

# 数据可视化

# 提纲

- **1 可视化概述**
- **2 可视化工具**
- **3 可视化典型案例**
- **4 可视化工具实践**

# 1 可视化概述

- 1.1 什么是数据可视化
- 1.2 可视化的发展历程
- 1.3 可视化的重要作用

# 1.1 什么是数据可视化

- 数据可视化是指将大型数据集中的数据以图形图像形式表示，并利用数据分析和开发工具发现其中未知信息的处理过程
- 数据可视化技术的基本思想是将数据库中每一个数据项作为单个图元素表示，大量的数据集构成数据图像，同时将数据的各个属性值以多维数据的形式表示，可以从不同的维度观察数据，从而对数据进行更深入的观察和分析

## 1.2 可视化的发展历程

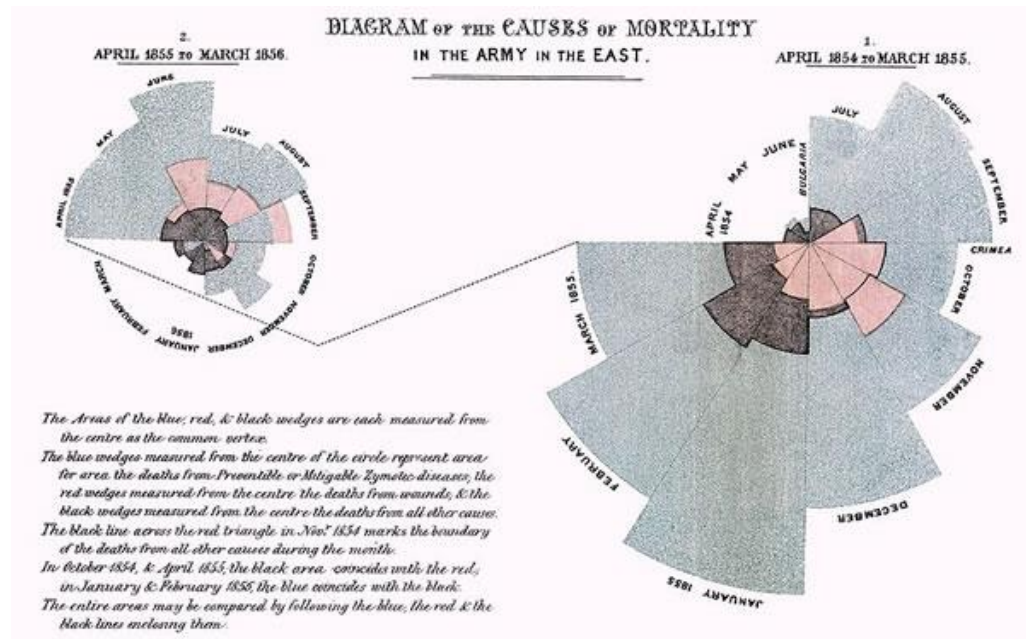
霍乱地图分析了霍乱患者分布与水井分布之间的关系，发现在有一口井的供水范围内患者明显偏多，据此找到了霍乱爆发的根源是一个被污染的水泵



图1 反映霍乱患者分布与水井分布的地图

# 1.2 可视化的发展历程

数据可视化历史上的另一个经典之作是1857年“提灯女神”南丁格尔设计的“鸡冠花图”(又称玫瑰图),它以图形的方式直观地呈现了英国在克里米亚战争中牺牲的战士数量和死亡原因,有力地说明了改善军队医院的医疗条件对于减少战争伤亡的重要性



## 1.2 可视化的发展历程

- 20世纪50年代，随着计算机的出现和计算机图形学的发展，人们可以利用计算机技术在电脑屏幕上绘制出各种图形图表，可视化技术开启了全新的发展阶段。最初，可视化技术被大量应用于统计学领域，用来绘制统计图表，比如圆环图、柱状图和饼图、直方图、时间序列图、等高线图、散点图等，后来，又逐步应用于地理信息系统、数据挖掘分析、商务智能工具等，有效促进了人类对不同类型数据的分析与理解
- 随着大数据时代的到来，每时每刻都有海量数据在不断生成，需要我们对数据进行及时、全面、快速、准确的分析，呈现数据背后的价值，这就更需要可视化技术协助我们更好地理解和分析数据，可视化成为大数据分析最后的一环和对用户而言最重要的一环

# 1.3 可视化的重要作用

在大数据时代，可视化技术可以支持实现多种不同的目标：  
(1) 观测、跟踪数据

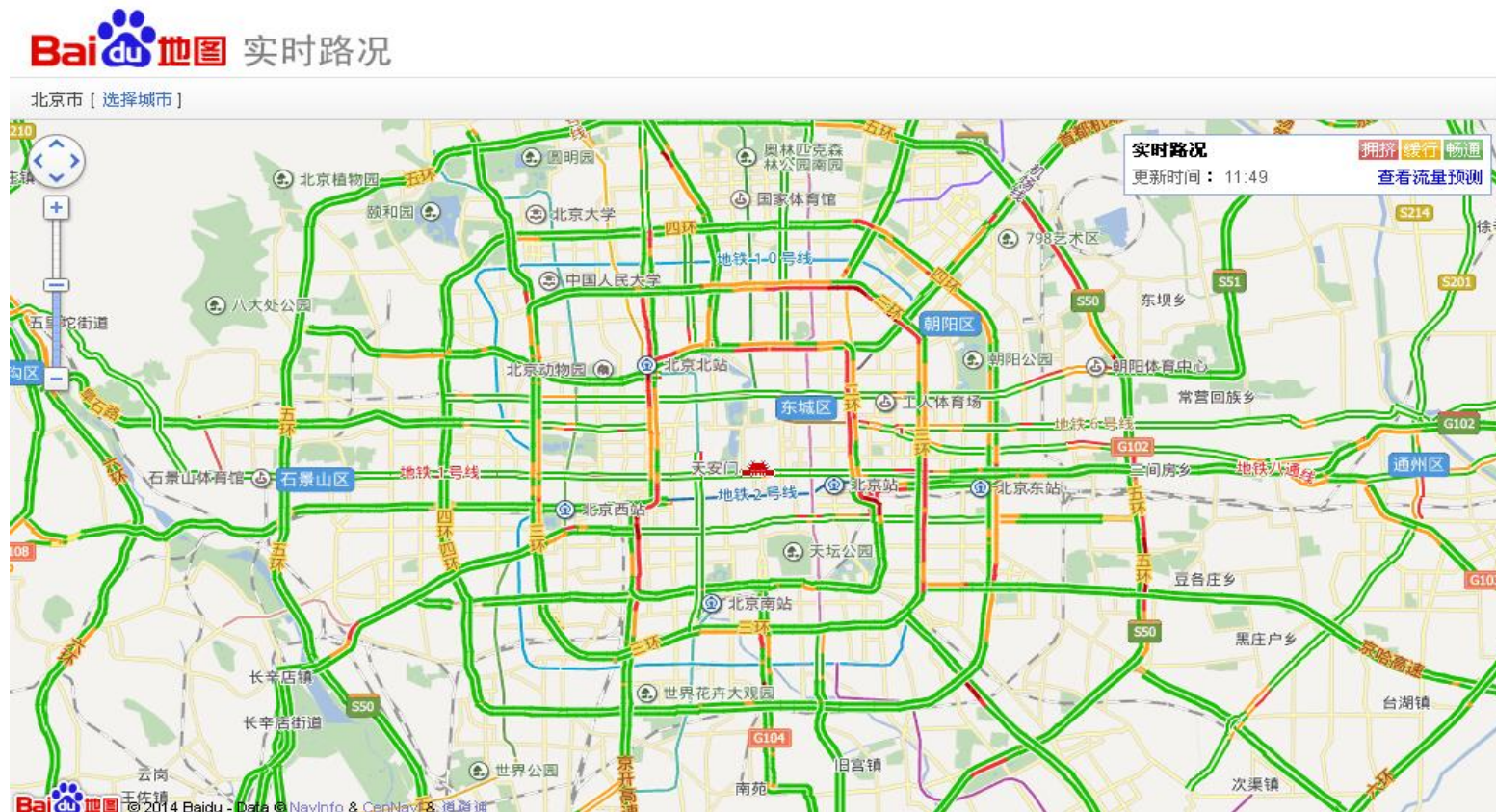


图3 百度地图显示的北京市实时交通路况信息



# 1.3 可视化的重要作用

## (2) 分析数据

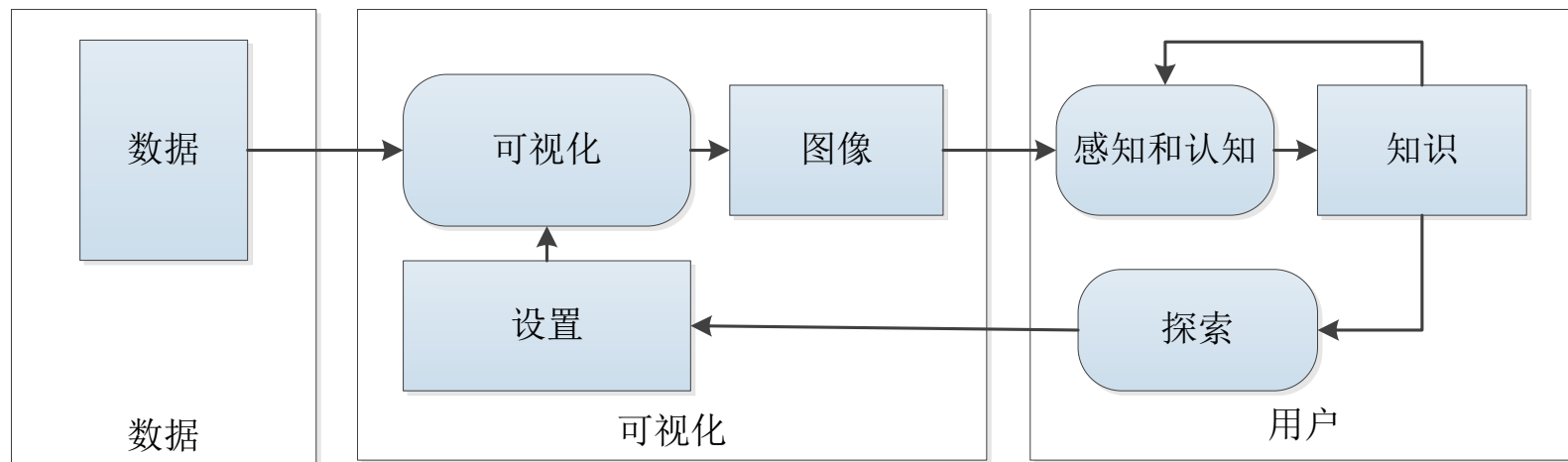


图4 用户参与的可视化分析过程

# 1.3 可视化的重要作用

## (3) 辅助理解数据



图5 微软“人立方”展示的人物关系图

# 1.3 可视化的重要作用

## (4) 增强数据吸引力



图6 一个可视化的图表新闻实例

## 2 可视化工具

- 2.1 入门级工具
- 2.2 信息图表工具
- 2.3 地图工具
- 2.4 时间线工具
- 2.5 高级分析工具

## 2.1 入门级工具

- **Excel**是微软公司的办公软件**Office**家族的系列软件之一，可以进行各种数据的处理、统计分析和辅助决策操作，已经广泛地应用于管理、统计、金融等领域

## 2.2 信息图表工具

信息图表是信息、数据、知识等的视觉化表达，它利用人脑对于图形信息相对于文字信息更容易理解的特点，更高效、直观、清晰地传递信息，在计算机科学、数学以及统计学领域有着广泛的应用。

### 1. Google Chart API

谷歌公司的制图服务接口Google Chart API，可以用来为统计数据并自动生成图片，该工具使用非常简单，不需要安装任何软件，可以通过浏览器在线查看统计图表。

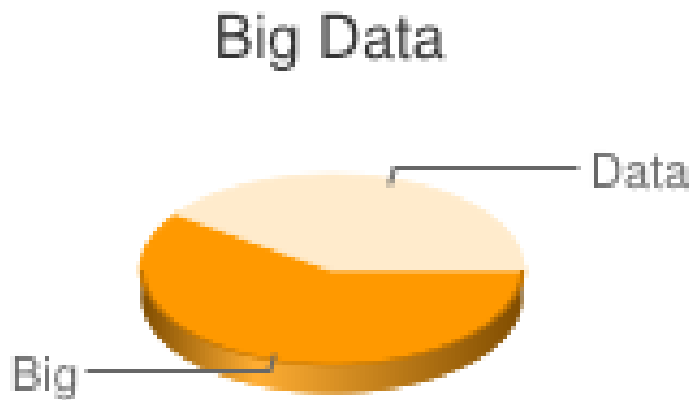


图7 通过浏览器在线查看Google Chart统计图表

## 2.2 信息图表工具

### 2. D3

D3是最流行的可视化库之一，是一个用于网页作图、生成互动图形的JavaScript函数库，提供了一个D3对象，所有方法都通过这个对象调用。D3能够提供大量线性图和条形图之外的复杂图表样式，例如Voronoi图、树形图、圆形集群和单词云等（如图8所示）。

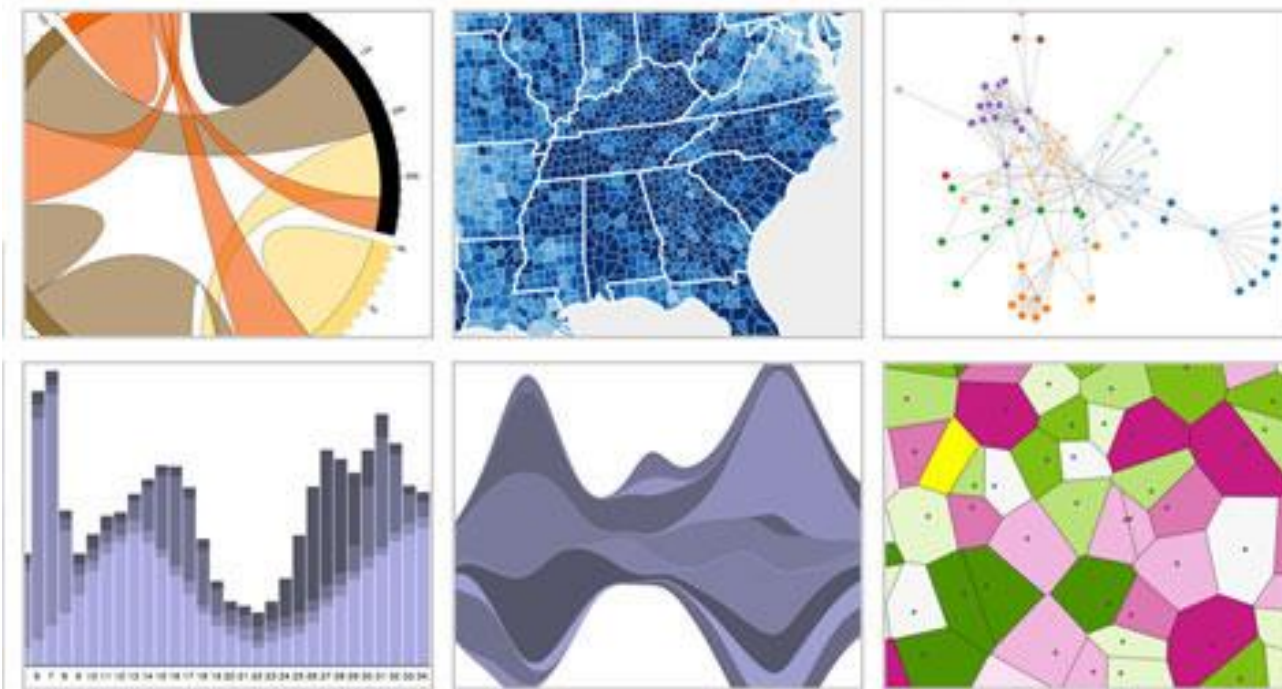


图7 通过浏览器在线查看Google Chart统计图表

## 2.2 信息图表工具

### 3. Visual.ly

Visual.ly是一款非常流行的信息图制作工具，非常好用，不需要任何设计相关的知识，就可以用它来快速创建自定义的、样式美观且具有强烈视觉冲击力的信息图表。

### 4. Tableau

Tableau是桌面系统中最简单的商业智能工具软件，更适合企业和部门进行日常数据报表和数据可视化分析工作。Tableau实现了数据运算与美观的图表的完美结合，用户只要将大量数据拖放到数字“画布”上，转眼间就能创建好各种图表。

### 5. 大数据魔镜

大数据魔镜是一款优秀的国产数据分析软件，它丰富的数据公式和算法可以让用户真正理解探索分析数据，用户只要通过一个直观的拖放界面就可创造交互式的图表和数据挖掘模型。



## 2.3 地图工具

•地图工具在数据可视化中较为常见，它在展现数据基于空间或地理分布上有很强的表现力，可以直观地展现各分析指标的分布、区域等特征。当指标数据要表达的主题跟地域有关联时，就可以选择以地图作为大背景，从而帮助用户更加直观地了解整体的数据情况，同时也可以根据地理位置快速地定位到某一地区来查看详细数据。

•图9就是以数据地图形式呈现的世界各国GDP数据，图中，颜色越深的国家，其GDP越高。

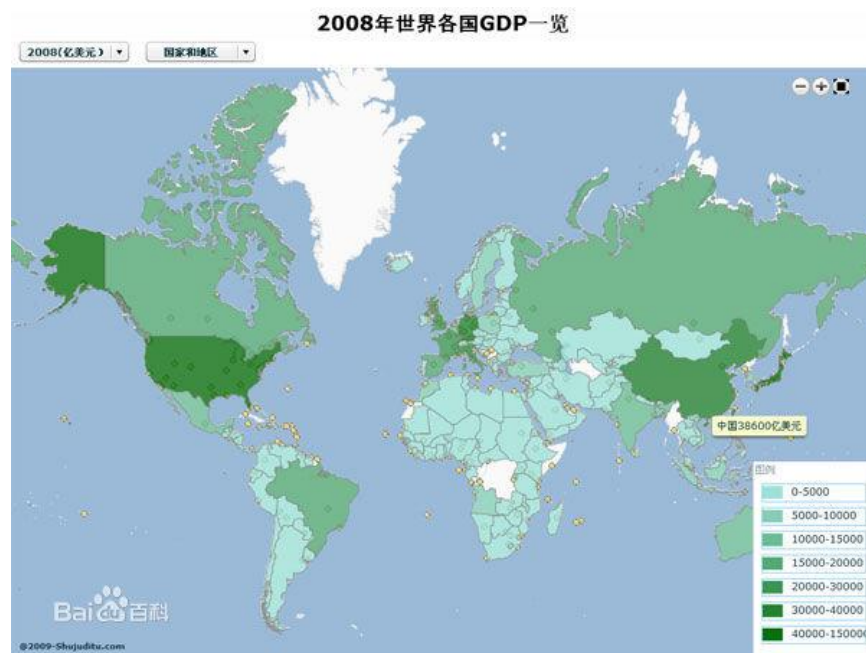


图9 世界各国GDP数据地图

## 2.3 地图工具

- **1. Google Fusion Tables**

Google Fusion Tables让一般使用者也可以轻松制作出专业的统计地图。该工具可以让数据表呈现为图表、图形和地图，从而帮助发现一些隐藏在数据背后的模式和趋势。

- **2. Modest Maps**

Modest Maps是一个小型、可扩展、交互式的免费库，提供了一套查看卫星地图的API，只有10KB大小，是目前最小的可用地图库，它也是一个开源项目，有强大的社区支持，是在网站中整合地图应用的理想选择。

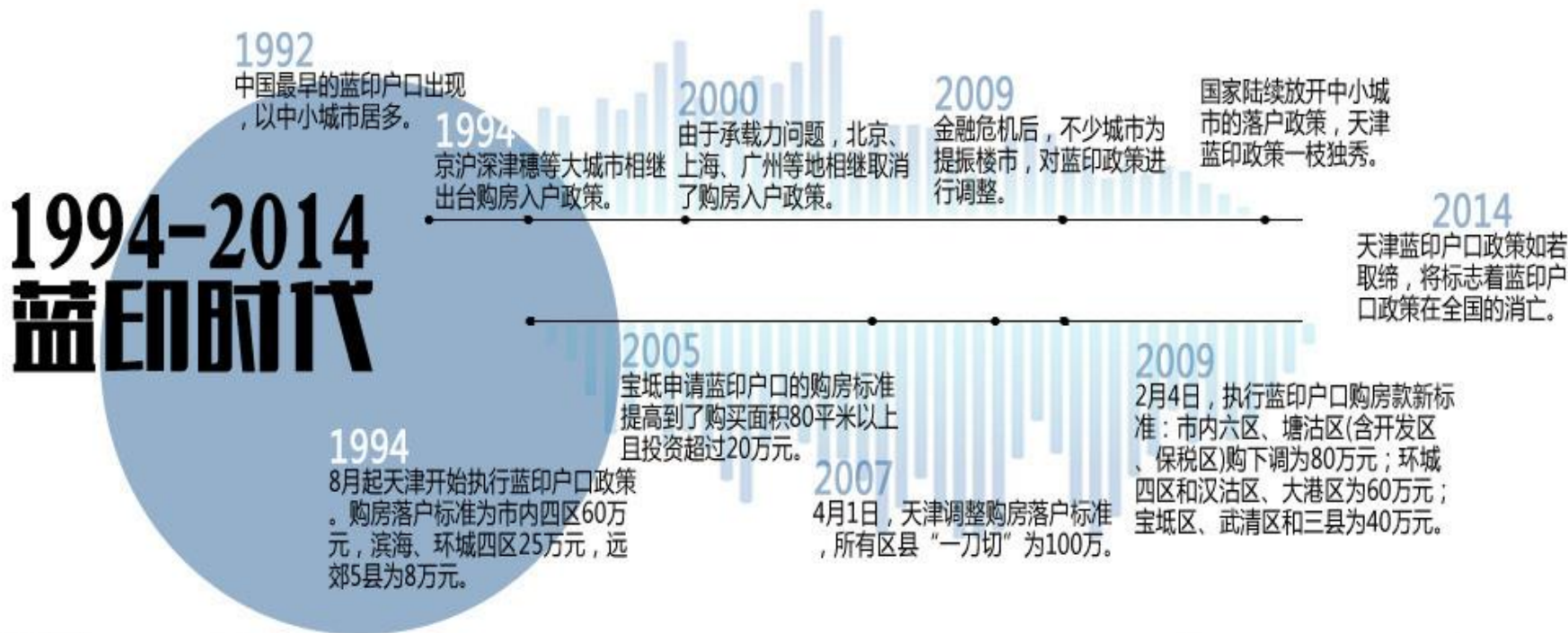
- **3. Leaflet**

Leaflet是一个小型化的地图框架，通过小型化和轻量化来满足移动网页的需要。

## 2.4 时间线工具

时间线是表现数据在时间维度的演变的有效方式，它通过互联网技术，依据时间顺序，把一方面或多方面的事件串联起来，形成相对完整的记录体系，再运用图文的形式呈现给用户。时间线可以运用于不同领域，最大的作用就是把过去的事物系统化、完整化、精确化。自2012年Facebook在F8大会上发布了以时间线格式组织内容的功能后，时间线工具在国内外社交网站中开始大面积流行。

图10显示了我国户籍制度在1994年到2014年间随时间的演变情况，



## 2.4 时间线工具

- **1. Timetoast**

Timetoast是在线创作基于时间轴事件记载服务的网站，提供个性化的时间线服务，可以用不同的时间线来记录你某个方面的发展历程、心理路程、进度过程等等。Timetoast基于 flash 平台，可以在类似 flash 时间轴上任意加入事件，定义每个事件的时间、名称、图像、描述，最终在时间轴上显示事件在时间序列上的发展，事件显示和切换十分流畅，随着鼠标点击可显示相关事件，操作简单。

- **2. Xtimeline**

Xtimeline 是一个免费的绘制时间线的在线工具网站，操作简便，用户通过添加事件日志的形式构建时间表，同时也可给日志配上相应的图表。不同于Timetoast的是，Xtimeline是一个社区类型的时间轴网站，其中加入了组群功能和更多的社会化因素，除了可以分享和评论时间轴外，还可以建立组群讨论所制作的时间轴。

## 2.5 高级分析工具

- **1. R**

R是属于GNU系统的一个自由、免费、源代码开放的软件，它是一个用于统计计算和统计制图的优秀工具，使用难度较高。R的功能包括数据存储和处理系统、数组运算工具（具有强大的向量、矩阵运算功能）、完整连贯的统计分析工具、优秀的统计制图功能、简便而强大的编程语言，可操纵数据的输入和输出，实现分支、循环以及用户可自定义功能等，通常用于大数据集的统计与分析。

- **2. Weka**

Weka是一款免费的、基于Java环境的、开源的机器学习以及数据挖掘软件，不但可以进行数据分析，还可以生成一些简单图表。

- **3. Gephi**

Gephi是一款比较特殊也很复杂的软件，主要用于社交图谱数据可视化分析，可以生成非常酷炫的可视化图形。

# 3 可视化典型案例

- 3.1 全球黑客活动
- 3.2 互联网地图
- 3.3 编程语言之间的影响力关系图
- 3.4 百度迁徙
- 3.5 世界国家健康与财富之间的关系
- 3.6 3D可视化互联网地图APP

# 3.1 全球黑客活动

安全供应商Norse打造了一张能够反映全球范围内黑客攻击频率的地图 (<http://map.ipviking.com>)，它利用Norse的“蜜罐”攻击陷阱显示出所有实时渗透攻击活动。如图11所示，地图中的每一条线代表的都是一次攻击活动，借此可以了解每一天、每一分钟甚至每一秒世界上发生了多少次恶意渗透。



图11 一张能够反映全球范围内黑客攻击频率的地图

## 3.2 互联网地图

为了探究互联网这个庞大的宇宙，俄罗斯工程师 Ruslan Enikeev 根据 2011 年底的数据，将全球 196 个国家的 35 万个网站数据整合起来，并根据 200 多万个网站链接将这些“星球”通过关系链联系起来，每一个“星球”的大小根据其网站流量来决定，而“星球”之间的距离远近则根据链接出现的频率、强度和用户跳转时创建的链接来确定，由此绘制得到了“互联网地图”（<http://internet-map.net>），如图12所示。

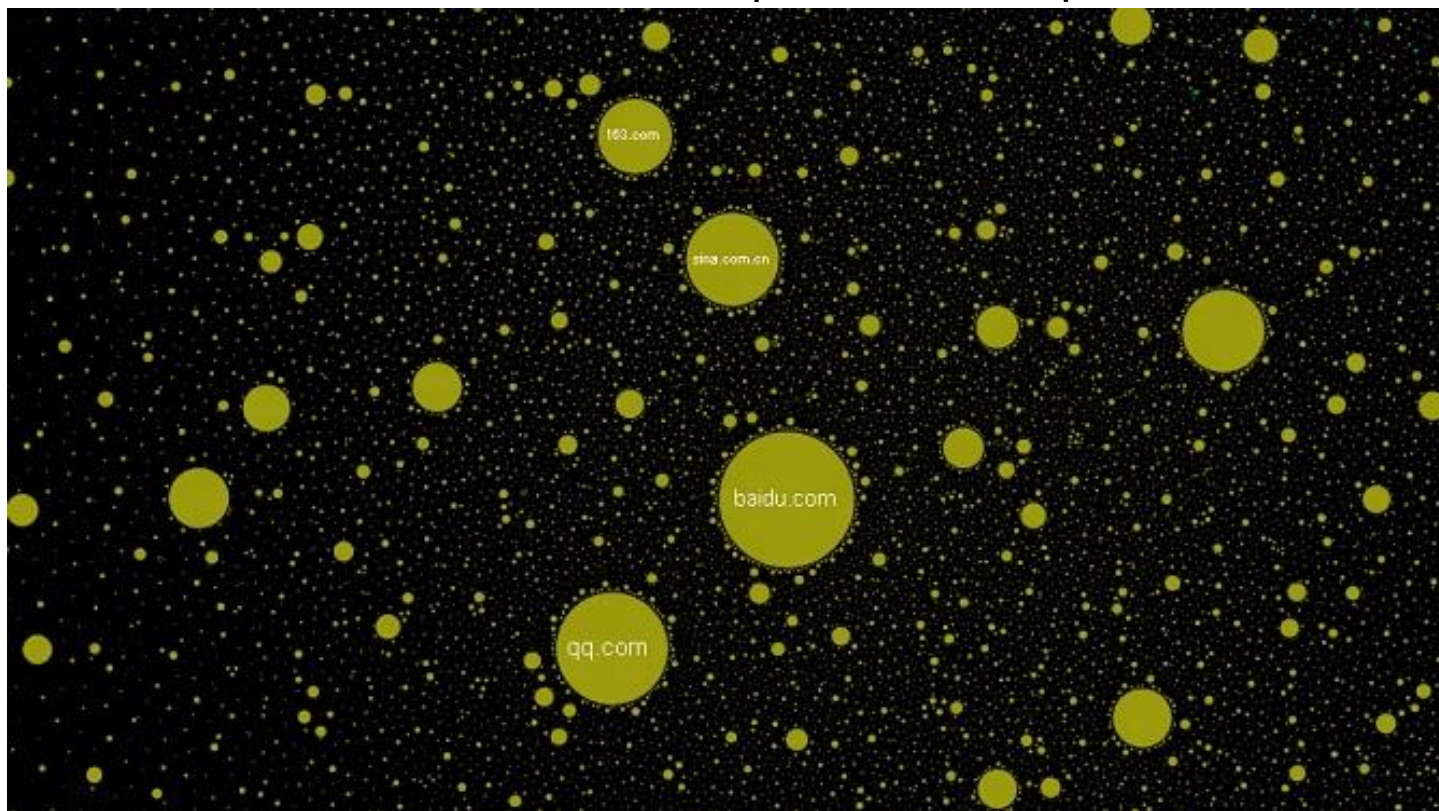


图12俄罗斯工程师绘制的“互联网地图”



## 3.3 编程语言之间的影响力关系图

Ramio Gómez利用来自Freebase上的编程语言维护表里的数据，绘制了编程语言之间的影响力关系图，如图13所示，图中的每个节点代表一种编程语言，之间的连线代表该编程语言对其他语言有影响，有影响力的语言会连线多个语言，相应的节点也会越大。

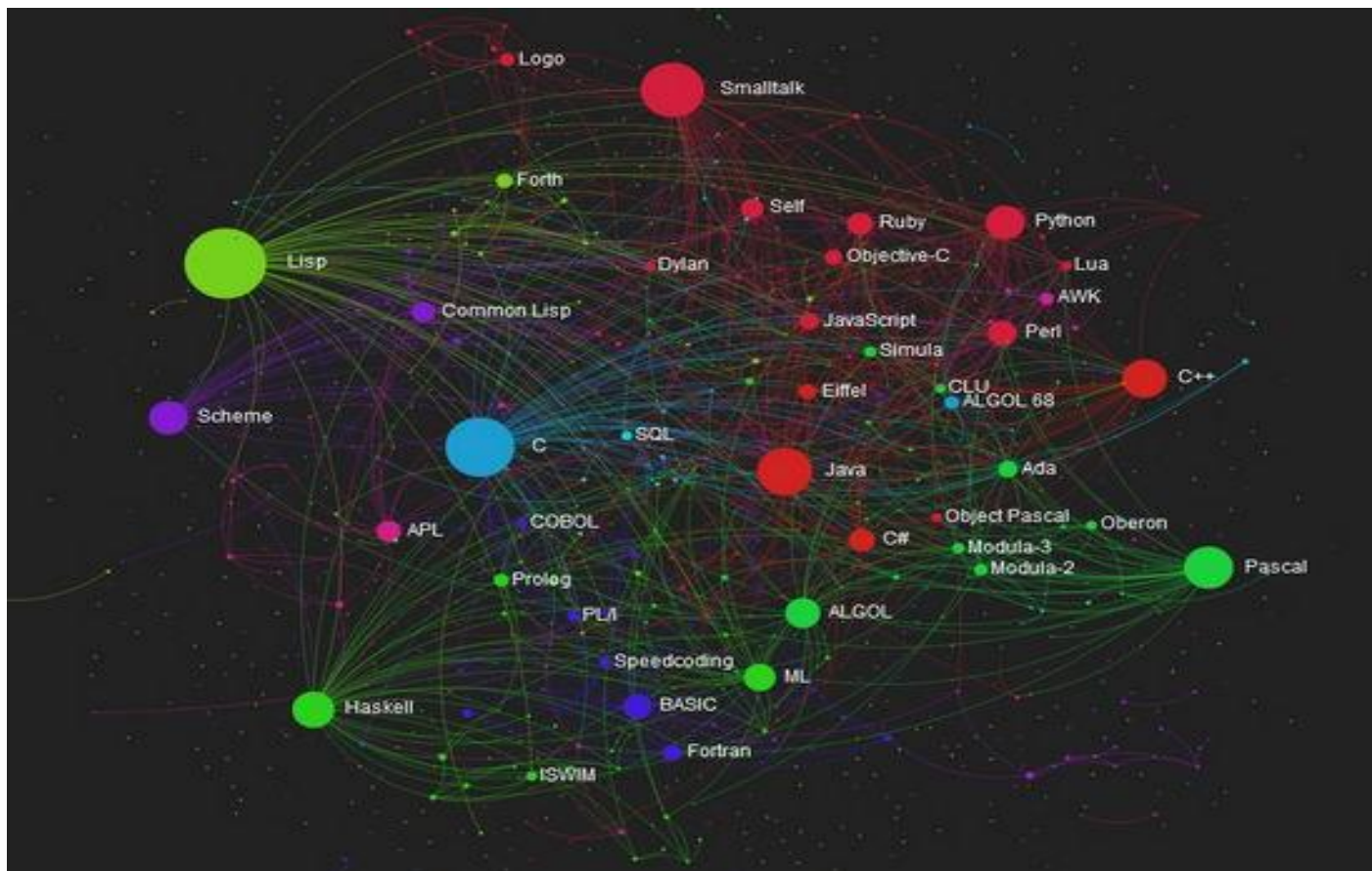


图13 编程语言之间的影响力关系图

## 3.4 百度迁徙

央视与百度合作，启用百度地图定位可视化大数据播报春节期间全国人口迁徙情况（如图14所示），引起广泛关注。



图14 百度迁徙展示的全中国8小时迁徙图

## 3.5 世界国家健康与财富之间的关系

如图15所示，“世界国家健康与财富之间的关系”利用可视化技术，把世界上200个国家，从1810年到2010年历时200年其各国国民的健康、财富变化数据（收集了1千多万数据）制作成三维动画进行了直观展示（<http://www.moojnn.com/Index/whn>）。

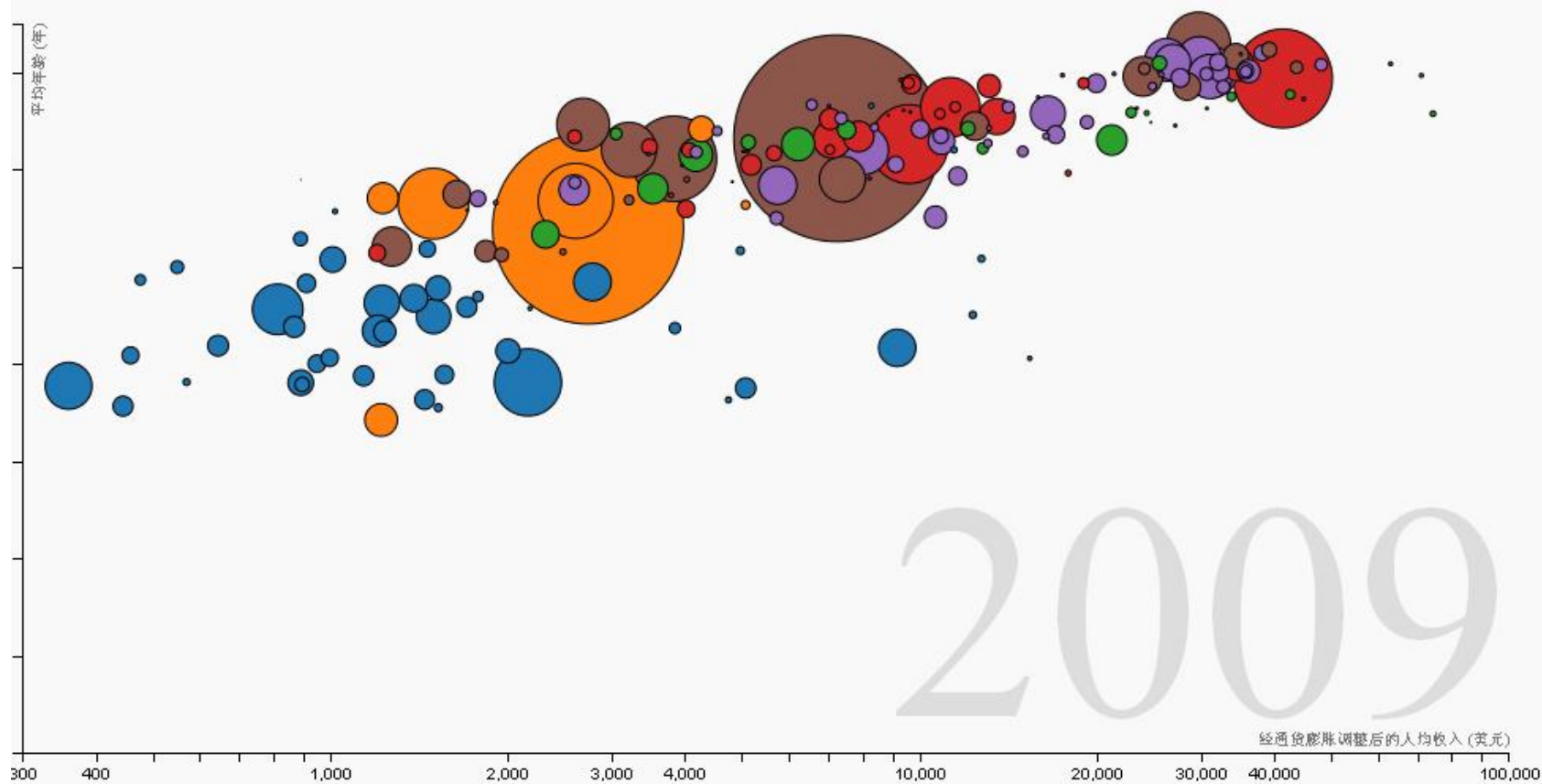


图15 世界国家健康与财富之间的关系图

## 3.6 3D可视化互联网地图APP

3D可视化是描绘和理解数据的一种手段，是数据的一种表征形式，并非模拟技术。3D可视化以一种独特的立体视角为用户呈现数据，可以帮助用户发现一些在2D模式下无法察觉的内容。Peer 1开发了一个称为“互联网地图”的APP（如图16所示），这是一个建立在小盒子形式上的3D地图。

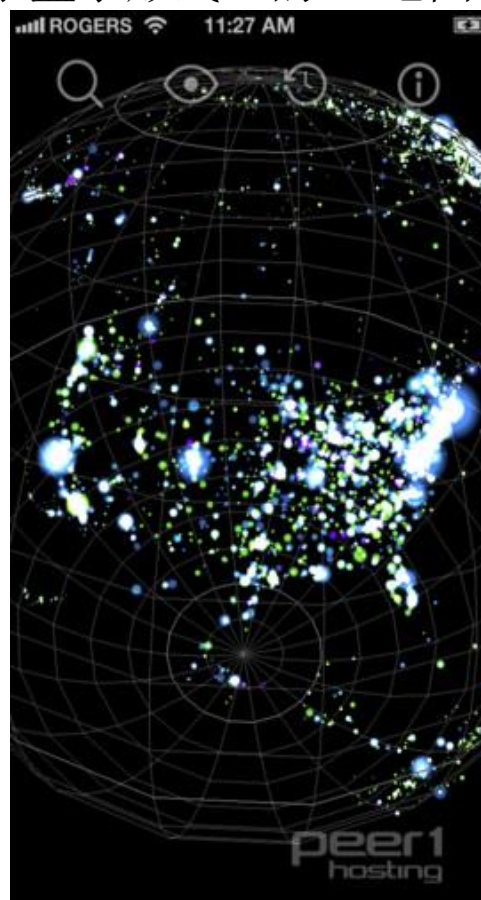


图16 Peer 1开发的“互联网地图”

## 4 可视化工具实践

- 4.1 D3可视化库使用指南
- 4.2 Easel.ly信息图制作实践
- 4.3 数据可视化之魔镜
- 4.4 数据可视化之ECharts

# 4.1 D3可视化库使用指南

- 一、D3可视化库介绍
- 二、D3可视化库的安装
- 三、预备知识
- 四、基本操作

# 4.1 D3可视化库使用指南

## 一、D3可视化库介绍

- D3 的全称是（Data-Driven Documents），顾名思义，它是一个被数据驱动的文档。听名字有点抽象，说简单一点，其实就是一个 JavaScript 的函数库，使用它主要是用来做数据可视化的
- 学习 D3 最好的地方是：<http://d3js.org/>

# 4.1 D3可视化库使用指南

## 二、D3可视化库的安装

D3 是一个 JavaScript 函数库，并不需要通常所说的“安装”。它只有一个文件，在 HTML 中引用即可。有两种方法：

- 方法一：下载 D3.js 的文件，解压后，在 HTML 文件中包含相关的 js 文件即可。

方法二：可以直接包含网络的链接，这种方法较简单：

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

但使用的时候要保持网络连接有效，不能在断网的情况下使用。



# 4.1 D3可视化库使用指南

## 三、预备知识

学习D3可视化库需要的一些基础知识如下：

- HTML：超文本标记语言，用于设定网页的内容，比如<body>和<p>标签
- CSS：层叠样式表，用于设定网页的样式
- JavaScript：一种直译式脚本语言，用于设定网页的行为
- DOM：文档对象模型，用于修改文档的内容和结构
- SVG：可缩放矢量图形，用于绘制可视化的图形

# 4.1 D3可视化库使用指南

## 四、基本操作 1、添加元素

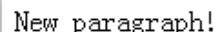
```
1. d3.select("body").append("p").text("New paragraph!");
```

为选择body标签，为之添加一个p标签，并设置它的内容为New paragraph!

源代码：

```
1. <html>
2.   <head>
3.     <meta charset="utf-8">
4.     <title>D3测试</title>
5.     <script type="text/javascript" src="http://d3js.org/d3.v3.min.js"> </script>
6.   </head>
7.   <body>
8.     <script type="text/javascript">
9.       d3.select("body").append("p").text("New paragraph!");
10.    </script>
11.  </body>
12. </html>
```

效果：



效果图

# 4.1 D3可视化库使用指南

## 2、数据绑定

D3可以处理哪些类型的数据？

它会接受几乎任何数值数组，字符串，或对象（本身包含其他数组或键/值对）。它可以处理JSON和GeoJSON

源代码：

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>testD3-1.html</title>
5.     <script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>
6.   </head>
7.
8.   <body>
9.     This is my HTML page. <br>
10.  </body>
11.  <script type="text/javascript">
12.    var dataset = [ 5, 10, 15, 20, 25 ];
13.
14.    d3.select("body").selectAll("p")
15.      .data(dataset)
16.      .enter()
17.      .append("p")
18.      .text("New paragraph!");
19.  </script>
20. </html>
```

# 4.1 D3可视化库使用指南

## 语法说明

- `d3.select("body")`

查找DOM中的body。

- `selectAll("p")`

选择DOM中的所有段落。由于没有存在，这将返回一个空的选择。这个选择为空，代表段落很快就会存在。

- `data(dataset)`

计数和分析我们的数据值。有五个值，之后我们的数据集执行了5次，每个值一次。

- `enter()`

绑定数据和DOM元素。这个方法将数据传递到DOM中。如果数据值比相应的DOM元素多，就用`enter()`创建一个新元素的占位符。

- `append("p")`

通过`enter()`创建的占位符 在DOM中插入一个p元素。

- `text("New paragraph!")`

为新创建的p标签插入一个文本值。

运行效果如下：

```
This is my HTML page.
```

```
New paragraph!
```

```
New paragraph!
```

```
New paragraph!
```

```
New paragraph!
```

```
New paragraph!
```

# 4.1 D3可视化库使用指南

## 3、用层画条形图

(1)、为同类层添加样式

```
1. div.bar {  
2.     display: inline-block;  
3.     width: 20px;  
4.     height: 75px; /* We'll override this later */  
5.     background-color: teal;  
6. }
```

(2)、声明要为某类层设置属性

```
1. .attr("class", "bar")
```

(3)、为每个特定的层设置属性

```
1. .style("height", function(d) {  
2.     var barHeight = d * 5; //Scale up by factor of 5  
3.     return barHeight + "px";  
4. });
```

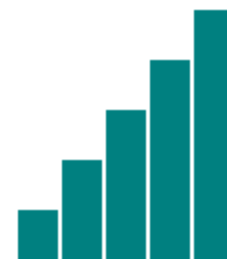
(4)、设置层之间的间隔

```
1. margin-right: 2px;
```

# 4.1 D3可视化库使用指南

## 源代码

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>testD3-3-drawingDivBar</title>
6.     <script type="text/javascript" src="http://d3js.org/d3.v3.min.js"> </script>
7.     <style type="text/css">
8.
9.       div.bar {
10.        display: inline-block;
11.        width: 20px;
12.        height: 75px; /* Gets overridden by D3-assigned height below */
13.        margin-right: 2px;
14.        background-color: teal;
15.      }
16.
17.    </style>
18.  </head>
19.  <body>
20.    <script type="text/javascript">
21.
22.      var dataset = [ 5, 10, 15, 20, 25 ];
23.
24.      d3.select("body").selectAll("div")
25.        .data(dataset)
26.        .enter()
27.        .append("div")
28.        .attr("class", "bar")
29.        .style("height", function(d) {
30.          var barHeight = d * 5;
31.          return barHeight + "px";
32.        });
33.
34.    </script>
35.  </body>
36. </html>
```



运行效果图

# 4.1 D3可视化库使用指南

## 4、SVG概要

### (1) 简单形状

SVG标签包含一些视觉元素，包括矩形，圆形，椭圆形，线条，文字和路径等。基于像素的坐标系统，其中浏览器的左上角是原点(0,0)。x,y的正方向分别是右和下。

- 矩形。使用x和y的指定左上角的坐标，width和height指定的尺寸。绘制SVG的矩形可以这样写：

```
<rect x="0" y="0" width="500" height="50"/>
```

- 圆。使用cx和cy，指定指定半径的中心的坐标，和r表示半径。例如：

```
<circle cx="250" cy="25" r="25"/>
```

- 椭圆。使用cx和cy，指定指定半径的中心的坐标，rx和ry分别指定x方向和y方向上圆的半径，例如：

```
<ellipse cx="250" cy="25" rx="100" ry="25"/>
```

# 4.1 D3可视化库使用指南

- 线。使用x1和Y1到指定线的一端的坐标，x2和y2指定的另一端的坐标。stroke指定描边使得线是可见的。例如：

```
<line x1="0" y1="0" x2="500" y2="50" stroke="black"/>
```

- 文本。使用 x和y指定文本的位置。例如：

```
<text x="250" y="25">Easy-peasy</text>
```

可以给文本设置样式。例如：

```
<text x="250" y="155" font-family="sans-serif" font-size="25"  
fill="gray">Easy-peasy</text>
```



# 4.1 D3可视化库使用指南

## (2) SVG的默认样式

SVG的默认样式是黑色填充。如果想换颜色，就必须将样式应用到相应的元素。常见的SVG性质：

1. 填充（fill） -颜色值。正如用CSS，颜色可以被指定为

- \* 命名的颜色- green

- \* 十六进制值- #3388aa或38A

- \* RGB值- RGB(10,150,20)

- \* RGB与Alpha透明度- RGBA(10,150,20,0.5)

2. 描边（stroke） -颜色值。

3. 描边宽度（stroke-width） -数字（通常以像素为单位）。

4. 不透明度（opacity） – 0.0（完全透明）和1.0（完全不透明）之间的数值。

5. 有了文字，也可以使用CSS样式，如：

- \* 字体类型（font-family）

- \* 字体大小（font-size）

# 4.1 D3可视化库使用指南

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>testD3-6-SVG.html</title>
    <script type="text/javascript"
src="http://d3js.org/d3.v3.min.js"></script>
    <style type="text/css">
      .pumpkin {
        fill: yellow;
        stroke: orange;
        stroke-width: 5;
      }
    </style>
  </head>
```

备注：左边和右边代码是一段完整的代码，垂直放不下，所以分开左右两段写

```
<body>
  <script type="text/javascript"></script>
  <svg width=500 height=960>
    <rect x="0" y="0" width="500" height="50"/>
    <ellipse cx="250" cy="225" rx="100" ry="25"/>
    <line x1="0" y1="120" x2="500" y2="50" stroke="black"/>
    <text x="250" y="155" font-family="sans-serif"
      font-size="25" fill="gray">Easy-peasy</text>
    <circle cx="25" cy="80" r="20"
      fill="rgba(128, 0, 128, 0.75)"
      stroke="rgba(0, 255, 0, 0.25)"
      stroke-width="100"/>
    <circle cx="75" cy="80" r="20"
      fill="rgba(0, 255, 0, 0.75)"
      stroke="rgba(0, 0, 255, 0.25)" stroke-width="10"/>
    <circle cx="125" cy="80" r="20"
      fill="rgba(255, 255, 0, 0.75)"
      stroke="rgba(255, 0, 0, 0.25)" stroke-width="10"/>
    <rect x="0" y="300" width="30" height="30" fill="purple"/>
    <rect x="20" y="305" width="30" height="30" fill="blue"/>
    <rect x="40" y="310" width="30" height="30" fill="green"/>
    <rect x="60" y="315" width="30" height="30" fill="yellow"/>
    <rect x="80" y="320" width="30" height="30" fill="red"/>
    <circle cx="25" cy="425" r="22" class="pumpkin"/>
    <circle cx="25" cy="525" r="20" fill="rgba(128, 0, 128, 1.0)"/>
    <circle cx="50" cy="525" r="20" fill="rgba(0, 0, 255, 0.75)"/>
    <circle cx="75" cy="525" r="20" fill="rgba(0, 255, 0, 0.5)"/>
    <circle cx="100" cy="525" r="20" fill="rgba(255, 255, 0, 0.25)"/>
    <circle cx="125" cy="525" r="20" fill="rgba(255, 0, 0, 0.1)"/>
    <circle cx="25" cy="625" r="20" fill="purple"
      stroke="green" stroke-width="10"
      opacity="0.9"/>
    <circle cx="65" cy="625" r="20" fill="green"
      stroke="blue" stroke-width="10"
      opacity="0.5"/>
    <circle cx="105" cy="625" r="20" fill="yellow"
      stroke="red" stroke-width="10"
      opacity="0.1"/>
  </svg>
</body>
</html>
```

# 4.1 D3可视化库使用指南

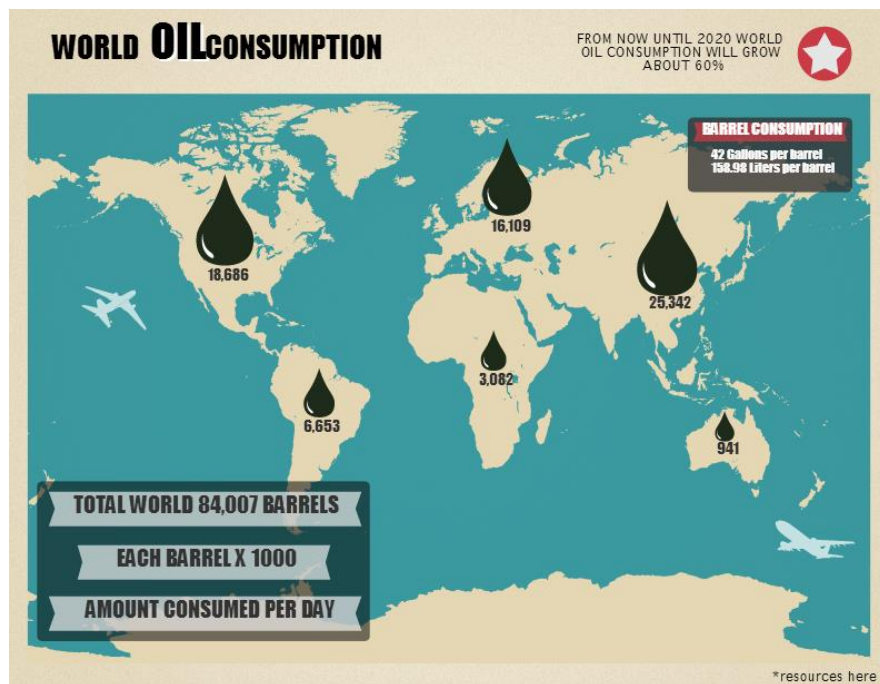


运行效果图

## 4.2 Easel.ly信息图制作实践

Easel.ly 是当下非常流行的信息图制作软件之一，其具有界面简洁、操作简便、图片精美等特点。用户只需要登录[Easel.ly官方网站](http://Easel.ly)即可开始进行信息图制作。

信息图形是指数据、信息或知识的可视化表现形式。信息图形主要应用于必须有一个清楚准确的解释或表达甚为复杂且大量的信息，例如在各式各样的文件档案上、各个地图及标志、新闻或教程文件，表现出的设计是化繁为简。下图是一个典型的信息图，显示的是全球各大洲的石油消费情况。



## 4.2 Easel.ly信息图制作实践

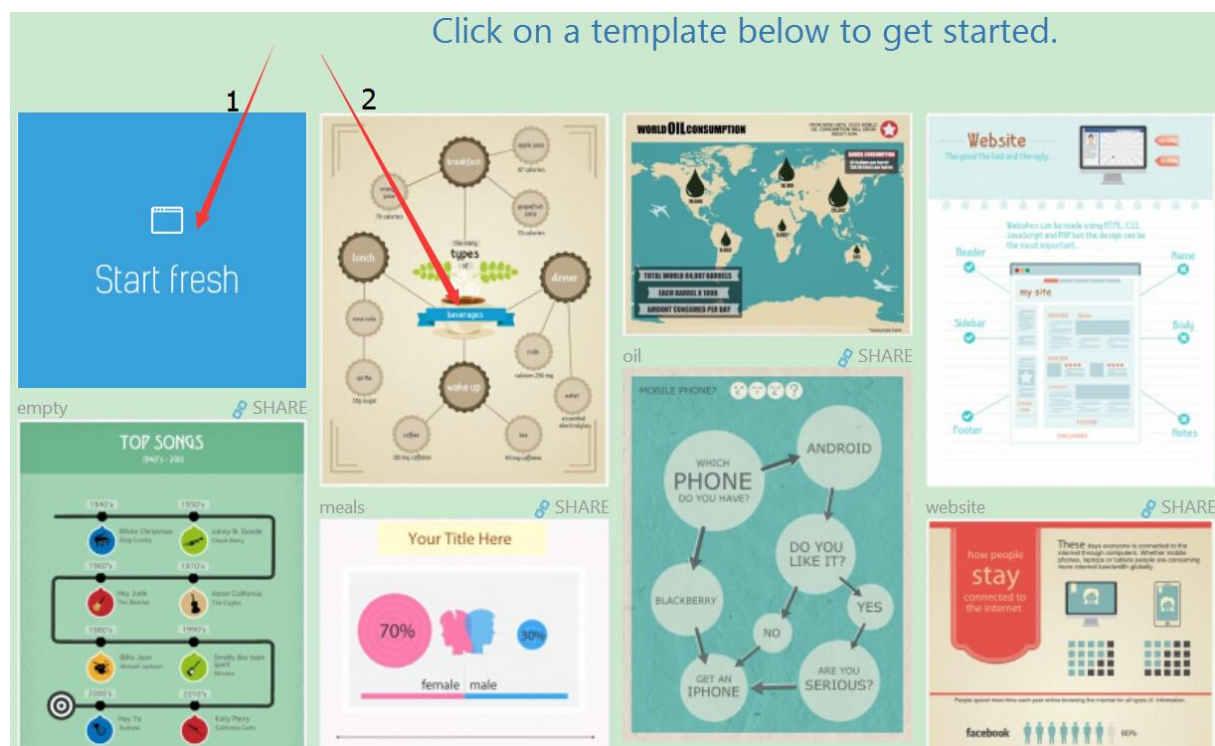
Easel.ly信息图制作步骤:

- 第一步: 选择模板
- 第二步: 编辑内容
- 第三步: 导出图片

# 4.2 Easel.ly信息图制作实践

## •第一步：选择模板

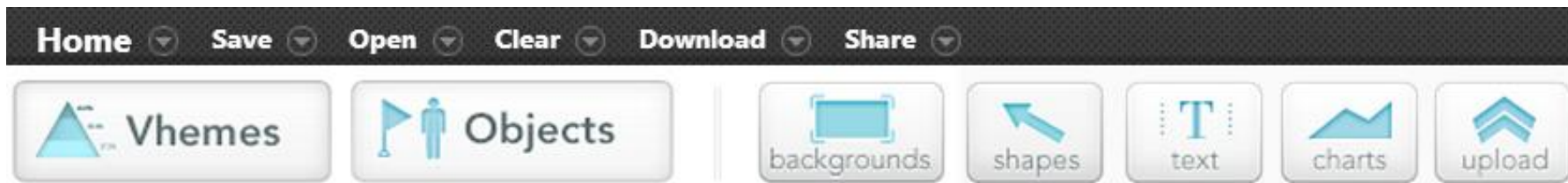
进入Easel.ly页面后，可以开始创建信息图，如下图所示，可以选择两种方法创建信息图：方法1：创建空白模版；2、选择一个已有模版，在其基础上进行制作，模版内的内容可以修改。目前Easel.ly官网上已经有数量超过2百万的模版，且可以通过选项进行筛选，推荐使用模版进行制作。



## 4.2 Easel.ly信息图制作实践

### •第二步：编辑内容

选好模版后便可以开始进行信息图的制作。下图展示的是Easel.ly的操作台，其展示了Easel.ly的所有功能，从左到右依次是：1、“Vhemes”模版选择；2、“Objects”添加元素；3、“backgrounds”选择背景；4、“shaps”添加线条；5、“text”添加文本；6、“charts”添加数据图表，可选直方图、曲线图、卫星图、雷达图、饼状图；7、“upload”上传图片。只要将元素拖到需要添加的地方便能完成元素的添加，过程十分简便。



## 4.2 Easel.ly信息图制作实践

### •第三步：导出图片

在运用Easel.ly制作出信息图后，便可以进行图片的导出：1、先点击操作台最右侧的“Present”；2、用鼠标右键点击完成的图片选择“图片另存为”，便可将制作的图片保存到本机。



## 4.2 Easel.ly信息图制作实践

这是中文你好, CA BND

这是中文你好, Comic  
Sans

这是中文你好, courier

这是中文你好, DejaVu

这是中文你好, Delicious

这是中文你好, Encient

### 中文不兼容问题

【问题】Easel.ly确实存在导出图片时中文字体显示错乱问题。这是由于其网站内部的问题，目前暂无完美解决方案，这里给出一个不完全的解决方案。

问题：图片中的中文有粗有细，不美观

## 4.2 Easel.ly信息图制作实践

### 中文不兼容问题

【方案】在信息图制作完成后，点击顶部菜单中的“Save”选项保存，再点击“Share”选项选择“View in Browser”。此时，你制作好的信息图就可以在浏览器中显示，此时中文是显示正常的。最后把浏览器中的显示效果进行截屏得到图片，或者直接把浏览器中的图片点击右键另存为，就可以保存成图片。但是这种方法存在清晰度不高的问题。



图片中的中文粗细一致

## 4.3 数据可视化之魔镜

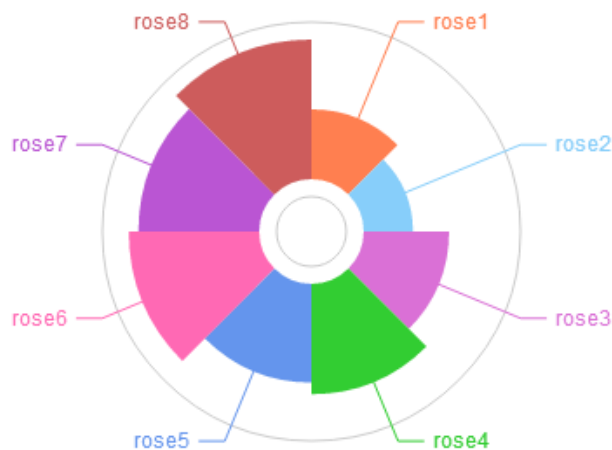
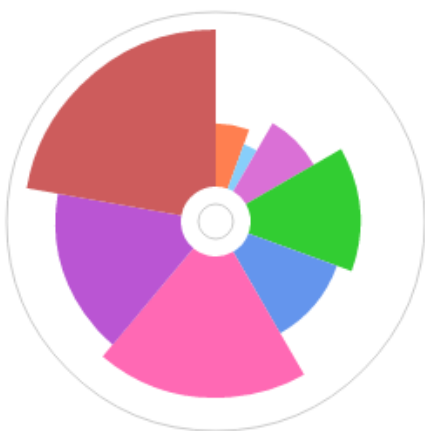
- 魔镜是国内国云数据旗下的数据可视化产品，底层封装了可视化库接口，用户通过一个直观的拖放界面就可以创建交互式的图表和完美的仪表盘。通过强大的筛选器分类整理和探索自己ed数据。简单拖拽的操作、上选下旋转的功能，更为用户实现更自由化、探索化的分析挖掘。
- 大数据魔镜同时拥有市面上最丰富的可视化效果，500多种图形库，涵盖了各种主流的可视化展现形式，绚丽而易读。探索式的分析结合丰富的可视化效果，给用户前所未有的产品体验和视觉冲击。以下四个图形效果选自于魔镜官网，展示魔镜制作图形的效果。

# 4.3 数据可视化之魔镜

来自魔镜官网的可视化效果图

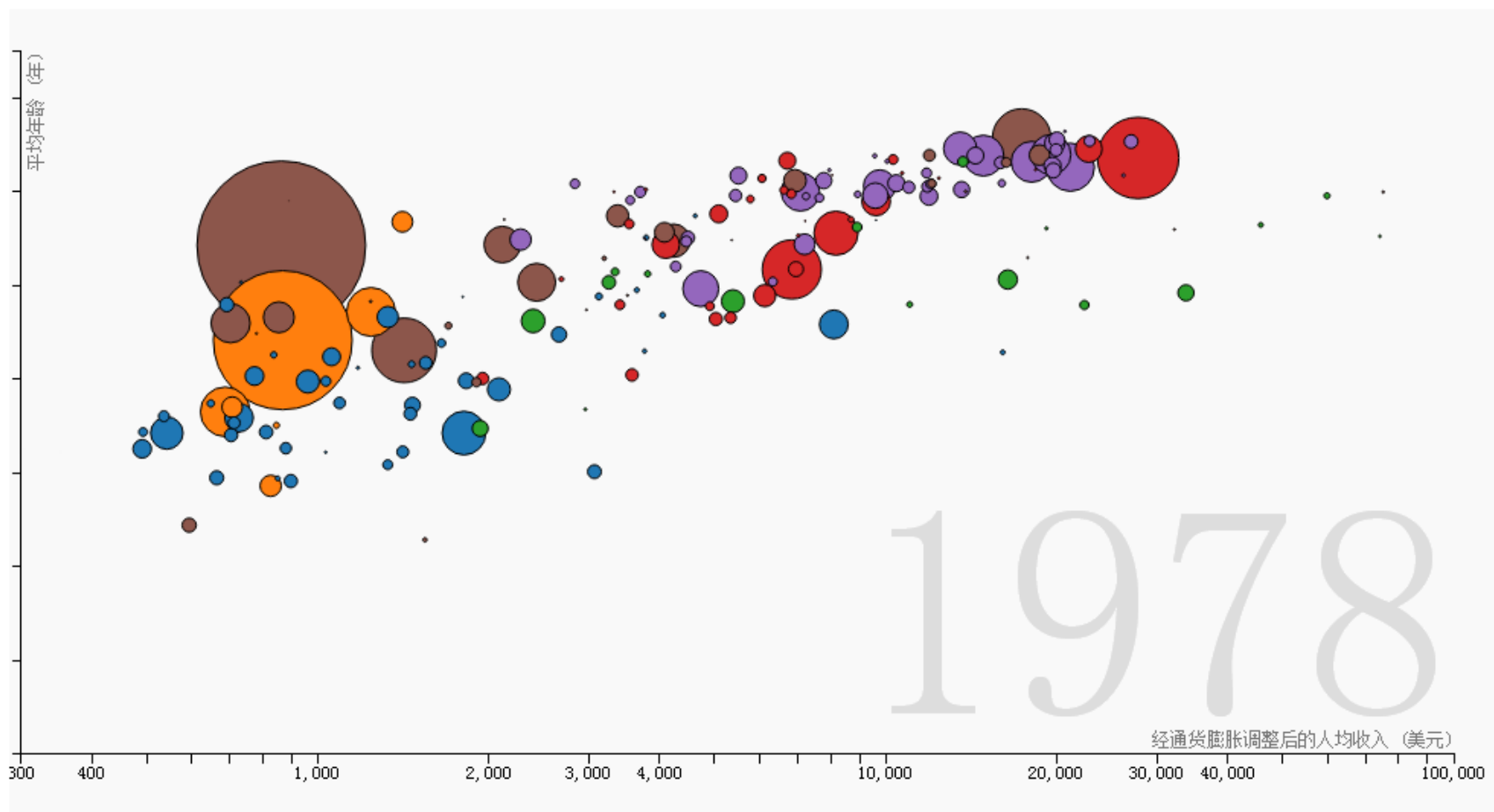
南丁格尔玫瑰图

纯属虚构



## 4.3 数据可视化之魔镜

来自魔镜官网的可视化效果图



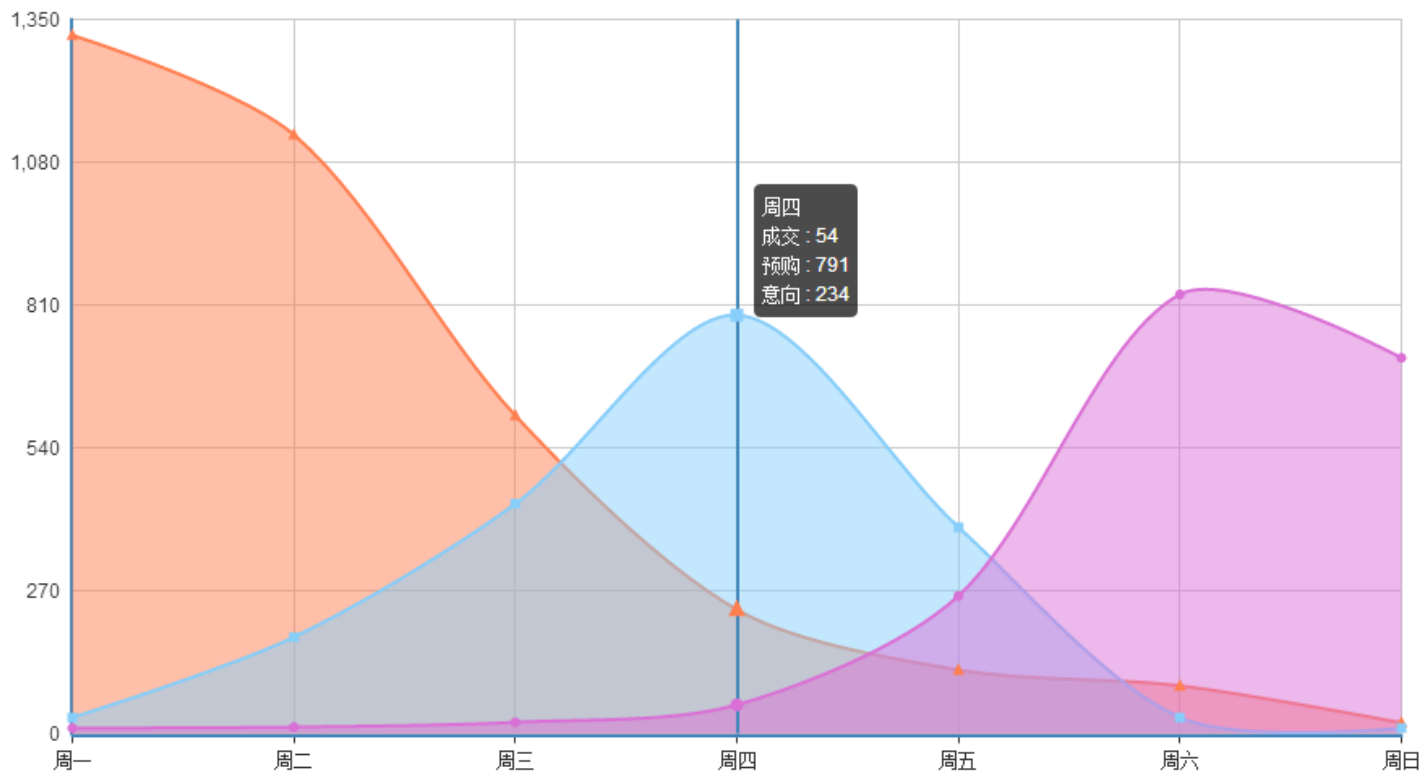
# 4.3 数据可视化之魔镜

来自魔镜官网的可视化效果图

某楼盘销售情况

纯属虚构

—▲— 意向 —■— 预售 —●— 成交



# 4.3 数据可视化之魔镜

来自魔镜官网的可视化效果图



# 4.3 数据可视化之魔镜

## 使用魔镜制作 可视化图的简单步骤

首先登陆[魔镜官方网站](#) 点击右上角“登陆/注册”，先注册一个账号，然后再用用户账号登陆。登陆之后界面如右侧所示：



数据源 · 数据处理 · 数据分析 · 数据挖掘 · 仪表盘 · 数据工厂 · 设置 · 我的账户 ·

项目列表名/数据源名/仪表盘名

我的项目  开启编辑

新建项目



示例项目



人力资源部门



销售部门



某超市销售分析



学生成绩分析



全省医疗卫生

入门视频

更多 >>



魔镜产品介绍



概念性介绍



魔镜项目使用



产品全功能使用入门



## 4.3 数据可视化之魔镜

首先点击新建项目，接下来输入项目名，数据源选择Excel表格，导入数据，点击保存即可出现个人制作管理后台：



## 4.3 数据可视化之魔镜

然后点击新建图表，即可出现制作图形的工作区：

其中左边为数据的两个维度，维度(X轴)为文本属性，度量(Y轴)为数值属性。当我们需要制作某个图表时，只需要用鼠标将维度拖至箭头指向的行，将度量拖至箭头指向的列（也可以点击维度或度量后面的箭头），然后点击右边框中的各种图表，即可生成相应的图表类型。

魔镜

数据可视化分析台

某超市销售数... 数据源 数据处理 数据分析 数据挖掘 仪表盘 数据工厂 设置 我的账户

拖拽模式 代码模式 路径模式

报表名: 图表-N 保存

行(X轴):

列(Y轴):

筛选器:

限制条数: 100

请将左侧数据源维度、度量拖入分析台行或列，开始可视化数据分析

维度(X轴)

- 订单 ID
- 订单日期
- 发货日期
- 邮寄方式
- 客户 ID
- 客户名称
- 细分
- 城市
- 省/自治区
- 国家
- 地区
- 产品 ID
- 类别
- 子类别
- 产品名称

度量(Y轴)

- 行 ID
- 销售额

列表

饼图

散点图

线图

柱状图

条形图

区域图

树状图

地图

混合图

数字

仪表盘

更多

# 4.3 数据可视化之魔镜

我们来分析下每个地区的销售额和利润，首先将地区拖至行，然后销售额和利润拖至列，然后点击某种图表即可生成，如下：



## 4.4 数据可视化之ECharts

- Echarts简介
- Echarts开发环境
- Echarts例子讲解

## 4.4 数据可视化之ECharts

### •Echarts简介

•[ECharts](#)是由百度商业前端数据可视化团队研发的图表库，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（IE8/9/10/11，Chrome，Firefox，Safari等），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 提供了非常丰富的图表类型，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，满足用户绝大部分用户分析数据时的图表制作需求

# 4.4 数据可视化之ECharts

## •Echarts开发环境

ECharts是一款可视化开发库，底层用的是javascript封装，所以可以在网页HTML中嵌入ECharts代码显示数据图表。在网页中嵌入ECharts步骤如下：

1.下载ECharts

2.HTML引入 ECharts

因为ECharts底层是javascript，所以可以像javascript一样，直接嵌入到HTML中，如下：

```
<!DOCTYPE html>
<html>
<header>
  <meta charset="utf-8">
  <!-- 引入 ECharts 文件 -->
  <script src="echarts.js"></script>
</header>
</html>
```

## 4.4 数据可视化之ECharts

### •Echarts开发环境

#### 3.绘制一个简单的图表

(1)在绘图前我们需要为 ECharts 准备一个具备高宽的 dom 容器。

```
<body>
```

```
  <!-- 为 ECharts 准备一个具备大小（宽高）的Dom -->
```

```
  <div id="main" style="width: 600px;height:400px;"></div>
```

```
</body>
```

(2)然后就可以通过 `echarts.init` 方法初始化一个 `echarts` 实例并通过 `setOption` 方法生成一个简单的柱状图，下面是完整代码

## 4.4 数据可视化之ECharts

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>ECharts</title>
  <!-- 引入 echarts.js -->
  <script src="echarts.js"></script>
</head>
```

备注：左边和右边代码是一段完整的代码，垂直放不下，所以分开左右两段写

```
<body>
  <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
  <div id="main" style="width: 600px;height:400px;"></div>
  <script type="text/javascript">
    // 基于准备好的dom，初始化echarts实例
    var myChart = echarts.init(document.getElementById('main'));

    // 指定图表的配置项和数据
    var option = {
      title: {
        text: 'ECharts 入门示例'
      },
      tooltip: {},
      legend: {
        data:['销量']
      },
      xAxis: {
        data: ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
      },
      yAxis: {},
      series: [{
        name: '销量',
        type: 'bar',
        data: [5, 20, 36, 10, 10, 20]
      }]
    };

    // 使用刚指定的配置项和数据显示图表。
    myChart.setOption(option);
  </script>
</body>
</html>
```



## 4.4 数据可视化之ECharts

上面代码是ECharts官网提供的一个简单示例，显示一个柱状图。在官网下载echarts.js文件，然后在echarts.js同一个文件夹下面新建一个html文件（注意改后缀为.html），将上述示例代码复制进去，然后双击html文件即可在浏览器显示。



## 4.4 数据可视化之ECharts

### (3) 图片导出

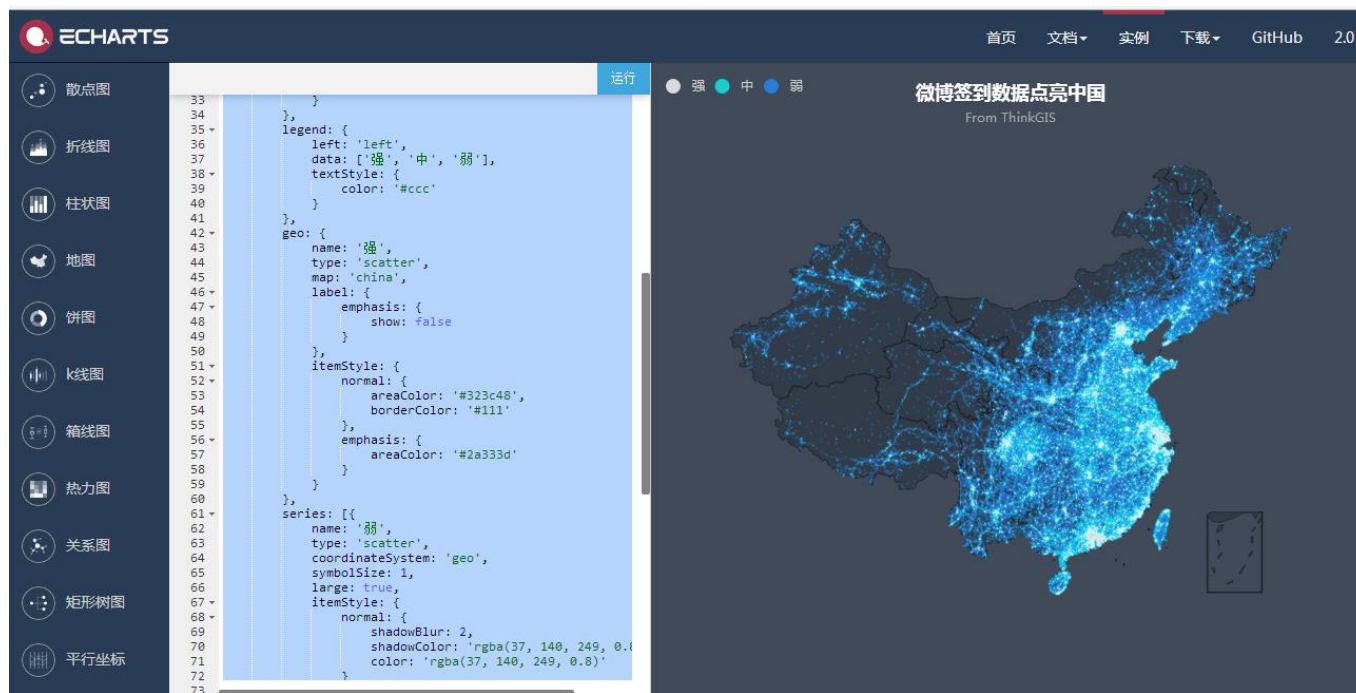
ECharts可以很方便的导出制作的图表只要在代码中加入如下代码，即可显示上图右上角的工具栏，其中最右边蓝框中的图标即为导出图表。

```
toolbox: {  
  show : true,  
  feature : {  
    dataZoom: {},  
    dataView: {readOnly: false},  
    magicType: {type: ['line', 'bar']},  
    restore: {},  
    saveAsImage: {}  
  }  
}
```

## 4.4 数据可视化之ECharts

如果只是简单的制作图表，而不是用于网页中，最简单的办法是打开ECharts提供的一个示例，将示例代码清除，直接在示例的编辑器编辑即可，而且不会出现乱码，图片导出操作和上述所说一样。

比如，在Echarts中打开“微博签到数据点亮中国”这个示例，如下图所示，将左边的代码清除，写入显示柱状图的代码，就可以运行得到柱状图（见下页）



The screenshot shows the ECharts online editor interface. On the left is a sidebar with various chart types: 散点图, 折线图, 柱状图, 地图, 饼图, k线图, 箱线图, 热力图, 关系图, 矩形树图, and 平行坐标. The main area is split into a code editor and a preview window. The code editor contains the following JSON configuration:

```
33     },
34   },
35   legend: {
36     left: 'left',
37     data: ['强', '中', '弱'],
38     textStyle: {
39       color: '#ccc'
40     }
41   },
42   geo: {
43     name: '强',
44     type: 'scatter',
45     map: 'china',
46     label: {
47       emphasis: {
48         show: false
49       }
50     },
51     itemStyle: {
52       normal: {
53         areaColor: '#323c48',
54         borderColor: '#111'
55       },
56       emphasis: {
57         areaColor: '#2a333d'
58       }
59     }
60   },
61   series: [
62     {
63       name: '弱',
64       type: 'scatter',
65       coordinateSystem: 'geo',
66       symbolSize: 1,
67       large: true,
68       itemStyle: {
69         normal: {
70           shadowBlur: 2,
71           shadowColor: 'rgba(37, 140, 249, 0.8)',
72           color: 'rgba(37, 140, 249, 0.8)'
73         }
74       }
75     }
76   ]
77 }
```

The preview window on the right displays a map of China titled "微博签到数据点亮中国" (Weibo Check-in Data Lights Up China) from ThinkGIS. The map shows a dense distribution of blue points across the country, representing check-in data. A legend at the top indicates three levels of emphasis: 强 (Strong), 中 (Medium), and 弱 (Weak).

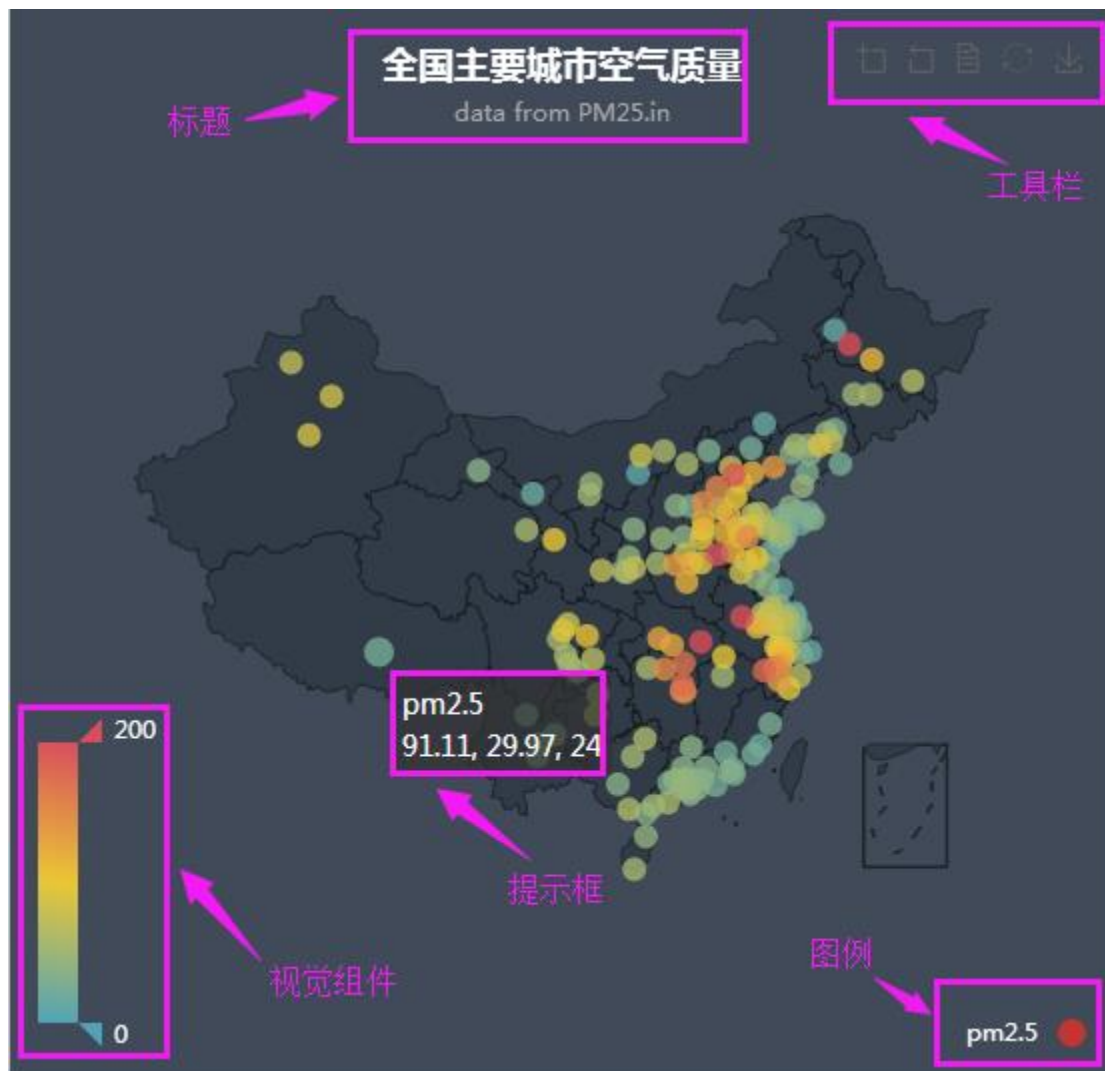
# 4.4 数据可视化之ECharts

运行得到的柱状图



# 4.4 数据可视化之ECharts

下面用全国主要城市空气质量图的示例来解释ECharts图表制作过程



## 4.4 数据可视化之ECharts

**标题组件** 即为这幅图表所展示的主题，标题有分为主标题和副标题，而且还可以为标题设置连接，文字属性等等，相应的代码为：

```
title: {  
  text: '全国主要城市空气质量', //主标题的文字  
  subtext: 'data from PM25.in', //子标题的文字  
  sublink: 'http://www.pm25.in', //子标题的链接  
  left: 'center', //标题置于中间  
  textStyle: { //设置主标题文字的风格  
    color: '#fff' //设置主标题的颜色  
  }  
},
```

## 4.4 数据可视化之ECharts

**工具栏组件** 即针对制作的图表的一些操作，例如导出图片，数据视图，动态类型切换，数据区域缩放，重置五个工具。相应的代码如下：

```
toolbox: {  
  show : true, //是否显示工具栏，默认显示  
  feature : {  
    mark : {show: true}, //这个在ECharts2.0版本中是用于画辅助线，3.0版取消了  
    dataZoom : {show: true}, //数据区域缩放  
    dataView : {show: true, readOnly: false}, //数据视图工具  
    restore : {show: true}, //配置项还原  
    saveAsImage : {show: true} //保存为图片  
  }  
},
```

## 4.4 数据可视化之ECharts

**提示框组件** 当鼠标点击或者滑过图表中的点线时，弹出关于这点线的数据信息。  
相应的代码如下：

```
tooltip: {  
    trigger: 'item'//这个为触发类型，ECharts支持两种触发类型，'item'为  
    数据项图形触发，'axis'为坐标轴触发  
},
```



## 4.4 数据可视化之ECharts

**图例组件** 图例组件展现了不同系列的标记(symbol), 颜色和名字。可以通过点击图例控制哪些系列不显示。相应代码为:

```
legend: {  
  orient: 'vertical', //图例布局方向, 有水平和垂直两种  
  top: 'bottom', //置于底端  
  left: 'right', //同时置于右边  
  data: ['pm2.5'], //这个主要用于显示图例的文本和图标, 图标默认为圆形  
  textStyle: {  
    color: '#fff' //图例字体的颜色, 默认白色  
  }  
},
```

//data代码写的较为简略, 还可以如下详细的设置:

```
data: [{  
  name: 'pm2.5', //图例的文本  
  icon: 'circle', //图例的图标, 还可以有'diamond', 'triangle'等等  
  textStyle: {  
    color: 'white' //文本的颜色  
  }  
}],
```

## 4.4 数据可视化之ECharts

**视觉组件** 视觉组件是用于进行【视觉编码】，也就是将数据映射到视觉元素（视觉通道），在图中表现为将数据根据视觉组件映射到不同颜色的图例。相应代码如下：

```
visualMap: {  
  min: 0, //视觉组件最小值,  
  max: 200, //视觉组件最大值,  
  calculable: true, //是否启用值域漫游, 即视觉组件中的最大和最小值是可以拖动的  
  color: ['#d94e5d', '#eac736', '#50a3ba'], //数据组件的三种颜色, 有柔和的过渡  
  textStyle: {  
    color: '#fff' //文本的颜色, 即数字0和200的颜色  
  }  
},
```

## 4.4 数据可视化之ECharts

**地图坐标系组件** 用于地图的绘制，支持在地里坐标系上绘制散点图，线集。  
相应代码如下：

```
geo: {  
    map: 'china', //地图类型，这里用的是中国地图，ECharts还支持其他地图  
    //图形上的文本标签，用于说明图形的一些数据信息，比如值，名称等。在这个地图中，即为省  
    的名称  
    label: {  
        emphasis: {  
            show: false //如果改为true，鼠标浮在省份上，会显示省份的名称  
        }  
    },  
    //地图区域的多边形 图形样式，有 normal 和 emphasis 两个状态，normal 是图形正常的样式，  
    emphasis 是图形高亮的样式，比如鼠标悬浮或者图例联动高亮的时候会使用 emphasis 作为图形的样  
    式。  
    itemStyle: {  
        normal: {  
            areaColor: '#323c48', //正常情况显示的省份颜色  
            borderColor: '#111' //正常情况下省份的边界颜色  
        },  
        emphasis: {  
            areaColor: '#2a333d' //鼠标浮在省份上时，即高亮时省份的颜色  
        }  
    }  
},
```

## 4.4 数据可视化之ECharts

**series**（系列列表），即在同一个图表上可以制作一系列的多个图表，然后叠在一起，形成一个图表。每个系列通过**type**决定自己的图表类型。**ECharts**支持多种图表类型，折线图（**line**），柱状图（**bar**），饼图（**pie**），散点（气泡）图（**scatter**）等。全国主要城市空气质量图采用的散点图（**scatter**）。系列组件中还将数据映射到地图的图标，相应代码如下：

## 4.4 数据可视化之ECharts

```
series: [  
  {  
    name: 'pm2.5',//这个系列图表的名称  
    type: 'scatter'//图表的类型，这里使用的是散点（气泡）图  
    coordinateSystem: 'geo',//使用地理坐标系，还有二维坐标系和极坐标系  
    data: convertData([//这是个函数，将数据拼接成指定格式。  
      //城市空气质量数据请看附录B  
    ]),  
    symbolSize: 12,//图标的大小  
    label: {//图形上的文本标签，可用于说明图形的一些数据信息。  
      normal: {  
        show: false//图标上是否显示数值，如果改为true，则每个图标上有相应的数值  
      },  
      emphasis: {  
        show: false  
      }  
    },  
    itemStyle: {//图标的样式  
      emphasis: {//图的高亮样式  
        borderColor: '#fff',//高亮时的边界颜色  
        borderWidth: 1//高亮时边界宽度  
      }  
    }  
  }  
]
```

## 4.4 数据可视化之ECharts

其中用到数据转换函数定义为：

```
var convertData = function (data) {  
    var res = [];  
    for (var i = 0; i < data.length; i++) {  
        var geoCoord = geoCoordMap[data[i].name]; //取出城市的坐标  
        if (geoCoord) {  
            res.push(geoCoord.concat(data[i].value)); //将空气质量值拼接在坐标后面  
        }  
    }  
    return res;  
};
```

制作一张图表，最重要的是数据，没有数据，图表无从谈起，所以在**series**里面，需要先将数据转换为指定的格式，在全国空气质量图中，格式为：省份名称+省份坐标+每个省份的空气质量值，然后**ECharts**能自动将数据映射到地图的图标上。至此，全国主要城市空气质量图分析完毕。