

D3.JS

互動式資料視覺化

Lecturer: LinJer 林哲

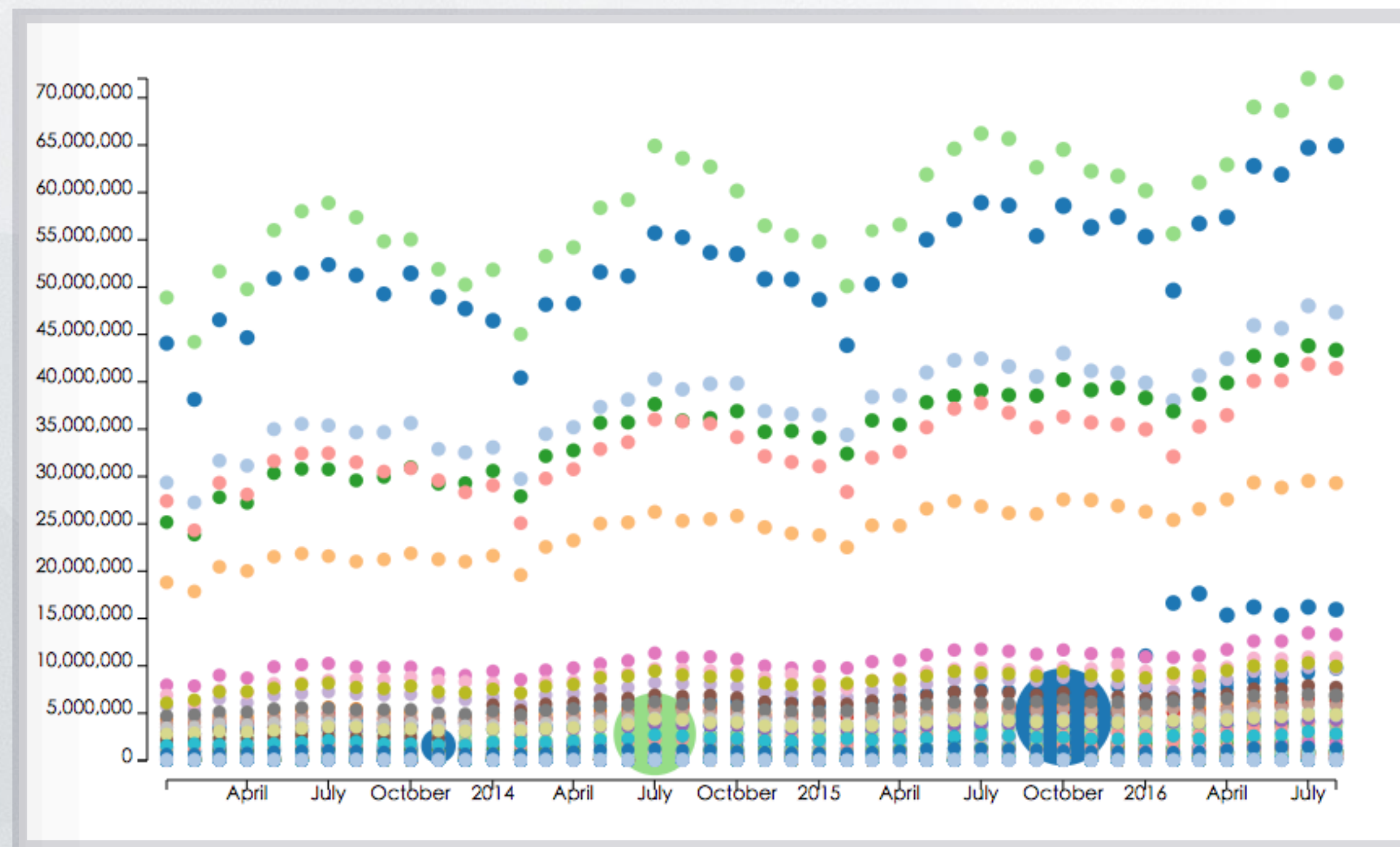
evin92@gmail.com



該換真實資料上場了

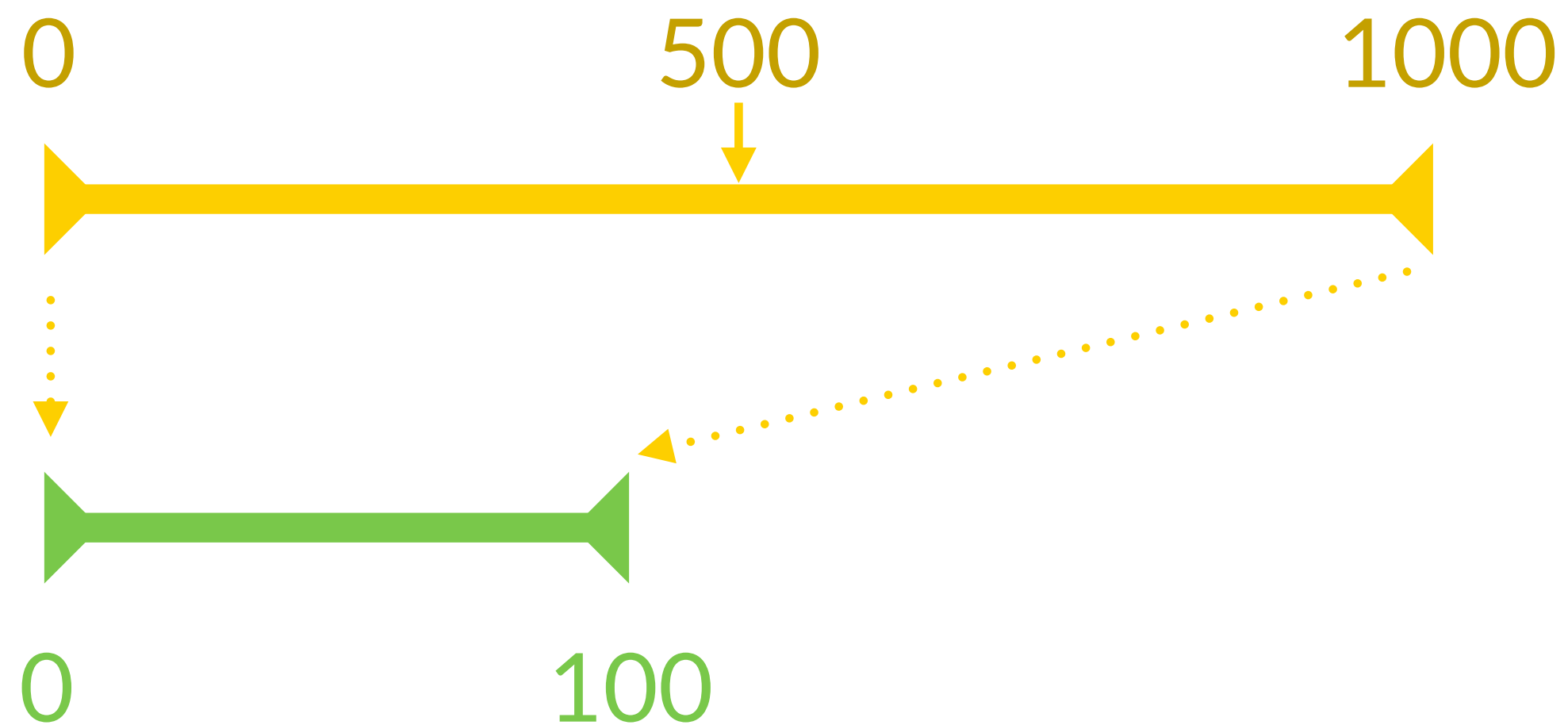
發票發行數量與日期
2D散佈圖

先了解: 比例尺 & 軸線



Scale-比例尺

拿來做範圍變換



線性：d3.scale.linear()

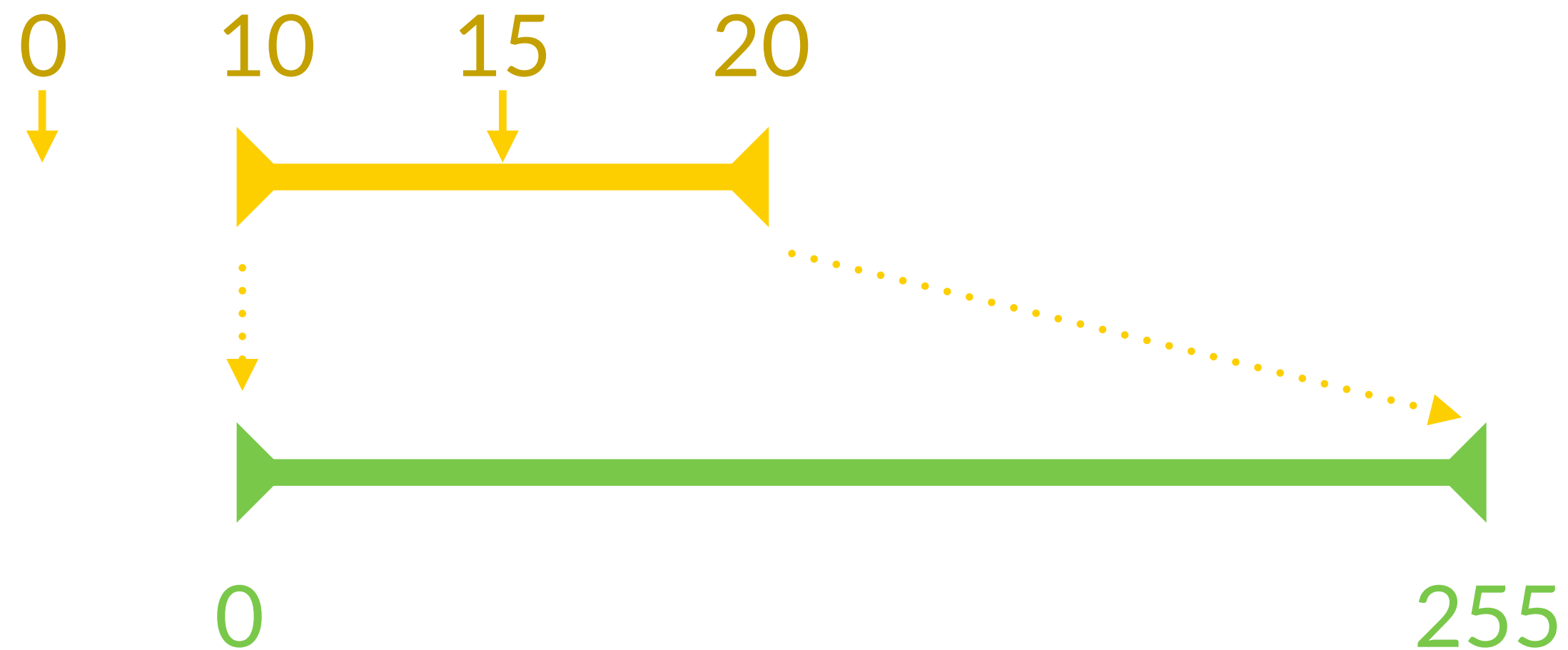
```
var xScale = d3.scale.linear()  
  .domain([0, 1000])  
  .range([0, 100]);
```

D: 輸入
R: 輸出

在console中，用xScale(455)試試看

Scale-比例尺

拿來做範圍變換



線性：d3.scale.linear()

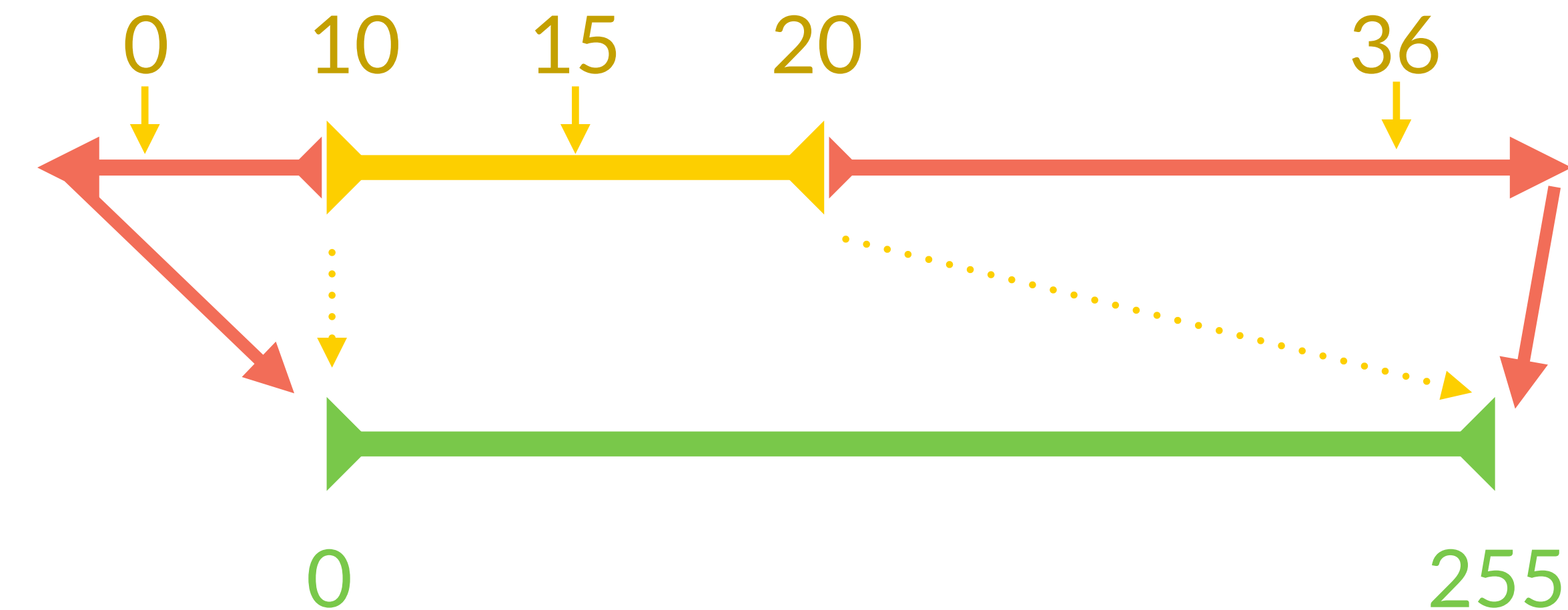
```
var xScale = d3.scale.linear()  
  .domain([10, 20])  
  .range([0, 255]);
```

D: 輸入
R: 輸出

在console中，用xScale(30)試試看

Scale-比例尺

clamp(true) 左右夾緊



線性：d3.scale.linear()

```
var xScale = d3.scale.linear()  
  .domain([10, 20])  
  .range([0, 255])  
  .clamp(true);
```

D: 輸入
R: 輸出

在console中，用xScale(30)試試看

動眼時間

修改random(n,m)亂數函式
套用d3.scale.linear()

```
function random(n, m){  
    var rScale = d3.scale.linear()  
        .domain([??])  
        .rangeRound([??]);  
    return rScale(Math.random());  
}
```

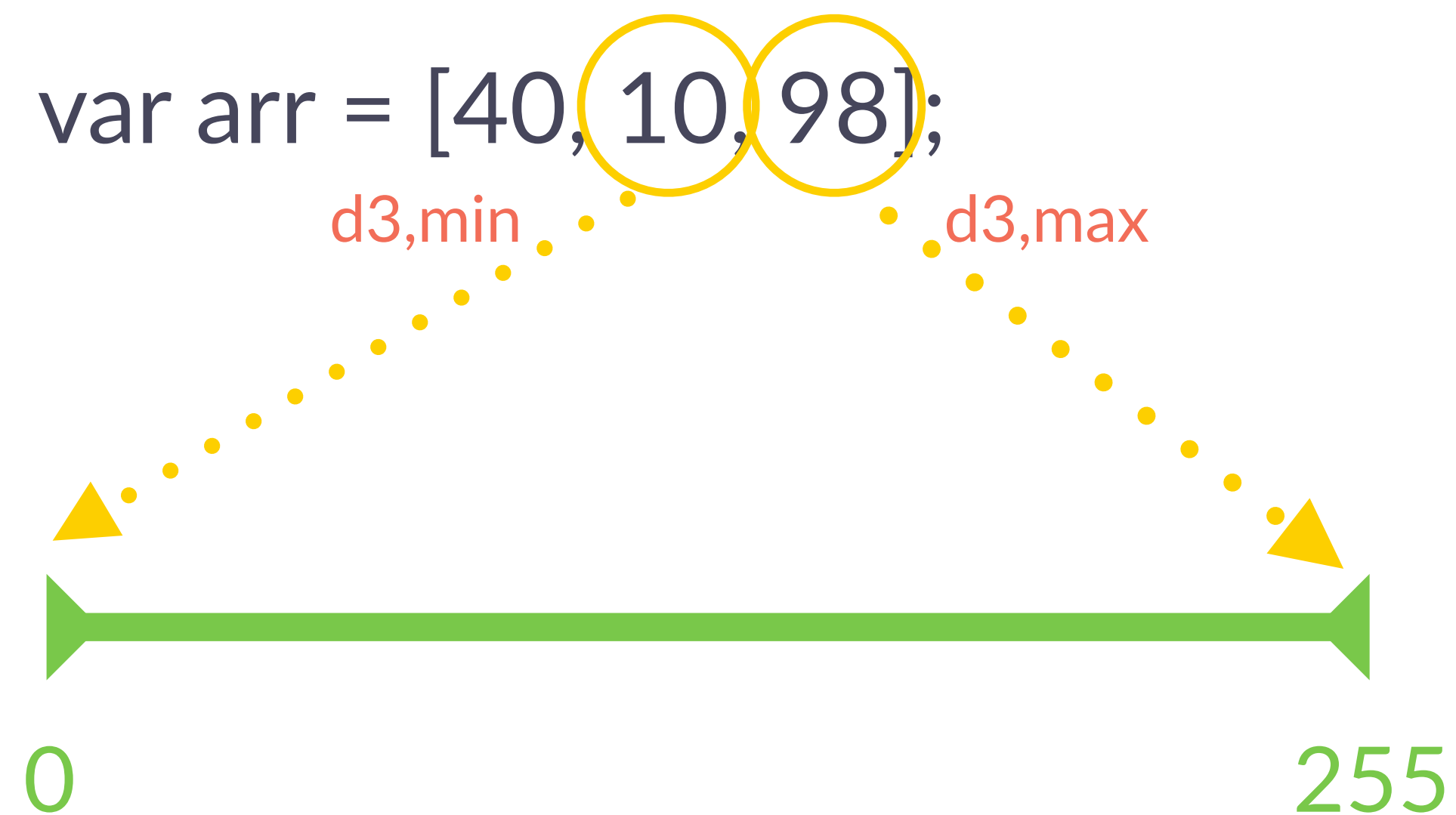

解答時間

```
function random(n, m){  
    var rScale = d3.scale.linear()  
        .domain([0, 1])  
        .rangeRound([n, m]);  
    return rScale(Math.random());  
}
```

如果資料量大，又不知其
範圍怎麼辦？

Scale-比例尺

拿好朋友來使用



`d3.min()`, `d3.max()`

```
var xScale = d3.scale.linear()  
  .domain([d3.min(arr), d3.max(arr)])  
  .range([0, 255]);
```


比例尺的好朋友

```
var arr = [40, 10, 98];
var dataSet = [
  {name: "Kevin", tall: 180, age: 32},
  {name: "Helen", tall: 166, age: 15},
  {name: "Gary", tall: 172, age: 27}
];
```

d3.max, d3.min

狀態	寫法	結果
得到最大值 max	d3.max(arr)	?
	d3.max(dataSet, function(d){ return d.tall; })	?
得到最小值 min	d3.min(arr)	?
	d3.min(dataSet, function(d){ return d.age; })	?

比例尺的好朋友

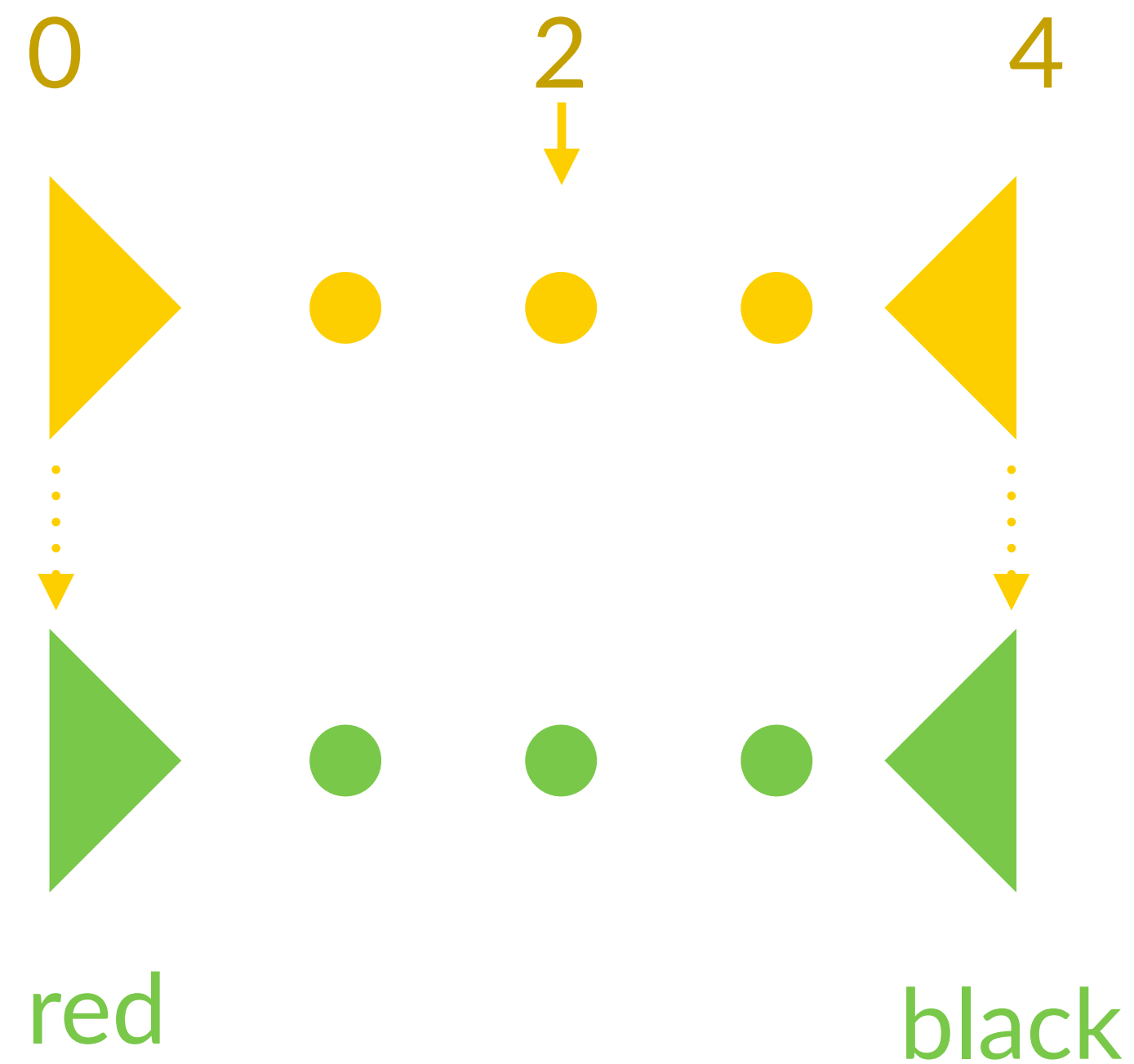
```
var arr = [40, 10, 98];
var dataSet = [
  {name: "Kevin", tall: 180, age: 32},
  {name: "Helen", tall: 166, age: 15},
  {name: "Gary", tall: 172, age: 27}
];
```

d3.max, d3.min

狀態	寫法	結果
得到最大值 max	d3.max(arr)	98
	d3.max(dataSet, function(d){ return d.tall; })	180
得到最小值 min	d3.min(arr)	10
	d3.min(dataSet, function(d){ return d.age; })	15

Scale-比例尺

拿來做範圍變換



序數：d3.scale.ordinal()

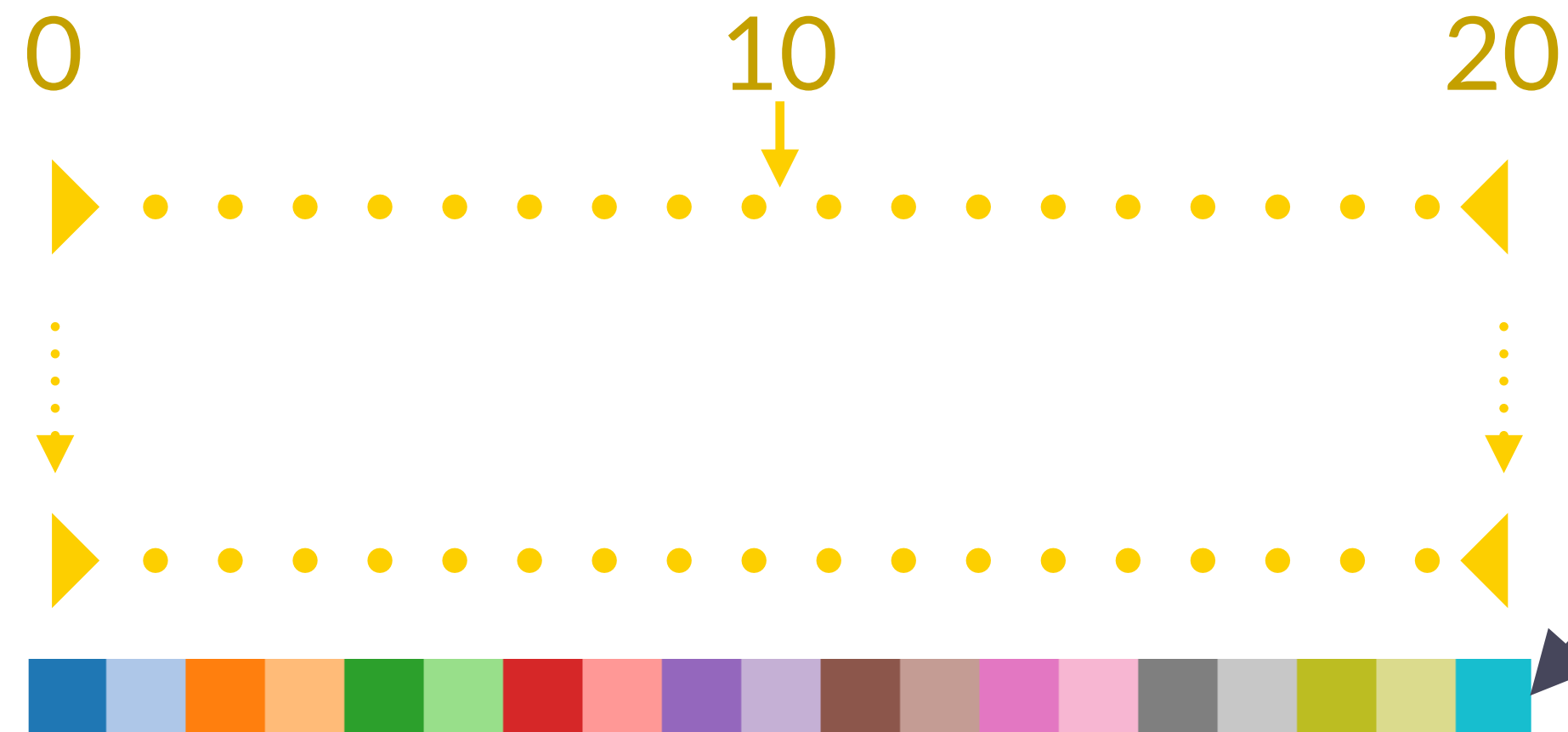
```
var index = [0, 1, 2, 3, 4];  
var color = ["red", "blue", "green", "yellow", "black"]
```

```
var xScale = d3.scale.ordinal()  
  .domain(index)  
  .range(color);
```

D: 輸入
R: 輸出

Scale-比例尺

拿來做範圍變換



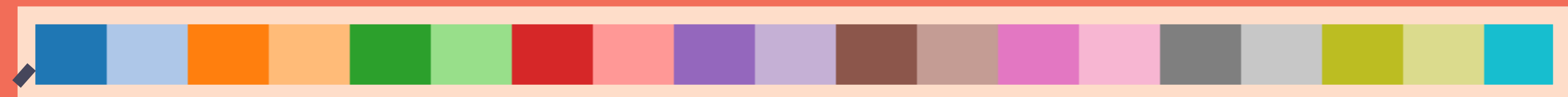
用console試試看

d3內建填色序數：
`d3.scale.category20()`

```
var fScale = d3.scale.category10();
```



```
d3.scale.category20();
```



```
d3.scale.category20b();
```

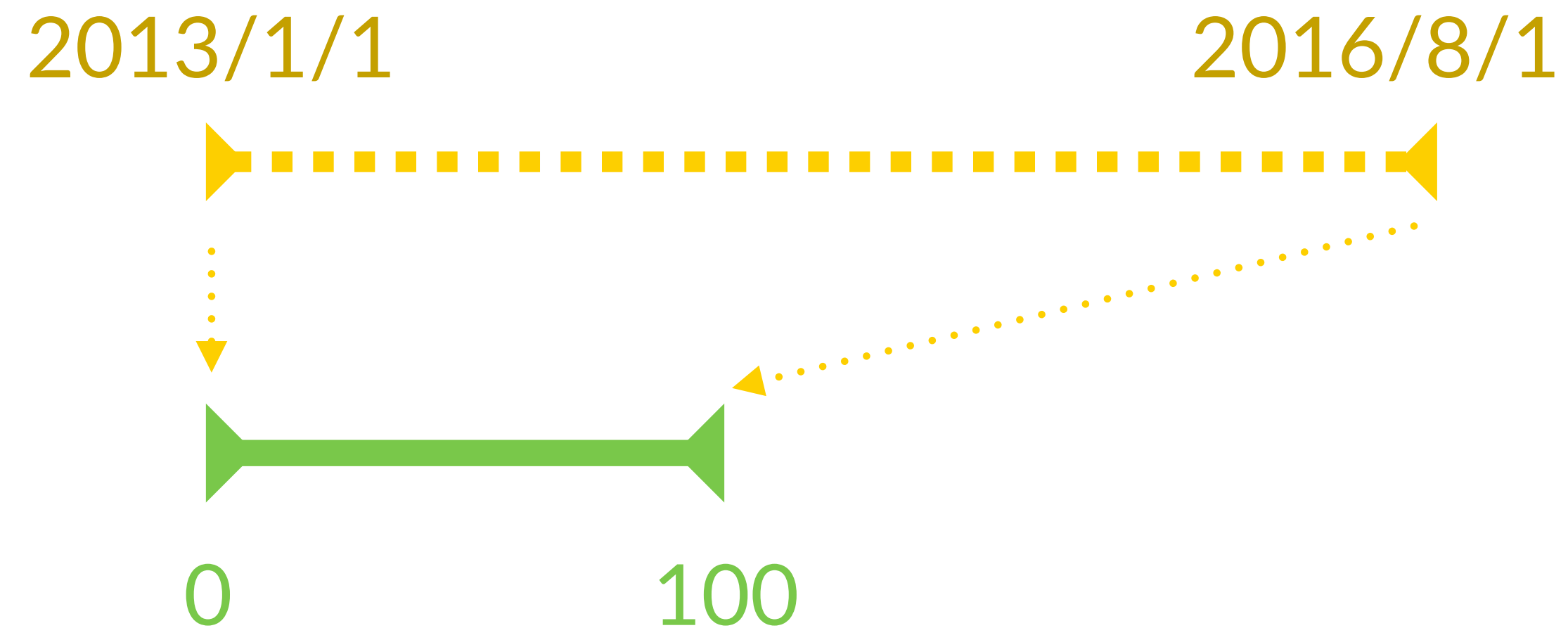


```
d3.scale.category20c();
```



Scale-比例尺

拿來做範圍變換



日期：d3.time.scale()

```
var xScale = d3.time.scale()  
    .domain([  
        new Date("2013-01-01"),  
        new Date("2016-08-01")  
    ])  
    .range(0, 100);
```

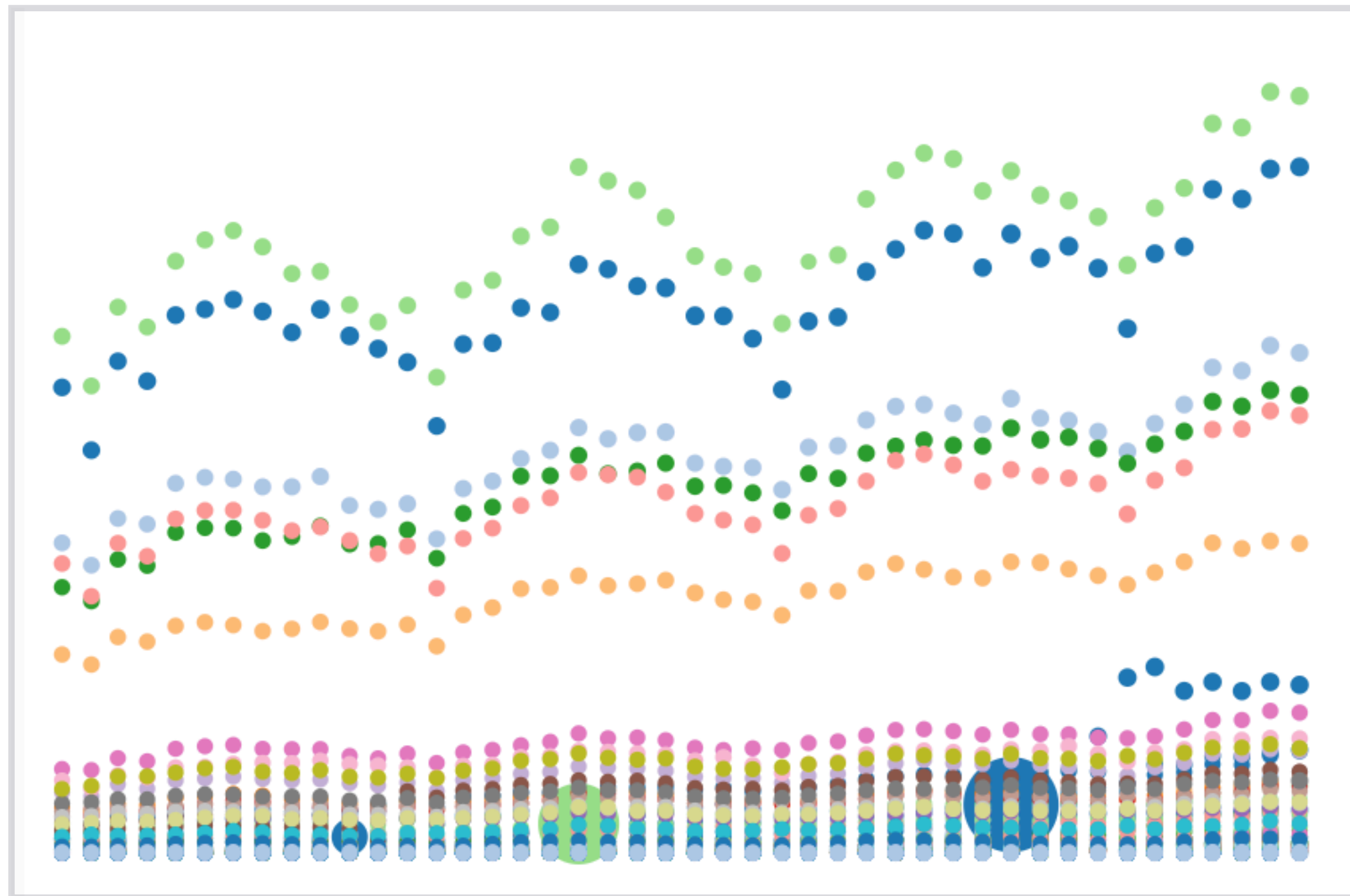
D: 輸入

R: 輸出

在console中，
用xScale(new Date("2014-10-30"))試試看



來畫畫2D散佈圖(無軸線版)

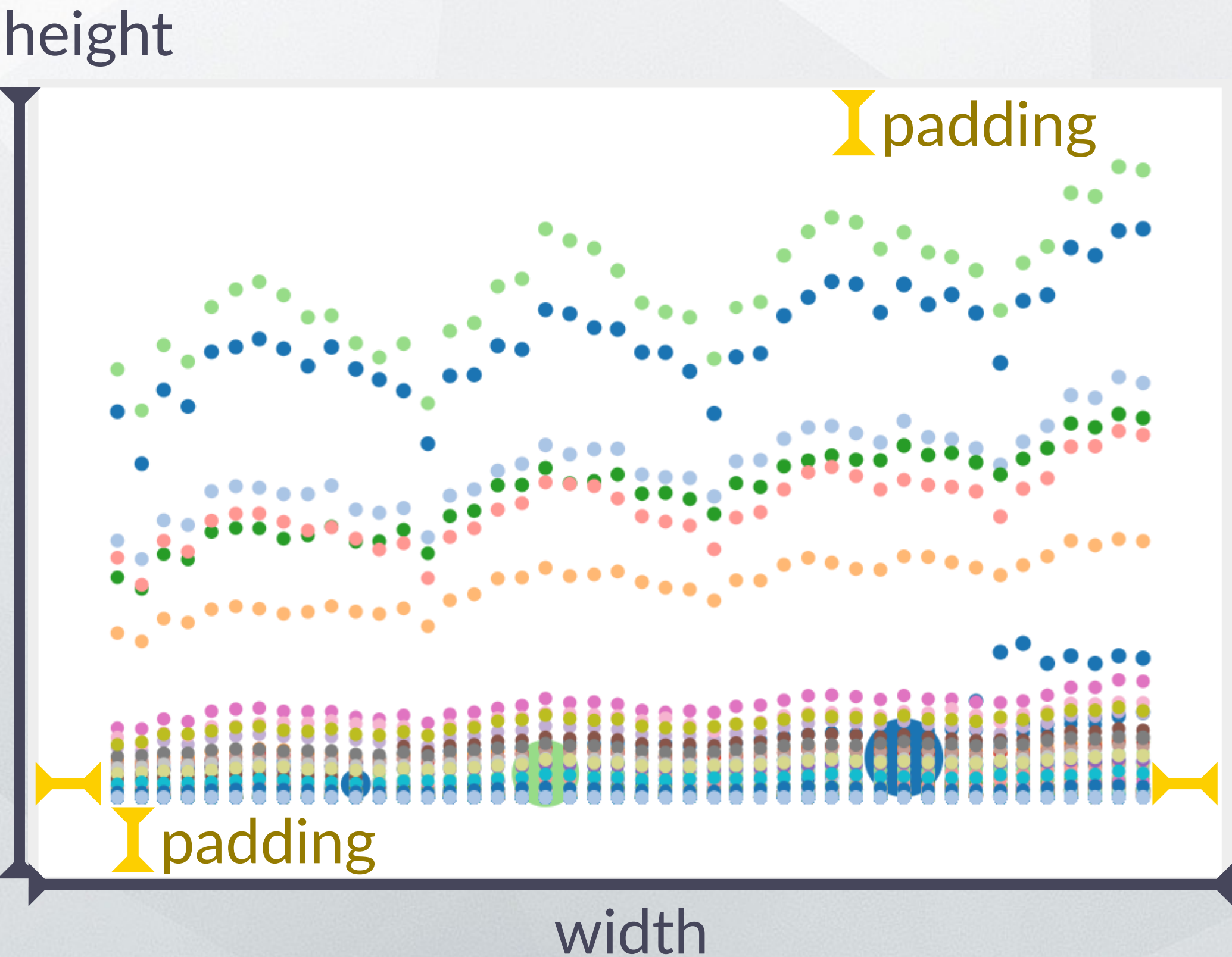


[按我下載範本](#)

建議步驟：

1. 定義width, height, padding, letterList變數
2. 建立svg()畫布環境
3. 用d3讀取csv
4. 建立bind()
5. 定義xScale, yScale, rScale, fScale比例尺(前三者range目的在決定在svg上位置)
6. 建立render()繪圖

定義xScale, yScale, rScale, fScale比例尺



比例尺們	類別	Domain(資料)	Range(svg上位置)
xScale	日期(date)	[日期起, 日期迄]	[padding, width-padding]
yScale	數量(number)	[0, 最大數量]	[height-padding, padding]
rScale	金額(amount)	[最小金額, 最大數量]	[5,30]
fScale	縣市(amount)	序號 ex: "C" -> 2	輸出為顏色，非位置