Кросс-платформенная разработка

Lecture 8

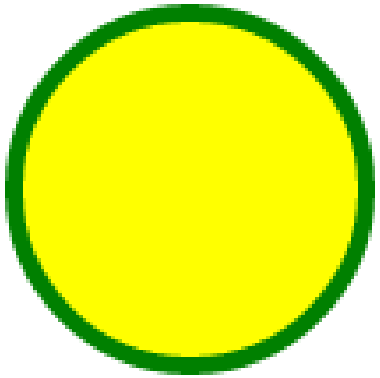
Topics

- SVG
- D3.js

SVG

- Scalable Vector Graphics
- Можно создавать и редактировать в текстовых редакторах
- Можно генерировать на лету
- Можно искать по изображению как по тексту
- Zoom и Scale без потери качества
- XML

Пример



```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

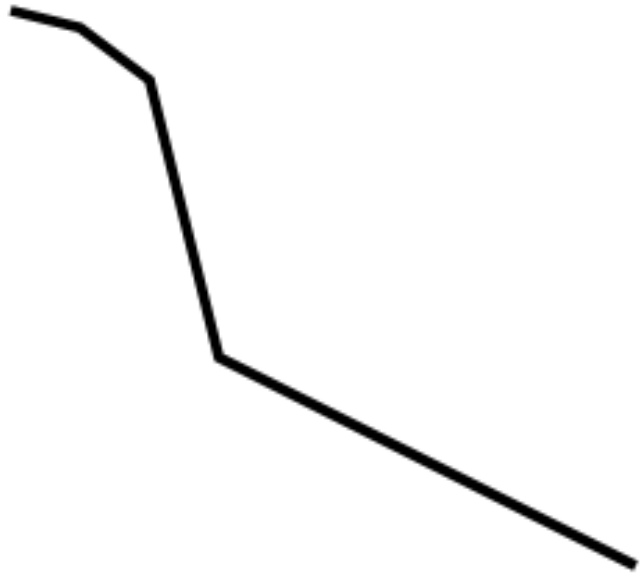
Доступные примитивы

- Rectangle <rect>
- Circle <circle>
- Ellipse <ellipse>
- Line <line>
- Polyline <polyline>
- Polygon <polygon>
- Path <path>

Используя style

```
<svg width="400" height="110">
  <rect width="300" height="100"
        style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
</svg>
```

Polyline



```
<svg height="200" width="500">
  <polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
  style="fill:none;stroke:black;stroke-width:3" />
</svg>
```

Text

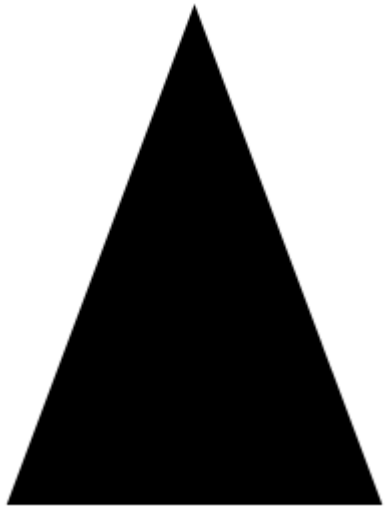

```
<svg height="30" width="200">
  <text x="0" y="15" fill="red">I love SVG!</text>
</svg>
```

```
<svg height="60" width="200">
  <text x="0" y="15" fill="red" transform="rotate(30 20,40)">I love
SVG</text>
</svg>
```

Path

- M = move to
- L = line to
- H = horizontal line to
- V = vertical line to
- C = curve to
- S = smooth curve to
- Q = quadratic Bézier curve
- T = smooth quadratic Bézier curve to
- A = elliptical Arc
- Z = close path

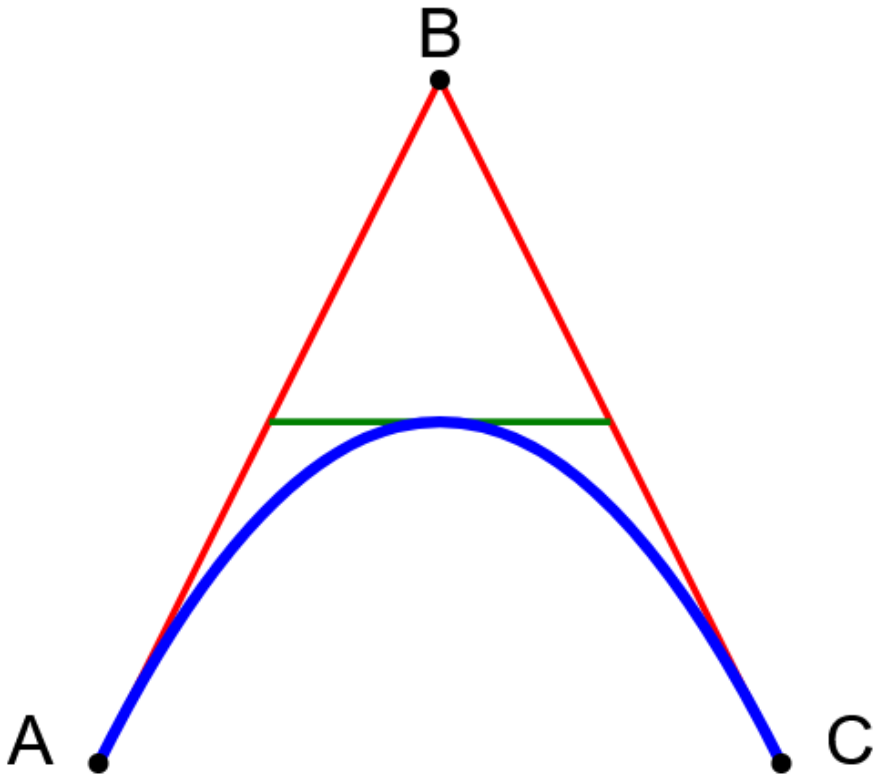
Треугольник



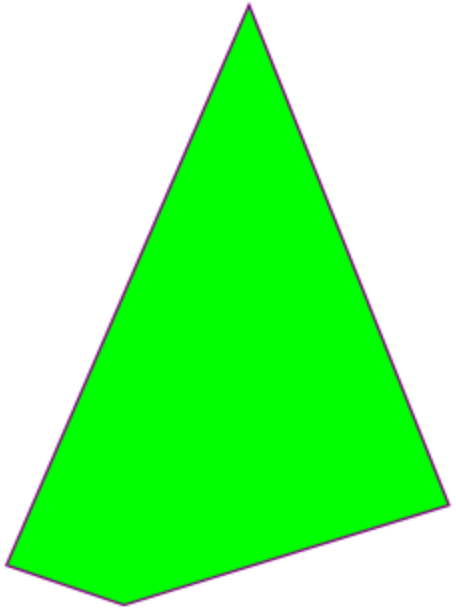
- <svg height="210" width="400">
  <path d="M150 0 L75 200 L225 200 Z" />
  </svg>

Кривые

```
<svg height="400" width="450">
  <path id="lineAB" d="M 100 350 l 150 -300" stroke="red"
  stroke-width="3" fill="none" />
  <path id="lineBC" d="M 250 50 l 150 300" stroke="red"
  stroke-width="3" fill="none" />
  <path d="M 175 200 l 150 0" stroke="green" stroke-width="3"
  fill="none" />
  <path d="M 100 350 q 150 -300 300 0" stroke="blue"
  stroke-width="5" fill="none" />
  <!-- Mark relevant points -->
  <g stroke="black" stroke-width="3" fill="black">
    <circle id="pointA" cx="100" cy="350" r="3" />
    <circle id="pointB" cx="250" cy="50" r="3" />
    <circle id="pointC" cx="400" cy="350" r="3" />
  </g>
  <!-- Label the points -->
</svg>
```

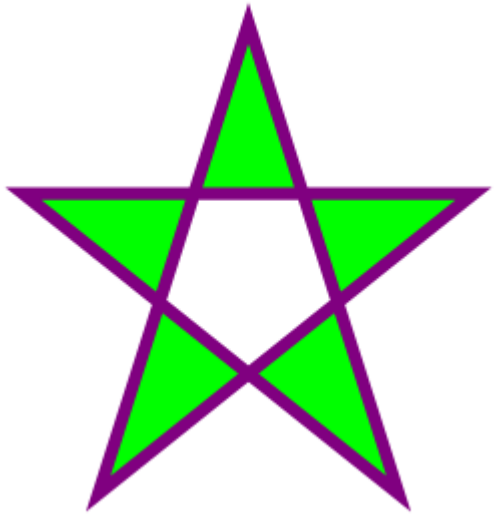


Polygon



```
<svg height="250" width="500">
  <polygon points="220,10 300,210 170,250 123,234"
        style="fill:lime;stroke:purple;stroke-width:1" />
</svg>
```

Модификаторы



```
<svg height="210" width="500">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>
```

SVG Filter Elements

- <feBlend> - filter for combining images
- <feColorMatrix> - filter for color transforms
- <feComponentTransfer>
- <feComposite>
- <feConvolveMatrix>
- <feDiffuseLighting>
- <feDisplacementMap>
- <feFlood>
- <feGaussianBlur>

•<feImage>
•<feMerge>
•<feMorphology>
•<feOffset> - filter for drop shadows
•<feSpecularLighting>
•<feTile>
•<feTurbulence>
•<feDistantLight> - filter for lighting
•<fePointLight> - filter for lighting
•<feSpotLight> - filter for lighting

Пример фильтра


```
<svg height="110" width="110">
  <defs>
    <filter id="f1" x="0" y="0">
      <feGaussianBlur in="SourceGraphic" stdDeviation="15" />
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3"
  fill="yellow" filter="url(#f1)" />
</svg>
```

Градиенты

```
<svg height="150" width="400">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-
opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-
opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
</svg>
```

Анимации

```
<rect id="cool_shape" ...="">
  <animate xlink:href="#cool_shape" ...=""></animate>
</rect>

<rect id="cool_shape" ...="">
  <animate ...=""></animate>
</rect>
```

Анимация через CSS

```
<rect>
  <animate
      attributetype="CSS"
      attributename="opacity"
      from="1"
      to="0"
      dur="5s"
      repeatcount="indefinite">
  </animate>
</rect>
```

Анимация через атрибуты элементов

```
<circle id="my-circle" r="30" cx="50" cy="50" fill="orange">
        <animate xlink:href="#my-circle"
                 attributename="cx"
                 from="50"
                 to="450"
                 dur="1s"
                 begin="click"
                 fill="freeze">
        </animate>
</circle>
```

Анимация через атрибуты элементов + 1

```
<circle id="my-circle" r="30" cx="50" cy="50" fill="orange">
        <animate xlink:href="#my-circle"
                 attributename="cx"
                 from="50"
                 to="450"
                 dur="1s"
                 begin="click + 1s"
                 fill="freeze">
        </animate>
</circle>
```

Последовательные анимации

```
<animate
    xlink:href="#orange-circle"
    attributename="cx"
    from="50"
    to="450"
    dur="5s"
    begin="click"
    fill="freeze"
    id="circ-anim">
</animate>
```

```
<animate
    xlink:href="#blue-rectangle"
    attributename="x"
    from="50"
    to="425"
    dur="5s"
    begin="circ-anim.begin + 1s"
    fill="freeze"
    id="rect-anim">
</animate>
```

D3.js

- Модульная библиотека для визуализации данных
- Работает с SVG и Canvas
- Поддерживает базовые визуализации

Селекторы

```
const paragraphs = document.getElementsByTagName("p");
for (let i = 0; i < paragraphs.length; i++) {
    var paragraph = paragraphs.item(i);
    paragraph.style.setProperty("color", "blue", null);
}
```

```
d3.selectAll("p").style("color", "blue");
```

Динамические свойства

```
d3.selectAll("p").style("color", function() {
    return "hsl(" + Math.random() * 360 + ",100%,50%)";
});
```

Связь с данными

```
d3.selectAll("p")
    .data([4, 8, 15, 16, 23, 42])
    .style("font-size", function(d) { return d + "px"; });
```

Enter

```
d3.select("body")
    .selectAll("p")
    .data([4, 8, 15, 16, 23, 42])
    .enter().append("p")
    .text(function(d) { return "I’m number " + d + "!"; });
```

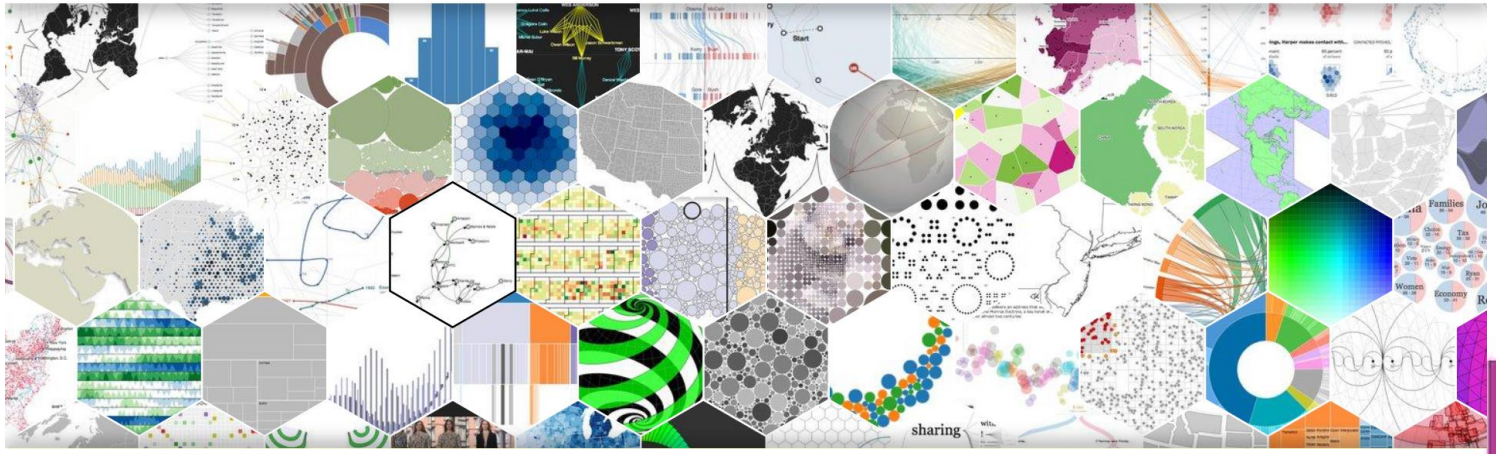
Exit

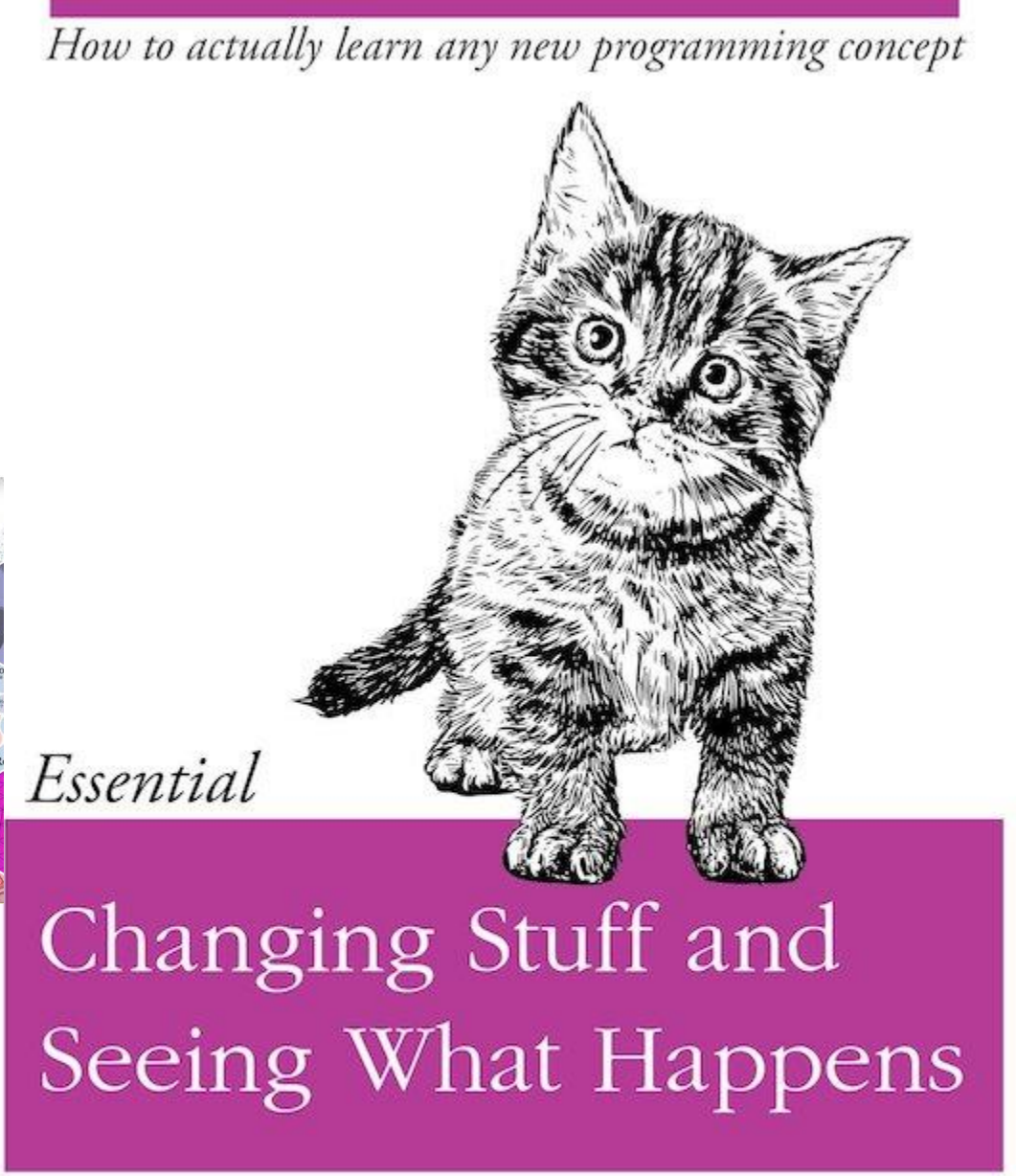
```
// Update…
var p = d3.select("body")
    .selectAll("p")
    .data([4, 8, 15, 16, 23, 42])
    .text(function(d) { return d; });

// Enter…
p.enter().append("p")
    .text(function(d) { return d; });

// Exit…
p.exit().remove();
```

Transition

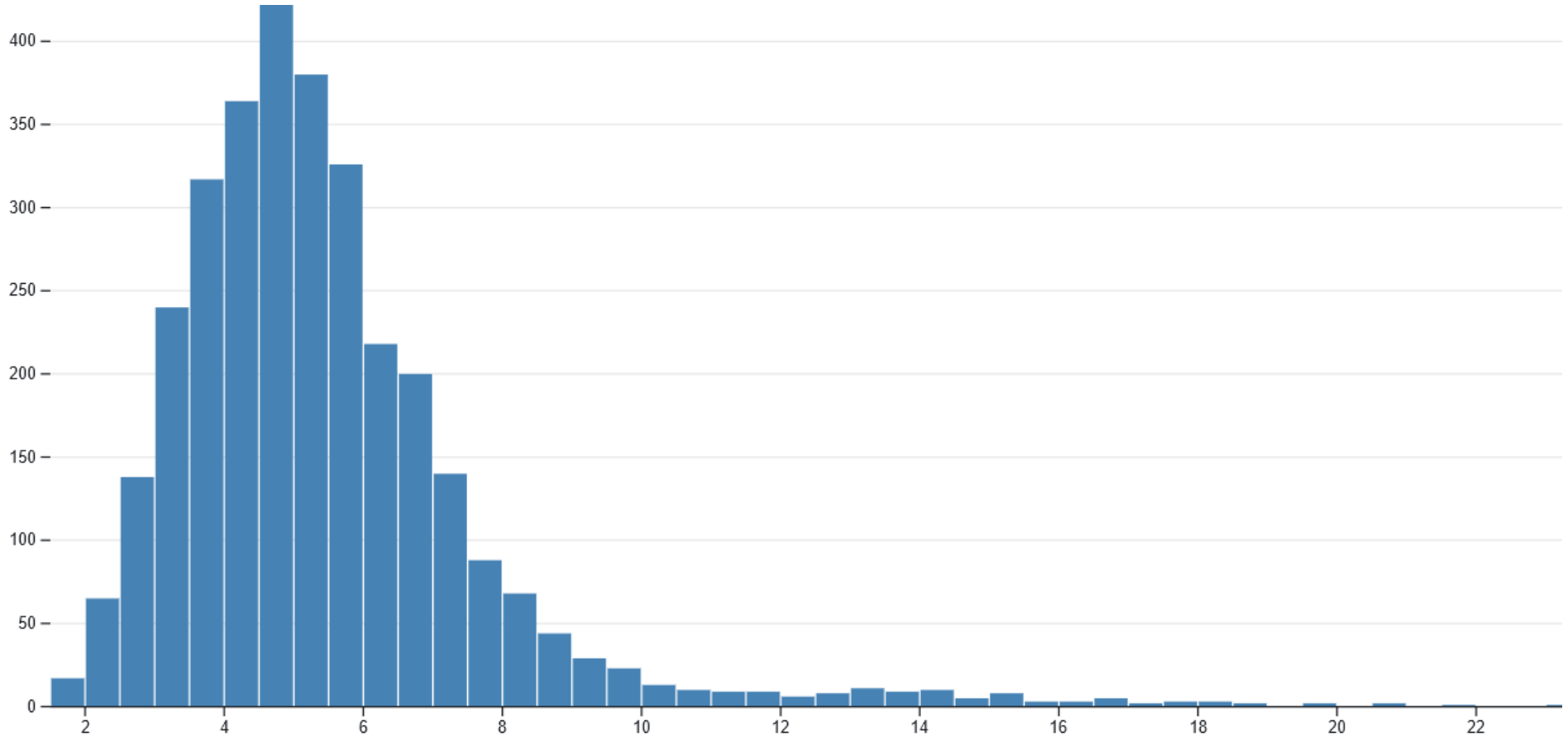
```
d3.select("body").transition()
    .style("background-color", "black");


d3.selectAll("circle").transition()
    .duration(750)
    .delay(function(d, i) { return i * 10; })
    .attr("r", function(d) { return Math.sqrt(d * scale); });
```

Как научиться?

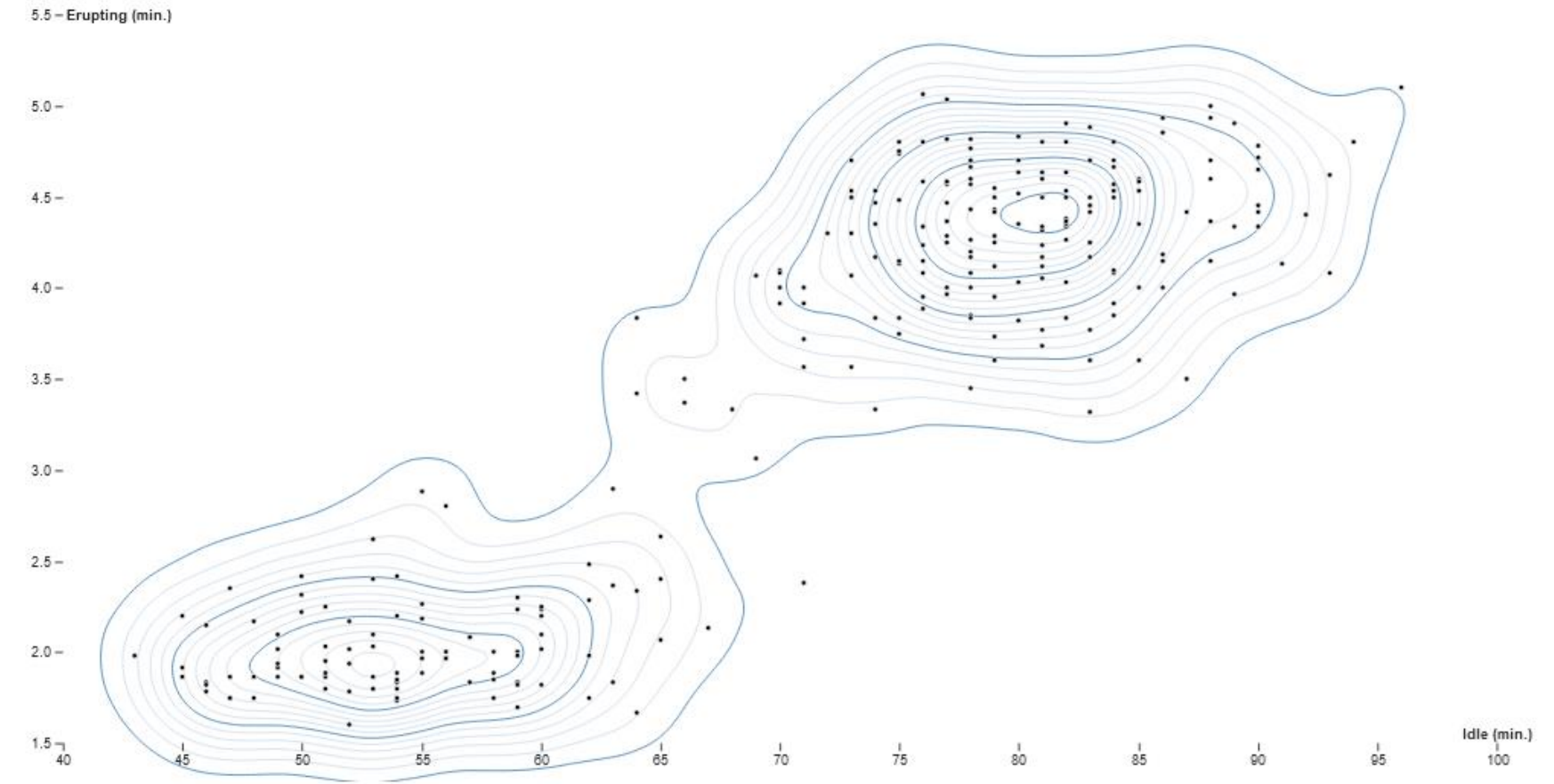




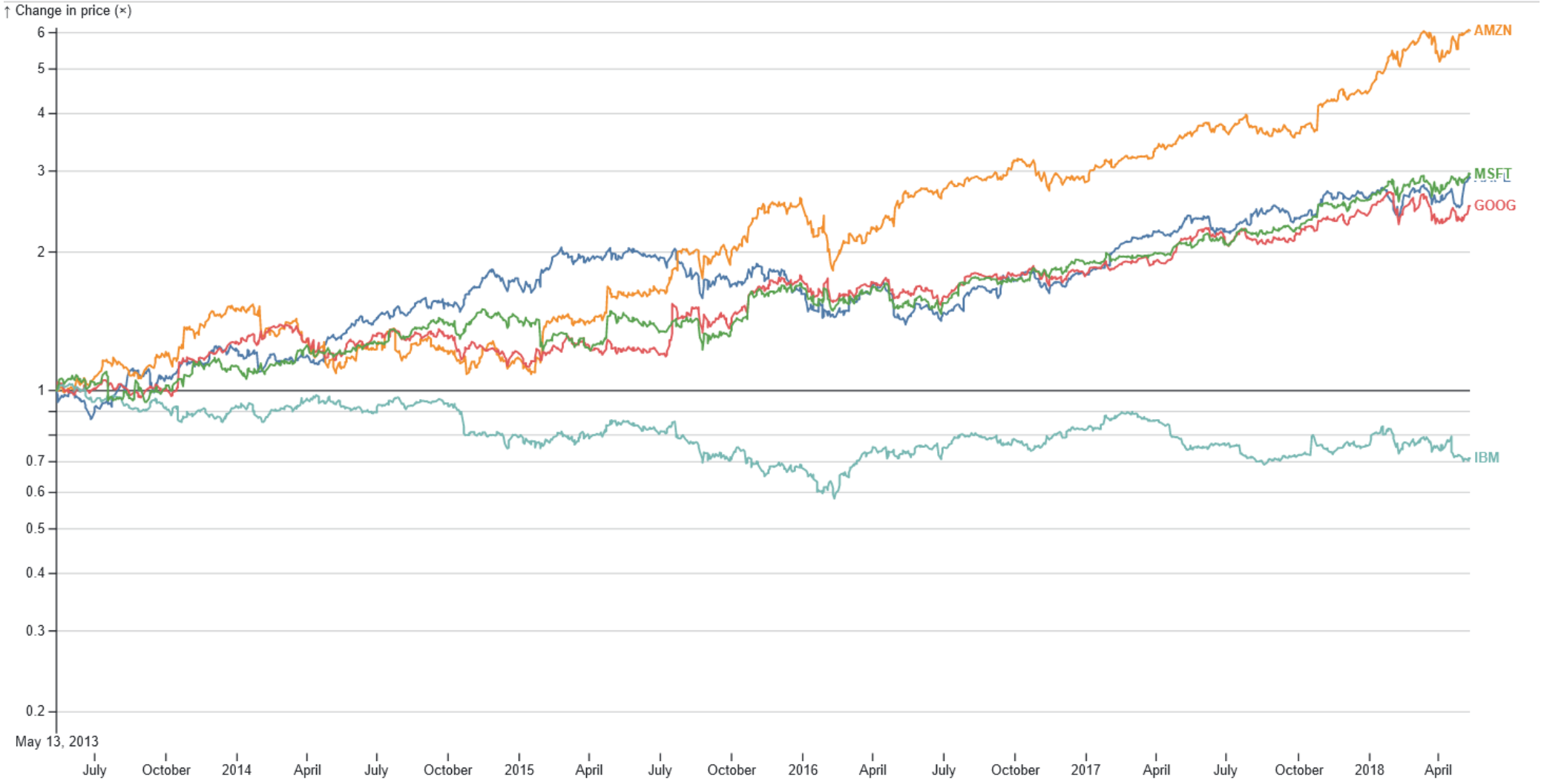
Элементы: Анализ: Гисторграммы



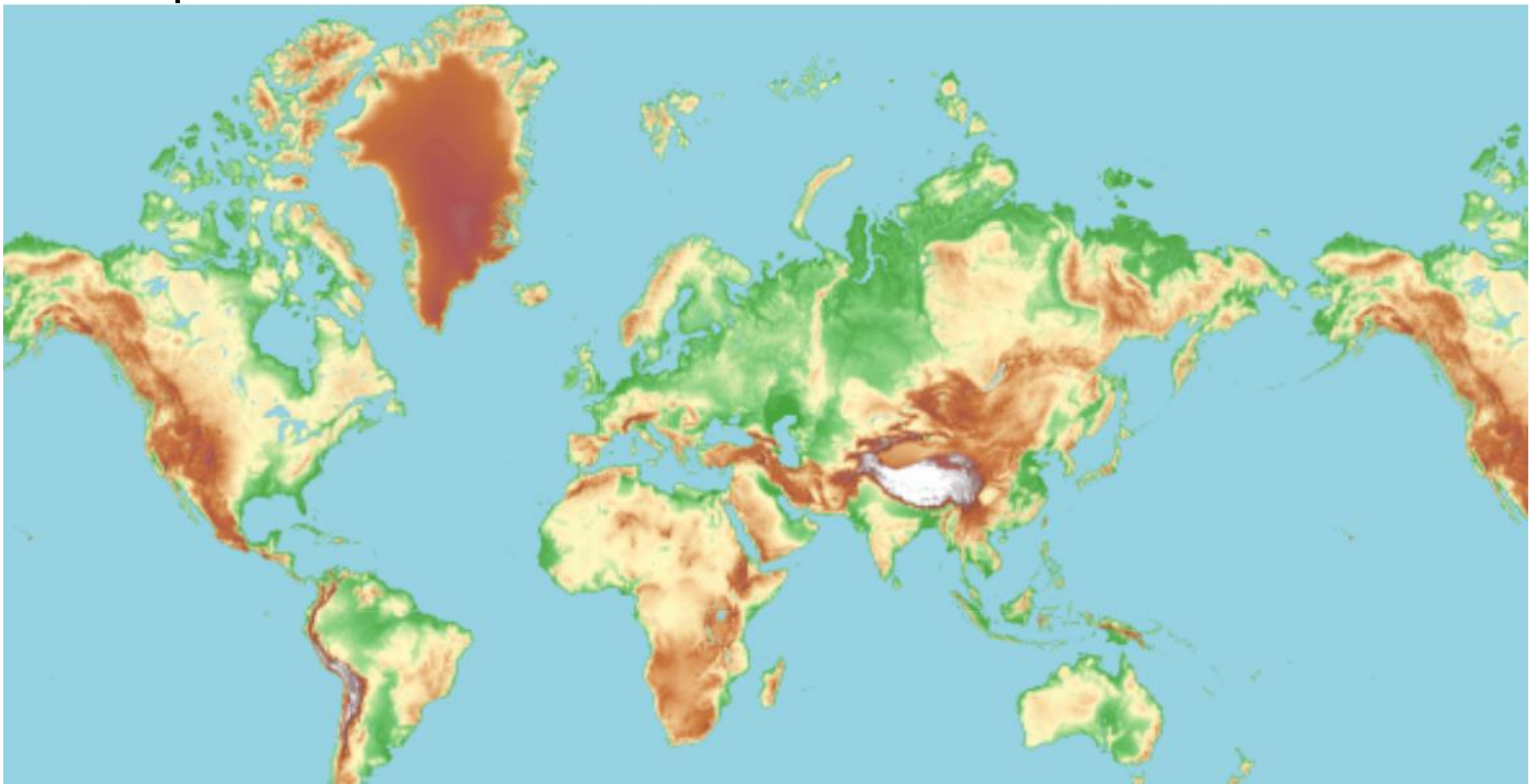
Контуры + Плотность



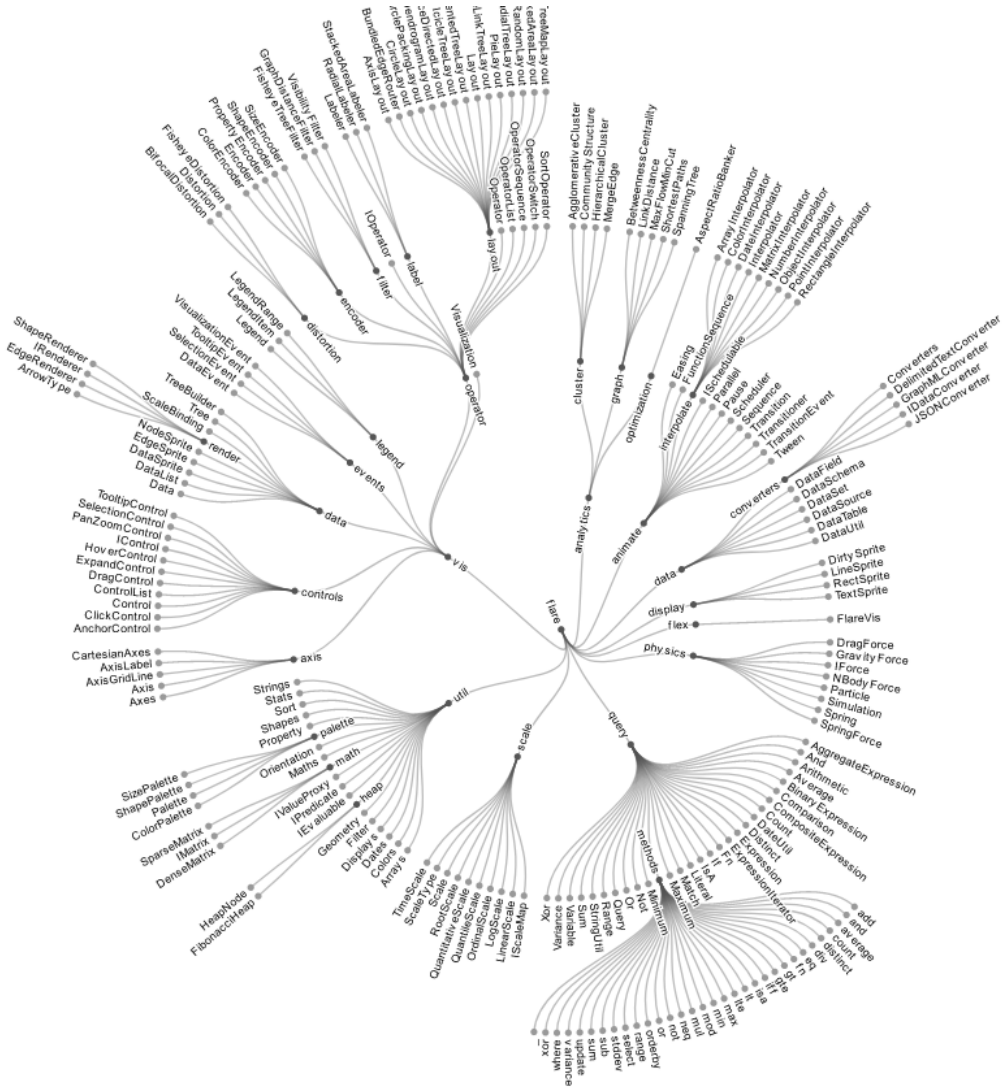
Графики


↑ Change in price (×)
6
5
4
3
2
1
0.7
0.6
0.5
0.4
0.3
0.2
AMZN
MSFT
GOOG
IBM
May 13, 2013
July
October
2014
April
July
October
2015
April
July
October
2016
April
July
October
2017
April
July
October
2018
April

Карты



Иерархические структуры



References

- https://javascript.info/
- https://developer.mozilla.org/
- https://www.w3schools.com