

# 数据可视化Echarts

一手微信study322

王红元 coderwhy

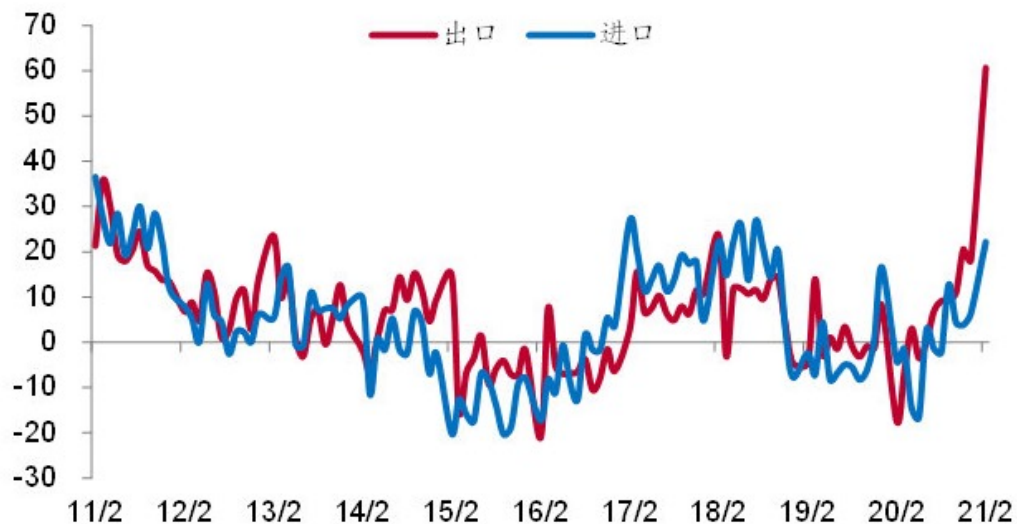
# 什么是数据可视化？

## ■ 什么是数据可视化呢？

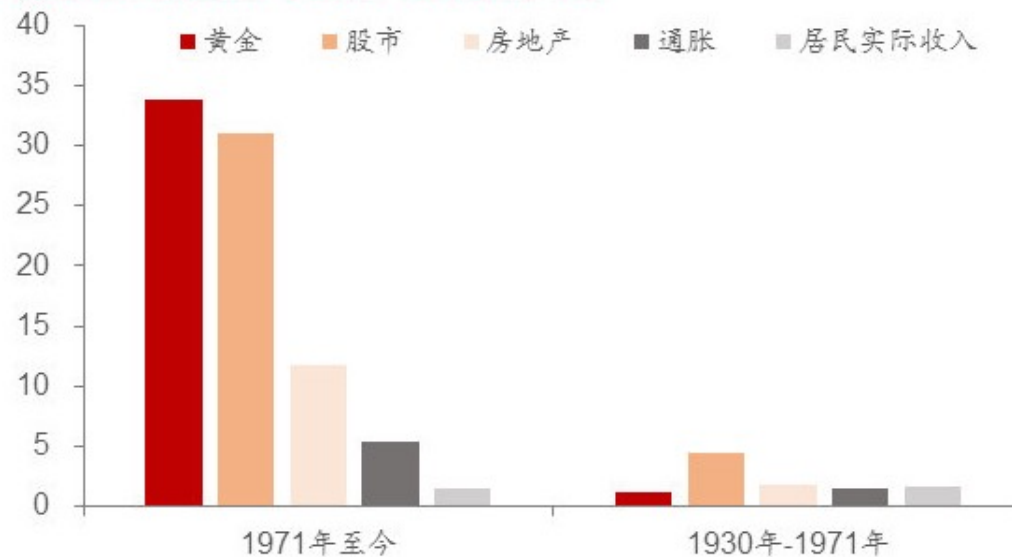
- 为了清晰有效地传递信息，数据可视化使用统计图形、图表、信息图表和其他工具；
- 可以使用点、线或条，对数字数据进行编码；
- 数据可视化让我们可以更加清晰的去认识、理解、表达数据；

一手微信study322

图 我国出口和进口金额同比增速（%）



美国历史上各大类资产、通胀、收入的涨幅（倍）



## ■ 前端进行数据可视化的工具非常多：

□ 常见的框架：ECharts、g2、d3、vis、hightChart等等；

□ g2框架封装：bizcharts(react) viser(vue)；

□ 地理可视化：g2、L7、高德의 Loca、菜鸟的 鸟图；

□ 3D可视化：three.js；

一手微信study322

## ■ 目前使用最多的还是ECharts

- 什么是ECharts呢？官方的解释是：一个基于 JavaScript 的开源可视化图表库；
- ECharts的历史：
  - ECharts由百度团队开源；
  - 2018年初，捐赠给Apache基金会，成为ASF（*Apache Software Foundation*，简称为*ASF*，*Apache*软件基金会）孵化级项目；
  - 2021年1月26日晚，Apache基金会官方宣布 ECharts项目正式毕业，成为Apache顶级项目；
  - 2021年1月28日，ECharts 5线上发布会举行；  
一手微信study322
- ECharts的特点：
  - 丰富的图表类型：提供开箱即用的 20 多种图表和十几种组件，并且支持各种图表以及组件的任意组合；
  - 强劲的渲染引擎：Canvas、SVG 双引擎一键切换，增量渲染、流加载等技术实现千万级数据的流畅交互；
  - 专业的数据分析：通过数据集管理数据，支持数据过滤、聚类、回归，帮助实现同一份数据的多维度分析；
  - 优雅的可视化设计：默认设计遵从可视化原则，支持响应式设计，并且提供了灵活的配置项方便开发者定制；
  - 健康的开源社区：活跃的用户保证了项目的健康发展，也贡献了丰富的第三方插件满足不同场景的需求；
  - 友好的无障碍访问：智能生成的图表描述和贴花图案，帮助视力障碍人士了解图表内容，读懂图表背后的故事；



# 使用Echarts的步骤

## ■ 第一步：我们可以通过以下方式获取ECharts：

- 从 [Apache ECharts 官网下载界面](#) 获取官方源码包后构建；
- 在 ECharts 的 [GitHub](#) 获取；
- 通过 npm 获取 echarts , `npm install echarts --save`；
- 通过 [jsDelivr](#) 等 CDN 引入；

一手微信study322

## ■ 第二步：引入Echarts

- 可以通过不同的方式引入

## ■ 第三步：初始化Echarts对象，并且设置配置进行绘制

- 通过`echarts.init(dom, theme, options)`初始化；
- 通过`setOption`方法设置绘制的数据；

# ECharts配置信息

```
const 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]  
    }  
  ]  
}
```

一手微信study322



# canvas vs svg

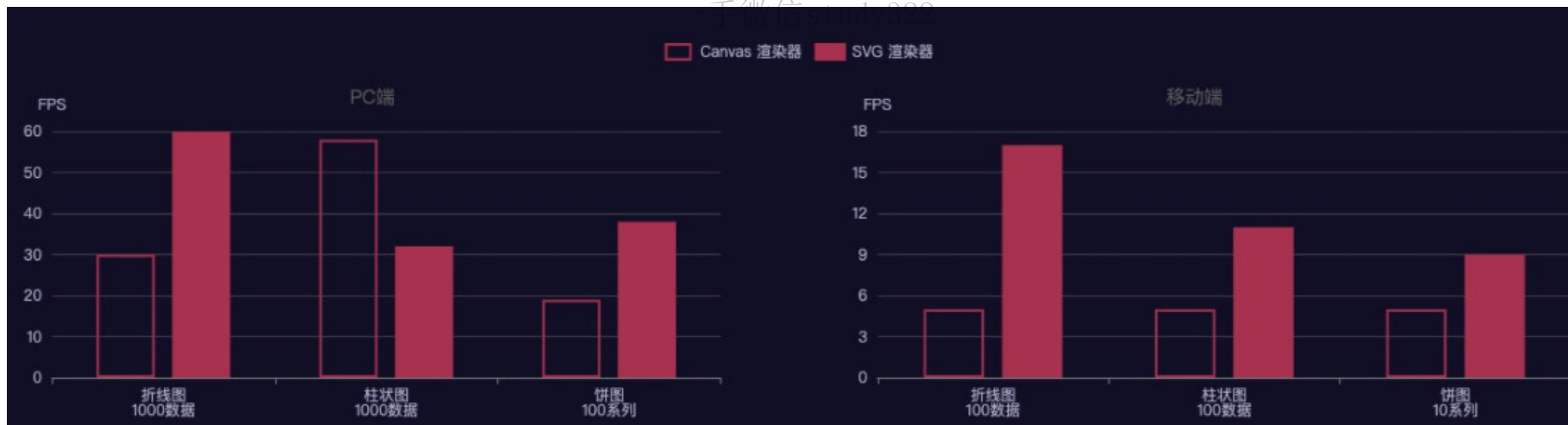
- 通常在渲染图表是我们会选择 SVG或者canvas进行渲染：
  - 通常情况下，这两种渲染模式是比较相近的，并且是可以相互替换的；
  - 但是在一些场景中，它们的表现和能力有一定的差异；
  - 对于它们之间的取舍，一直是没有一个明确、标准的答案的，也是一个经常被拿到讨论的话题；
- ECharts最初采用的是canvas绘制图表，从ECharts4.x开始，发布了SVG渲染器，提供了另外一种选择。
- 那么它们之间到底如何选择呢？
  - 一般来说，Canvas 更适合绘制图形元素数量非常大（这一般是由数据量大导致）的图表（如热力图、地理坐标系或平行坐标系上的大规模线图或散点图等），也利于实现某些视觉特效；
  - 但是，在不少场景中，SVG 具有重要的优势：它的内存占用更低（这对移动端尤其重要）、渲染性能略高、并且用户使用浏览器内置的缩放功能时不会模糊；

# canvas vs svg性能测试

■ ECharts在不同的设备上，进行了性能的测试：

□ 从图片来看，在这些场景中，SVG 渲染器相比 Canvas 渲染器在移动端的总体表现更好；

□ 当然，这个实验并非是全面的评测，在另一些数据量较大或者有图表交互动画的场景中，目前的 SVG 渲染器的性能还比不过 Canvas 渲染器；





# canvas vs svg的选择

## ■ 那么到底选择哪一个渲染器呢？

- 在软硬件环境较好，数据量不大的场景下（例如 PC 端做商务报表），两种渲染器都可以适用，并不需要太多纠结；
- 在环境较差，出现性能问题需要优化的场景下，可以通过试验来确定使用哪种渲染器；
  - 比如在须要创建很多 ECharts 实例且浏览器易崩溃的情况下（可能是因为 Canvas 数量多导致内存占用超出手机承受能力），可以使用 SVG 渲染器来进行改善；
  - 大略得说，如果图表运行在低端安卓机，或者我们在使用一些特定图表如 水球图等，SVG 渲染器可能效果更好；
  - 数据量很大、较多交互时，可以选用 Canvas 渲染器；