一. 实验目的

学会使用 html、css、JavaScript、svg、d3.js 进行数据可视化,掌握设计 html 页面框架、数据绑定、定义比例尺、柱状图和地图等视图的绘制、视图之间的交 互等方法,并学会利用绘制的可视化视图对数据进行分析。

二. 实验步骤

1. 配置和安装网络编程环境

1.1 下载并安装代码编辑器: sublime Text

	Sublime	Text	File	Edit	Selection	Find	View	Goto	Tools	Project	Window	Help
	•				index-r	o1-incon	nplete.ht	Iml		UN	NREGISTERE	D
	▶ index	-p1-inco	molete	.html ×								
			in nprise									
		2									III Street	-
	2 2 chear	4									Some second	-
	4	-										
	5 < <u>scr</u>	iot>										
	6 var	vpn	host	name_d	lata = "wvi	pn.szu	.edu.c	n";			Teres.	
	7 var	vpn	prot	ocol_d	iata = "ht	tps";					and a second	
	8 var	vpn_	prot	ocol_h	iost = "ht	tps://	wvpn.sz	zu.edu.	cn";		4	
	9 var	vpn_	_app_l	hostna	me_data =	"7772	6476706	5e69737	468656	26573742	1f5	
ų .	fb449	9d353e	e615e	79469a	b68d1b263	10a639	204";					
	.0 var	_vpn_	_app_i	protoc	ol_data =	"http:	s";					
	12 var	vpn_	_app_	ile -	ata = ;	-von/i	c/main	is2ver	20100	000-01"		
6 1	ia var	vpn_	work	er mod	P = 2:	-vpii/J	57 110 1111	1311001	-20150	905-01 ,		
1	14 var	vpn	host	crypt	= true:							
1	15 <th>ript></th> <th></th>	ript>										
1	.6 <scr< b="">.</scr<>	ipt sı	rc="/	wengin	ie-vpn/js/i	main.j	s?ver=2	2019090	9-01"	charset=		
	utf-8	8"> <th></th> <th>t> <st< th=""><th>yle></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></st<></th>		t> <st< th=""><th>yle></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></st<>	yle>							
1	.7	oody	t i									
1	.8	font	:-fam:	ily: H	elvetica,	Arial	, sans-	-serif				
	.9	}										
	21	h1 {										st47
2	22	back	karou	nd-col	or: steel	blue:						1
2	23		rodo									
2	24			ange t								
2	25											
2	26											- DAR
2	27	}										0.2
	Line 1, Col	umn 1							Spaces	:: 2	HTML	

1.2 下载解压 d3.js

●●● 33 D3/3-Deta-Driven Document x + ← → C △ # d3/s.org Ⅲ 用用 22 ▲ 10 ⊂ ◎ M 型 型 目 ▲ 9 1	名称
	API.md
Like visualization and creative coding? Try interactive JavaScript notebooks in Observable!	CHANGES.md
D3.js is a JavaScript library for manipulating documents based on data. D9 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 fring yourself to a proprietary framework, combring youreful visualization	d3.js
components and a data-driven approach to DOM manipulation. Download the latest version (5.16.0) here:	d3.min.js
To link directly to the latest release, copy this snippet:	LICENSE
<pre><script src="https://d3js.org/d3.v5.min.js"></script> The full source and tests are also available for download on OitHub.</pre>	README.md

1.3 MacOS - 搭建本地服务器

MacOS 自带 Apatch 服务器。首先要保证服务器是打开的,在 Google 中输入 http://127.0.0.1, 点击确定,成功显示: It Works! 如果访问失败了的话,在终端输入 sudo apachectl start 启动服务器。

	🕆 n	nac –	b	ash –	- 8	0×	24				
Ferry:~ mac\$ [恢复于 20 <u>20年5月23日 下午</u>	6:11	:26]									
Ferry:~ mac\$ <mark>sudo apachect1</mark> [Password: Ferry:~ mac\$	star	t									
● ● ● ● 127.0.0.1 × - ← → C △ ① 127.0.0.1	+						[57]				
It works!	知豆	ā 🔮	•		8	S.	学	Ø	e	R	A

前往 Apache 服务器的本地文件地址: /Library/WebServer/Documents。把自己做的文件,比如 一张照片或者一个 html 放到该地址下,比如照片名字: PoweredByMacOSX.gif.



在 Google 中输入 http://127.0.0.1/ PoweredByMacOSX.gif 显示如图:



2.完成页面总体框架

2.1 修改标题颜色为白色,并加上内边距为 5pt。

修改标题颜色很简单,用 color: white 代码即可将颜色设置为白色。

内边距:处于父元素和子元素之间,设置在父元素上,可以理解成物品和盒子中间塞的那层泡沫塑料,使用 padding 属性来设置。有两种方法,第一种是单独写法,分别使用 padding-top、padding-bottom、padding-left、padding-right 设置上、下、左、右内边距,第二种是复合写法,即 padding 后面可跟多个值,中间用空格分离。这里使用复合写法,用 padding:5px 代码表示上下左右内边距 均为 5px。代码如下图所示。



2.2 给每个 svg 元素添加一个灰色的实线边框, 边框粗细为 2pt

CSS 边框属性允许指定一个元素边框的样式和颜色,可以用 border-width 属性为边框指 定宽度, border-style 用来定义边框的样式, border-color 属性用于设置边框的颜色。也可以 在一个属性中设置边框,即同时在"border"属性中设置 border-width、border-style、border-color。 这里 border: 2px solid gray 表示将边框宽度设置为 2px,样式为实线 solid,颜色为灰色 gray,且为每个 svg 元素都如此设置。代码如下图所示:



2.3 将类 Mainwindow 中的元素调整方式改为 display flex

网页布局(layout)是 CSS 的一个重点应用,其中,Flex 布局,可以简便、完整、响应 式地实现各种页面布局。Flex 是 Flexible Box 的缩写,意为"弹性布局",用来为盒状模型提 供最大的灵活性。任何一个容器都可以指定为 Flex 布局。这里将类 Mainwindow 中的元素 调整方式设置为 display flex。代码如下图所示:



2.4 修改后的 index-w1-incomplete.html 文件效果如下图所示,可以看到,与所要求的目标

效果一致:

•••	0	inde	x-p1-	incon	nplete	e.html	ľ	×	+																				
$\leftarrow \ \rightarrow $	С	企	()	文件	= /I	Users	s/ma	c/Des	sktop	/inde	x-p1	-inco	mple	te.ht	ml														
应用	٢	*	G	ω	\odot	۲	Μ	知	知	묘	â	٢	-		Π	8	۲	学	0	Ø	R	/A	0	ö	-	×	秀米	0	conda pkg
Glo	b	al	S	Sa	le	S																							
Cate	go	rie	es					_	\r e	as								0	the))							_		

3.绘制柱状图

(1) 在 id 为 CategoryChart 的 svg 容器中添加三个<g>元素,分别用来存放矩形 块、水平坐标系和垂直坐标系,并设置这个 svg 容器的长度、宽度等信息。



(2) 导入 csv 文件中的数据,其中匿名函数的参数 d 代表数据中的每一个数据 元素。由于从 csv 文件中读入的是字符串,因此需要用"+"将字符串转为数值。 数据读入后赋值给变量 barData,根据 barData 来调用 drawBarChart 函数绘制柱 状图。



(3) 将数据绑定到元素。map 可按照指定方式遍历数组,此处利用 map 返回销售数据。

```
function drawBarChart(){
  var bindings = d3.select("#rects").selectAll("rect").data(barData.map(function(d){
    return d.sales;
  }));
  var updateRects = bindings;
  var enterRects = bindings.enter();
  var exitRects = bindings.exit();
```

(4)分别对 enter、update 和 exit 部分进行处理。Enter 部分是选择出未绑定元素 的数据,并使用 append 添加元素后赋予新元素属性值。与 Enter 部分相比, Update 部分不需要添加新元素,因此不需要 append,只需用 attr 和 style 修改元素属性 即可。Exit 部分选择出未绑定数据的多余元素,用 remove 删掉未绑定的元素即 可。



(5) 定义水平方向比例尺,将销量映射为水平方向的坐标;定义垂直方向比例尺,将类别映射为垂直方向的坐标。其中,domain为输入值域,range和rangeRound 是与输入值域对应的输出范围。

```
var xScale = d3.scaleLinear()
.domain([0, maxSales])
.range([barChartPadding.left, barChartWidth - barChartPadding.right]);
var yScale = d3.scaleBand()
.domain(barData.map( function(d){
   return d.subCategory;
})) // 得到所有类别信息
.rangeRound([barChartPadding.top, barChartHeight - barChartPadding.bottom])
output range
.paddingInner(0.3);
```

(6) 定义颜色比例尺,将利润值映射为颜色。利用 scaleDiverging 创建一个发散 比例尺,并使用 interpolateRdBu 这一发散配色方案。

```
var colorScale = d3.scaleDiverging()
.domain([-maxProfit, 0, maxProfit])
.interpolator(d3.interpolateRdBu);
```

(7)添加 x 和 y 坐标轴。其中, ticks(3)可将坐标轴划分成等大的三段。"axisLeft" 表明要在左边绘制坐标轴。select("#xAxis")和 select("#yAxis")分别表示选择 id 为 xAxis 和 yAxis 的元素,并用 call 对元素进行操作。

如果不对坐标轴移位,坐标轴会默认紧挨 svg 元素的 0 坐标,因此要用 transform 属性的 translate 对坐标轴进行移位。由于 barChartHeight、 barChartPadding.bottom、barChartPadding.left 为数值型数据,需要用加号转化为 字符串。

<pre>// 绘制X坐标轴 var xAxis = d3.axisBottom(xScale).ticks(3); d3.select("#xAxis").call(xAxis) .attr("transform","translate(0," +(barChartHeight-barChartPadding.bottom)+ ")" .attr("class","axis");</pre>)
//绘制Y坐标轴。 var yAxis = d3.axisLeft(yScale); d3.select("#yAxis").call(yAxis) .attr("transform","translate(" +barChartPadding.left+ ",0)") .attr("class","axis");	

(8) 柱状图最终绘制效果如图。

● ● ● ● ② 127.0.0.1/实验2.2b.html × +		
← → C 介 ① 127.0.0.1/実验2.2b.html		
ⅲ 应用 🔤 🍪 🔤 🕞 🔿 🎱 M 🚂 🧾 🚍 🍦	📀 🛸 🚳 🖬 💿 🍪 🖻 🥸	R /A 💶 🛅 📣 X 秀米 🔾 conda pk
Global Sales		
Ciobal Sales		
Ostanarias	A	Other
Categories	Areas	Other
7.914		
Chairs -		
Bookcases -		
Supplies -		
Storage -		-
Paper –		
Labels -		
Fasteners -		
Envelopes -		
Binders -		
Art -		
Appliances -		
Phones -		
Machines -		
Copiers -		
Accessories		

(9) 有趣的发现:我们可以通过改变颜色映射表来改变颜色映射效果,比如使用 interpolateRainbow,得到的可视化效果如图所示





Global Sales



4.绘制地图

4.1 对 HTML 文件(原实验二(b)的基础上)进行重修改。

(1) 首先要让 Other 区域在最下方,而不是像实验二(b)的结果一样排在最右方。 我们已经知道: Flex 是 Flexible Box 的缩写,意为"弹性布局",用来为盒状模型提供最大的灵活性,这里通过 display: flex 命令将类 mainView 中的元素设置为弹性盒对象的元素。而 flex-wrap 属性规定 flex 容器是单行或者多行,同时横轴的方向决定了新行堆叠的方向。(注 意:如果元素不是弹性盒对象的元素,则 flex-wrap 属性不起作用。)

所以这里用 flex-wrap: wrap 命令规定灵活的项目在必要的时候拆行或拆列,从而实现让让 Other 区域在最下方

svg{ border: 2px grey soli }	20 svg{ 21 border: 2px grey solid; 22 } 23
<pre>.mainView{ display: flex: flex-wrap: wrap; }</pre>	24 .mainView{ 25 display: flex; 26 } 27
 实验 C	28 29 30 实验 b 31 <body></body>

(2) 修改各区域的标题。将 Categories 修改为 sales and profit,将 Areas 修改为 profit of all categories。由于需要使用 topojson-client 来实现 TopoJSON 格式到 GeoJSON 的格式转换,所以需要添加 <script src="https://unpkg.com/topojson-client@3"></script>代码进行引用



(3)要想地图绘制如何在 Area 区域,而不是压在直方图上方。正如前面绘制条形图已定义的全局变量 var barChart = d3.select("#CatergoryChart"),而 Area 区域的 id 是 Map,所以绘制 地图之前要用 var mapUS = d3.select("#Map")选择对应的 id。同时定义好后面要用到的变量。



4.2 读入文件(sub-categories-states-sales.csv), 重新绘制第一个的视图的柱状图。

(1) 首先,我们打开 sub-categories-states-sales.csv 文件,查看数据文件的含义,可以看到各行分别代码家具类别、国家、州、子类别、利润、销量

category	country	state	subCategory	profit	sales
Furniture	Switzerland	Zürich	Chairs	1303.5	3159.12
Furniture	Venezuela	Zulia	Furnishings	-49.192	267.228
Furniture	Venezuela	Zulia	Chairs	-39.804	198.576
Furniture	Venezuela	Zulia	Bookcases	-76.368	569.772
Furniture	Senegal	Ziguinchor	Furnishings	12.66	31.74
Furniture	China	Zhejlang	Tables	59.166000000	2076.606
Furniture	China	Zhejlang	Furnishings	126.06	353.73

(2) 读入 csv 文件后,首先用 filter 方法筛选出美国的数据并转换为对象数组。

filter 方法是用来过滤对应 selection 中的元素。我们通过传入一个函数给.filter 方法来告诉 d3, 如果该函数返回 true 则该元素将被包含住, 如果返回 false,则不被包含在 filter 的返回 selection 中, 而.filter 则返回 filtered selection.



(3) 然后由于 csv 文件读入的数据是字符串,因此需要将利润和销售数据通过"+"运算符转换成数值形式。

```
dataUS.forEach((item,i)=>{//转换为数值类型
    item.profit=+item.profit;
    item.sales=+item.sales;
});
```

(4)使用 d3. nest 方法将数据进行切片,由我们决定以什么样的方式将这些数据切片,这里 按类别重新组织数据,使切片后的数据的 key 为 subCategory,并将按类别重新组织后的数 据存储为变量 nestBySubCategory。这里 nestBySubCategory 为一个对象数组. 用 key function 来组成 value,每个对象都有一个 key 属性和 value 属性,这些 entries 的 values 属性是一个数组,是所有的原始数据对象。

```
nestBySubCategory = d3.nest()//按类别重新组织数据
.key(function(d) {
    return d.subCategory;
}).entries(dataUS);
```

(5) 按类别重新组织数据后,我们可以用如下图命令方便地计算每一类别的总利润和总销量。这里使用 d3.map 方法,会自动给每个元素加上以索引为 key 的对象,并且可以用 map.keys()获取所有的 key,用 map.values()获取所有的值,用 map.entries()获取键值对数组。这里返回值为类别名称、总利润、总销量,存在外部变量 barData 中。

```
//类别重新组织数据后,我们可以方便地计算总销量和总记录(存在外部变量barData中)
barData = nestBySubCategory.map(function(d) {
    var sumProfit = d3.sum(d.values, function(item) {
        return item.profit;
    });
    var sumSales = d3.sum(d.values, function(item) {
        return item.sales;
    });
    return {
        subCategory: d.key,
        profit: sumProfit,
        sales: sumSales
        <u>};
    };
})</u>
```

(6)将 nestBySubCategory 转换为映射表形式,方便后续快速检索指定类别。 最后根据外部变量 barData,进行绘制柱状图。这里因为重新设置了代码,实验二(b)中的 drawBarChart 函数可以无需修改,所以直接调用 drawBarChart 函数来绘制柱状图。

```
// 转换为映射表形式,方便后续快速检索指定类别
nestBySubCategory = d3.map(nestBySubCategory, function(d) {
    return d.key;
});
drawBarChart();
```

(7) 用服务器打开 HTML 文件。可以看到直方图如下图所示:



4.3 在右侧的第二个视图,绘制等值区域图,显示该超市在美国每个州的利润情况。

(1) 读入地图文件 states-albers-10m. json,存储为外部变量 mapInfo。然后用 filter 筛选出美国的数据。解释同 2.2

(2)由于 csv 文件读入的数据是字符串类型,所以通过 "+" 运算符将利润和销售数据转换 成数值类型。



(3) 使用 d3. nest 方法将数据进行切片,这里按州重新组织数据,使切片后的数据的 key 为 state,并将按州重新组织后的数据存储为变量 nestByState。方法同 2.4

```
//按每个州的情况重新组织数据,存为外部变量nestByState;
nestByState = d3.nest()
.key(function(d) {
    return d.state;
}).entries(dataUS);
```

(4) 按州重新组织数据后,我们可以用如下图命令方便地计算每一个州的总利润和总销量。 这里返回值为州名称、总利润、总销量,存在外部变量 mapData 中。



(5)将 mapData 转换为映射表,以方便根据州的名称来得到对应的总利润,该外部变量记录为 profitInfoMap。最后实现调用 drawMap()来根据外部变量(MapInfo profitInfoMap) 来绘制等值区域图,



(6) 在 drawMap 函数中,先计算每个州最大利润和最小利润,然后以最大利润和最小利润 为范围用 d3. scaleDiverging 定义一个发散比例尺,使用的颜色为 d3. interpolateRdBu。



(7) 接着绘制地图区域。

- 使用 topojson 中的 feature 方法,将地图由 topojson 格式转换为 geojson 格式。这里 通过 d3 的路径生成器可以把 geojson 格式转换成对应的 svg 元素。
- 这里使用 join 方法可以代替 enter、update 和 exit 的处理。
- 利用所定义的发散比例尺,将各州的利润按不同的颜色映射在地图上,这里通过 properties.name 可以得到州名,get 方法可以通过州名来获取各州利润,由于源数据 中有一些州的利润数据是缺失的,这里将这些州的地图填充为灰色。



(8) 这里利用 transform 属性,对地图进行大小缩放和位置移动。scale 可以对地图进行 缩放,变量数值小于 1 则为缩小,这里使用 scale(0.5,0.5)进行缩小。然后通过调节,使 用 translate(0,180)将地图移动到 svg 元素合适的位置。注意这里 scale 和 translate 要 写在同一个字符串中,否则缩小和移动的效果会重叠。最后给每个区域加上 title 标签,使 得鼠标移动到某个州的区域上方时可以显示州的名称和此州的利润。



(9) 接着绘制区域的边界。之后对其进行同样的大小缩放和位置移动。



(10) 通过服务器打开 HTML 文件, 可以看到地图效果如下:

← → C ① ① 127.0.0.1/实验c.html

🗰 应用 🔤 📸 🔩 🟳 🔵 🙆 M 🛃 冠 🖬 🤹 🐓 🖤 🍇 📅 🧶 💥 😤 🏵

Global Sales

Categories: sales and profit

Areas: profit of all categories



Other



4.4 在右侧的地图上方添加图例(选做,加分项)。

(1) 图例的绘制需要使用 scaleQuantize 比例尺,所以这里原来绘制等值区域地图时使用 的 scaleDiverging 比例尺注释掉,并且修改为 scaleQuantize 比例尺。



(2) 定义 legend 变量用于绘制图例,并且设置绘制标题区域。

	会制图例		
var	<pre>legend = mapUS.app</pre>	end("g")	
	.attr("id", "legen	d")	
	<pre>.attr("transform",</pre>	"translate(220,	30)");
var	<pre>legendWidth = 250,</pre>		
	legendHeight = 25,		
	<pre>legendTop = 16; //</pre>	绘制标题区域	

(3) 定义一个线性比例尺,将颜色编号映射成图例上的位置。

```
// 从颜色的编号映射成图例上的位置
var x = d3.scaleLinear()
.domain([0, color.range().length])
.rangeRound([0, legendWidth]);
```

(4) 绘制图例的矩形块。



(5) 绘制图例的标题。



(6) 定义一个线性比例尺,将颜色编号映射成利润数据。

var values = d3.scaleLinear() .domain(x.domain()) .range(color.domain());

(7) 绘制图例的坐标轴。此处用 d3.format("d")将坐标轴上刻度数值设置为整数,用 ticks(9) 将设置坐标轴有 9 个刻度(即 8 种颜色)。



Global Sales



Categories: sales and profit

Areas: profit of all categories

X 秀米 🔘

5.实现柱状图和地图的交互

5.1 鼠标点击柱状图任意条形时,给该条形加上红色边框,并根据该条形对应的 子类别来更新等值区域图

(a) 修改 drawBarChart() 函数,给所有条形添加交互响应。设置一个监听器,由鼠标点击 来触发,使得点击该条形时,之前点击其他条形产生的红色边框被去掉,被点击的条形的边 框变为红色。而且点击时调用函数,更新地图。



(b) 定义函数 updateMap。当传入参数为"all"时,等值区域图显示所有产品在每个州的利润;当传入参数为产品类别时,显示该类别的产品在每个州的利润。该函数中,具体的更新等值区域图的操作是通过函数 updateMapByNewProfit()来完成的。



(c)在 updateMapByNewProfit 函数中,根据输入数据重新定义颜色映射表,并根据新的颜色映射表修改地图上各州的颜色。此外,需要根据输入的数据更新各州的标签。因为地图没有改变,此处不需要重新绑定数据。



5.2 鼠标点击柱状图的空白区域时,更新等值区域图,显示所有产品的利润信息 (a)为了能实现响应,我们需要在柱体下方添加一个透明矩形块来监测事件响 应。修改 drawBarChart()函数,来添加该矩形块。为方便显示,我们把该矩形块 设为半透明。我们把该矩形块插入到柱状图之前。



(b) 在 drawBarChart 函数内,给该矩形块添加事件响应。即点击时,之前选中的条形边缘不再高亮。调用 updateMap 函数更新等值区域图,输入参数为"all",显示所有产品的利润信息。



5.3 鼠标点击右侧任何州时,该州地图边缘高亮显示,同时柱状图上更新显示该 州不同类型的产品的销量和利润

(a)修改 drawMap()函数,给所有条形添加交互响应。使得点击地图上某个区域时,之前所选中区域产生的红色边框被去掉,被新选中区域的边框变为红色。为了使边框更明显,需要把边框设置得更粗一些。



(b)定义函数 updateBar()。当传入参数为"all"时,条形图显示全球市场下每个产品的销量 情况;当传入参数为州名时,条形图显示选中的州中每个产品的销量情况该函数中,具体的 更新等值区域图的操作是通过函数 updateBarByNewProfit ()来完成的。



(c)在 updateBarByNewProfit 函数中,根据输入的数据重新定义颜色映射表和水平 方向比例尺。该函数的输入为数组,代表该区域的销售情况。这里我们需要重新 绑定数据,并根据新的数据来修改每个条形。这里注意条形图的条形首次和重复 绑定数据时都应该使用键函数。



5.4 鼠标点击地图的空白区域时,更新条形图,显示所有区域的销售和利润信息 (a)在地图下方添加一个透明矩形块进行事件响应。在 drawMap 函数中添加如 下代码以添加矩形块。



(b)在 drawMap 函数内,给该矩形块添加事件响应。即点击时,之前选中的区域 地图边缘不再高亮。调用 updateBar 函数更新条形图,输入参数为 "all",显示所 有区域的销售和利润信息。



5.5 最后效果如下图所示:

Categories: sales and profit

Area: profits of Binders





Categories: sales and profit

Area: profits of all categories





Categories: sales and profit in ArizonaArea: profits of all categories



Categories: sales and profit

Area: profits of all categories



6.用折线图表示全球销量的变化

6.1 数据处理

(1) 将原本的全球销售的 xls 数据文件转换成 csv 文件,导入此 csv 文件。首先 将文件中字符串形式的日期数据用 new Date 转换成 JavaScript 的日期数据。



(2) 将初步处理后的数据以 Market 为 key 进行切片。



(3)对切片后的数据,计算每一个市场下,2011年至2014年各年的利润总和。 利用 item.OrderDate.getFullYear()返回 item 的订购日期中的年份(年份为数值型 数据),并用 d3.sum 得到每年的利润总和。最后以类的形式返回数据,存在变量 linechart 中。



6.2 绘制折线图

(1) 编写函数 drawLineChart()来实现折线图的绘制。首先,用 d3.max 计算出 2011 年至 2014 年各年的利润最大值,再计算这四个利润中的最大值,得到这四 年所有市场中的利润最大值。

```
function drawLineChart()
```

```
ł
 var maxProfit2011 = d3.max(linechart, function(d){
   return d.profit2011;
 });
 var maxProfit2012 = d3.max(linechart, function(d){
   return d.profit2012;
 }):
 var maxProfit2013 = d3.max(linechart, function(d){
   return d.profit2013;
 });
 var maxProfit2014 = d3.max(linechart, function(d){
   return d.profit2014;
 });
 var datayear=[maxProfit2011,maxProfit2012,maxProfit2013,maxProfit2014];
 var maxProfit=d3.max(datayear, function(d) {
   return d;
 });
```

(2) 定义颜色映射表。

var color1 = d3.scaleOrdinal(d3.schemeRdBu[7]);

(3) 定义水平和垂直方向比例尺。水平方向比例尺的输入值域为年份,垂直方向的输入值域为0 到最大利润。

```
xScaleLine = d3.scaleLinear()
.domain([2011,2014])
.range([barChartPadding.left, barChartWidth - barChartPadding.right]);
yScaleLine = d3.scaleLinear()
.domain([0, maxProfit])
.range([barChartHeight - barChartPadding.bottom,barChartPadding.top]);
```

(4) 绘制 x、y 坐标轴。x 轴刻度为 4。使用 translate 将坐标轴移动到合适位置。

(5)将利润转化成数组对象的形式,每个数组元素也是一个数组,其中第一个 元素为年份,第二个元素为对应的利润。转化后的数据存储在 dataset 变量中。

var dataset=linechart.map(function(d) {
 return [[2011,d.profit2011],[2012,d.profit2012],[2013,d.profit2013],[2014,d.profit2014]];
});

(6) 用 d3.line()新建一个线生成器, line.x([x])设置 x 坐标访问器, line.y([y])设置 y 坐标访问器。

var linepath=d3.line().x(function(d){return xScaleLine(d[0])})
.y(function(d){return yScaleLine(d[1])});

(7) 遍历 dataset,用 dataset 中的每一个数据元素即按年份划分的不同市场的利

润绘制折线。添加绘制折线的 g 元素和 path,再用线段生成器绘制折线。stroke 修改线条粗细和颜色。



(8) 遍历 dataset, 给每年的数据在折线上的位置加上圆点标示用 transform 将圆 点移动到对应位置。



(9) 遍历 dataset,为折线图添加图例。每遍历一次,添加一个绘制图例的矩形 块,每个矩形块在纵坐标方向上与上一个矩形块相隔同样的距离,横坐标相同,颜色与其表示的折线相同。,此处 title 从 linechart 中获得。

```
oth.append("g")
.append("rect")
.attr("id", "legend")
.attr("x",barChartPadding.left+5)
.attr("y",4+10*i)
.attr("width",30)
.attr("width",30)
.attr("height",4)
.attr("fill",function(d){
    return color1(i)
});
oth.append("text").attr("class","title")
.text(linechart[i].subMarket)
.attr("font-size","10px")
.attr("y",9)
.attr("transform",`translate(120, ${10*i})`);
```

(10) 修改折线图的标题。



(11) 最终绘制出的各个市场的利润随年份变化折线图如图所示。



Market: Profit by years

7.数据分析

(1) 柱状图的颜色表示某子类别商品利润的大小,颜色越蓝表明利润越大,越 红表明利润越小,长度则表示销售量大小。由柱状图可以看出,Copiers 产品利 润最高,其次 Phones 产品、Accessories 产品利润也比较高; Phones 产品的销量 最高,其次 Chairs 产品的销量也较高,综合利润和销量, Phones 产品都处于较 高的位置,所以未来可着重发展,投入更多的资源,从而获得更多的收入。

Categories: sales and profit



(2)我们前面已经知道 Copiers 产品的利润在所有子类别是最高的,通过交互可以看到 Copiers 产品在 Washington、Indiana 和 California 三个州的利润是最高的,分别为 9442.42、8849.97 和 7889.85,所以应该加入对这三个州的资源投入,重点发展 Copiers 产品。





(3)我们前面已经知道综合利润和销量, Phones 产品都处于较高的位置, 所以 未来可着重发展, 投入更多的资源, 那么应该在哪些州重点发展呢?通过交互我 们可以看到, Phones 产品在 New York 州的利润是最高的, 为13399.19 所以应 该加入对纽约州的资源投入, 重点发展 Phones 产品。



(4)等值区域地图上每个州的颜色表示全部类别的商品在此州的销售利润,颜 色越蓝表明利润越大,越红表明利润越小。由等值区域地图可以看出,California 和 New York 颜色最深,说明这两个州的利润最高,分别为 76381.39 和 74038.55, 而 Texas 的红色最深,说明这个州的利润最低,甚至是亏损的状态,为-25729.36。 这启示公司应该重点在 California 和 New York 发展业务,加入投入,获得更多 的利润,并且考虑如何在 Texas 等处于亏损状态的州及时止损。



Area: profits of all categories

(5)我们前面已经知道 California 和 New York 蓝色最深,说明这两个州的利润 最高,分别为 76381.39 和 74038.55,那么应该重点应该在这两个州发展什么业 务呢?通过交互可以看到,在 California 州,各类产品的利润都比较高,其中 Accessories 产品利润最高,而销量最高的还是 Phones 商品。在 New York 州,利 润最高的是 Machines 产品,销量最高的也是 Phones 产品。说明应该重点发展这

些产品。



Categories: sales and profit in New York

Area: profits of all categories



(6)我们前面已经知道 Texas 州的红色最深,说明这个州的利润最低,甚至是亏损的状态,为-25729.36。通过交互可以看到,在这个州,各类产品的利润普遍都很低,甚至都是亏损的,但是 Phones 的利润和销量都很高,所以应该考虑放弃其他亏损的产品线,重点发展 Phones。



(7)除了上面对 US 市场的分析,我们还可以通过折线图分析全球市场的利润。 从图中可以看出,APAC 市场和 EU 市场的利润都是高的,且从 2011-2014 年 一直处于增长阶段,所以应该重点关注这两个区域的市场。而 Canada 市场的利 润处于最低,且从 2011-2014 年没有明显的增长,所以应该考虑对市场进行调 研,进行产品升级,从而获得更多的利润。



Market: Profit by years

三. 实验结果

此次实验实现了 html 界面框架的绘制、以销售量为横坐标、利润映射为条形 的颜色的柱状图的绘制,以及由颜色表示各州利润的等值区域地图及其图例的绘 制,并在这两个视图之间成功建立了交互。此外,还以年份为横坐标,绘制了表 明各类商品全球销量随时间变化的折线图及其图例。最后,根据可视化的结果对 数据进行了七项分析。

四. 实验心得

由于之前没有接触过 D3,因此在实验时遇到了很多麻烦,也有很多不懂的 地方,但在完成从(a)-(e)几个练习的过程中,遇到不会的地方通过去看老师提供 的课件、视频和代码,看课本和谷歌搜索资料学习,与同学进行讨论交流,这样 就逐渐学会了可视化的很多知识。实验的过程还是比较困难的,但是当视图成功 做出来的时候特别有成就感的也有一些新的体悟。

如何美观而又一目了然地展示这些结果,并能清晰地实现课题项目的目标, 以及完整、直观、生动地呈现产品要表达的"故事",这是可视化和交互设计的 重点内容。回忆以前学习和制作的统计图,都是最基本的只有 X 轴、Y 轴的单 因素变量数据的静态图。现在通过对可视化课程的学习发现大数据之美就在于可 以选择多种维度、多种角度去发现数据变化规律,并通过比较分析,又能得到很 多不同的结果。

按照我的理解数据可视化,常用统计图来展现,包括折线图、柱状图、饼图、 扇形图、散点图、雷达图、统计地图、仪表盘、漏斗图、字符云等,这些类型经 过不同的定义和交互设计,能够衍生出多种统计图的表达形式。但是想要熟练应 用和衍生出统计图,需要以清晰掌握各类型统计图的含义和作用为前提,才能不 仅满足需求,更能满足审美、及行业或学科的特性。

最后,学期即将结束,特别幸运也很感谢能跟着周虹老师学习可视化这么课程,第一次上课就被老师可爱到了哈哈,所以这学期上课也很愉快,希望老师继续保持开开心心的状态~