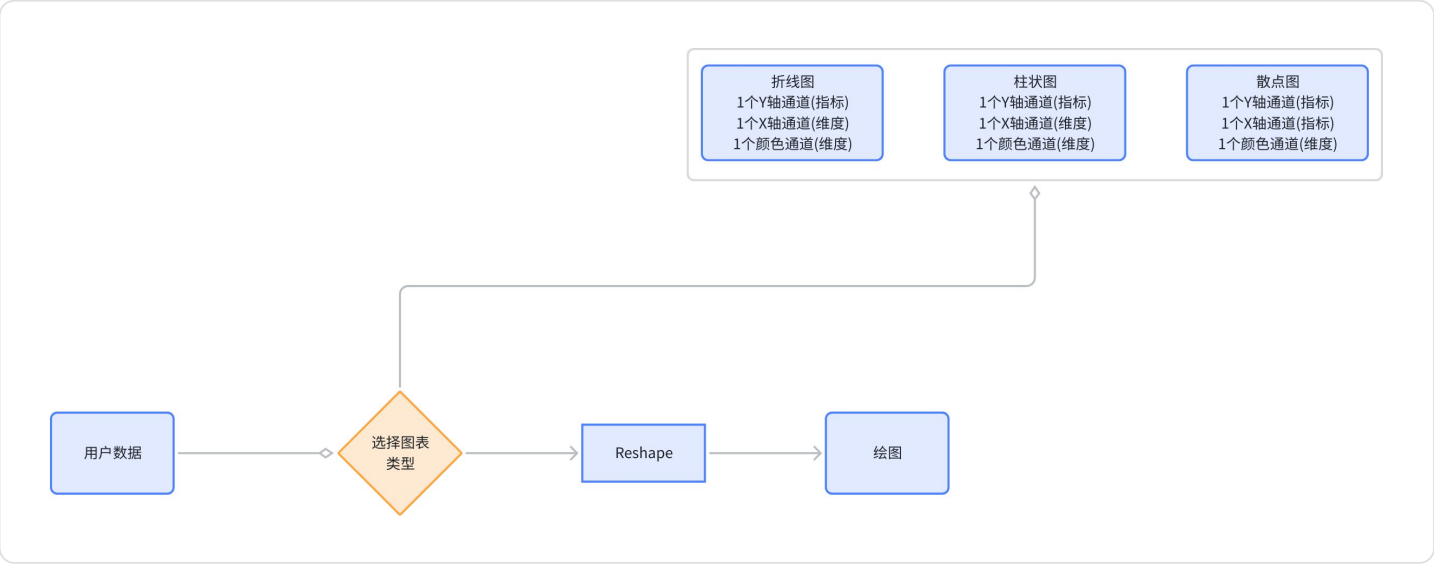


# 图表空间映射设计指南

 按图表类型，设计映射规则，从而突出不同图表的空间维度特性。增强指标的消费作用



## 图表类型

### 表格

图表类型	指标通道	行列维度通道	指标数量	维度数量
表格	$\infty$	$\infty$	$[0, \infty]$	$[0, \infty]$
透视表	$\infty$	$\infty$	$[0, \infty]$	$[0, \infty]$

### 笛卡尔坐标系

图表类型	指标通道	维度通道	颜色通道	指标数量	维度数量
折线图	1	1	1	$[1, \infty]$	$[0, \infty]$
堆叠柱状图	1	1	1	$[1, \infty]$	$[0, \infty]$
分组柱状图	1	1	1	$[1, \infty]$	$[0, \infty]$
百分比柱状图	1	1	1	$[1, \infty]$	$[0, \infty]$

堆叠条形图	1	1	1	$[1, \infty]$	$[0, \infty]$
分组条形图	1	1	1	$[1, \infty]$	$[0, \infty]$
百分比条形图	1	1	1	$[1, \infty]$	$[0, \infty]$
堆叠面积图	1	1	1	$[1, \infty]$	$[0, \infty]$
百分比面积图	1	1	1	$[1, \infty]$	$[0, \infty]$
圆视图	1	1	1	$[1, \infty]$	$[0, \infty]$
散点图	2	0	1	$[2, \infty]$	$[0, \infty]$
双轴图	2	1	1	$[1, \infty]$	$[0, \infty]$

## 极坐标系

图表类型	指标通道	维度通道	颜色通道	指标数量	维度数量
饼图	1	1	1	$[1, \infty]$	$[0, \infty]$
环形图	1	1	1	$[1, \infty]$	$[0, \infty]$
玫瑰图	1	1	1	$[1, \infty]$	$[0, \infty]$
雷达图	1	1	1	$[1, \infty]$	$[0, \infty]$

## 层级关系

图表类型	指标通道	维度通道	颜色通道	指标数量	维度数量
旭日图	1	1	0	$[1, \infty]$	$[0, \infty]$
嵌套圆图	1	1	0	$[1, \infty]$	$[0, \infty]$
矩形树图	1	1	0	$[1, \infty]$	$[0, \infty]$
漏斗图	1	1	0	$[1, \infty]$	$[0, \infty]$
桑基图	1	3	0	$[1, \infty]$	$[0, \infty]$

## 其它

图表类型	指标通道	维度通道	颜色通道	指标数量	维度数量
------	------	------	------	------	------

水波图	1	0	0	[1, ∞]	[0, ∞]
环形进度图	1	0	0	[1, ∞]	[0, ∞]
条形进度图	1	0	0	[1, ∞]	[0, ∞]
词云	1	1	1	[0, ∞]	[0, ∞]
地图	1	1	1	[1, ∞]	[0, ∞]

# 维度重塑

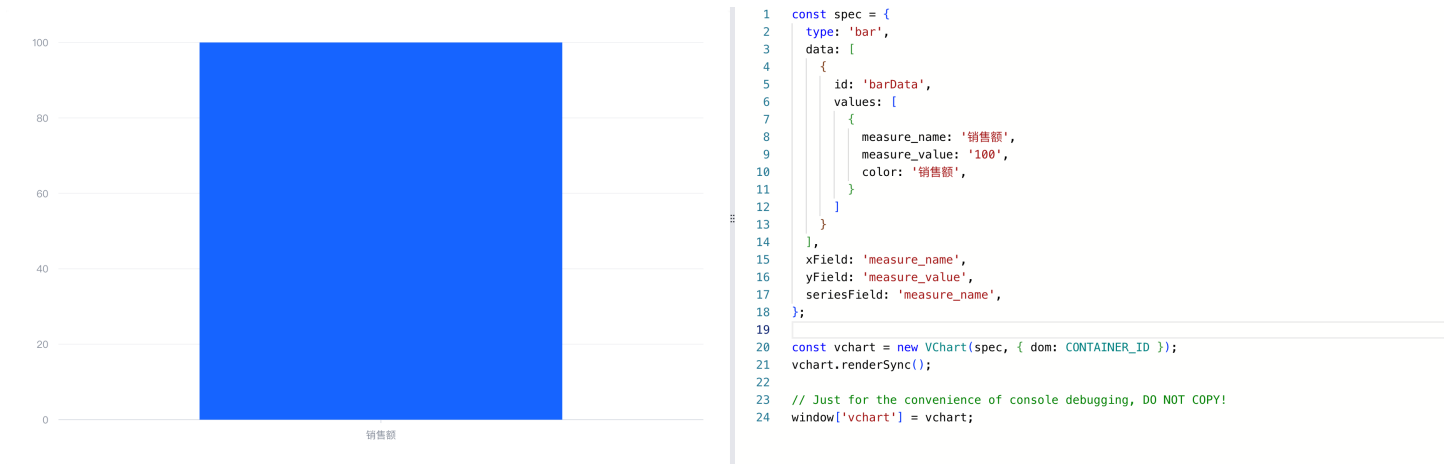
柱状图举例，所有的数据转换结构都是 2 个维度 1 个指标，从而任意字段数据（≥1 个指标），都能绘制出柱状图

## 单指标重塑



### 代码块

```
1  const input = [  
2    { 销售额: 100 }  
3  ]  
4  
5  const result = [  
6    {  
7      measure_name: '销售额',  
8      measure_value: '100',  
9      color: '销售额',  
10   }  
11  ]
```



## 多指标重塑



### 代码块

```
1  const input = [
2    { 销售额: 100, 利润: 30, 成本: 70 }
3  ]
4
5  const result = [
6    {
7      measure_name: '销售额',
8      measure_value: '100',
9      color: '销售额',
10     },
11    {
12      measure_name: '利润',
13      measure_value: '30',
14      color: '利润',
15     },
16    {
17      measure_name: '成本',
18      measure_value: '70',
19      color: '成本',
20     }
21  ]
```



## 单维度单指标



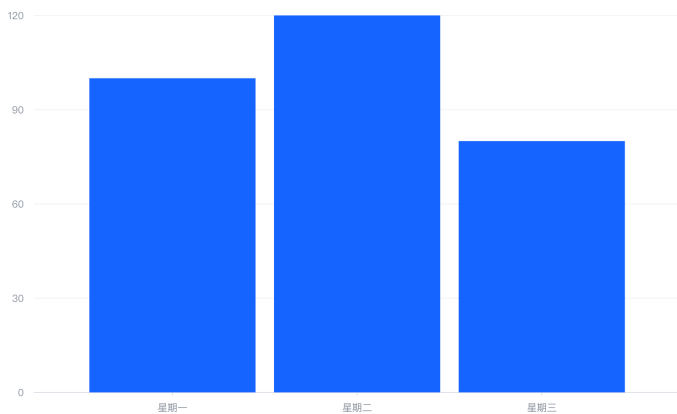
代码块

```
1  const input = [
2    { 销售额: 100, 日期: '星期一' },
3    { 销售额: 120, 日期: '星期二' },
4    { 销售额: 80, 日期: '星期三' },
5  ]
6
7  const result = [
8    {
9      "日期": '星期一',
10     measure_value: '100',
11     color: '销售额',
12   },
```

```

13     {
14         "日期": '星期二',
15         measure_value: '120',
16         color: '销售额',
17     },
18     {
19         "日期": '星期三',
20         measure_value: '80',
21         color: '销售额',
22     }
23 ]

```



```

6     values: [
7     {
8         "日期": '星期一',
9         measure_value: '100',
10        color: '销售额',
11    },
12    {
13        "日期": '星期二',
14        measure_value: '120',
15        color: '销售额',
16    },
17    {
18        "日期": '星期三',
19        measure_value: '80',
20        color: '销售额',
21    }
22    ],
23    },
24    ],
25    xField: '日期',
26    yField: 'measure_value',
27    seriesField: 'color',
28    };
29
30 const vchart = new VChart(spec, { dom: CONTAINER_ID });
31 vchart.renderSync();

```

## 单维度多指标

销售额	利润	日期
100	30	星期一
120	90	星期二
80	50	星期三



日期	颜色	指标值
星期一	销售额	100
星期二	销售额	120
星期三	销售额	130
星期一	利润	30
星期二	利润	90
星期三	利润	50

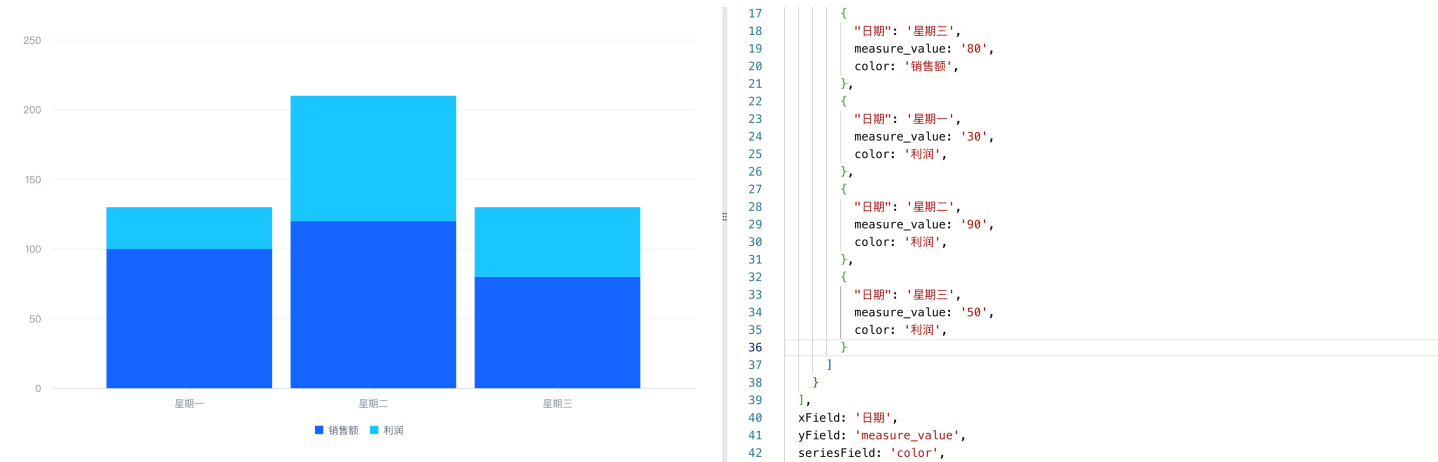
### 代码块

```

1  const input = [
2      { 销售额: 100, 日期: '星期一', 利润: 30 },

```

```
3      { 销售额: 120, 日期: '星期二', 利润: 90 },
4      { 销售额: 80, 日期: '星期三', 利润: 50 },
5  ]
6
7  const result = [
8      {
9          "日期": '星期一',
10         measure_value: '100',
11         color: '销售额',
12     },
13     {
14         "日期": '星期二',
15         measure_value: '120',
16         color: '销售额',
17     },
18     {
19         "日期": '星期三',
20         measure_value: '80',
21         color: '销售额',
22     },
23     {
24         "日期": '星期一',
25         measure_value: '30',
26         color: '利润',
27     },
28     {
29         "日期": '星期二',
30         measure_value: '90',
31         color: '利润',
32     },
33     {
34         "日期": '星期三',
35         measure_value: '50',
36         color: '利润',
37     }
38 ]
```



## 多维度单指标



```
代码块

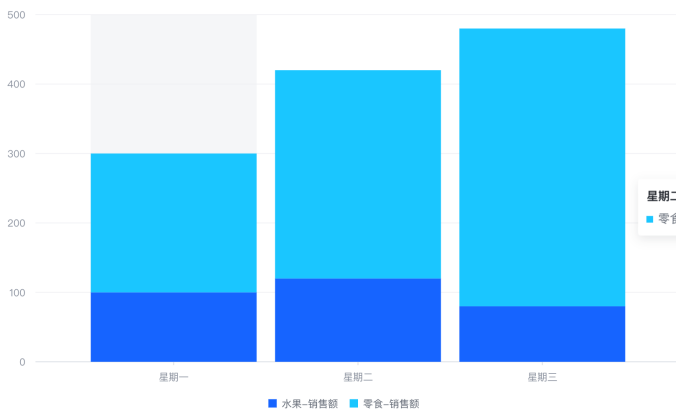
1  const input = [
2    { 销售额: 100, 日期: '星期一', 类型: '水果' },
3    { 销售额: 200, 日期: '星期一', 类型: '零食' },
4    { 销售额: 120, 日期: '星期二', 类型: '水果' },
5    { 销售额: 300, 日期: '星期二', 类型: '零食' },
6    { 销售额: 80, 日期: '星期三', 类型: '水果' },
7    { 销售额: 400, 日期: '星期三', 类型: '零食' },
8  ]
9
10 const result = [
11   {
12     "日期": '星期一',
13     measure_value: '100',
14     color: '水果-销售额',
15   },
16   {
17     "日期": '星期二',
18     measure_value: '120',
19     color: '水果-销售额',
20   },
21   {
```



```

22     "日期": '星期三',
23     measure_value: '80',
24     color: '水果-销售额',
25 },
26 {
27     "日期": '星期一',
28     measure_value: '200',
29     color: '零食-销售额',
30 },
31 {
32     "日期": '星期二',
33     measure_value: '300',
34     color: '零食-销售额',
35 },
36 {
37     "日期": '星期三',
38     measure_value: '400',
39     color: '零食-销售额',
40 }
41 ]

```



```

27     {
28         "日期": '星期二',
29         measure_value: '300',
30         color: '零食-销售额',
31     },
32     {
33         "日期": '星期三',
34         measure_value: '400',
35         color: '零食-销售额',
36     }
37 ],
38 xField: '日期',
39 yField: 'measure_value',
40 seriesField: 'color',
41 legends: {
42     visible: true
43 },
44 };
45
46 const vchart = new VChart(spec, { dom: CONTAINER_ID });
47 vchart.renderSync();
48
49 // Just for the convenience of console debugging, DO NOT COPY!
50 window['vchart'] = vchart;

```

## 多维度多指标

销售额	利润	日期	类型
100	70	星期一	水果
200	50	星期一	零食
120	90	星期二	水果
300	50	星期二	零食
80	50	星期三	水果
400	50	星期三	零食

升维

日期	类型	指标名称	指标值
星期一	水果	销售额	100
星期一	零食	销售额	200
星期二	水果	销售额	120
星期二	零食	销售额	300
星期三	水果	销售额	80
星期三	零食	销售额	400
星期一	水果	利润	70
星期一	零食	利润	50
星期二	水果	利润	90
星期二	零食	利润	50
星期三	水果	利润	50
星期三	零食	利润	50

降维

日期	水果-销售额	零食-销售额	水果-利润	零食-利润
星期一	100	200	70	50
星期二	120	300	90	50
星期三	80	400	50	50

升维

日期	颜色	指标值
星期一	水果-销售额	100
星期二	水果-销售额	120
星期三	水果-销售额	80
星期一	零食-销售额	200
星期二	零食-销售额	300
星期三	零食-销售额	400
星期一	水果-利润	70
星期二	水果-利润	90
星期三	水果-利润	50
星期一	零食-利润	50
星期二	零食-利润	50
星期三	零食-利润	50

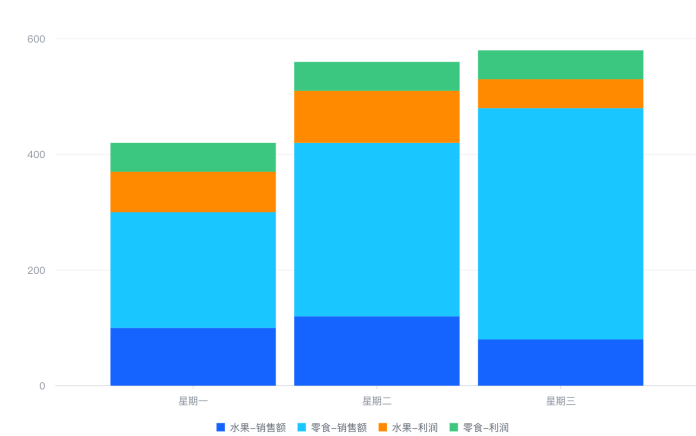
## 代码块

```

1  const input = [
2    { 销售额: 100, 利润: 70, 日期: '星期一', 类型: '水果' },
3    { 销售额: 200, 利润: 50, 日期: '星期一', 类型: '零食' },
4    { 销售额: 120, 利润: 90, 日期: '星期二', 类型: '水果' },
5    { 销售额: 300, 利润: 50, 日期: '星期二', 类型: '零食' },
6    { 销售额: 80, 利润: 50, 日期: '星期三', 类型: '水果' },
7    { 销售额: 400, 利润: 50, 日期: '星期三', 类型: '零食' },
8  ]
9
10 const result = [
11   {
12     "日期": '星期一',
13     measure_value: '100',
14     color: '水果-销售额',
15   },
16   {
17     "日期": '星期二',
18     measure_value: '120',
19     color: '水果-销售额',
20   },
21   {
22     "日期": '星期三',
23     measure_value: '80',
24     color: '水果-销售额',
25   },
26   {
27     "日期": '星期一',
28     measure_value: '200',
29     color: '零食-销售额',

```

```
30     },
31     {
32         "日期": '星期二',
33         measure_value: '300',
34         color: '零食-销售额',
35     },
36     {
37         "日期": '星期三',
38         measure_value: '400',
39         color: '零食-销售额',
40     },
41     {
42         "日期": '星期一',
43         measure_value: '70',
44         color: '水果-利润',
45     },
46     {
47         "日期": '星期二',
48         measure_value: '90',
49         color: '水果-利润',
50     },
51     {
52         "日期": '星期三',
53         measure_value: '50',
54         color: '水果-利润',
55     },
56     {
57         "日期": '星期一',
58         measure_value: '50',
59         color: '零食-利润',
60     },
61     {
62         "日期": '星期二',
63         measure_value: '50',
64         color: '零食-利润',
65     },
66     {
67         "日期": '星期三',
68         measure_value: '50',
69         color: '零食-利润',
70     }
71 ]
```



```

51   },
52   {
53     "日期": '星期一',
54     measure_value: '50',
55     color: '零食-利润',
56   },
57   {
58     "日期": '星期二',
59     measure_value: '50',
60     color: '零食-利润',
61   },
62   {
63     "日期": '星期三',
64     measure_value: '50',
65     color: '零食-利润',
66   },
67   ]
68 }
69 },
70 xField: '日期',
71 yField: 'measure_value',
72 seriesField: 'color',
73 legends: {
74   visible: true
75 }
76 };

```

# 规律总结

维度数量	指标数量
0	1
0	任意
1	1
1	任意
任意	1
任意	任意

经过维度重  
塑

维度数量	指标数量
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$
$[0, \infty]$	$[1, \text{unique (维值笛卡尔积) }]$

1. 维度归零：降维，可以将所有维度转指标，从而维度可以归 0
2. 指标归一：升维，所有的指标可以合并为一个指标
3. 维度无穷多：升维一次多一个维度，因此维度可以无穷多
4. 指标最大值：降维可以把所有的维值笛卡尔积的去重结果转指标，因此指标最多取决于维值笛卡尔积结果

# 示例

只要确定字段的映射位置，施加规范，即可保证任意有效字段可以转换为图表所需的结构。

图表类型	图表通道	数据维度数量	数据指标数量	升维次数	降维次数

折线图	1 个指标 (Y)	0	大于 1	2	0
	2 个维度 (X+Color)	1	大于 1	1	0
		2	大于 1	2	1
		任意	大于 1	2	1

## 图表空间映射设计



目标：让任意数据可以一键绘制任意图表类型；

每一个图表类型约定数据映射结构

任意数据均可通过维度重塑，转换为图表类型目标的映射结构

图表类型	图表通道	数据约定	图表约定
折线图	X 指标 Y 维度 Color 维度	至少 1 个指标	多维度：除第一个维度，其余维度标准重塑（升维-降维-升维） 单维度：升维 1 次 无维度：升维 2 次  第一个维度映射到 X 轴 第二个维度映射到颜色
散点图	X 指标 Y 指标 Color 维度	至少 1 个指标	多维度：所有维度标准重塑（升维-降维-升维） 单维度：升维 1 次 无维度：升维 2 次  第一个指标用于 X 轴 第二个及其之后到指标，均用于 Y 轴
双轴图	LY 指标 RY 指标 Color 维度	至少 1 个指标	多维度：所有维度标准重塑（升维-降维-升维） 单维度：升维 1 次 无维度：升维 2 次  第一个指标用于左轴 第二个及其之后到指标，均用于右轴

