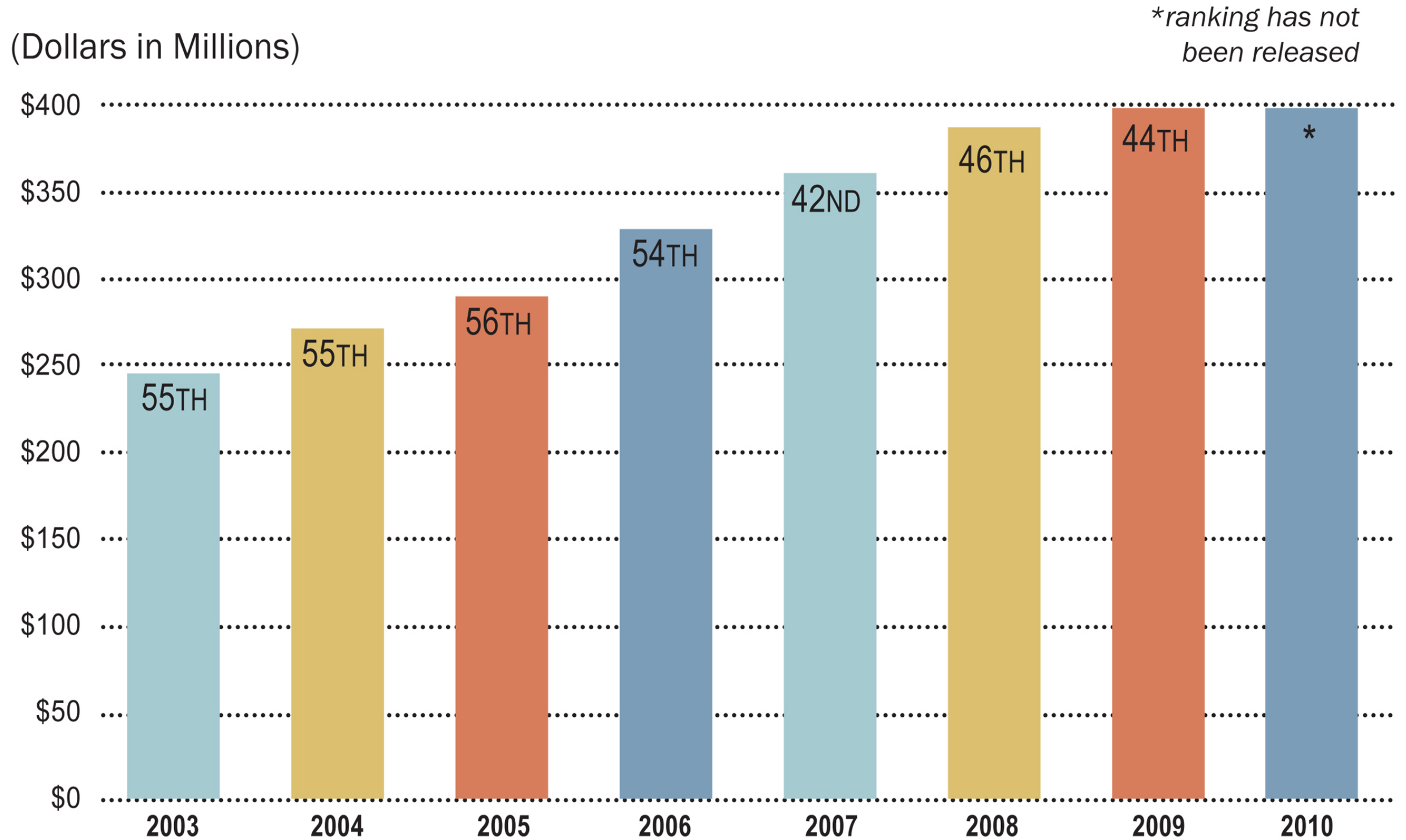


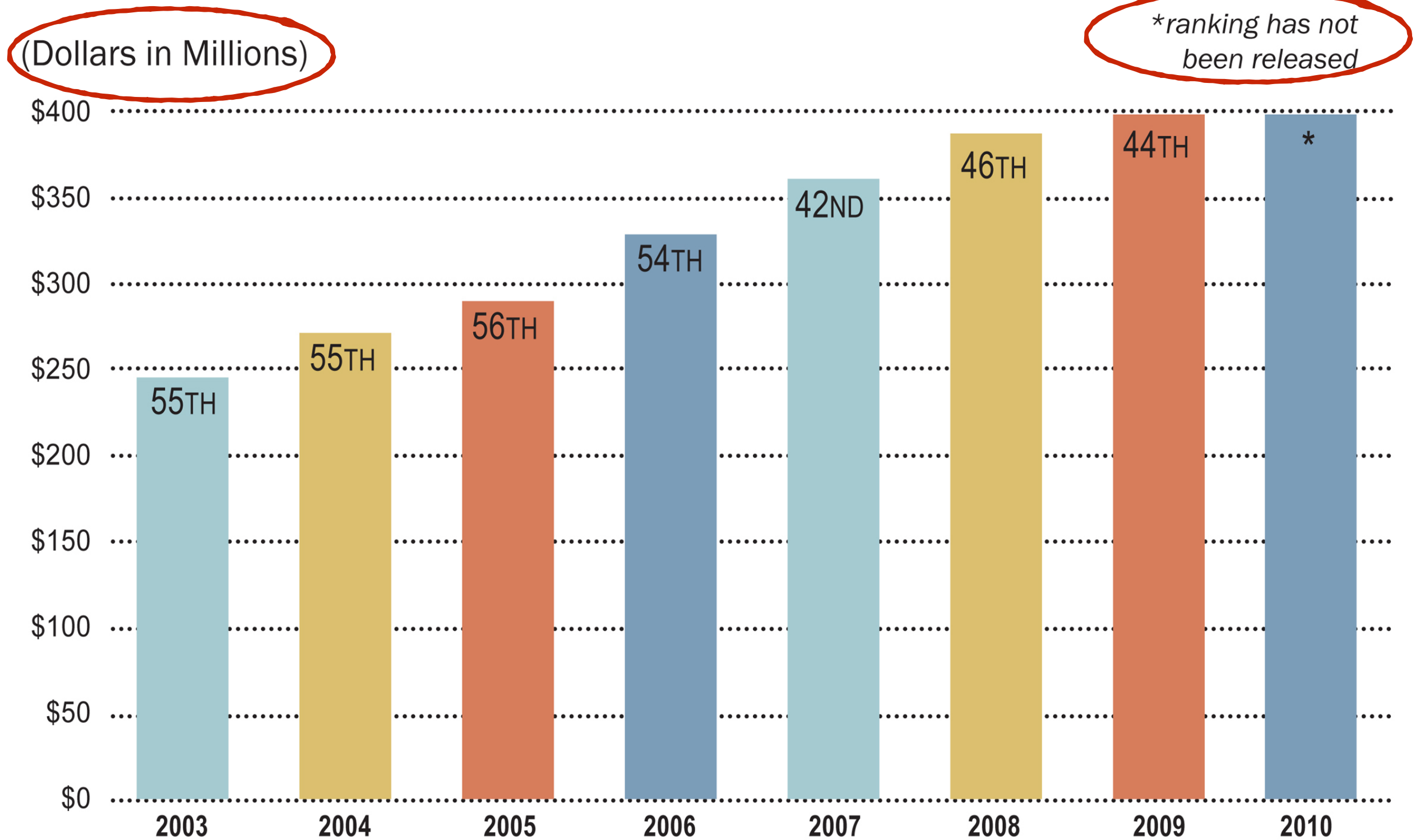
Data Driven Documents

```
01: d3.select('.container')
02:   .selectAll('div')
03:   .data(BIG_DATA)
04:   .enter()
05:   .append('div')
06:   .attr('class', 'bar')
07:   .style('width', function(d) {
08:     return d + 'px';
09:   })
10:   .text(function(d) {
11:     return d;
12:   })
13:   .exit();
```

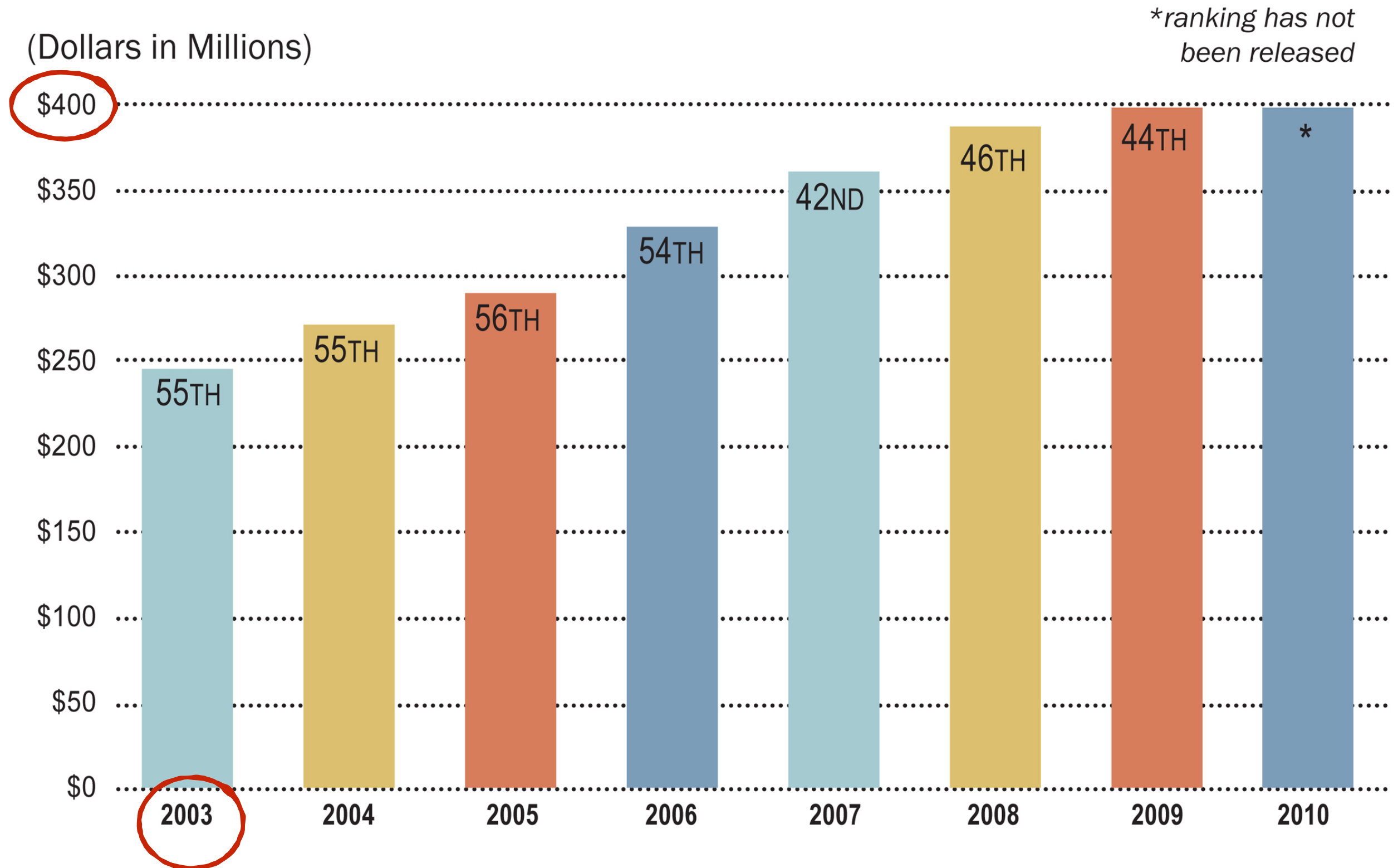
Barchart



Легенда



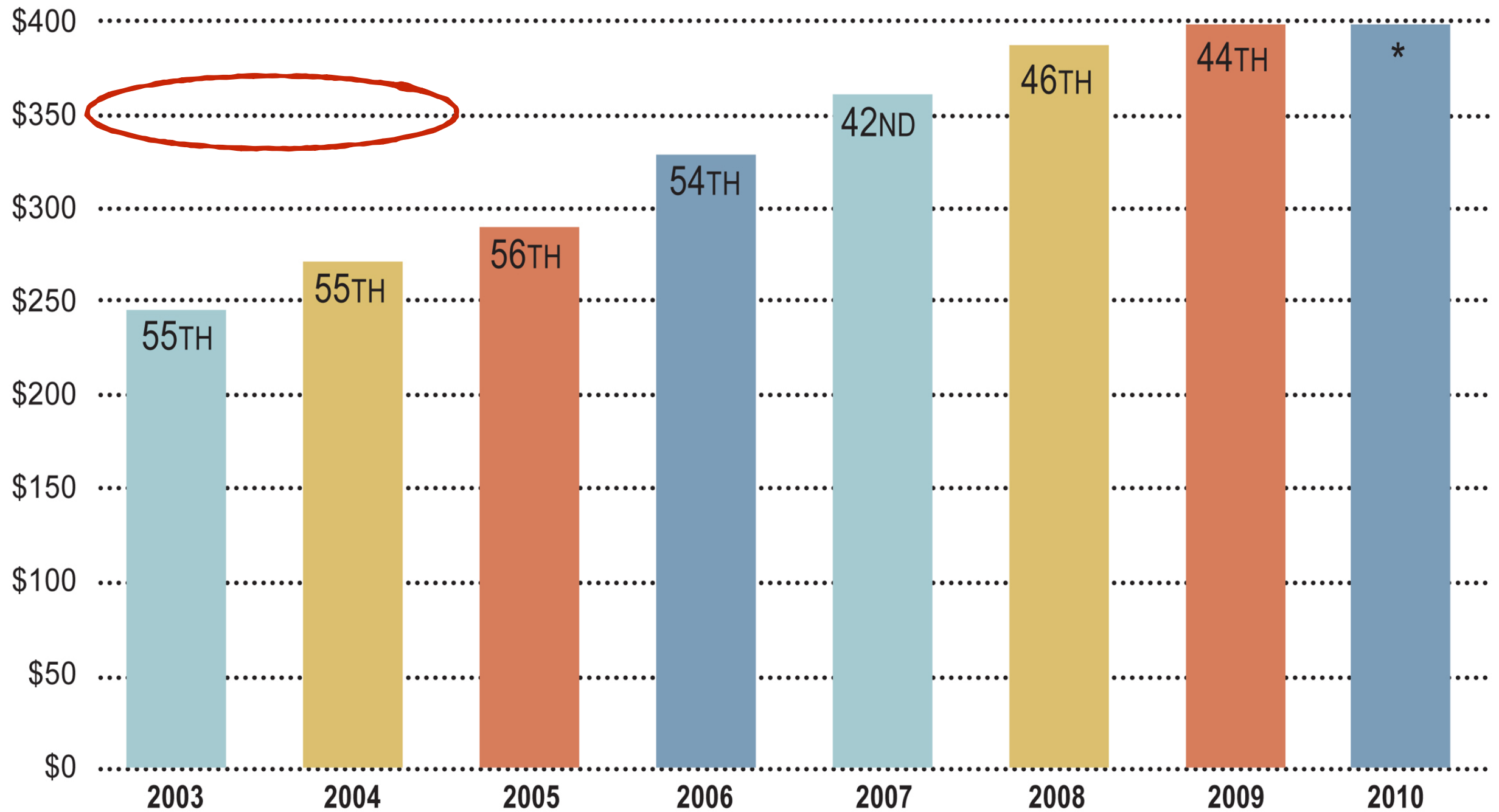
Отметки на шкалах



Разметка для удобства

(Dollars in Millions)

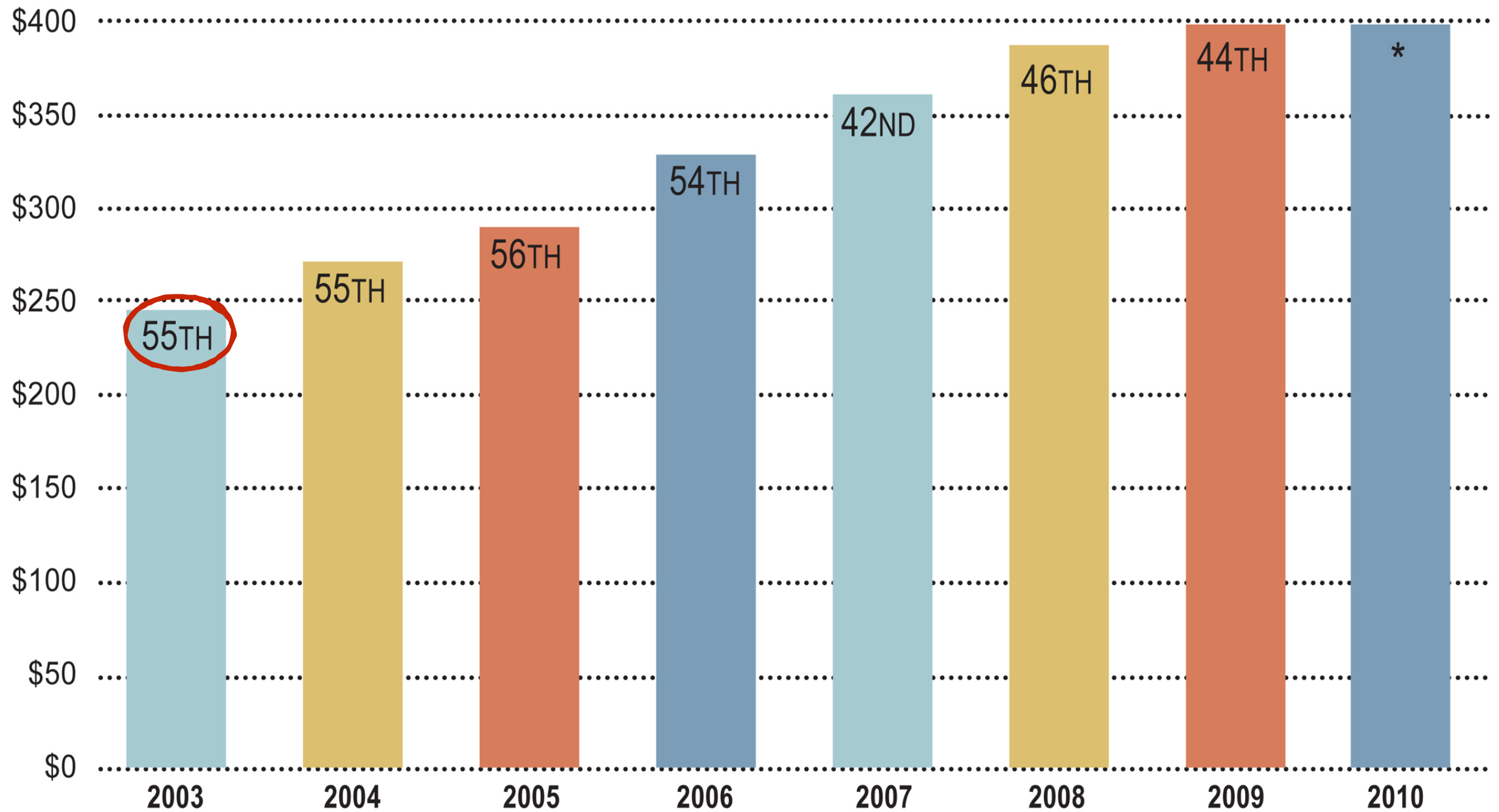
**ranking has not
been released*



Какие-то еще данные

(Dollars in Millions)

**ranking has not
been released*

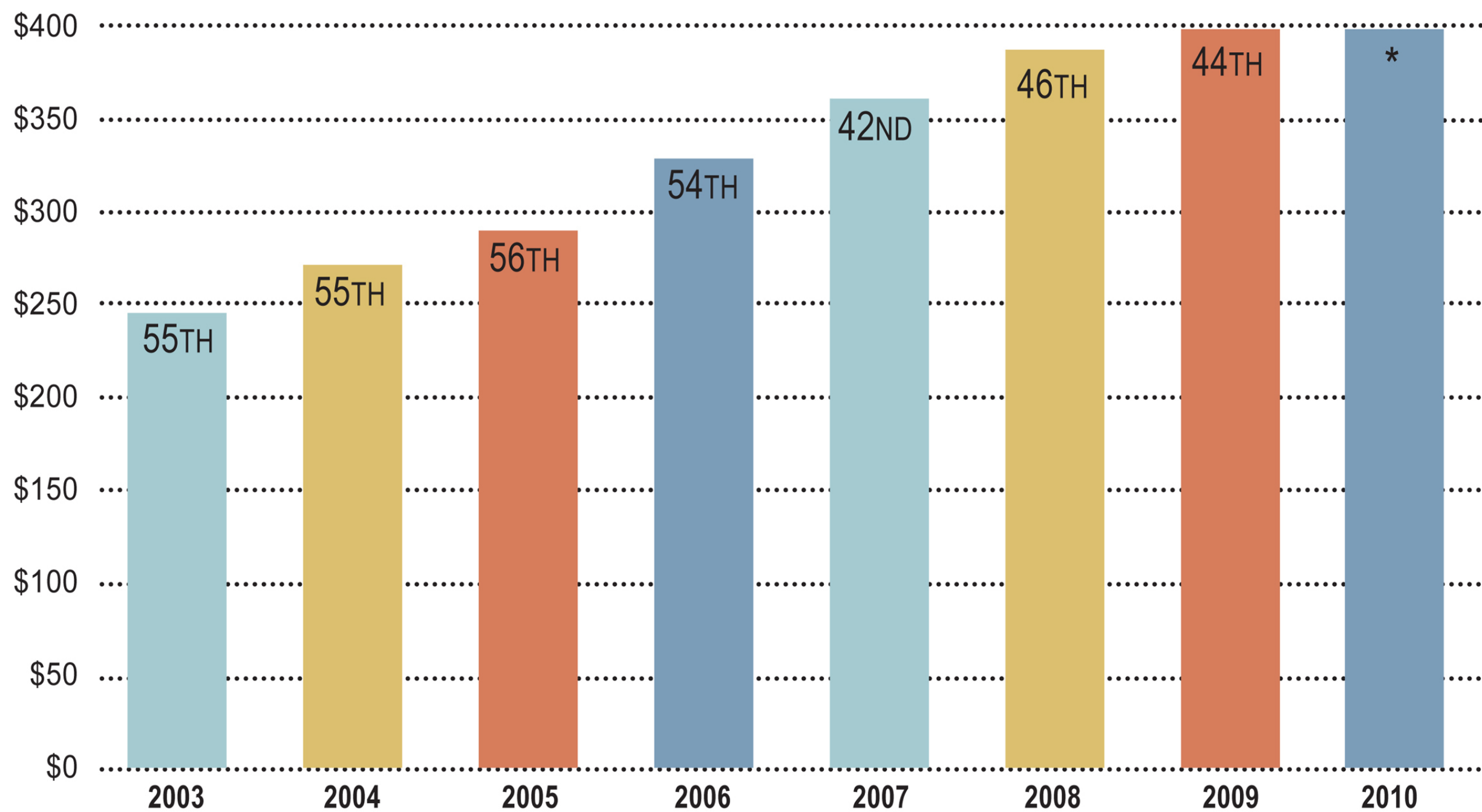


Bar-chart на SVG

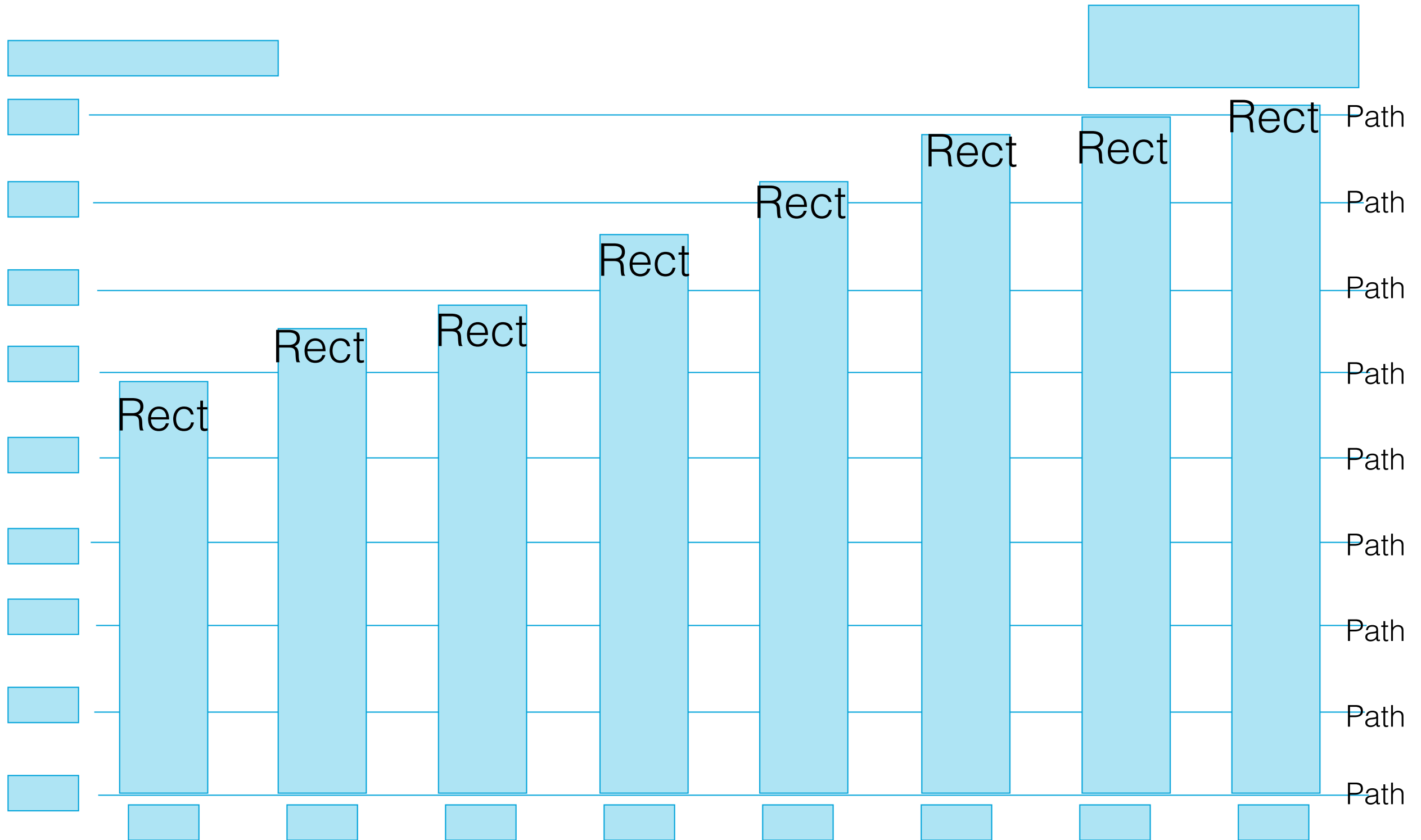
Элементы SVG

(Dollars in Millions)

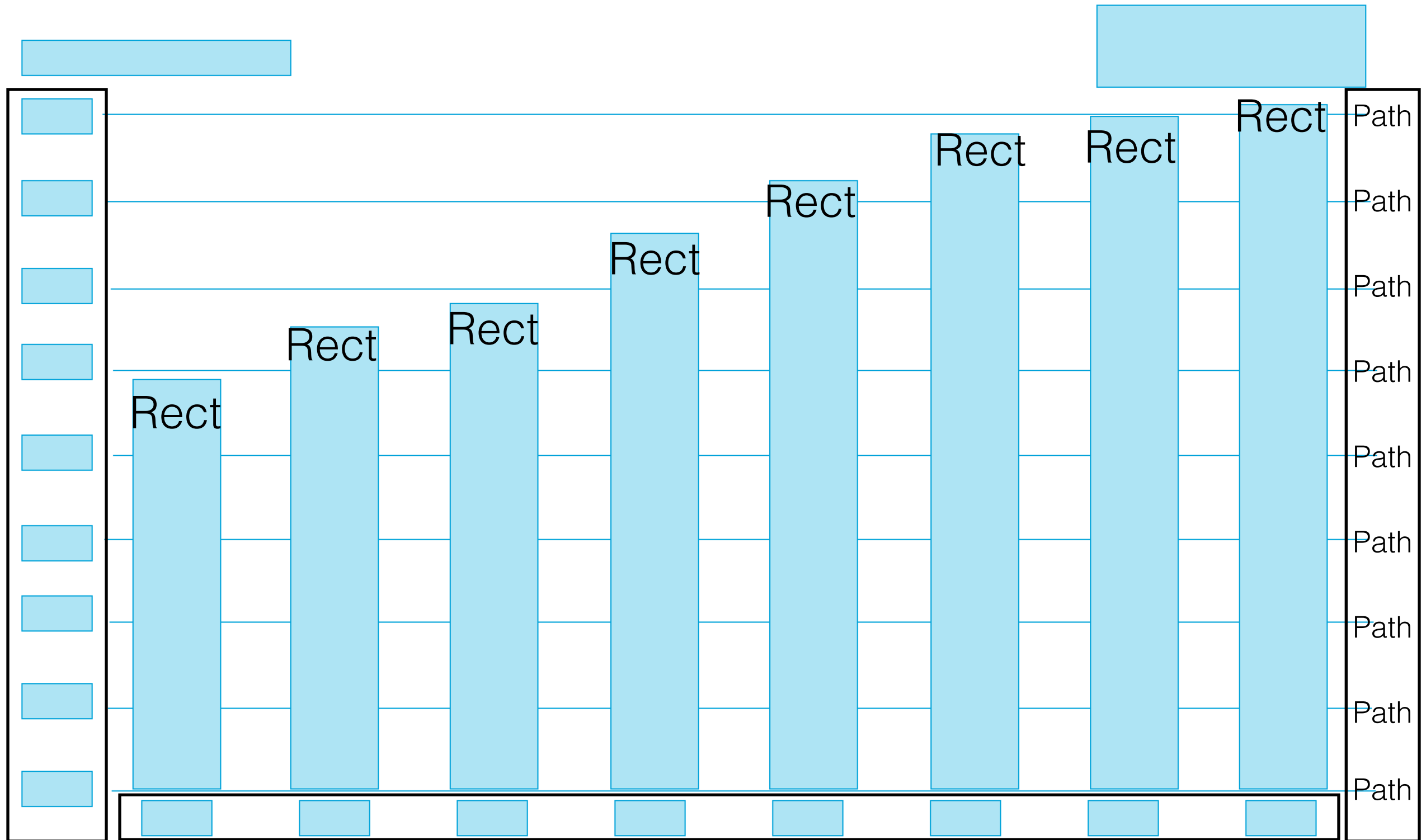
**ranking has not
been released*



SVG

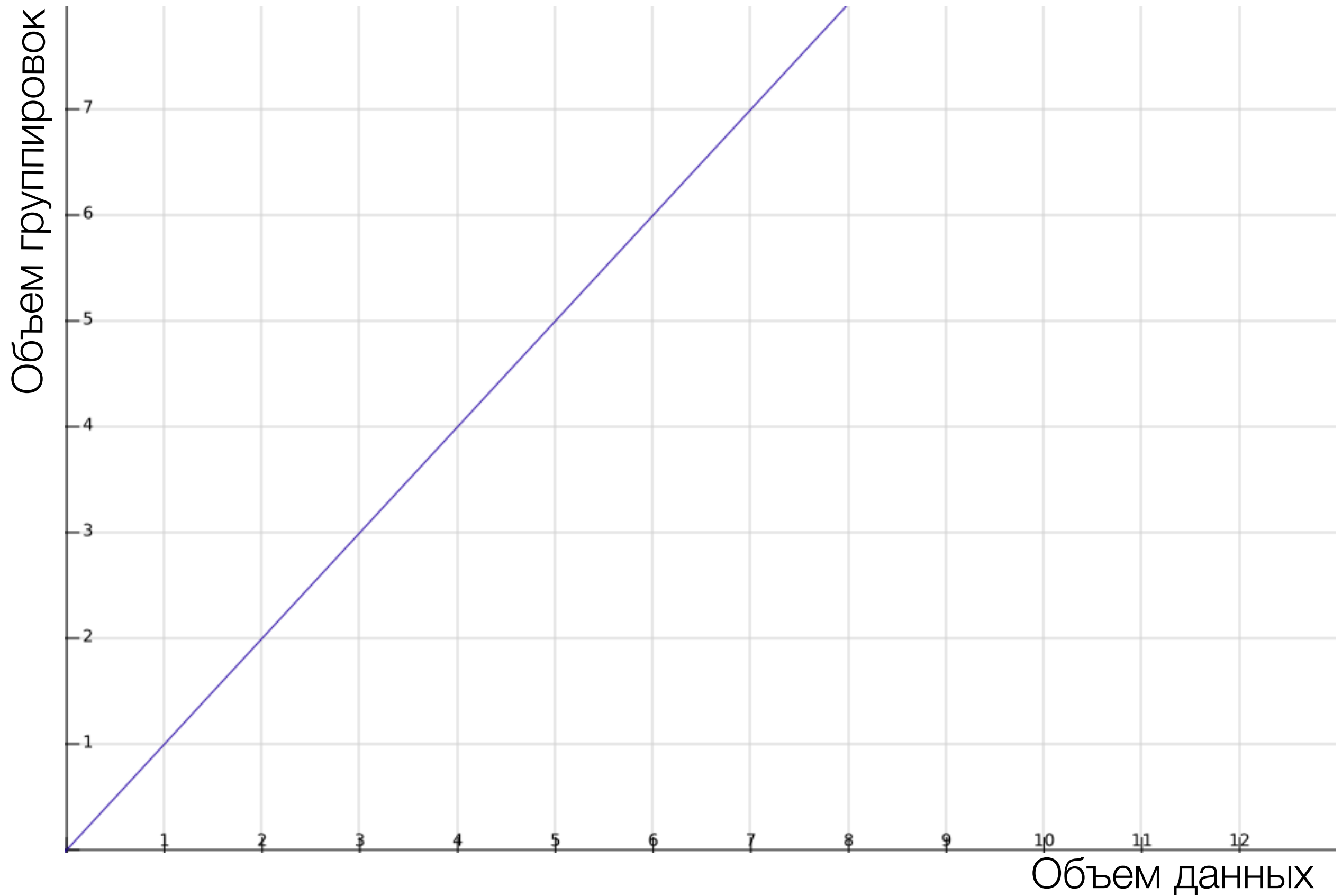


SVG

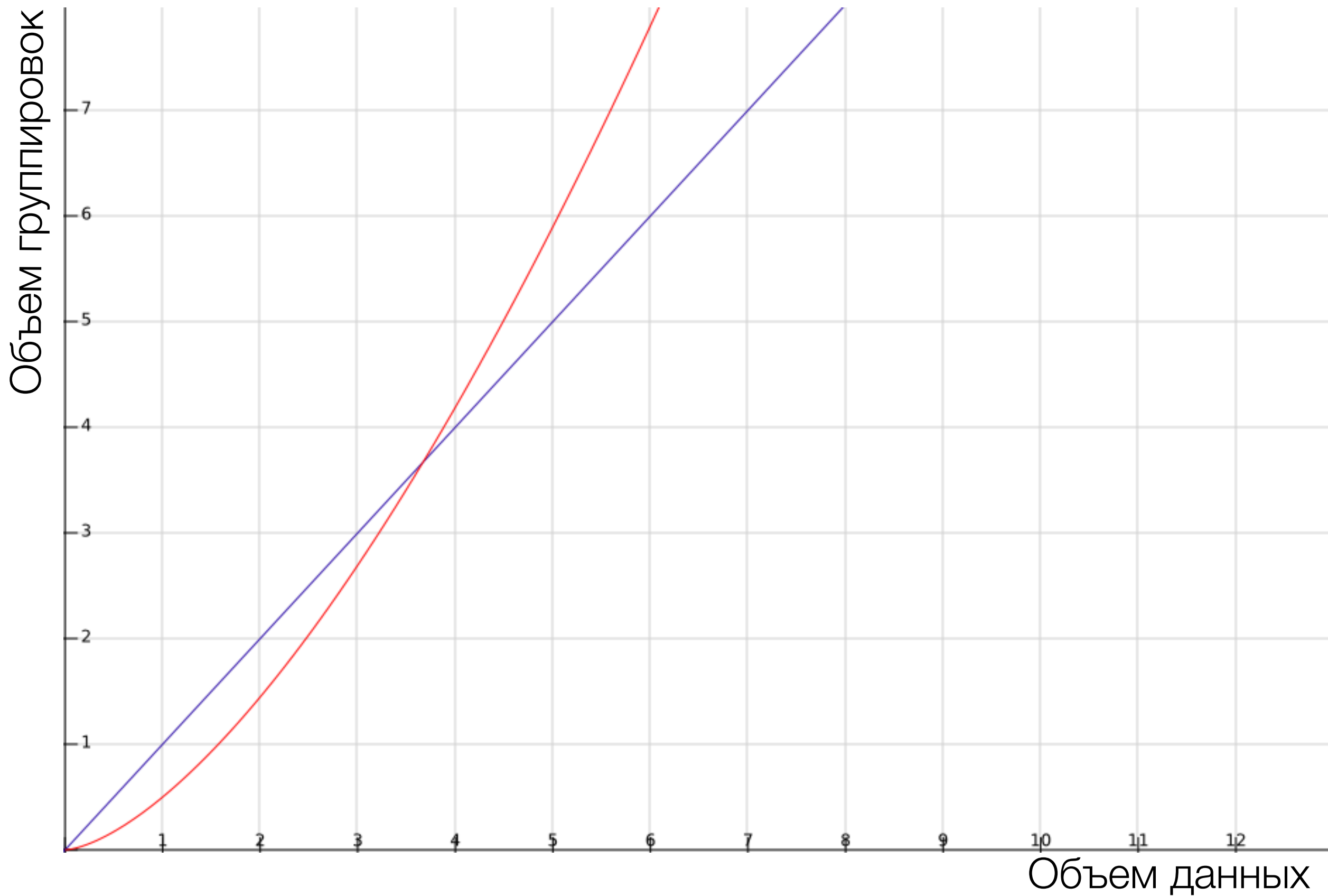


Масштабирование

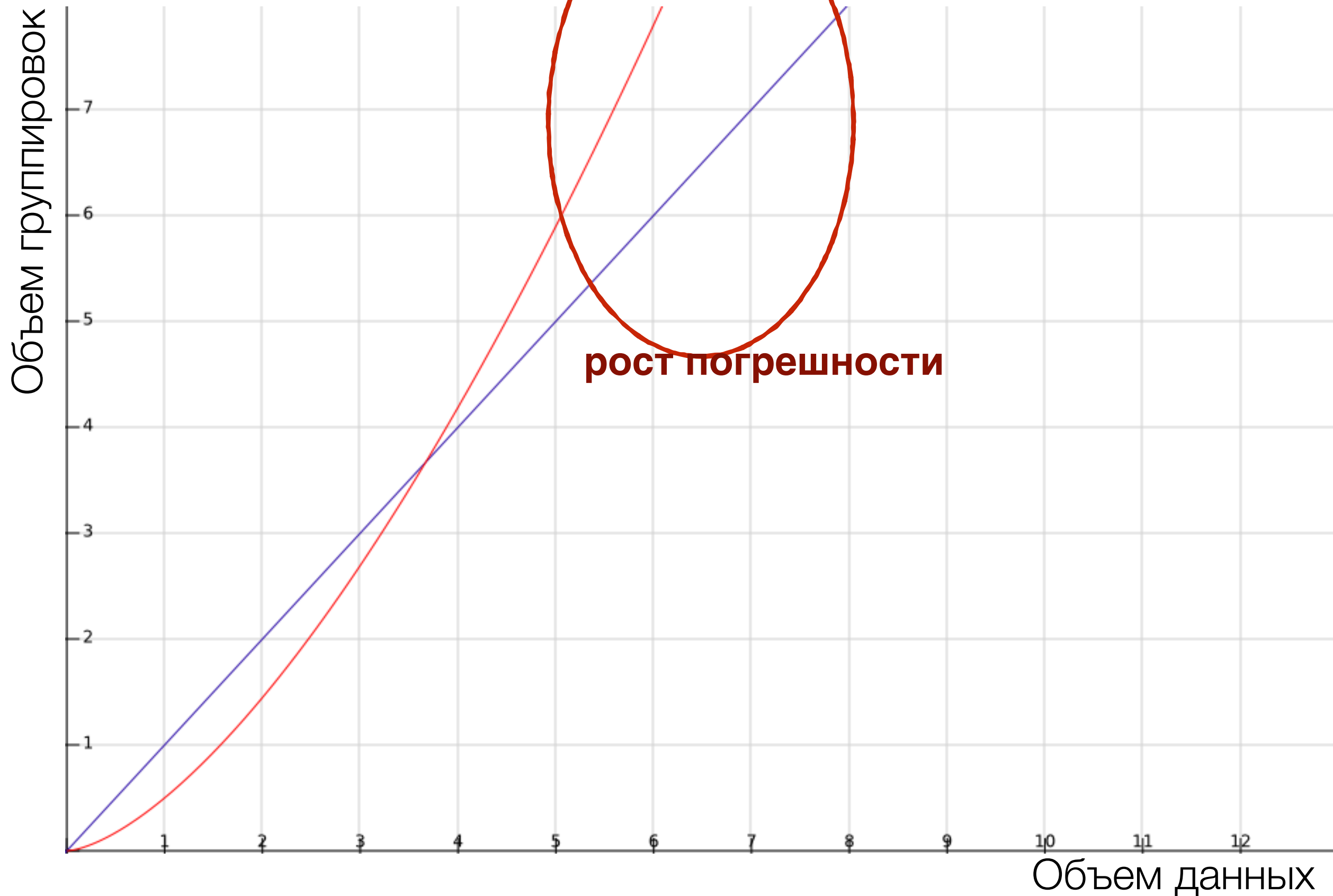
Общий линейный случай



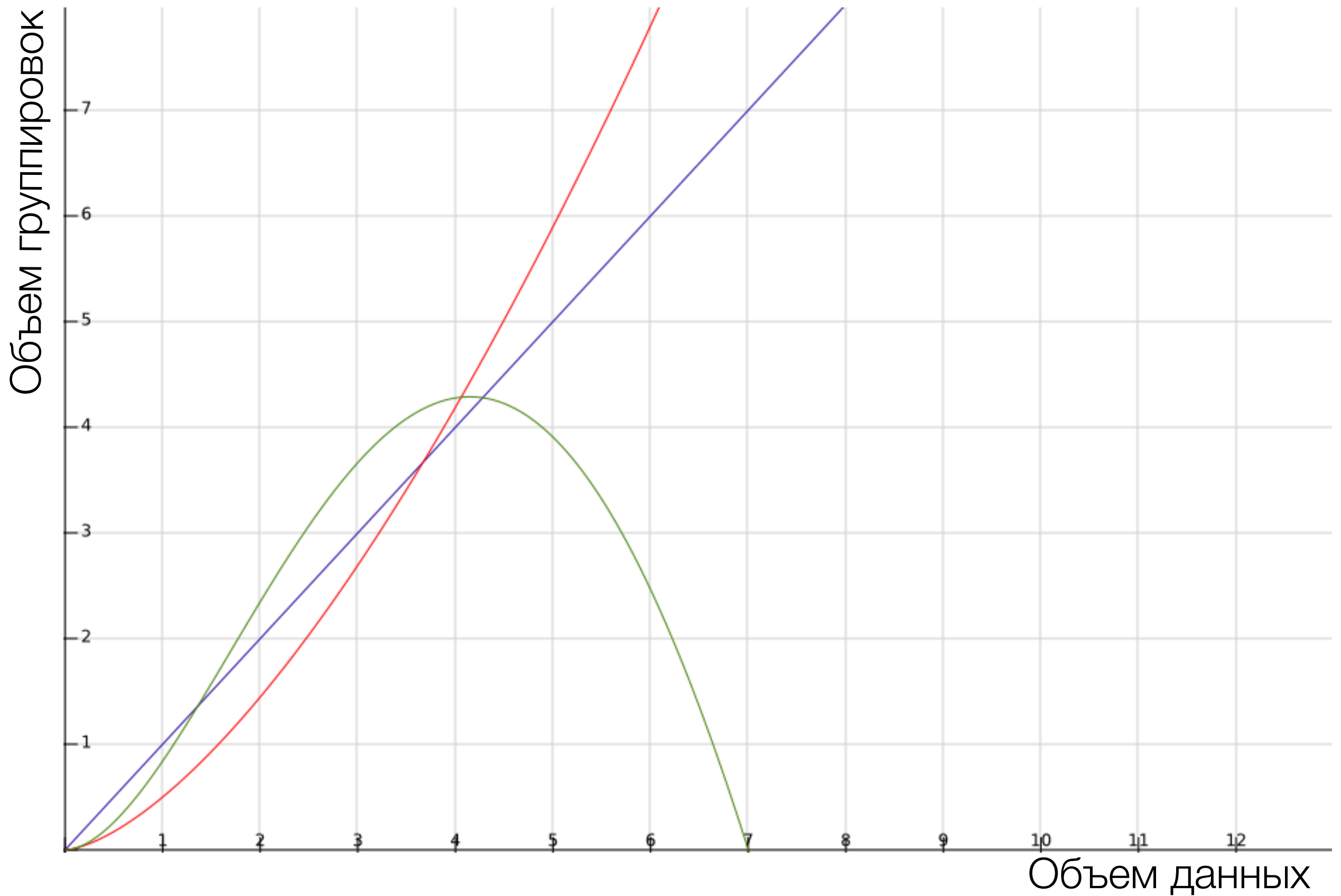
Общий линейный случай

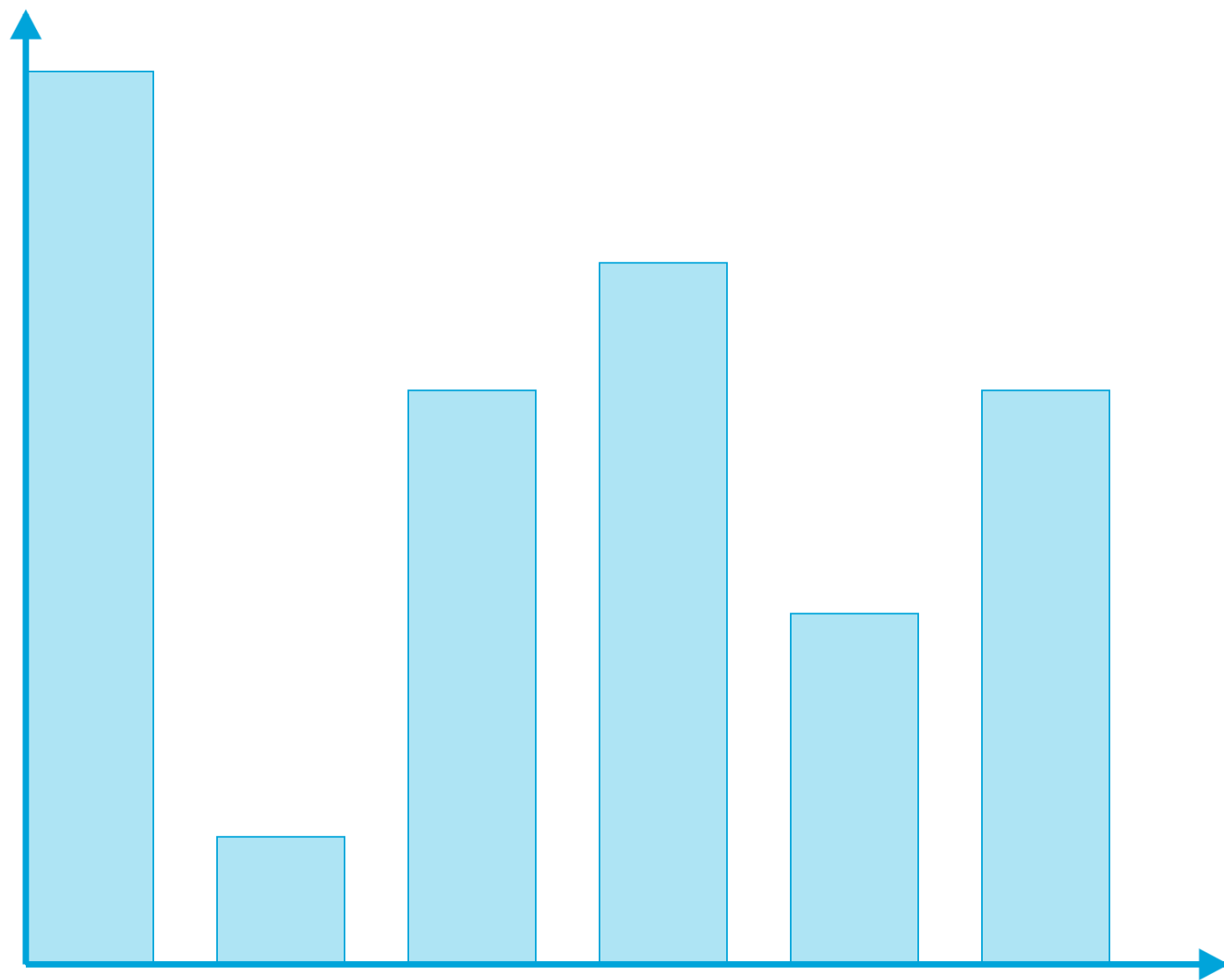


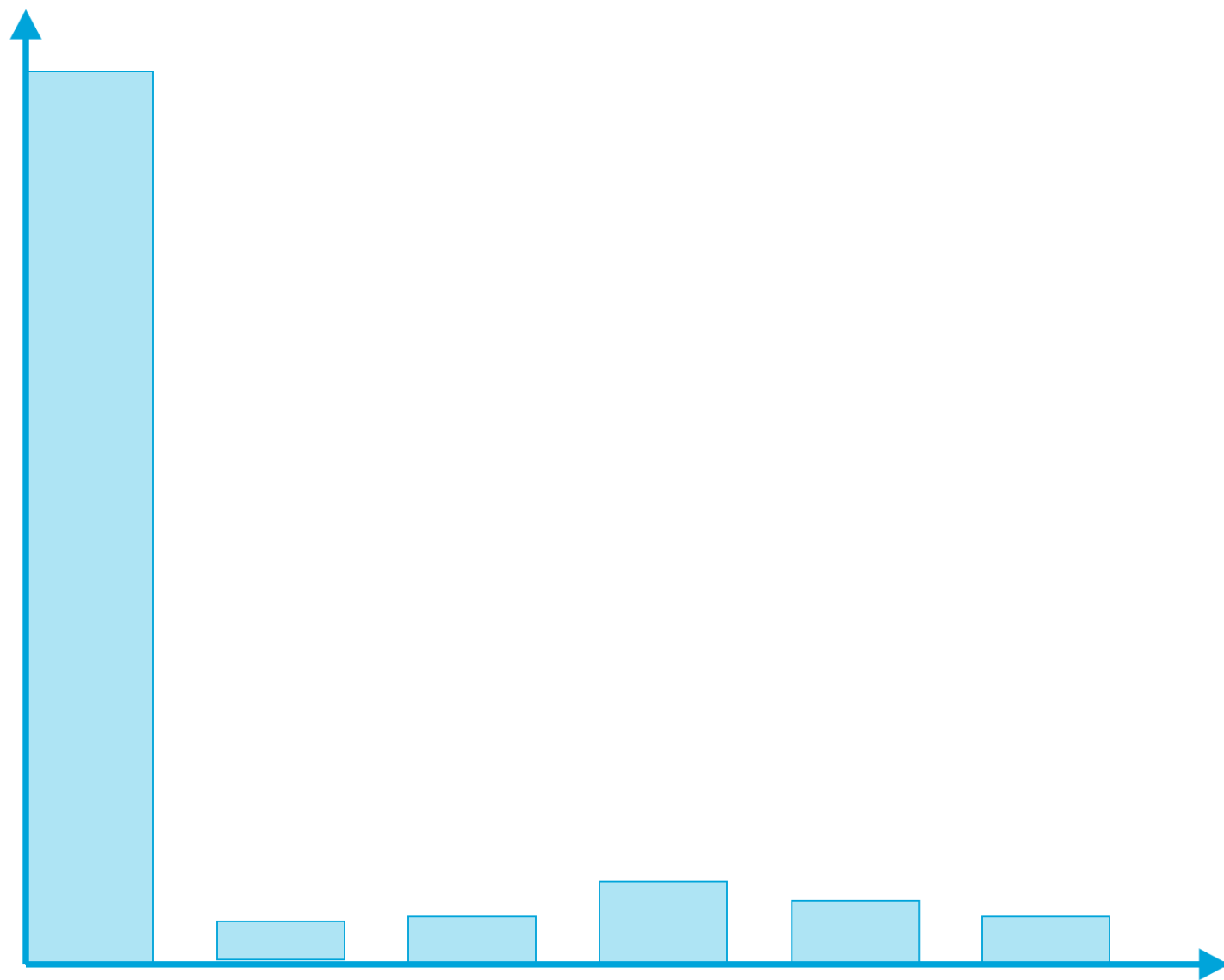
Общий линейный случай



Общий линейный случай







Шкалы

Линейные (linear)

```
01: var scale = d3.scale.linear();
```

```
02:
```

```
03: scale(10); // -> 10
```

Линейные (linear)

```
01: var scale = d3.scale.linear();
```

```
02:
```

```
03: scale(10); // -> 10
```

масштабирование не задано

Линейные (linear)

```
01: var scale = d3.scale.linear();  
02:  
03: scale(10); // -> 10  
04:  
05: scale.domain([0, 100]);  
06:  
07:  
08: scale.range([1, 5]);
```

Линейные (linear)

```
01: var scale = d3.scale.linear();
```

```
02:
```

```
03: scale(10); // -> 10
```

```
04:
```

```
05: scale.domain([0, 100]);
```

```
06: область входных данных
```

```
07:
```

```
08: scale.range([1, 5]);
```

```
область выходных данных
```


Линейные (linear)

```
01: var scale = d3.scale.linear();
```

```
02:
```

```
03: scale(10); // -> 10
```

```
04:
```

```
05: scale.domain([0, 100]);
```

```
06: область входных данных
```

```
07:
```

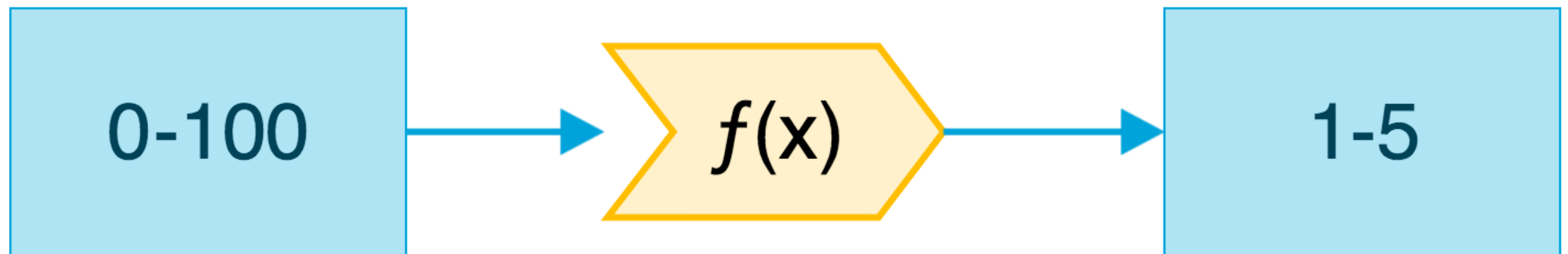
```
08: scale.range([1, 5]);
```

```
область выходных данных
```

Линейные (linear)

100-бальная шкала

5-бальная шкала



весь матан здесь

Линейные (linear)

```
01: var scale = d3.scale.linear();
02:
03: scale(10); // -> 10
04:
05: scale.domain([0, 100]);
06:
07: scale.range([1, 5]);
08:
09: scale(0); //-> 1
10: scale(100); //-> 5
11: scale(37); //-> 2.48
12: scale(64); //-> 3.56
13: scale(73); //-> 3.92
```

Порядковые (ordinal)

```
01: var scale = d3.scale.ordinal();
```

Порядковые (ordinal)

```
01: var scale = d3.scale.ordinal();
```



Порядковые (ordinal)

```
01: var scale = d3.scale.ordinal();
02:
03: var names = ['60', '70', '80',
04:             '90', '00', '10'].map(function(n){
05:     return n + "s";
06: });
07:
08: scale.domain(names);
```

Порядковые (ordinal)

```
01: var scale = d3.scale.ordinal();
02:
03: var names = ['60', '70', '80',
04:             '90', '00', '10'].map(function(n){
05:     return n + "s";
06: });
07:
08: scale.domain(names);
09:
10: scale.rangePoints([0, 100]);
```

Оси

Оси (axis)

```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis();
```

Оси (axis)

```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis();
```



пустая ось

Oси (axis)

```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis()  
06:     .scale(y)  
07:     .orient('left')  
08:     .ticks(10, .01);
```

Оси (axis)

```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis()  
06:     .scale(y) шкала оси  
07:     .orient('left')  
08:     .ticks(10, .01);
```

Оси (axis)

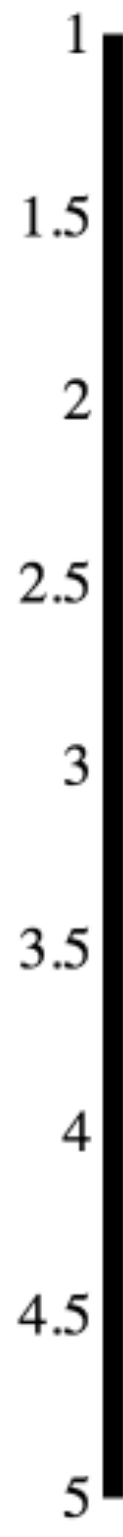
```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis()  
06:     .scale(y) шкала оси  
07:     .orient('left') ориентация на графике  
08:     .ticks(10, .01);
```

Оси (axis)

```
01: var y = d3.scale.linear()  
02:     .range([0, 100])  
03:     .domain([1, 5]);  
04:  
05: var yAxis = d3.svg.axis()  
06:     .scale(y) шкала оси  
07:     .orient('left') ориентация на графике  
08:     .ticks(10, .01); размер шага
```

Oси (axis)

```
01: var yAxis = d3.svg.axis()  
02:     .scale(y)  
03:     .orient('left')  
04:     .ticks(10, .01);  
05:  
06: var axisElement = d3.select('svg')  
07:     .attr('height', 500)  
08:     .attr('width', 50)  
09:     .append('g')  
10:     .attr('transform', 'translate(30, 50)')  
11:     .call(yAxis)
```



Oси (axis)

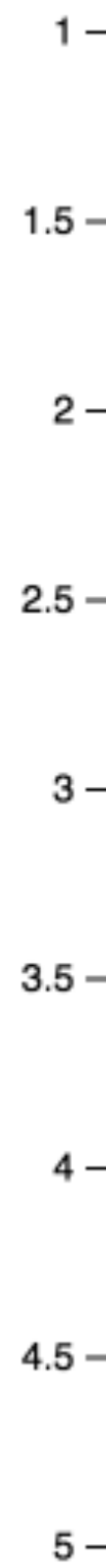
```
01: var yAxis = d3.svg.axis()  
02:     .scale(y)  
03:     .orient('left')  
04:     .ticks(10, .01);  
05:  
06: var axisElement = d3.select('svg')  
07:     .attr('height', 500)  
08:     .attr('width', 50)  
09:     .append('g')  
10:     .attr('transform', 'translate(30, 50)')  
11:     .attr('class', 'yAxis')  
12:     .call(yAxis)
```

Оси (axis)

```
01: var yAxis = d3.svg.axis()  
02:     .scale(y)  
03:     .orient('left')  
04:     .ticks(10, .01);  
05:  
06: var axisElement = d3.select('svg')  
07:     .attr('height', 500)  
08:     .attr('width', 50)  
09:     .append('g')  
10:     .attr('transform', 'translate(30, 50)')  
11:     .attr('class', 'yAxis')  
12:     .call(yAxis)           можем использовать CSS
```

Стилизация осей

```
01: .yAxis path,  
02: .yAxis line {  
03:     fill: none;  
04:     stroke: #000;  
05:     shape-rendering: geometricPrecision;  
06: }
```



Добавление легенды

```
01: axisElement.append('text')
02:     .attr('transform', 'rotate(-90)')
03:     .attr('y', 5)
04:     .attr('dy', 6)
05:     .style('text-anchor', 'end')
06:     .text('Legend');
```

Добавление легенды

```
01: axisElement.append('text')
02:     .attr('transform', 'rotate(-90)')
03:     .attr('y', 5)
04:     .attr('dy', 6)
05:     .style('text-anchor', 'end')
06:     .text('Legend');
```

**Поворачиваем на 90°
(потому что ось вертикальная)**

Добавление легенды

```
01: axisElement.append('text')
02:     .attr('transform', 'rotate(-90)')
03:     .attr('y', 5)
04:     .attr('dy', 6)
05:     .style('text-anchor', 'end')
06:     .text('Legend');
```

Задаем координаты

Добавление легенды

```
01: axisElement.append('text')
02:     .attr('transform', 'rotate(-90)')
03:     .attr('y', 5)
04:     .attr('dy', 6)
05:     .style('text-anchor', 'end')
06:     .text('Legend');
```

Позиционирует точку начала текста

Добавление легенды

```
01: axisElement.append('text')
02:     .attr('transform', 'rotate(-90)')
03:     .attr('y', 5)
04:     .attr('dy', 6)
05:     .style('text-anchor', 'end')
06:     .text('Legend');
```

Задаем контент

