

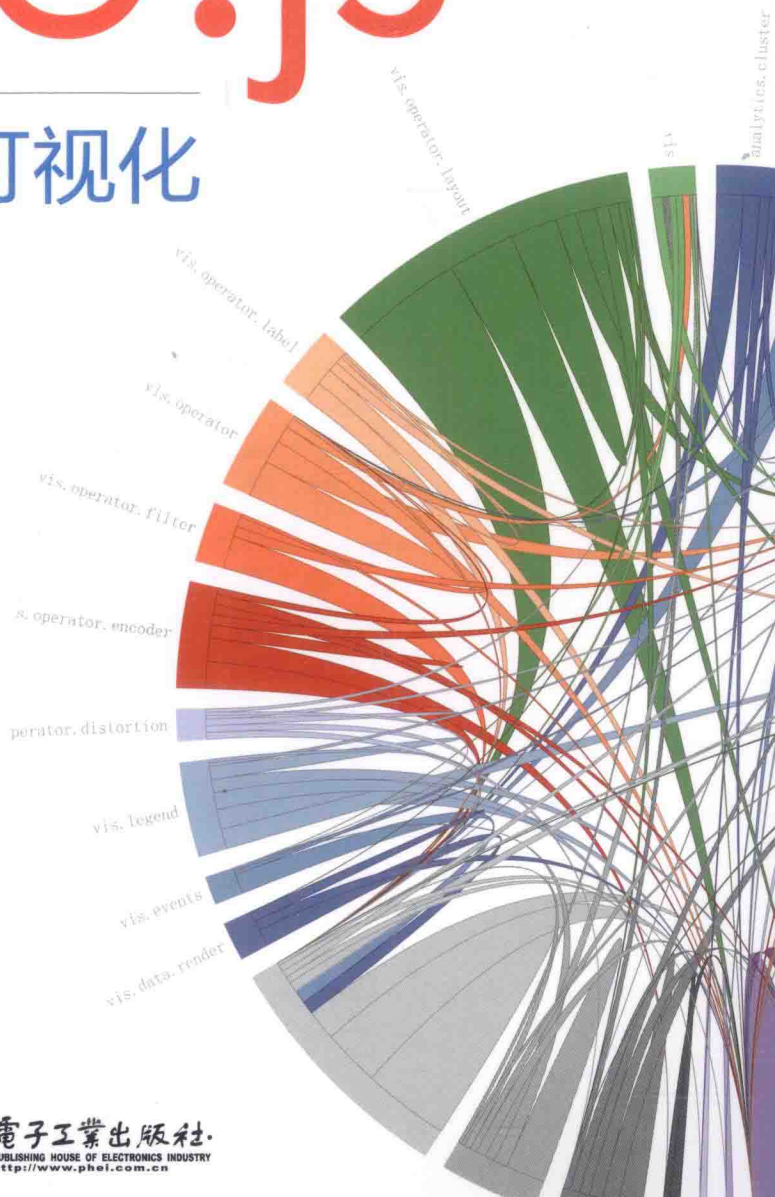
# 精通

· 吕之华 著 ·

# D3.js

## 交互式数据可视化 高级编程

大数据时代需要可视化工具，  
D3是世界最流行的可视化函数库。



# 精通 D3.js

• 吕之华 著 •

交互式数据可视化  
高级编程



电子工业出版社

Publishing House of Electronics Industry

## 内 容 简 介

本书以当前流行的数据可视化技术 D3.js 为主要内容，分为三大部分，共计 13 章。第一部分讲述基础知识，第二部分学习制作各种常见图表，第三部分讲解交互式图表及地图的进阶应用。本书作为一个相对完整的 D3.js 教程，讲解此技术所有重要的知识点，既有基础入门知识，又有相对深入的内容。笔者秉持以下原则：由易到难，循序渐进，图文并茂，清晰易懂。

本书适合有一定计算机基础的读者，需要熟悉 C、C++、Java、JavaScript 等至少一门编程语言，能够理解基础的数据结构和算法。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

图书在版编目（CIP）数据

精通 D3.js：交互式数据可视化高级编程/吕之华著. —北京：电子工业出版社，2015.8  
ISBN 978-7-121-26776-5

I. ①精... II. ①吕... III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字（2015）第 169494 号

责任编辑：付睿

印 刷：北京丰源印刷厂

装 订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：25.25 字数：562 千字 彩插：9

版 次：2015 年 8 月第 1 版

印 次：2015 年 8 月第 1 次印刷

定 价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：（010）88258888。

# 目 录

---

第 1 章 D3 简介	1
1.1 D3 是什么	1
1.1.1 D3 简史	2
1.1.2 D3 的优势	2
1.1.3 D3 的适用范围	3
1.2 数据可视化是什么	3
1.2.1 目的	4
1.2.2 构成要素	4
1.2.3 相关概念	6
1.3 图表种类	6
1.4 学习方法	11
第 2 章 Web 前端开发基础	13
2.1 浏览器和服务器	14
2.1.1 浏览器	14
2.1.2 服务器	15
2.2 HTML&CSS	16
2.2.1 HTML 元素	17
2.2.2 CSS 选择器	20
2.2.3 综合运用	23
2.3 JavaScript	25
2.3.1 在 HTML 中使用 JavaScript	26
2.3.2 语法	26
2.3.3 变量	27
2.3.4 数据类型	28

2.3.5	操作符	30
2.3.6	语句	32
2.3.7	函数	34
2.3.8	对象	34
2.3.9	数组	35
2.4	DOM	36
2.4.1	结构	37
2.4.2	访问和修改 HTML 元素	37
2.4.3	添加和删除节点	38
2.4.4	事件	39
2.5	SVG	40
2.5.1	位图和矢量图	40
2.5.2	图形元素	41
2.5.3	文字	46
2.5.4	样式	47
2.5.5	标记	48
2.5.6	滤镜	50
2.5.7	渐变	51
<b>第 3 章</b>	<b>安装和使用</b>	<b>53</b>
3.1	安装	53
3.1.1	下载文件	54
3.1.2	网络引用	54
3.2	搭建服务器	54
3.3	Hello, World	57
3.4	绘制矢量图	58
3.5	调试	59
<b>第 4 章</b>	<b>选择集与数据</b>	<b>61</b>
4.1	选择元素	61
4.2	选择集	63
4.2.1	查看状态	63
4.2.2	设定和获取属性	63
4.3	添加、插入和删除	66
4.4	数据绑定	67

4.4.1	datum()的工作过程	68
4.4.2	data()的工作过程	71
4.4.3	绑定的顺序	74
4.5	选择集的处理	76
4.5.1	enter 的处理方法	76
4.5.2	exit 的处理方法	77
4.5.3	处理模板	78
4.5.4	过滤器	79
4.5.5	选择集的顺序	79
4.5.6	each()的应用	80
4.5.7	call()的应用	80
4.6	数组的处理	81
4.6.1	排序	81
4.6.2	求值	82
4.6.3	操作数组	86
4.6.4	映射 (Map)	89
4.6.5	集合 (Set)	91
4.6.6	嵌套结构 (Nest)	92
4.7	柱形图的制作	96
4.7.1	矩形和文字	97
4.7.2	更新数据	101
<b>第 5 章</b>	<b>比例尺和坐标轴</b>	<b>105</b>
5.1	定量比例尺	105
5.1.1	线性比例尺	106
5.1.2	指数和对数比例尺	109
5.1.3	量子 and 分位比例尺	110
5.1.4	阈值比例尺	112
5.2	序数比例尺	113
5.3	坐标轴	118
5.3.1	绘制方法	119
5.3.2	刻度	121
5.3.3	各比例尺的坐标轴	122
5.4	柱形图的坐标轴	123

5.5	散点图的制作 .....	125
第 6 章	绘制 .....	128
6.1	颜色 .....	128
6.1.1	RGB .....	129
6.1.2	HSL .....	130
6.1.3	插值 .....	131
6.2	线段生成器 .....	132
6.3	区域生成器 .....	136
6.4	弧生成器 .....	137
6.5	符号生成器 .....	140
6.6	弦生成器 .....	142
6.7	对角线生成器 .....	144
6.8	折线图的制作 .....	145
第 7 章	动画 .....	151
7.1	过渡效果 .....	151
7.1.1	过渡的启动 .....	152
7.1.2	过渡的属性 .....	155
7.1.3	子元素 .....	158
7.1.4	each()和 call() .....	160
7.1.5	过渡样式 .....	162
7.2	定时器 .....	163
7.2.1	setInterval 和 setTimeout .....	163
7.2.2	d3.timer .....	164
7.3	应用过渡的场合 .....	165
7.4	简单的动画制作 .....	171
7.4.1	时钟 .....	171
7.4.2	小球运动 .....	172
第 8 章	交互 .....	174
8.1	交互式入门 .....	174
8.1.1	鼠标 .....	176
8.1.2	键盘 .....	178
8.1.3	触屏 .....	180

8.2	事件	182
8.3	行为	183
8.3.1	拖曳	184
8.3.2	缩放	186
<b>第 9 章</b>	<b>导入和导出</b>	<b>191</b>
9.1	文件导入	191
9.1.1	JSON	192
9.1.2	CSV	194
9.1.3	XML	198
9.1.4	TEXT	199
9.2	文件导出	200
9.2.1	导出为 SVG 文件	200
9.2.2	编辑矢量图	203
<b>第 10 章</b>	<b>布局</b>	<b>206</b>
10.1	布局是什么	206
10.2	饼状图	207
10.3	力导向图	213
10.4	弦图	221
10.5	树状图	228
10.6	集群图	234
10.7	捆图	238
10.8	打包图	245
10.9	直方图	248
10.10	分区图	255
10.11	堆栈图	261
10.12	矩阵树图	268
<b>第 11 章</b>	<b>地图</b>	<b>274</b>
11.1	地图的数据	274
11.1.1	获取数据	275
11.1.2	简化数据	278
11.1.3	GeoJSON	280
11.1.4	TopoJSON	284



11.2	中国地图 .....	285
11.2.1	基于 GeoJSON .....	285
11.2.2	基于 TopoJSON .....	289
11.3	地理路径 .....	297
11.3.1	地理路径生成器 .....	297
11.3.2	形状生成器 .....	301
11.4	投影 .....	306
11.5	球面数学 .....	315
<b>第 12 章</b>	<b>友好的交互 .....</b>	<b>317</b>
12.1	提示框 .....	317
12.1.1	饼状图的提示框 .....	318
12.1.2	提示框的样式 .....	321
12.2	坐标系中的焦点 .....	323
12.2.1	折线图的焦点 .....	323
12.2.2	为折线图添加提示框 .....	329
12.3	元素组合 .....	334
12.3.1	饼状图的拖曳 .....	335
12.3.2	移入和移出 .....	336
12.3.3	合并 .....	345
12.4	区域选择 .....	347
12.4.1	在 SVG 画板里选择一块区域 .....	348
12.4.2	散点图的区域选择 .....	350
12.5	开关 .....	353
12.5.1	思维导图的构造思路 .....	353
12.5.2	思维导图的制作 .....	356
<b>第 13 章</b>	<b>地图进阶 .....</b>	<b>363</b>
13.1	值域的颜色 .....	363
13.2	标注 .....	368
13.2.1	标注地点 .....	368
13.2.2	夜光图 .....	370
13.3	标线 .....	373
13.3.1	带有箭头的标线 .....	373
13.3.2	球状地图的标线 .....	377

13.4 拖动和缩放 .....	378
13.4.1 平面地图 .....	378
13.4.2 球面地图 .....	381
13.5 力导向地图 .....	383
13.5.1 Voronoi 图和 Delaunay 三角剖分 .....	383
13.5.2 力导向的中国地图 .....	387
附录 A 彩色插图 .....	393
附录 B 参考文献 .....	410

# 第 1 章

## D3 简介

本章内容包括：

D3 是什么

- 数据可视化是什么
- 常见可视化图表的种类
- 学习 D3 的方法

笔者第一次接触到 D3，首先是被绚丽多彩的图表吸引，然后陶醉于这些图表的可交互特性，而且动画流畅简洁、赏心悦目。近年来，可视化越来越流行，许多报刊杂志、门户网站、新闻媒体都大量使用可视化技术，使得复杂的数据和文字变得十分容易理解，有一句谚语“一张图片价值相当于一千个字”，的确是名副其实。对于信息爆炸式增长的今天，我们没有时间一条一条地阅读，希望的是一眼就能找到自己想要的信息。图片不仅容易理解，而且容易记忆，是值得推广的信息传达方式。

### 1.1 D3 是什么

D3 的全称是 Data-Driven Documents，直译为“数据驱动文档”。听名字有点抽象，简单概括为一句话：

**D3 是一个 JavaScript 的函数库，是用来做数据可视化的。**

文档指 DOM，即文档对象模型（Document Object Model）。D3 允许用户绑定任意数据到

DOM，然后根据数据来操作文档，创建可交互的图表。

JavaScript 文件的后缀名通常为.js，故 D3 也常称为 D3.js。D3 提供了各种简单易用的函数，大大简化了 JavaScript 操作数据生成图表的难度。由于它本质上是 JavaScript，所以用 JavaScript 也是可以实现所有功能的，但 D3 能大大减轻你的工作量，尤其是在数据可视化方面，D3 已经将生成可视化的复杂步骤精简到了几个简单的函数，你只需要输入几个简单的数据，就能够转换为各种绚丽的图形。有过 JavaScript 基础的朋友一定很容易理解它。

### 1.1.1 D3 简史

大数据时代蓬勃发展的今天，每天都有惊人的数据产生，怎么提取并显示有用的信息，变得越来越重要。在 Web 浏览器中进行可视化也成为迫切的需求，有许多项目以此为目标。2009 年，Mike Bostock、Jeff Heer、Vadim Ogievetsky 共同开发了 Protovis，可以算是 D3 的前身。2011 年，他们停止了 Protovis，使用 JavaScript 开发了一个新的项目，这就是 D3。

JavaScript 诞生于 1995 年，经过近 20 年的发展，如今已经成为 Web 浏览器上事实上的标准语言，其应用范围扩展到服务器端、移动领域等，并且越来越完善。因此，D3 采用 JavaScript 作为开发语言。

2011 年 2 月 18 日，Mike Bostock 发布了 v1.0 版本。

2011 年 8 月 25 日，v2.0 版本发布，功能大幅增强，应用逐渐增多。

2012 年 12 月 22 日，v3.0 版本发布，修复了大量 bug，功能更加稳健。

笔者写作时，最新版本为 v3.4.13。v3.0 版之后差别不大，本书以 v3.x 版为标准。

### 1.1.2 D3 的优势

可视化的库有很多，基于 JavaScript 开发的库也有很多，D3 有什么优势呢？

#### 1. 数据能够与 DOM 绑定在一起

D3 能够将数据与 DOM 绑定在一起，使得数据与图形成为一个整体，即图形中有数据、数据中有图形。那么在生成图形或更改图形时，就可以方便地根据数据进行操作。并且，当数据更改之后，图形的更新也会很方便。

#### 2. 数据转换和绘制是独立的

将数据变成图表，需要不少数学算法，一些可视化库的做法是：提供一个函数 `drawPie()`，输入数据，直接绘制出饼状图。

D3 的做法是：提供一个函数 `computePie()`，可将数据转换成饼状图的数据，然后开发者使用自己喜欢的方式来绘制饼状图。

看起来，好像 D3 使问题变麻烦了，但是在图表比较复杂的时候，直接绘制的饼状图往往达不到要求，细微的部分没有办法更改。将两者分开，就极大地提高了自由度，以至于开发者甚至可以使用其他的图形库来显示 D3 计算的数据。

### 3. 代码简洁

jQuery 是网页开发中很常用的库，其链式语法被很多人喜爱。D3 也采用了这一语法，能够一个函数套一个函数，使得代码很简洁。

### 4. 大量布局

饼状图、树形图、打包图、矩阵树图等，D3 将大量复杂的算法封装成一个一个“布局”，能够适用于各种图表的制作。

### 5. 基于 SVG，缩放不会损失精度

SVG，是可缩放的矢量图形。D3 大部分是在 SVG 上绘制的，并且提供了大量的图形生成器，使得在 SVG 上绘制图形变得简单。另外，由于 SVG 是矢量图，所以放大缩小不会有精度损失。

## 1.1.3 D3 的适用范围

D3 开发的应用是显示在网页上的。因此，开发者需将数据置于服务器端，并在网页文件（HTML）中插入 D3 代码。用户通过浏览器请求此网页文件，就会看见开发者希望让用户看到的可视化内容。

Ben Fry 在他的著作《Visualizing Data》中将数据可视化的过程分为七个步骤。

- (1) 获取——Acquire
- (2) 分析——Parse
- (3) 过滤——Filter
- (4) 挖掘——Mine
- (5) 表现——Represent
- (6) 改善——Refine
- (7) 交互——Interact

前四步不属于 D3 的处理范围，更多的是处理后三步，即表现、改善、交互。

## 1.2 数据可视化是什么

数据可视化（Data Visualization）起源于 18 世纪，William Playfair 在出版的书籍《The Commercial and Political Atlas》中第一次使用了柱形图和折线图，当时是为了表示国家的进出口

量,在今天依然这么使用。19 世纪初,出版了《Statistical Breviary》一书,里面第一次使用了饼状图,这三种图形都是至今最常用的最著名的可视化图形。19 世纪中叶,数据可视化主要被用于军事用途,用来表示军队死亡原因、军队的分布图等。进入 20 世纪,数据可视化有了飞跃性的发展。1990 年,在人机界面学会上,作为信息可视化原型的技术被发表。1995 年,IEEE Information Visualization 正式创立,信息可视化作为独立的学科被正式确立。近年,随着大数据时代的到来,数据可视化作为大量数据的呈现方式,成为当前重要的课题。

### 1.2.1 目的

The main goal of data visualization is its ability to visualize data, communicating information clearly and effectively.

数据可视化的目的,是要对数据进行可视化处理,以使得能够明确地、有效地传递信息。

— Vitaly Friedman

比起枯燥乏味的数值,人类对于大小、位置、浓淡、颜色、形状等能够有更好、更快的认识。经过可视化之后的数据能够加深人对于数据的理解和记忆。

例如有以下的数据,请找出最大值:

[ 321, 564, 1391, 245, 641, 798, 871 ]

数据量比较小,用肉眼也能找出来,但更好的办法是将数据进行可视化处理,如图 1-1 所示。

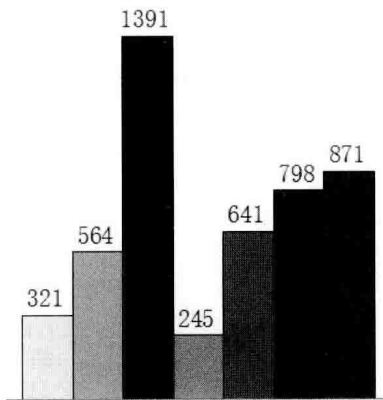


图 1-1 可视化后的柱形图

如图 1-1 所示,很明显,经过可视化之后,数据变得容易理解了。

### 1.2.2 构成要素

数据可视化的手法很多,其中有一些共通的视觉要素。

- 坐标。数值的位置被对应到直角坐标系或极坐标系上，如图 1-2 所示。

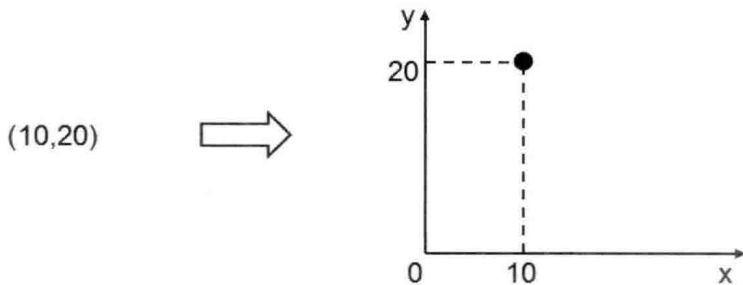


图 1-2 对应到坐标系上

- 大小。数值的大小被对应到图形的大小上，如图 1-3 所示。

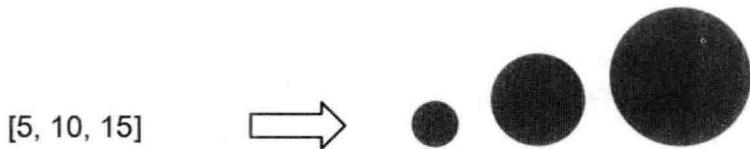


图 1-3 对应到大小上

- 色彩。数值的分类和界限等对应到不同的颜色上，如图 1-4 所示。

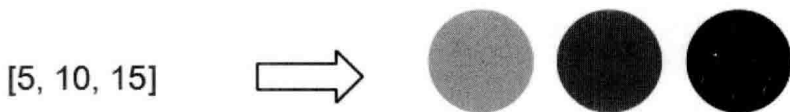


图 1-4 对应到颜色上

- 标签。数值的特征用标签来标记，如图 1-5 所示。



图 1-5 对应到标签上

- 关联。数值之间的联系，用关联线条等连接起来，如图 1-6 所示。

如图 1-2 至图 1-6 所示，列举了一些常见的将数据对应到视觉要素的方式。这些方式经过多数人的使用，是最容易被理解也是最容易制作的，但是视觉要素并非局限于此。

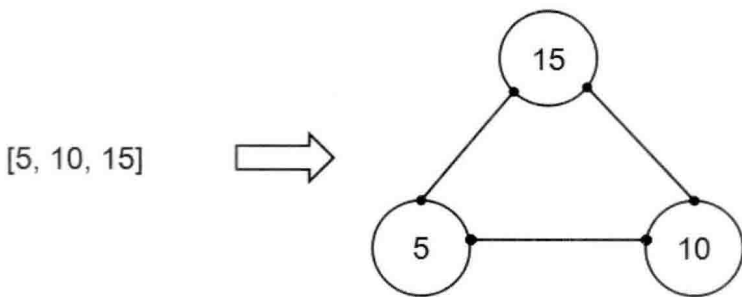


图 1-6 对应到关联线条上

### 1.2.3 相关概念

数据可视化 (Data Visualization) 和信息可视化 (Information Visualization) 很相近, 有时几乎可以等同。但严格来说它们是不同的, 它们的不同可以总结为一句话:

数据可视化是对数字信息进行可视化, 信息可视化是对**数字信息**和**非数字信息**进行可视化。

## 1.3 图表种类

用于数据可视化的图表种类相当多, 这里列举一些常用的图表。实际应用时可单独使用, 也可以多种联动。

一般来说, 图表要尽可能简单, 能用简单的就用简单的。有的人可能会觉得简单的图表太古老、不大气, 而追求复杂的图表, 这反而有点本末倒置。数据可视化的目的, 是要使数据明确地、有效地传递, 而简单的图表是能够最快被人认可的。凭感觉自创图表也是可以的, 但要注意此图表是否比原来的更简单易懂。

下面列举部分可视化图表。要注意, 一种图表对应的汉语名称可能有多种, 在本节中使用的名称将作为本书的标准。

### 1. 柱形图

柱形图是最常见、最容易理解的图表, 使用矩形的长短来表示数据的大小。数据类型一般是形如“时间—销售额”这样的二维数据集, 图表要表现的是“随着时间的变化, 销售额的变化情况”(如图 1-7 所示)。

如将图 1-7 的  $x$  轴和  $y$  轴替换, 得到横向的柱形图, 有时也称为条形图, 但本质是一样的, 都是用柱形的长短来表示数据的大小。此外, 还可用矩形的宽窄来表示第三维的数据, 使得一个图里的信息量更大, 但会加深理解的难度。



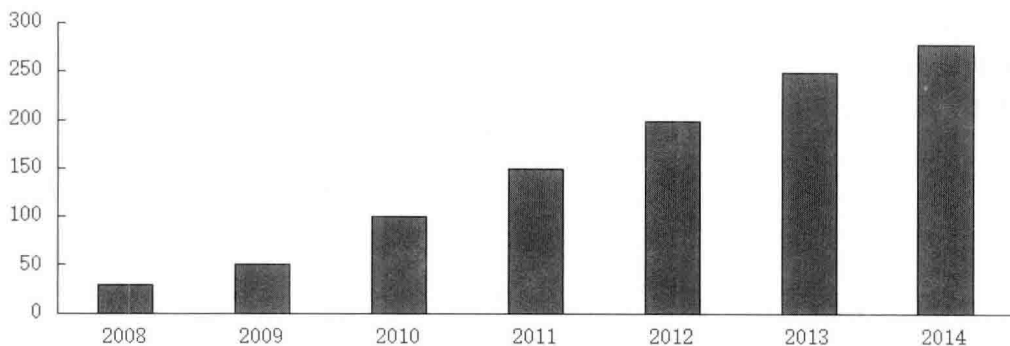


图 1-7 柱形图

## 2. 散点图

散点图使用三维数据集，将其中的二维数据分别对应到  $x$  轴和  $y$  轴，再将第三维用点表示，而第三维数据是对应前二维的。其直观表现为在  $x$  轴和  $y$  轴的坐标系中分布着很多点，如图 1-8 所示。

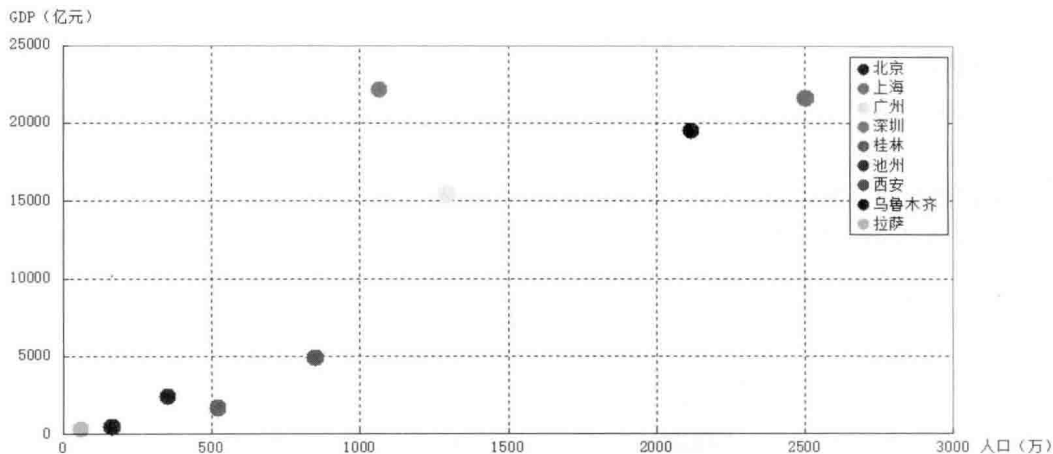


图 1-8 散点图

图 1-8 中  $x$  轴表示人口， $y$  轴表示 GDP，第三维数据为城市，所表示的内容为各城市在人口和 GDP 的二维坐标系中的分布。

## 3. 折线图

折线图的目的与柱形图类似，也适合表示在二维数据集中，某一维相对于另一维的变化趋势，不同的是：

- 折线图较适合连续的数据，柱形图较适合离散的数据。