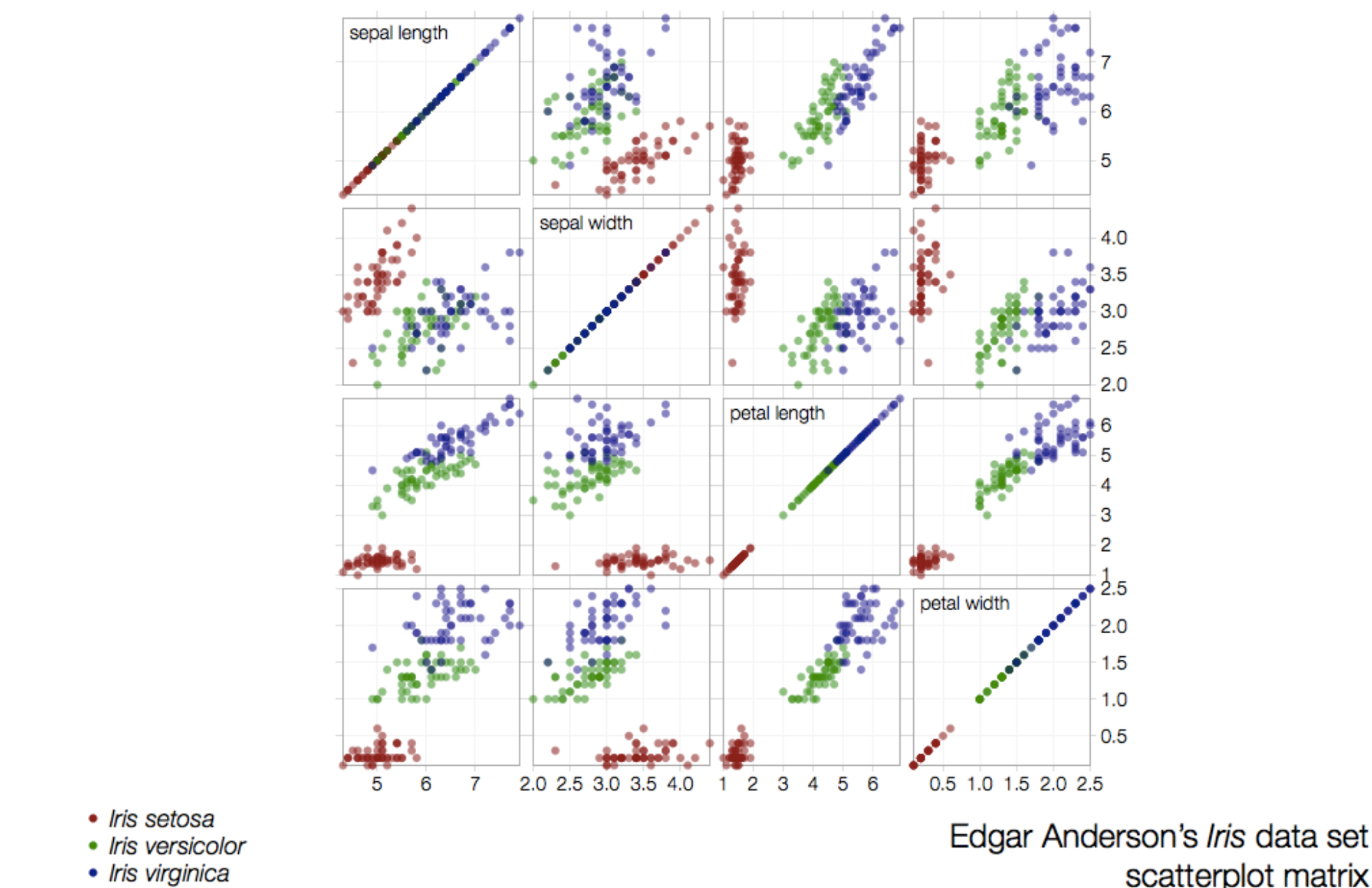


二.巫英才资料：<http://mbostock.github.io/d3/talk/20111116/iris-splom.html>



二.其他相关资料：

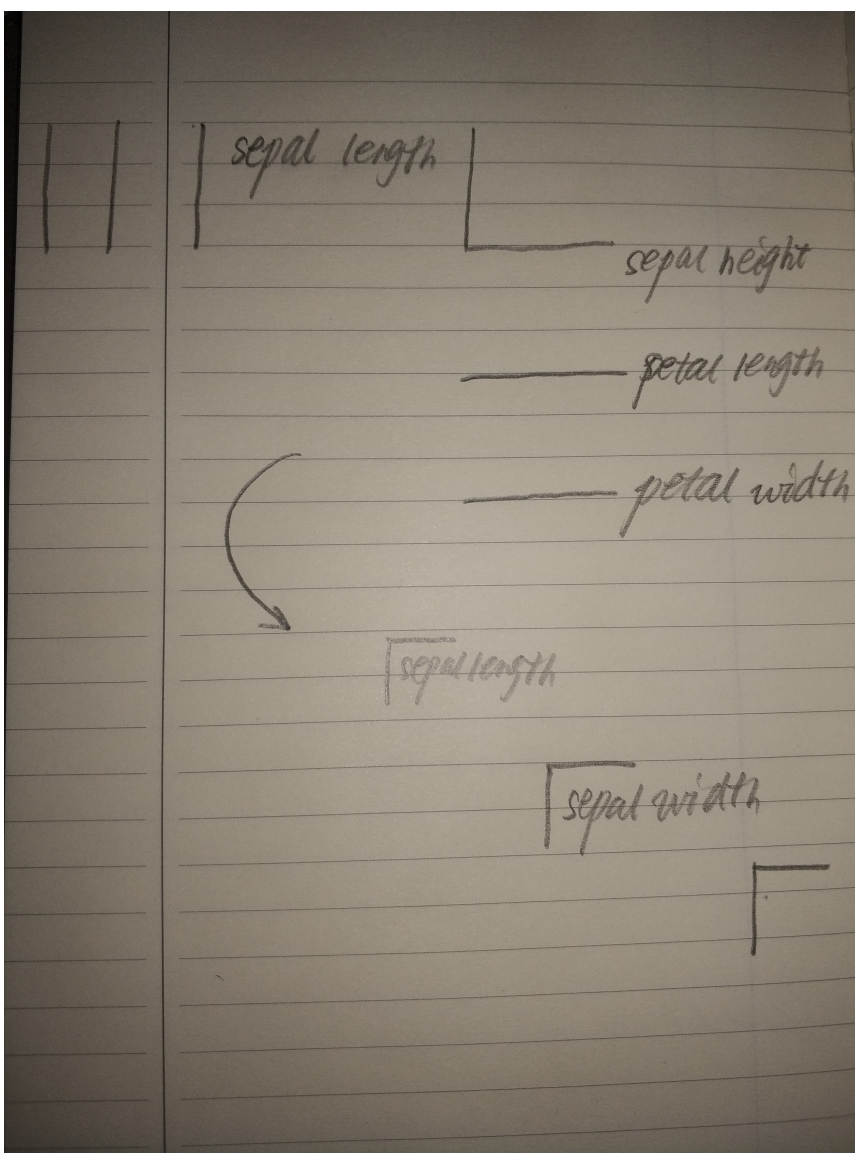
- 1.原资料d3.js相关代码分析：<https://blog.csdn.net/u014711869/article/details/72917861>
- 2.不同维度下的scatter matrix以及它的适用场合：<https://www.r-bloggers.com/scatterplot-matrices/>

In conclusion,

- Scatterplot matrices are **good** for determining rough linear correlations of metadata that contain continuous variables.
- Scatterplot matrices are **not so good** for looking at discrete variables.

- 3.D3.js实现：<https://bl.ocks.org/mbostock/4063663>
- 4.简介：<https://www.jianshu.com/p/06295967e6b4>

Scatter Plot Matrix 最初是由john and paul turkey 提出的, 如之前所提, 它能够让你一眼就看到所有的变量的两两相关性。但是这个技术有一个很大的缺陷, 就是单数据的维度变得很大的时候计算机的屏幕就无法容纳这么多的散点图。



- 5.简易版本（没有刷子功能的）：<https://bl.ocks.org/mbostock/3213173>

三.代码解读：

- 1.首先了解一下svg:

在 SVG 画布的预定义元素里, 有六种基本图形：

- 矩形 <rect>
- 圆形 <circle>
- 椭圆 <ellipse>
- 线段 <line>
- 折线 <polyline>
- 多边形 <polygon>

另外, 还有一种比较特殊, 也是功能最强的元素：

- 路径 <path>

画布中的所有图形, 都是由以上七种元素组成。

显然, 这里面没有坐标轴 **<axis>** 这种元素。如果有的话, 我们可以采用类似以下的方式定义：

我们需要用其他元素来组合成坐标轴, 最终使其变为类似以下的形式：

```
<g>
<!-- 第一个刻度 -->
<g>
<line></line> <!-- 第一个刻度的直线 -->
<text></text> <!-- 第一个刻度的文字 -->
</g>
<!-- 第二个刻度 -->
<g>
<line></line> <!-- 第二个刻度的直线 -->
<text></text> <!-- 第二个刻度的文字 -->
</g>
...
<!-- 坐标轴的轴线 -->
<path></path>
</g>
```

分组元素 <g>, 是 SVG 画布中的元素, 意思是 group。此元素是将其他元素进行组合的容器, 在这里是用于将坐标轴的其他元素分组存放。

如果需要手动添加这些元素就太麻烦了, 为此, D3 提供了一个组件: d3.svg.axis()。它为我们完成了以上工作。

二、定义坐标轴

要生成坐标轴, 需要用到比例尺, 它们二者经常是一起使用的。

```
// 数据
var dataset = [ 2.5 , 2.1 , 1.7 , 1.3 , 0.9 ];
//定义比例尺
var linear = d3.scale.linear()
    .domain([0, d3.max(dataset)])
    .range([0, 250]);

var axis = d3.svg.axis()
    .scale(linear) //指定比例尺
    .orient("bottom") //指定刻度的方向
    .ticks(7); //指定刻度的数量
```

第 1 – 2 行: 定义数组。

第 4 – 7 行: 定义比例尺, 其中使用了数组 dataset。

第 9 – 12 行: 定义坐标轴, 其中使用了线性比例尺 linear。其中：

- d3.svg.axis(): D3 中坐标轴的组件, 能够在 SVG 中生成组成坐标轴的元素。
- scale(): 指定比例尺。
- orient(): 指定刻度的朝向, bottom 表示在坐标轴的下方显示。
- ticks(): 指定刻度的数量。

三、在 SVG 中添加坐标轴

定义了坐标轴之后, 只需要在 SVG 中添加一个分组元素 <g>, 再将坐标轴的其他元素添加到这个 <g> 里即可。代码如下：

```
svg.append("g")
    .call(axis);
```

上面有一个 **call()** 函数, 其参数是前面定义的坐标轴 axis。

在 D3 中, call() 的参数是一个函数。调用之后, 将当前的选择集作为参数传递给此函数。也就是说, 以下两段代码是相等的。

```
function foo(selection) {
    selection
        .attr("name1", "value1")
        .attr("name2", "value2");
}
foo(d3.selectAll("div"))
```

和

```
d3.selectAll("div").call(foo);
```

因此,

```
svg.append("g").call(axis);
```

与

```
axis(svg.append(g));
```

是相等的

如何导入数据<https://blog.csdn.net/lzh1zz/article/details/39755649>  
D3.js坐标轴：<https://www.cnblogs.com/LO-ME/p/5384388.html>  
D3.js 比例尺与坐标：<https://www.cnblogs.com/chenjinxinlove/p/5608165.html>  
js中的call方法:<https://blog.csdn.net/longjuesi/article/details/29859599>