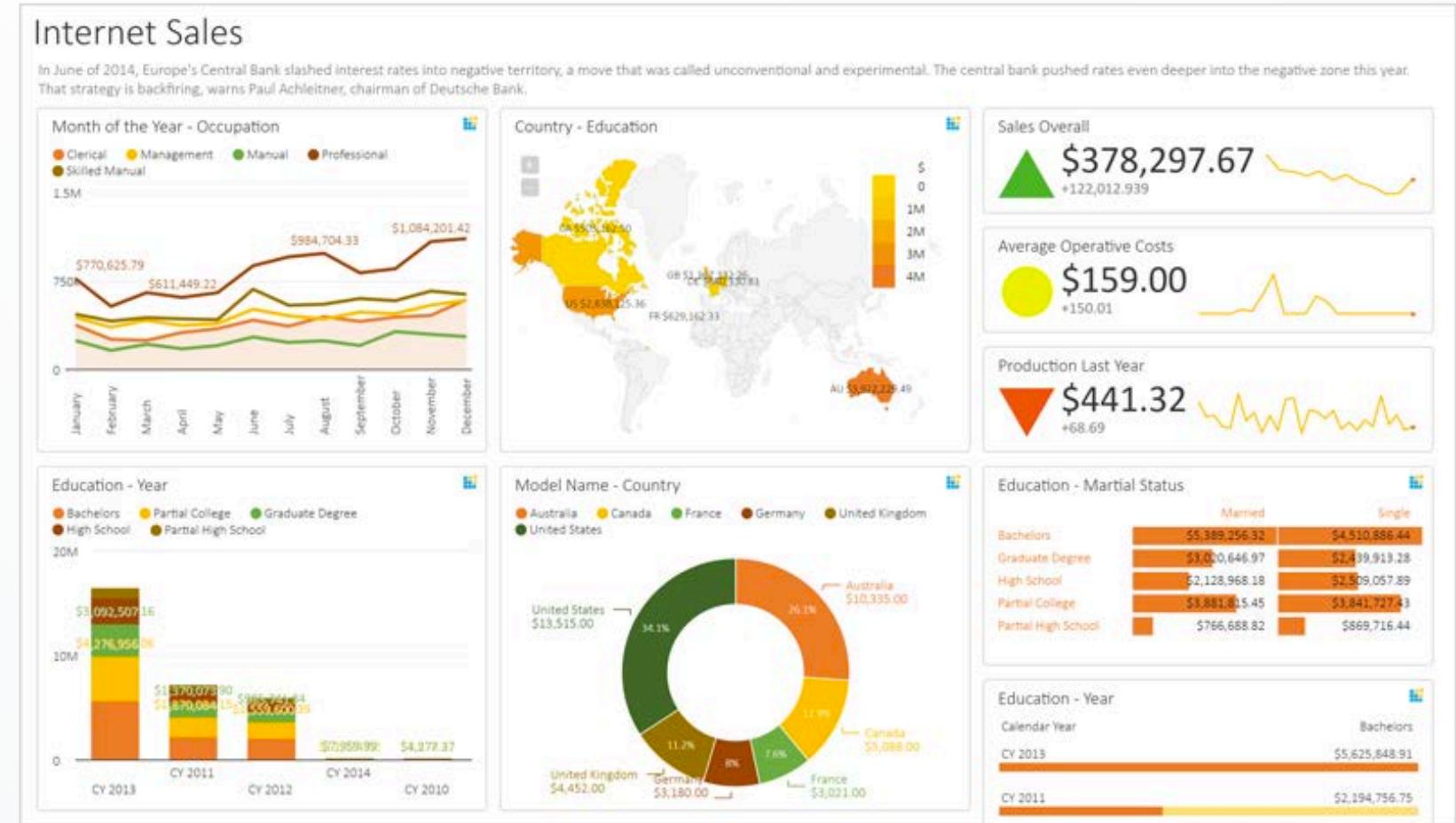
	Set A		Set B		Set C		Set D	
	X	Y	X	Y	X	Y	X	Y
0	10	8.04	10	9.14	10	7.46	8	6.58
1	8	6.95	8	8.14	8	6.77	8	5.76
2	13	7.58	13	8.74	13	12.74	8	7.71
3	9	8.81	9	8.77	9	7.11	8	8.84
4	11	8.33	11	9.26	11	7.81	8	8.47
5	14	9.96	14	8.10	14	8.84	8	7.04
6	6	7.24	6	6.13	6	6.08	8	5.25
7	4	4.26	4	3.10	4	5.39	19	12.50
8	12	10.84	12	9.13	12	8.15	8	5.56
9	7	4.82	7	7.26	7	6.42	8	7.91
10	5	5.68	5	4.74	5	5.73	8	6.89
mean	9.00	7.50	9.00	7.50	9.00	7.50	9.00	7.50
std	3.32	2.03	3.32	2.03	3.32	2.03	3.32	2.03
corr	0.82		0.82		0.82		0.82	
lin. reg.	$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$	

安斯庫姆四重奏 (Anscombe's Quartet)



为什么要可视化？

- 直观的展示信息
- 帮助数据分析与推理
- 促进对数据的沟通与交流

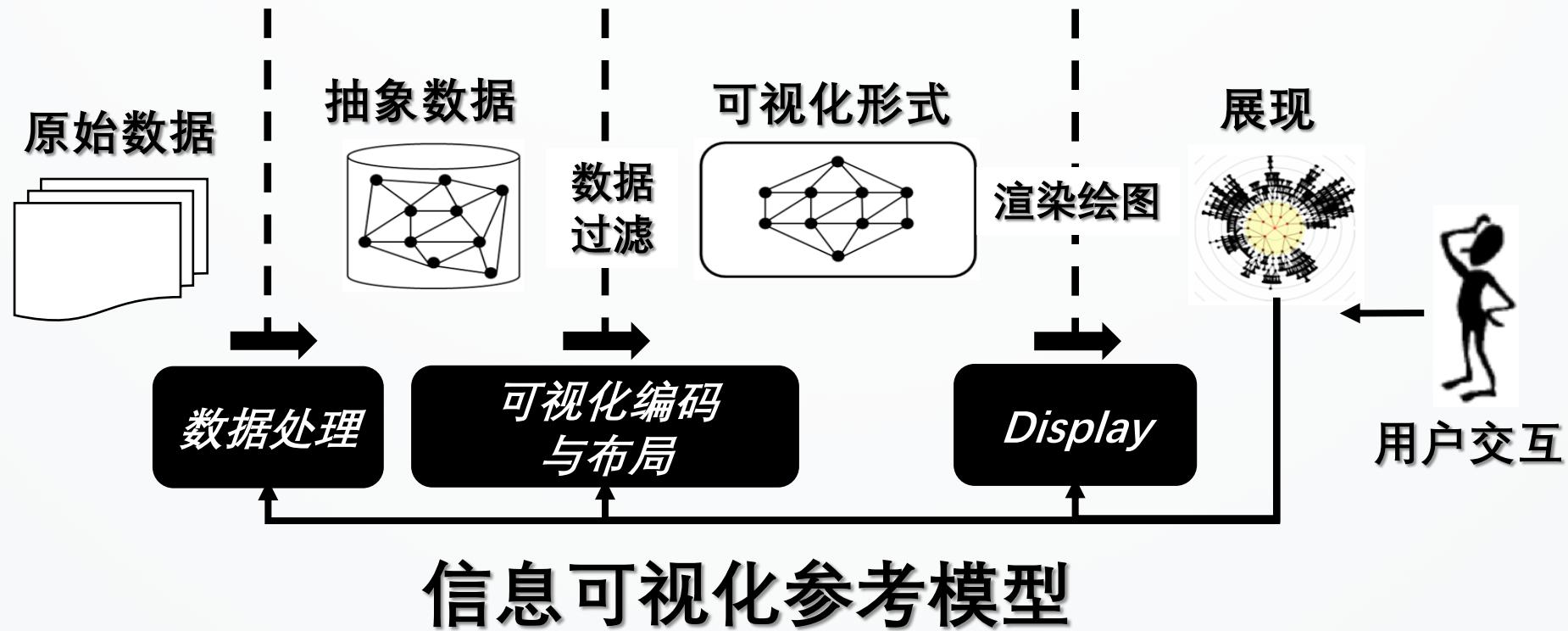


信息图、可视化大屏、及可视化报表，就好似对数据照了一张照片，让我们能够通过可视化展示便利的传达、沟通数据

课程大纲

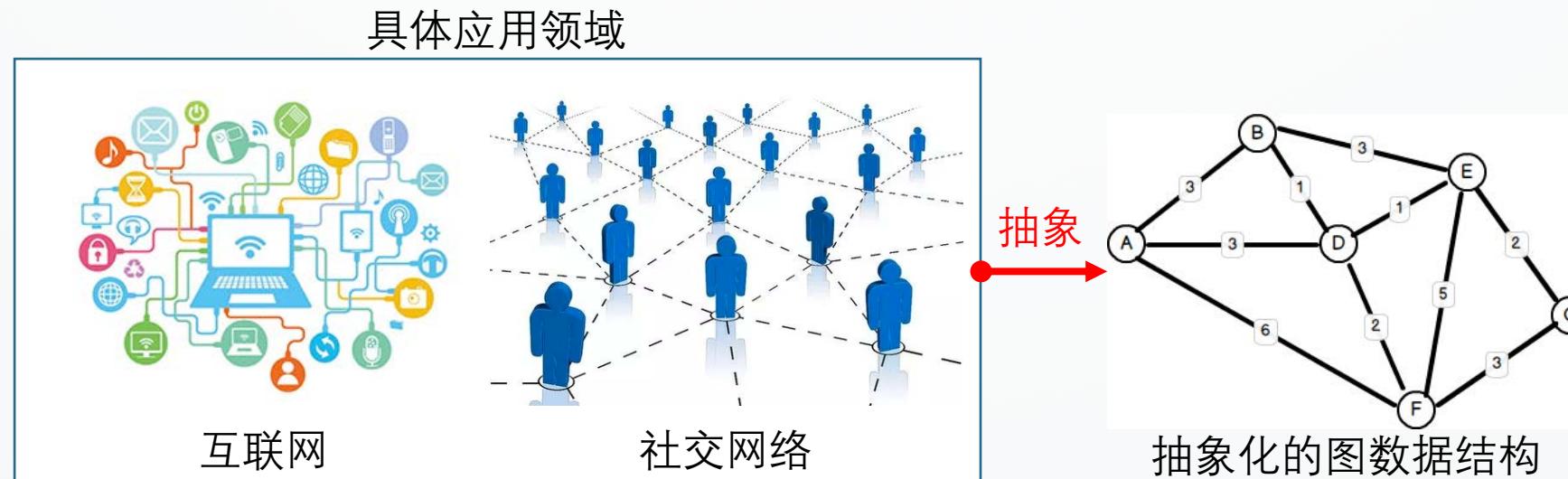
- 什么是可视化？
- 为什么要可视化？
- **怎样对数据进行可视化？**
- 如何对可视化进行评估？
- 怎样学习可视化？

怎样对数据进行可视化？



第一步：将数据转换为抽象形式

- **数据清洗**：借助于各种手段（过滤，采样，压缩，分类，聚类等）对数据进行清理，去除数据中的噪声，提取有用信息
- **数据变换**：改变数据的表达存储形式，例如 无结构数据的结构化转换
- **数据抽象**：提取应用数的抽象数据展现形式



第二步：可视化设计

- 通过设计确定数据的基本展现形式。可视化设计的基本准则：**信**（可视化需要准确表达数据）、**达**（可视化需要高效传达数据中的信息）、**雅**（可视化需要美观）
- 可视化编码：如何用图形符号（点、线、面）及属性（颜色、形状、大小等）展示数据元素及属性

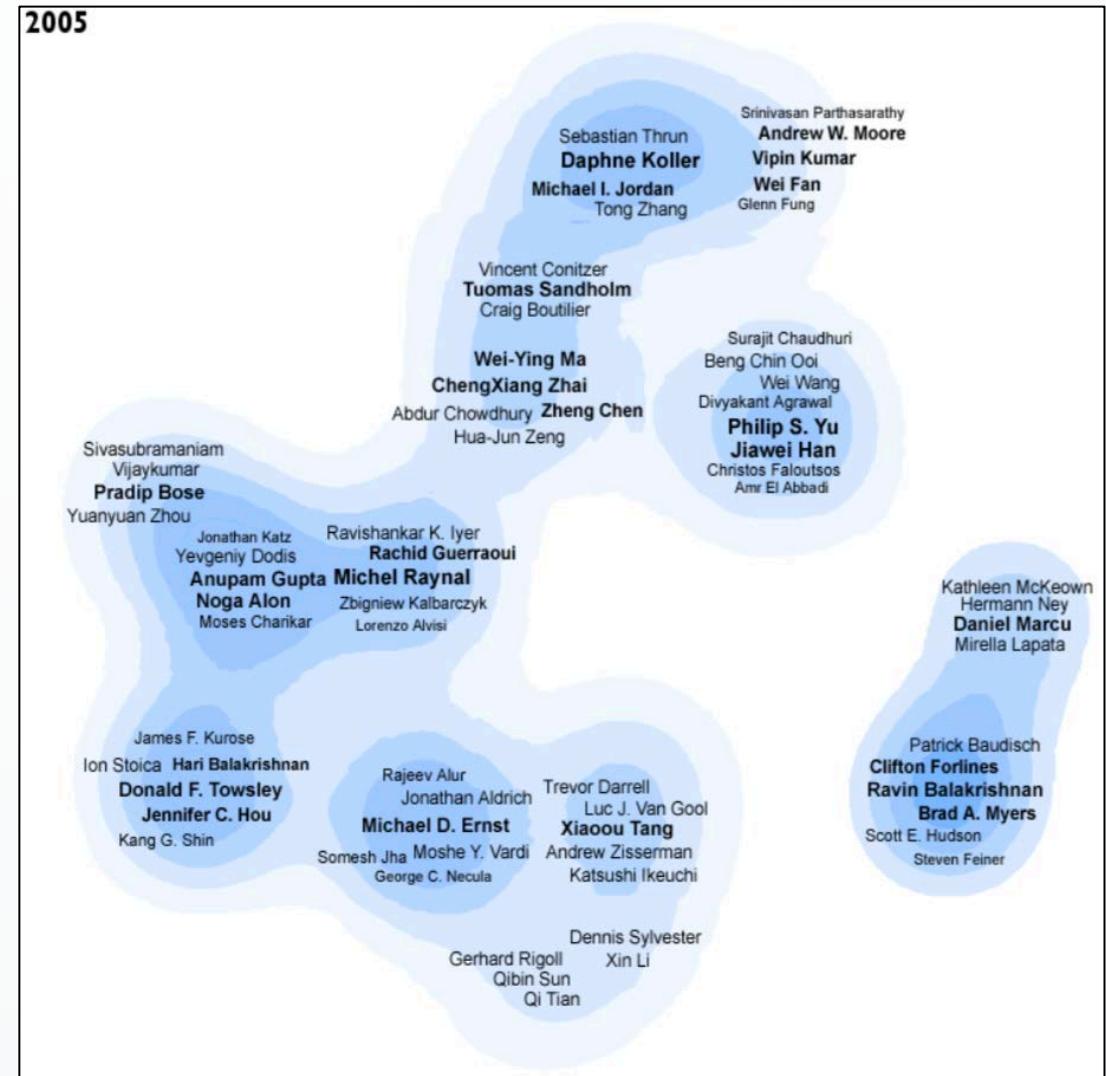
第三步：可视化布局



- 通过算法自动计算可视化元素在屏幕上的位置
- 往往是一个优化过程，需要考虑诸多信息呈现及美观方面的因素
- 例如左图所示的可视化布局需要避免点的遮挡与线的交叉

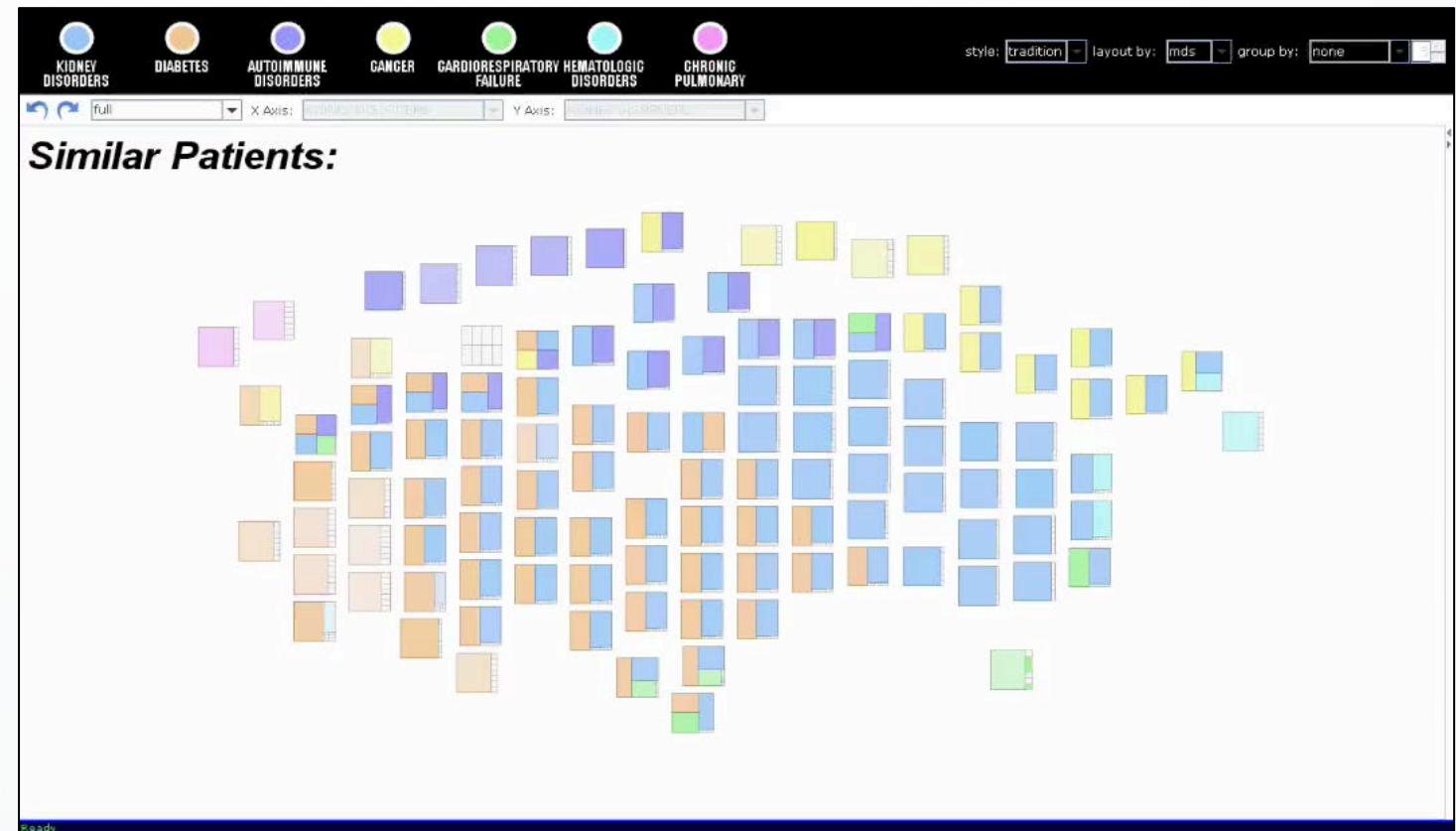
第四步：可视化渲染

- 根据设计及布局，将数据显示在屏幕上之上
- 渲染使用了计算机图形学中的相关技术，可视化往往是计算机前端的工作，渲染往往采用支持前端网页显示的渲染技术，例如，SVG，H5 Canvas, 以及 Web GL 等



第五步：可视化交互与动画

- 当数据量较大，无法完整显示在可视化视图当中时，交互能够帮助用户完整的浏览数据
- 在进行可是分析的过程中，当用户对可视化所展示的数据分析结果有异议时，交互能够帮助系统收集必要的用户反馈
- 当数据发生变化时，还需要设计开发相应的动画，用于平滑的展现数据中的变动，以便用户追踪变动的数据，并分析相应的原因



常用的可视化开发工具

D3.js

<https://d3js.org/>



The screenshot shows the official D3.js website. At the top, there is a navigation bar with links to 'Overview', 'Examples', 'Documentation', and 'Source'. Below the navigation is the D3.js logo, which consists of three stylized orange and red 'D's. To the right of the logo is the text 'Data-Driven Documents'. In the top right corner, there is an orange button with the text 'Fork me on GitHub'. The main content area features a large, dense collage of various data visualizations, including hexagonal maps, treemap diagrams, network graphs, and other complex data structures. Below this collage, there is a brief description of what D3.js is and a link to see more examples.

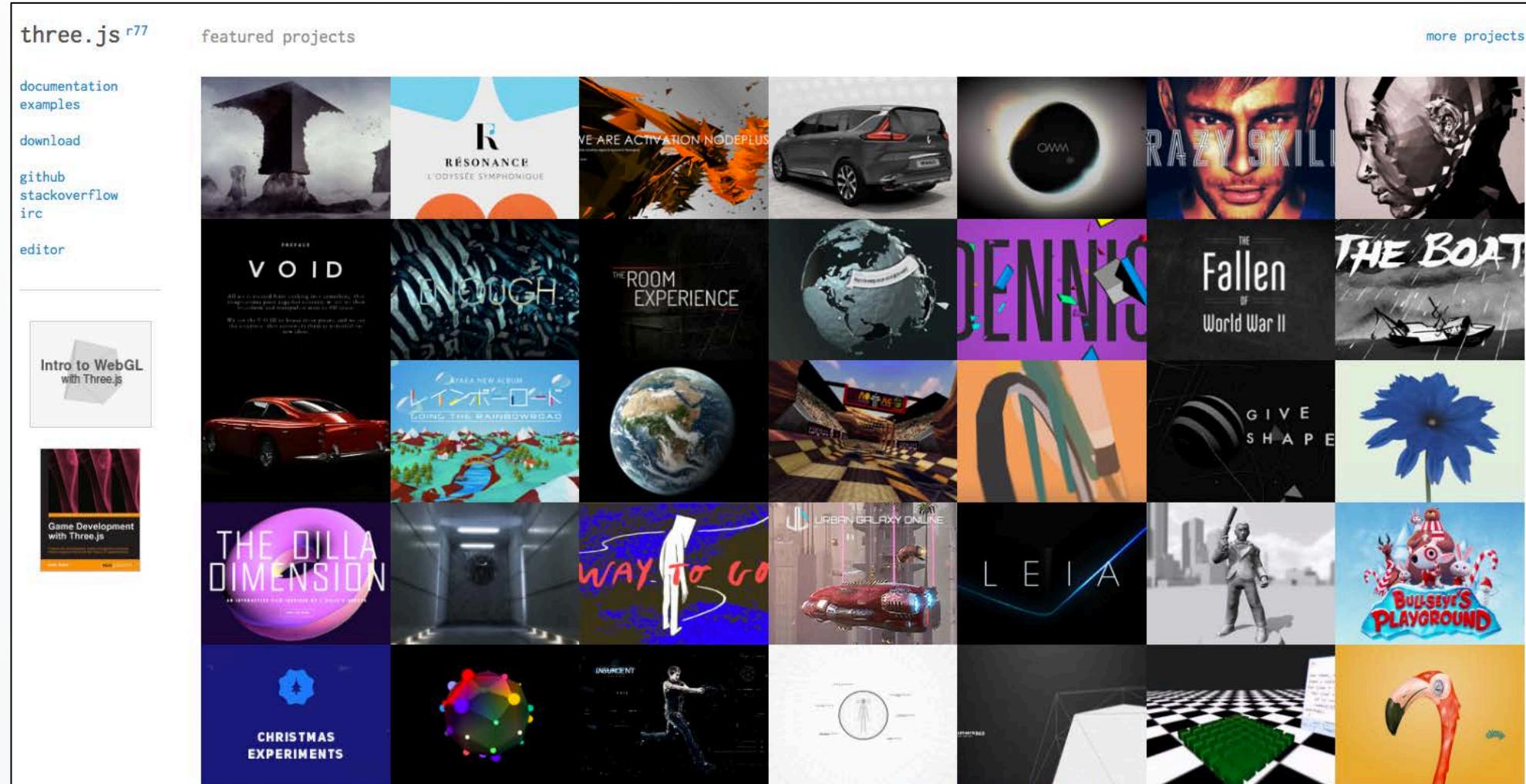
D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

[See more examples.](#)

常用的可视化开发工具

Three.js

<http://threejs.org/>

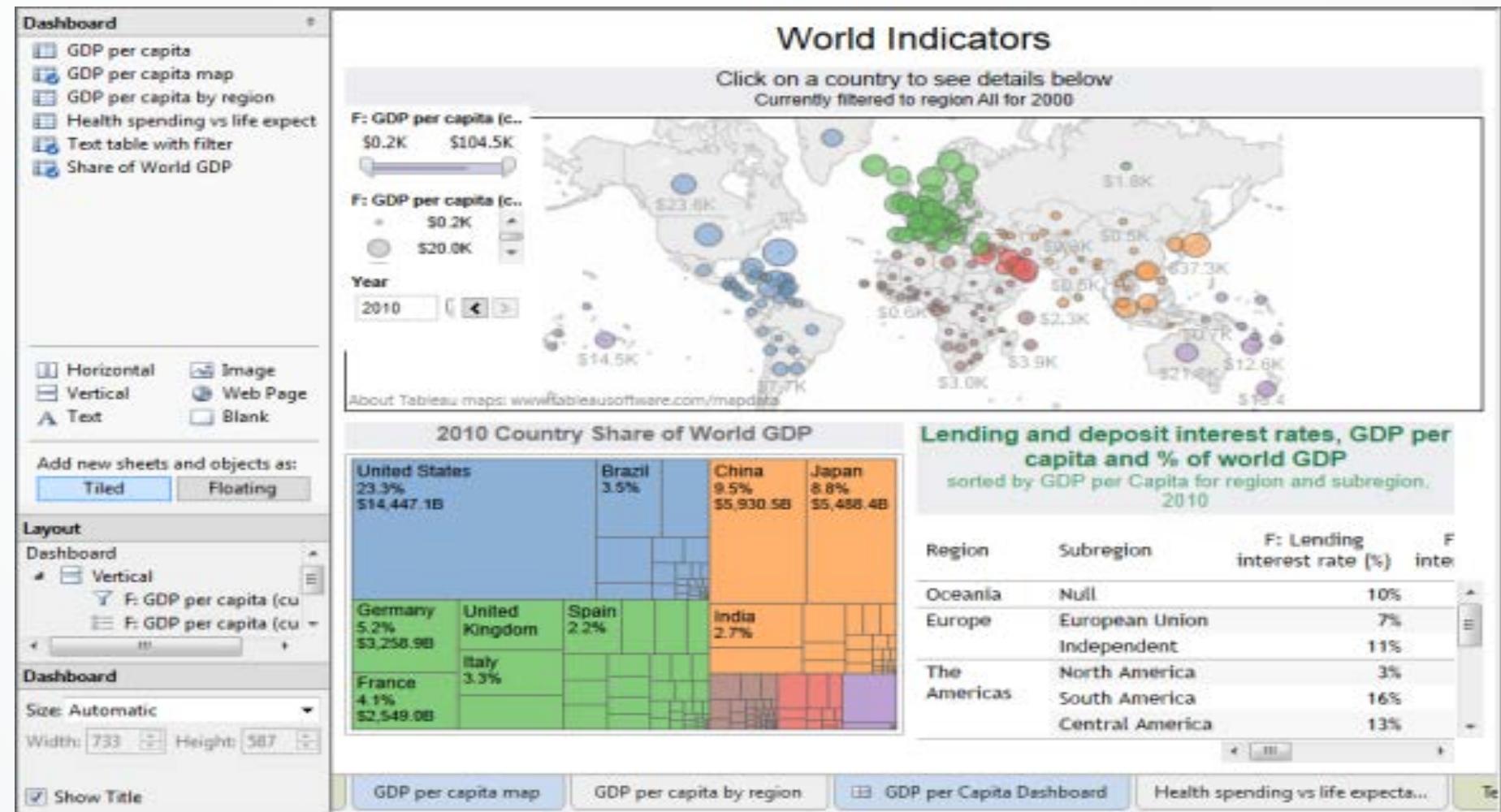


常用的可视化开发工具

Tableau

<https://www.tableau.com>

<https://public.tableau.com/s/>



课程大纲

- 什么是可视化？
- 为什么要可视化？
- 怎样对数据进行可视化？
- 如何对可视化进行评估？
- 怎样学习可视化？

如何对可视化进行评估

设计准则比对 (Design Principles)

非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)

如何对可视化进行评估

- 设计准则比对 (Design Principles)
- 非正式用户调研 (Informal Study)
- 受控用户调研 (Controlled User Study)
- 实际应用及案例分析 (Case Study)
- 理论分析 (Theoretical Analysis)
- 基准化分析法 (Benchmark)

检验可视化的设计
是否符合一般性的设计准则

如何对可视化进行评估

设计准则比对 (Design Principles)

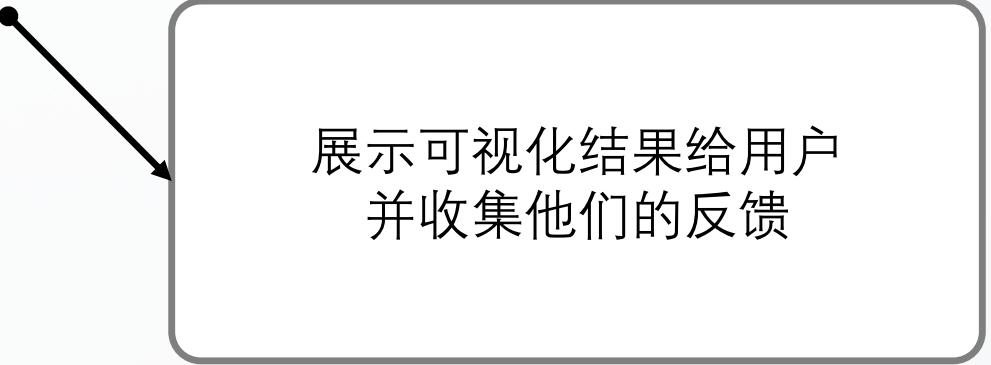
非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)



展示可视化结果给用户
并收集他们的反馈

如何对可视化进行评估

设计准则比对 (Design Principles)

非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)

设计比对试验，计算用户在完成相同任务时耗费的时间及任务的正确率是否有显著差异

如何对可视化进行评估

设计准则比对 (Design Principles)

非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)

将可视化应用于实际案例，通过展示所得到的数据模式，及有趣发现证明可视化设计的有用性

如何对可视化进行评估

设计准则比对 (Design Principles)

非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)

分析可视化布局算法复杂度、及
可视化系统的综合性能等

如何对可视化进行评估

设计准则比对 (Design Principles)

非正式用户调研 (Informal Study)

受控用户调研 (Controlled User Study)

实际应用及案例分析 (Case Study)

理论分析 (Theoretical Analysis)

基准化分析法 (Benchmark)

与现有最佳解决
方案进行直观对比

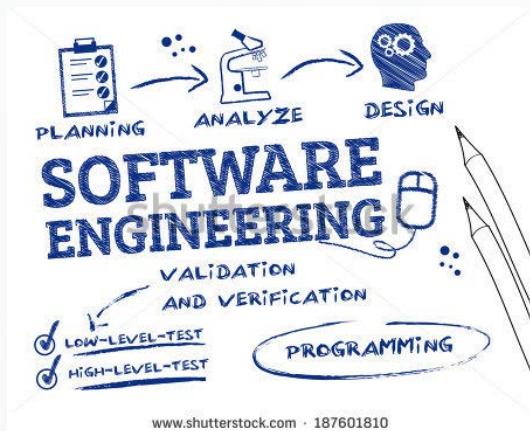
课程大纲

- 什么是可视化？
- 为什么要可视化？
- 怎样对数据进行可视化？
- 如何对可视化进行评估？
- 怎样学习可视化？

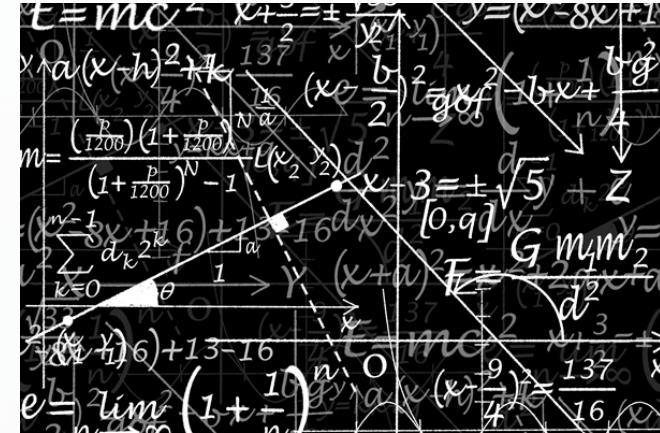
需要学习什么？



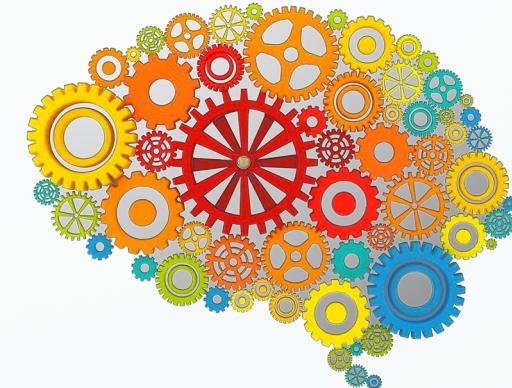
设计 (Web, UI & UX, Visual Design)



编程

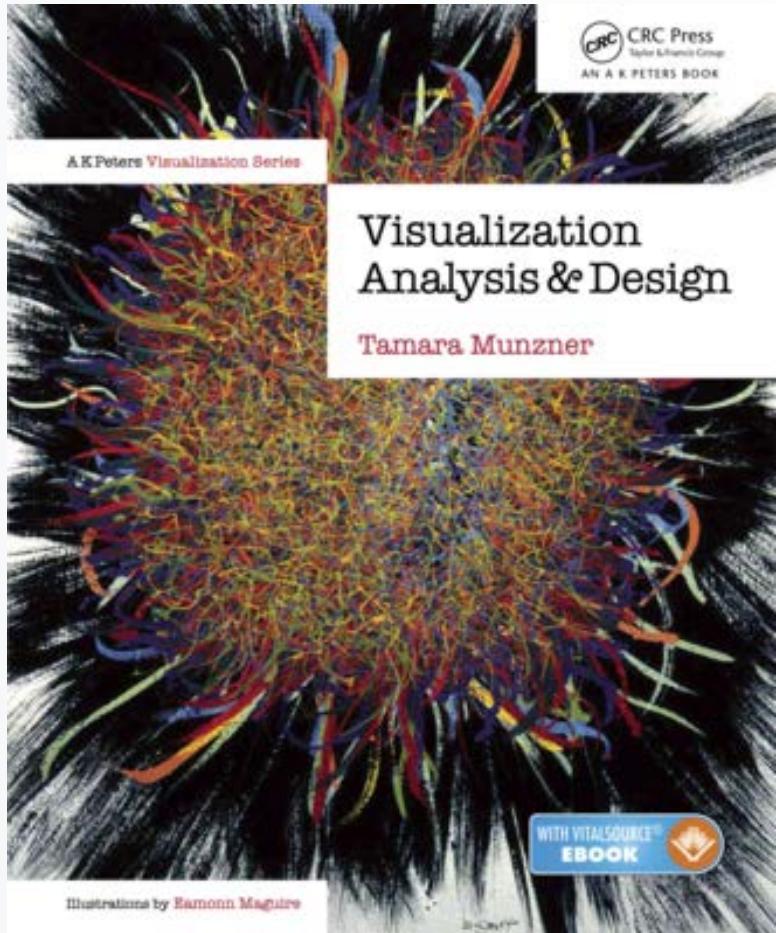


数学与算法

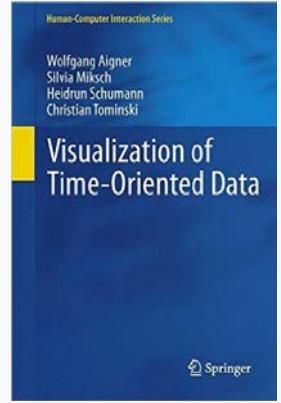
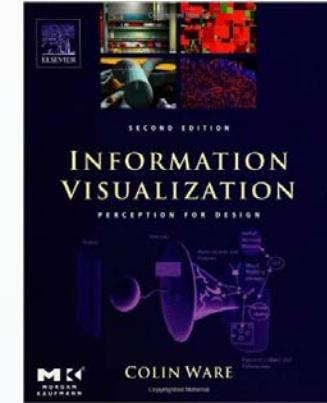
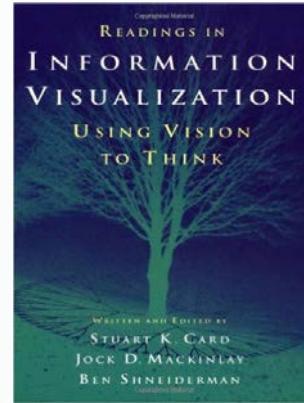
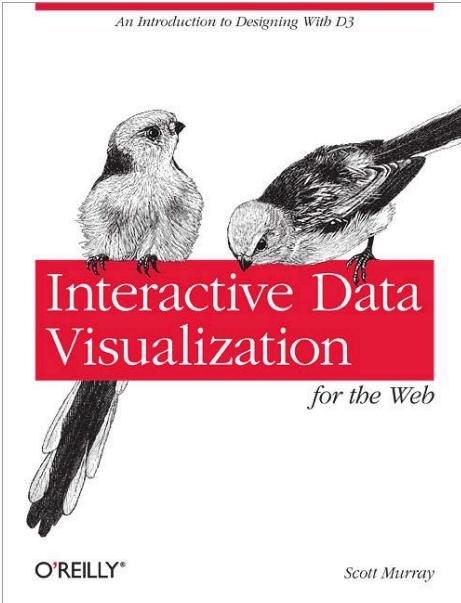


技术 (机器学习, 数据挖掘)

有哪些参考书？



可视化编程



一些其他的参考书

<http://www.cs.ubc.ca/~tmm/vadbook/>

如何获得前沿知识？

顶级学术期刊

年度学术会议



IEEE InfoVis / VAST / SciVis Conferences

IEEE TRANSACTIONS ON
VISUALIZATION AND COMPUTER GRAPHICS

A CATALYST FOR ADVANCED VISUALIZATION

From the September 2013 issue
Scheduling in Heterogeneous Computing Environments for Proximity Queries

By Duksu Kim, Jinkyu Lee, Junghwan Lee, Insik Shin, John Kim, Sung-Eui Yoon



We present a novel, linear programming (LP)-based scheduling algorithm that exploits heterogeneous multicore architectures such as CPUs and GPUs to accelerate a wide variety of proximity queries. To represent complicated performance relationships between heterogeneous architectures and different computations of proximity queries, we propose a simple, yet accurate model that measures the expected running time of these computations. Based on this model, we formulate an optimization problem that minimizes the largest time spent on computing resources, and propose a novel, iterative LP-based scheduling algorithm. Since our method is general, we are able to apply our method into various proximity queries used in five different applications that have different characteristics. Our method achieves an order of magnitude performance improvement by using four different GPUs and two hexa-core CPUs over using a hexa-core CPU only.

IEEE Transaction on Visualization and Computer Graphics

课程总结

- 本节课
 - 什么是可视化？
 - 为什么要可视化？
 - 怎样对数据进行可视化？
 - 如何对可视化进行评估？
 - 怎样学习可视化？
- 下节课
 - 可视化的基本设计准则