hci lab
HUMAN × COMPUTER INTERACTION

D3.js Tutorial (2)

조재민

Human-Computer Interaction Lab

Dept. of Computer Science & Engineering

Seoul National University

Find me on Github (http://github.com/e-) or

visit my blog (http://www.jaeminjo.com)

Git

- 파일을 수정한 후 동작이 이상해서 되돌리고 싶을 때
  - git checkout <file>
- 버전관리를 하고 싶지 않은 파일은
  - vim .gitignore
- 이미 add를 하고 난 후에 이상한 파일이 추가된 걸 알았다면
  - git rm --cached -r .
  - vim .gitignore
  - git add .

Github Pages

- https://pages.github.com/
- 정적인 웹사이트를 호스팅 할 수 있게 해줌
  - gh-pages 브랜치에 있는 파일을 웹으로 접근 가능
  - index.html가 시작 페이지가 됨

- https://github.com/snuhci2017/ExampleRepository
- git branch gh-pages (처음에 브랜치 생성)
- git push origin gh-pages (원격의 브랜치로 푸시)
- https://snuhci2017.github.io/ExampleRepository/

Event

- jQuery의 이벤트와 비슷함
- 일반적인 HTML 요소에도 적용 가능하므로 이벤트만을 위해서 jQuery를 따로 쓸 필요 없음
- this는 DOM 객체이므로 d3의 API를 쓰려면 d3.select로 감싸기

```
d3.selectAll('rect')
  .on('mouseenter', function(d, i) {
    var rect = d3.select(this)

    console.log('data', d)
    console.log('index', i)
    console.log('SVGElement', rect)
    console.log('x', d3.event.x, 'y', d3.event.y)
  })
```

Event

- 자주 쓰이는 이벤트

```
selection
    .on('mouseenter', function() { })
    .on('mouseleave', function() { })
    .on('mousemove', function() { })
    .on('click', function() { })
    .on('mouseenter', null)
```

Event

- “this” is hard

```
function log(){ console.log(this); }

d3.selectAll('rect')
  .on('mouseenter.e1', log)
  .on('mouseenter.e2', function() {
    log(); // bad
    log.bind(this)(); // good
  })
```

Transition

- d3.js에서는 쉽게 시각화에 트랜지션을 넣을 수 있음
- https://bl.ocks.org/mbostock/1256572
- https://bost.ocks.org/mike/transition/

```
function move(circles, x, y) {
  circles
    .attr('cx', x)
    .attr('cy', y)
}

function move2(circles, x, y) {
  circles
    .transition()
    .attr('cx', x)
    .attr('cy', y)
}
```

Transition

```
d3.selectAll('circle')
  .transition()
  .duration(300)
  .delay(function(_, i) { return i * 500 })
  .attr('cx', function(_, i) { return (i + 1) * 10 })
  .attr('cy', 40)
```

Transition

- 항상 시작점에 주의!
  - enter시 트랜지션이 시작될 때의 속성을 지정해줘야
- Bad

```
var update = svg.selectAll('circle')
                .data(data)

update
  .enter()
  .append('circle')

update
  .transition()
  .attr('cx', 20)
  .attr('cy', function(_, i) {return i * 30 })
  .attr('r', function(d){ return d * 10 })
  .attr('opacity', .3)
  .attr('fill', 'red')
```

Transition

- Good

```
var update = svg.selectAll('circle')
                .data(data)

update
  .enter()
  .append('circle')
  .attr('cx', 50)
  .attr('cy', 50)
  .attr('r', 0)
  .attr('opacity', 1)
  .attr('fill', 'white')

update
  .transition()
  .attr('cx', 20)
  .attr('cy', function(_, i) {return i * 30 })
  .attr('r', function(d){ return d * 10 })
  .attr('opacity', .3)
  .attr('fill', 'red')
```

Transition

- 트랜지션을 중첩하려면 이름을 붙여줘야

Bad

```
update
  .enter()
  .append('circle')
  .attr('cx', 50)
  .attr('cy', 50)
  .attr('fill', 'red')
  .attr('r', 10)

update
  .transition()
  .attr('cx', 20)

update
  .transition()
  .attr('cy', 20)
```

Good

```
update
  .enter()
  .append('circle')
  .attr('cx', 50)
  .attr('cy', 50)
  .attr('fill', 'red')
  .attr('r', 10)

update
  .transition('x transition')
  .attr('cx', 20)

update
  .transition('y transition')
  .attr('cy', 20)
```

Translation

```
selection
    .attr('transform', 'translate(' + x + ',' + (y / 30) + ')')

function translate(x, y){ return 'translate(' + x + ',' + y + ')'}

selection
    .attr('transform', translate(x, y / 30))
```

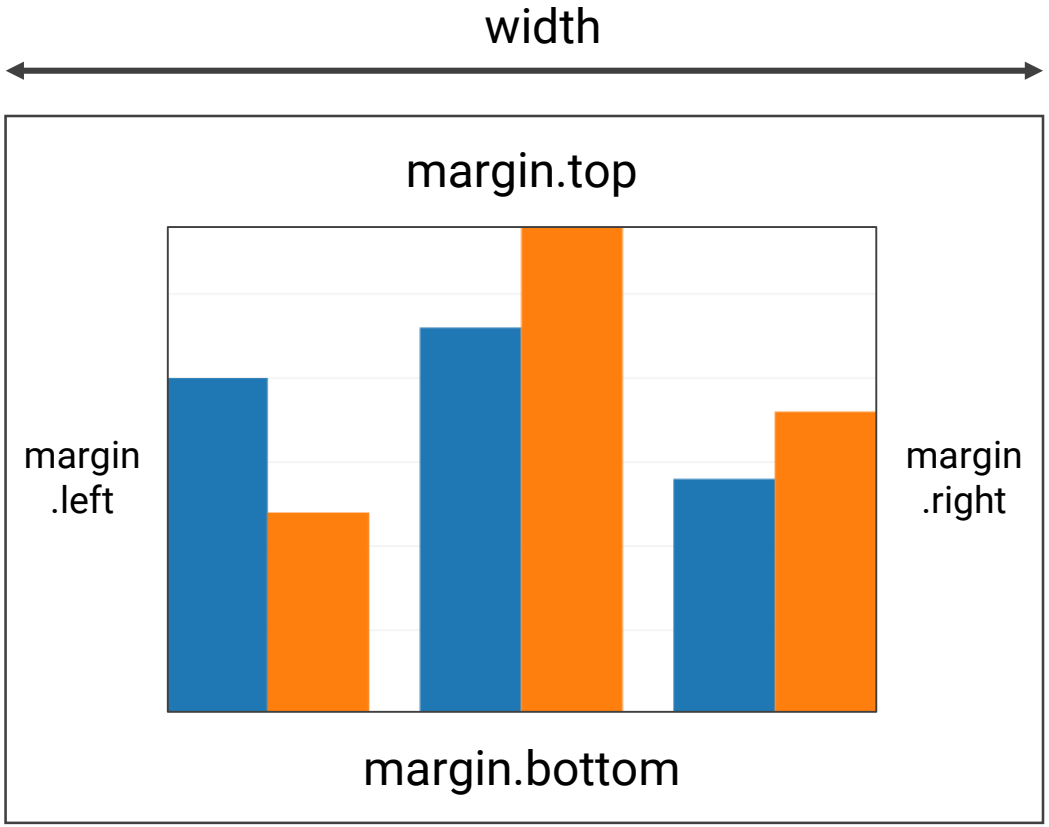
Selection

```
d3.selectAll(".bar").style("opacity", 1)
d3.selectAll("#barlabels").style("opacity", 1)
d3.selectAll("#barYAxis").style("opacity", 1)
d3.selectAll("#barXAxis").style("opacity", 1)
d3.selectAll("#bars").selectAll("text").style("opacity", 1);

d3.selectAll('.bar, #barlabels, #barYAxis, #barXAxis, #bars text')
  .style('opacity', 1);
```

Margin 처리

- 축 레이블 등을 위해 시각화에 여백을 넣는 것이 권장됨

```
var margin = {
    left: 10,
    right: 40,
    top: 20,
    bottom: 30
}
```



Margin 처리

1. 시각화를 margin을 제외하고 그린 후 시각화 전체를 평행이동

```
var x = d3.scale.linear().range([0, width - margin.left - margin.right])

// ...

group.attr('transform', translate(margin.left, margin.top))
```

2. 시각화 내 scale 자체에서 margin을 고려

```
var x = d3.scale.linear().range([margin.left, width - margin.right])

// ...
```

Margin 처리

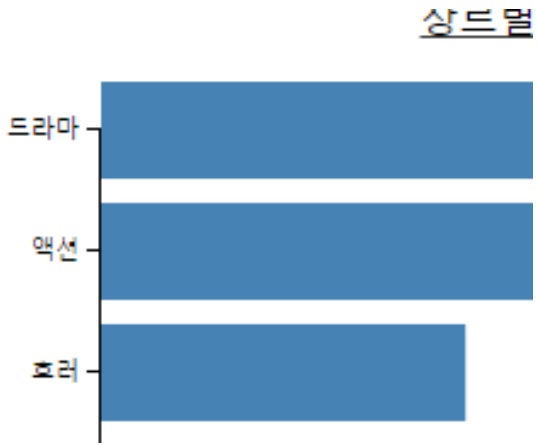
- 아래와 같은 코드는 피할 수 있다면 피하는 것을 권장

```
selection
  .attr('cx', function(d) { return x(d.grade) + margin.left; })
```

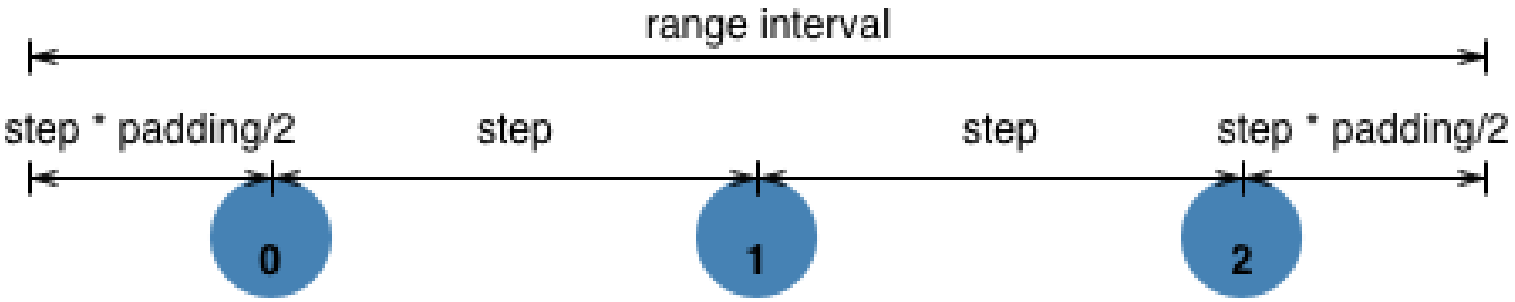
- scale 자체에 좌표에 관련된 정보를 모두 추상화 하는 것이 좋음

바 차트 Categorical Axis

- 범주형 스케일을 표현하는 d3.scale.ordinal()의 경우 주어진 데이터 값을 다양한 range에 매핑할 수 있음

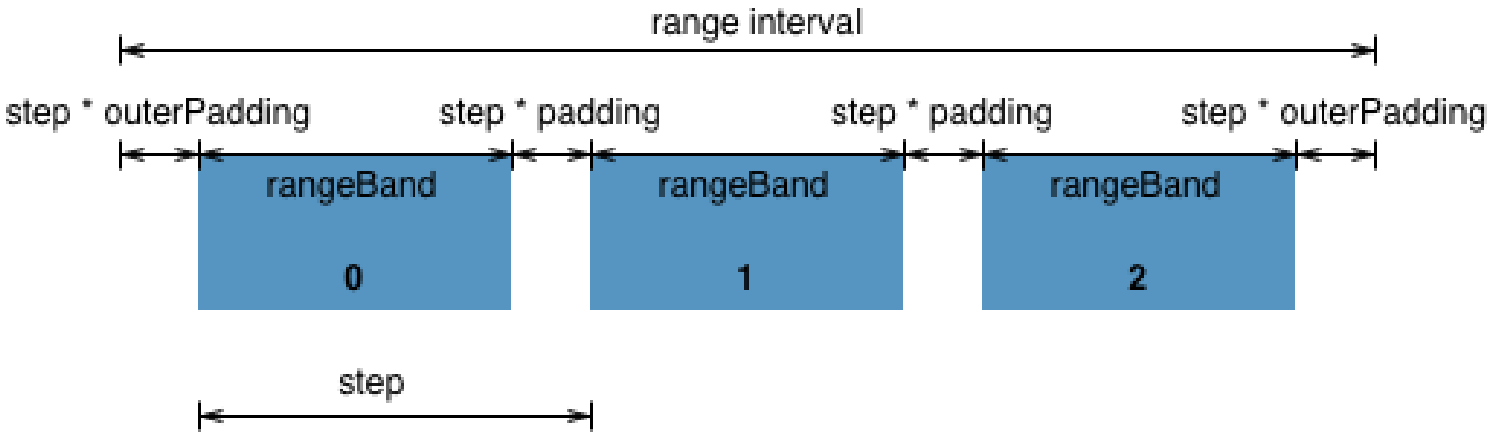


바 차트 Categorical Axis




```
var o = d3.scale.ordinal()
    .domain([1, 2, 3, 4])
    .rangePoints([0, 100]);

o.range(); // [0, 33.333333333333336, 66.66666666666667, 100]
```

바 차트 Categorical Axis




```
var o = d3.scale.ordinal()
    .domain([1, 2, 3])
    .rangeBands([0, 100]);

o.rangeBand(); // 33.333333333333336
o.range(); // [0, 33.333333333333336, 66.66666666666667]
o.rangeExtent(); // [0, 100]
```

소수점 픽셀 처리

- 길이 또는 좌표를 데이터에 매핑할 때 소수점 이하 값으로 인해 시각화가 뿌옇게 보일 수 있음

1. 시각화 요소의 css에 shape-rendering 속성에 crispEdges을 지정

2. “round” 버전의 API 이용
   - ordinal.rangeBands() -> ordinal.rangeRoundBands()

Enter & Update Selection

- Ugly

```
var svg = d3.select('svg')
var data = [1, 2];

var update = svg.selectAll('rect')
                .data(data)

update
   .enter()
   .append('rect')
   .attr('...', '...')

svg.selectAll('rect')
   .attr('...', '...')
```

Enter & Update Selection

- Bad

```
var svg = d3.select('svg')
var data = [1, 2];

var update = svg.selectAll('rect')
                .data(data)

update
   .enter()
   .append('rect')
   .attr('...', '...')

update
   .attr('...', '...')
```

Enter & Update Selection

- Good

```
var svg = d3.select('svg')
var data = [1, 2];

var update = svg.selectAll('rect')
                .data(data)

update
   .enter()
   .append('rect')

update
   .attr('...', '...')
```

Enter & Update Selection

- enter 셀렉션에 append 함수가 호출되는 순간 추가된 요소들이 자동적으로 update 셀렉션에 삽입됨

- 따라서,
  - enter 셀렉션을 처리할 때에는 요소만 추가하고 이 요소들의 속성 값을 설정해주지 않아도 됨 (설정은 update 셀렉션에서)
  - Enter 셀렉션에 요소를 추가하고 update 셀렉션에 이를 포함하기 위해 다시 selectAll 함수를 호출하지 않아도 됨

- https://bost.ocks.org/mike/selection/#enter-update

Use Arrow Functions

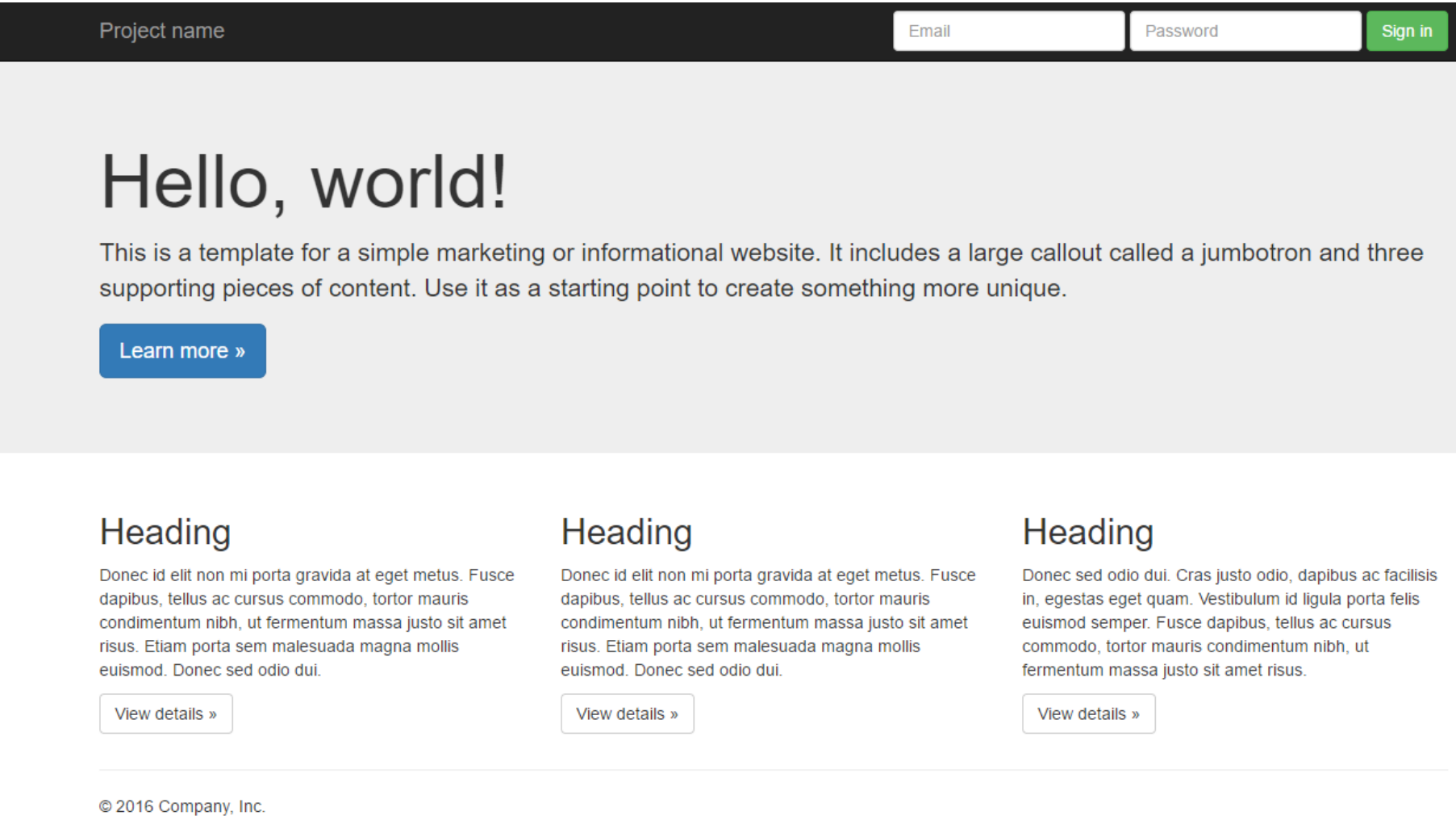
```
selection
    .attr('cx', function(d) { return x(d.price); })

selection
    .attr('cx', (d) => x(d.price))
```

Use Arrow Functions?

- Arrow functions are not syntactic sugar for ordinary functions.

```
var elements = []

selection
    .each(function(d) {
        var circle = d3.select(this)
        elements.push(circle)
    })

// vs

selection
    .each((d) => {
        var circle = d3.select(this)
        elements.push(circle)
    })

console.log(elements)
```

Bootstrap


Project name
Email
Password
Sign in
Hello, world!
This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.
Learn more »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.
View details »
© 2016 Company, Inc.

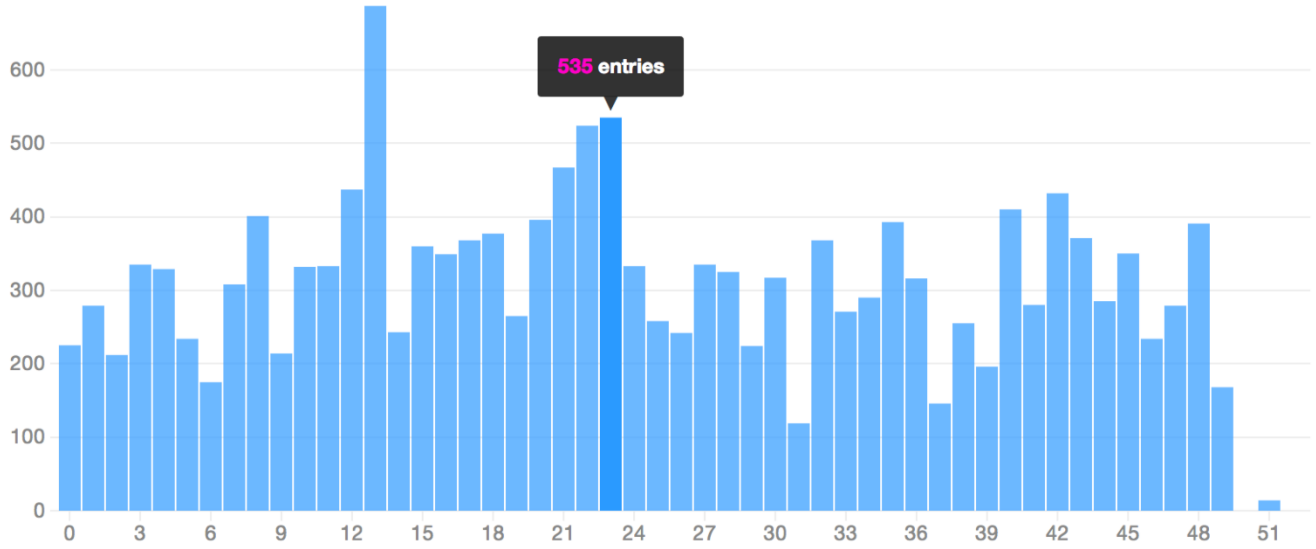
Bootstrap

- http://getbootstrap.com/getting-started/
- http://getbootstrap.com/css/
- http://getbootstrap.com/components/

Tooltip

- 시각화를 간결하게 유지하면서 추가적인 정보를 보여주고 싶을 때 툴팁을 이용

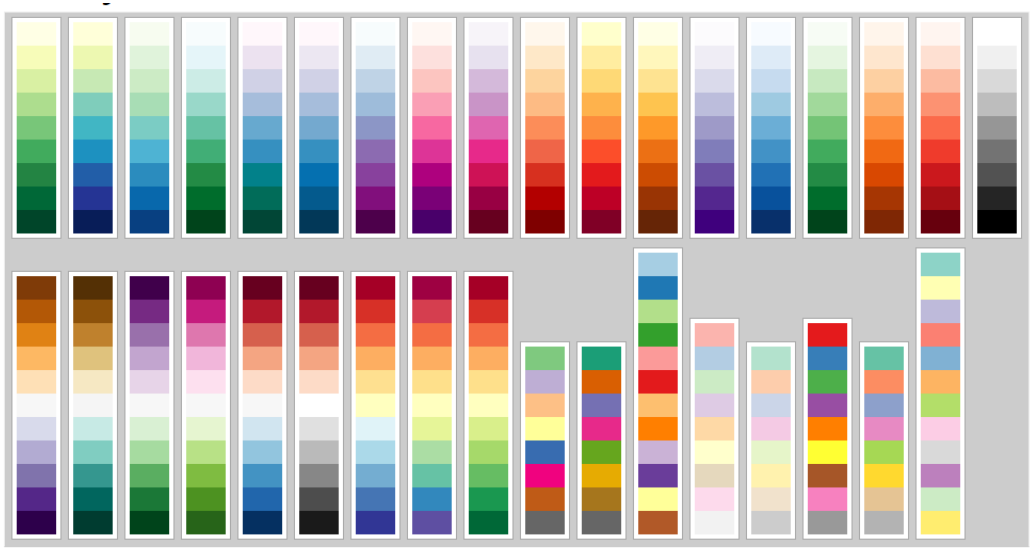
- https://github.com/Caged/d3-tip



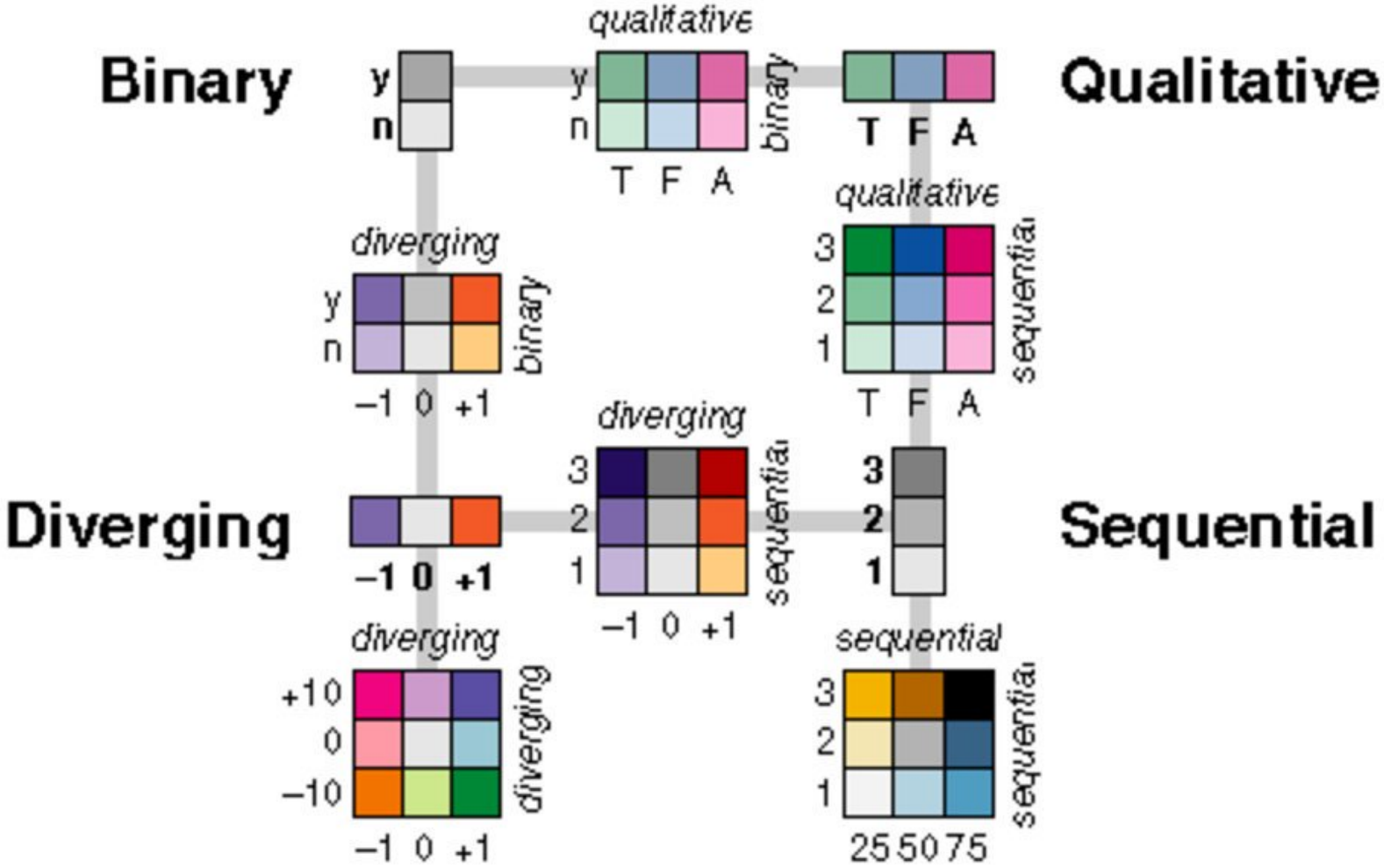
ColorScheme

- 인지적으로 효율적인 색을 사용

- http://colorbrewer2.org (by Brewer et al.)
- https://bl.ocks.org/mbostock/5577023

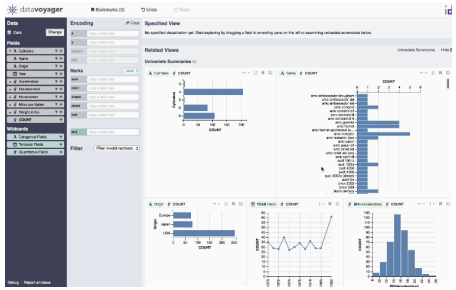


ColorScheme



추가 참고 자료

- By Dominik Moritz and Kanit “Ham” Wongsuphasawat
  - Part 1: http://uwdata.github.io/d3-tutorials/d3-01.html#/
  - Part 2: http://uwdata.github.io/d3-tutorials/d3-02.html#/



- By Michael Correll and Jane Hoffsell
  - https://docs.google.com/presentation/d/1YgWaiW7dfQ8C3a_LFiIA9heEaKthY27ajNG0uDuvpws/edit#slide=id.p