

Lab 9: Treemaps & Heatmaps

정보 비주얼라이제이션 2015 Fall

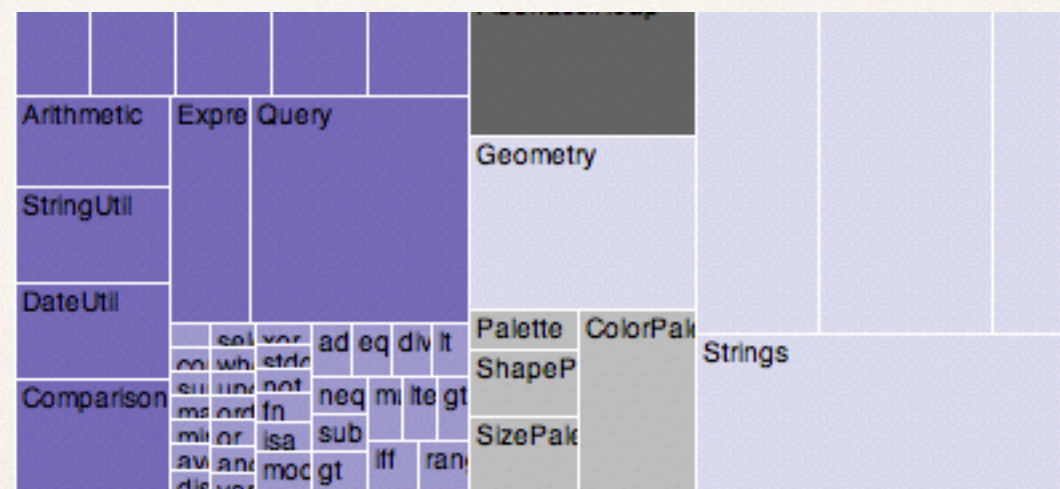
human-computer interaction + design lab.

Joonhwan Lee

Treemaps

Treemap Layout

- ✦ Documentation: <https://github.com/mbostock/d3/wiki/Treemap-Layout>
- ✦ Treemap을 그리기 위한 레이아웃
 - ✦ Treemap: Ben Shneiderman (1991)



Treemap Layout

- ✦ 3 Steps
 - ✦ Setup layout
 - ✦ Load data
 - ✦ Call position calculation function (d3 provides)

Lab 1: Treemap

- ♦ Data: 셰익스피어스의 작품 목록
 - ♦ http://www.opensourceshakespeare.org/views/plays/plays_numwords.php
 - ♦ Play's name, Genre, Word count

Lab 1: Treemap

```
* {
  "name": "Shakespeare",
  "children": [
    {
      "name": "Comedies",
      "children": [
        {"name": "All's Well That Ends Well", "size": 23009},
        {"name": "As You Like It", "size": 21690},
        {"name": "Comedy of Errors", "size": 14701}
      ]
    },
    {
      "name": "Histories",
      "children": [
        {"name": "Henry IV, Part I", "size": 24579},
        {"name": "Henry IV, Part II", "size": 25689},
        {"name": "Henry V", "size": 24629},
        {"name": "Henry VI, Part I", "size": 21607},
        {"name": "Henry VI, Part II", "size": 25439},
        {"name": "Richard III", "size": 29278}
      ]
    }
  ]
}
```

Lab 1: Treemap

♦ Step1: Setup Layout

```
var margin = {top: 40, right: 10, bottom: 10,  
              left: 10},  
width = 960 - margin.left - margin.right,  
height = 500 - margin.top - margin.bottom
```

```
var color = d3.scale.category20()
```

```
var treemap = d3.layout.treemap()  
  .size([width, height])  
  .sticky(true)  
  .value(function(d) { return d.size })
```

Lab 1: Treemap

♦ Step1: Setup Layout

```
var margin = {top: 40, right: 10, bottom: 10,  
              left: 10},  
width = 960 - margin.left - margin.right,  
height = 500 - margin.top - margin.bottom
```

```
var color = d3.scale.category20()
```

```
var treemap = d3.layout_treemap()
```

```
  .size([width, height])
```

```
  .sticky(true)
```

```
  .value(function(d)
```

Treemap의 각 cell의
위치를 트랜지션 시에 유지

Lab 1: Treemap

✦ Step1 (cont): Setup Layout

```
var div = d3.select("body").append("div")
    .style("position", "relative")
    .style("width", (width + margin.left +
        margin.right) + "px")
    .style("height", (height + margin.top +
        margin.bottom) + "px")
    .style("left", margin.left + "px")
    .style("top", margin.top + "px")
```

(참고) http://www.w3schools.com/css/css_positioning.asp

Lab 1: Treemap

♦ Step2: Loading data

```
d3.json("data/tree.json", function(error, root) {  
  if (error) throw error;
```

```
  var node = div.datum(root).selectAll(".node")  
    .data(treemap.nodes)  
    .enter().append("div")  
    .attr("class", "node")  
    .call(position)  
    .style("background", function(d) {  
      return d.children ? color(d.name) : null  
    })  
    .text(function(d) {  
      return d.children ? null : d.name +  
        " (" + d.size + ")"  
    })  
  })
```

Lab 1: Treemap

- ✦ Ternary Operator

- ✦ [https://en.wikipedia.org/wiki/%3F:](https://en.wikipedia.org/wiki/%3F)

- ✦ `var y = (x == 2 ? "yes" : "no");`

- ✦ 조건문을 확인한 후, true 면 왼쪽변의 값을, false 면 오른쪽의 값을 출력

Lab 1: Treemap

✦ Ternary Operator

```
.style("background", function(d) {  
    return d.children ? color(d.name) : null  
})  
.text(function(d) {  
    return d.children ? null : d.name +  
                                   " (" + d.size + ")"  
})
```

Lab 1: Treemap

- ✦ Step3: Call position calculation function

```
function position() {  
  this.style("left", function(d) {  
    return d.x + "px" })  
  .style("top", function(d) {  
    return d.y + "px" })  
  .style("width", function(d) {  
    return Math.max(0, d.dx - 1) + "px" })  
  .style("height", function(d) {  
    return Math.max(0, d.dy - 1) + "px" })  
}
```

Lab 1: Treemap

♦ Step4: Styling

```
body {  
    font-family: sans-serif;  
    margin: auto;  
    position: relative;  
    width: 960px;  
}  
  
.node {  
    border: solid 1px white;  
    font: 10px/12px sans-serif;  
    overflow: hidden;  
    position: absolute;  
    text-align: center;  
}
```

Lab 1: Treemap

- ♦ Step4 (cont): more styling

- ♦ 문자의 정렬을 cell 의 가운데에

```
function position() {  
    ...  
    .style("height", function(d) {  
        return Math.max(0, d.dy - 1) + "px" })  
    .style("line-height", function(d) {  
        return Math.max(0, d.dy - 1) + "px" })  
}  
  
.node {  
    ...  
    position:absolute;  
    text-align:center;  
}
```

Lab 1: Treemap

✦ Step5: count vs. size

- ✦ 각 node의 사이즈 별로 혹은 카운트 별로 cell의 크기 변환
- ✦ 사이즈: d.size, 카운트: 1 (각각의 셀 크기 동일)

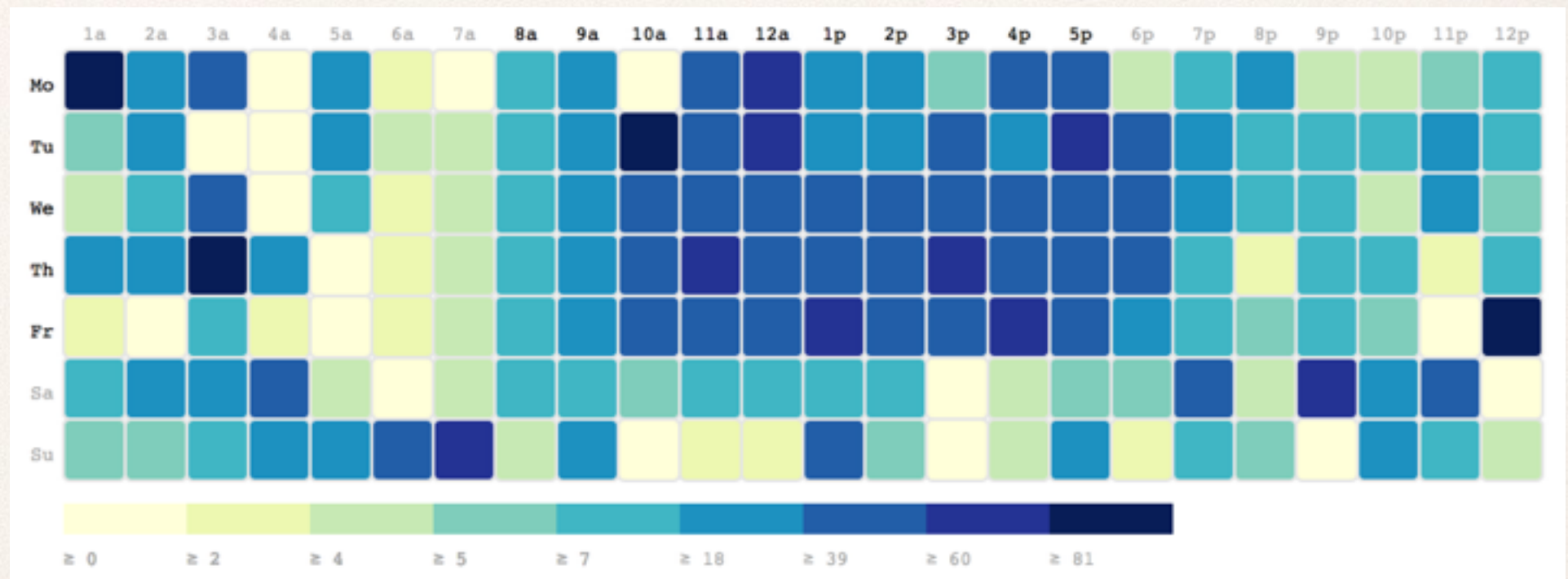
```
<form>
  <label><input type="radio" name="mode" value="size"
checked>Size</label>
  <label><input type="radio" name="mode" value="count">
Count</label>
</form>
```

```
d3.selectAll("input").on("change", function change() {
  var value = this.value == "count"
    ? function() { return 1 }
    : function(d) { return d.size }
  node.data(treemap.value(value).nodes)
    .transition()
    .duration(1500)
    .call(position)
})
```

Heatmaps

Heatmaps

- ♦ Heatmap은 시간에 따라 변화하는 패턴을 한눈에 파악하기 위해 사용함.



<http://bl.ocks.org/tjdecke/5558084>

Lab2: Heatmap

✦ Step1: Setup

- ✦ `gridSize = Math.floor(width / 24)`
- ✦ `legendElementWidth = gridSize*2`
- ✦ `colors = ["#ffffd9", "#edf8b1", ...`
- ✦ `days = ["Mo", "Tu", "We", ...`
- ✦ `times = ["1a", "2a", "3a", "4a", "5a", ...`
- ✦ `datasets = ["data/data1.tsv", "data/
data2.tsv"]`

Lab2: Heatmap

✦ Step2: Add Labels

```
var dayLabels = svg.selectAll(".dayLabel")
    .data(days)
    .enter().append("text")
    .text(function (d) { return d })
    .attr("x", 0)
    .attr("y", function (d, i) {
        return i * gridSize })
    .style("text-anchor", "end")
    .attr("transform", "translate(-6," +
        gridSize / 1.5 + ")")
    .attr("class", function (d, i) {
        return ((i >= 0 && i <= 4) ?
            "dayLabel mono axis axis-workweek" :
            "dayLabel mono axis") })
```

Lab2: Heatmap

- ◆ Step3: heatmapChart 메서드
 - ◆ 두개의 데이터를 번갈아 보여주기 위해서 데이터를 그리는 부분을 함수로 만들어 사용
 - ◆ `heatmapChart(datasets[0])` 등의 형식으로 사용
 - ◆ 이후에 버튼을 만들고 두개의 데이터를 `heatmapChart()`에 넘겨 사용

Lab2: Heatmap

✦ Step3 (cont):

```
cards.enter().append("rect")
    .attr("x", function(d) {
        return (d.hour - 1) * gridSize })
    .attr("y", function(d) {
        return (d.day - 1) * gridSize })
    .attr("rx", 4)
    .attr("ry", 4)
    .attr("class", "hour bordered")
    .attr("width", gridSize)
    .attr("height", gridSize)
    .style("fill", colors[0])
```

```
cards.transition().duration(1000)
    .style("fill", function(d) {
        return colorScale(d.value) })
```

Lab2: Heatmap

- ◆ Quantile Colorscale

- ◆

```
var colorScale = d3.scale.quantile()  
  .domain([0, buckets - 1, d3.max(data,  
    function (d) { return d.value } )])  
  .range(colors)
```

- ◆ domain: 값의 범위를 지정하면 `quantile()` 메서드가 자동으로 분위를 나누어 줌.

- ◆ <https://github.com/mbostock/d3/wiki/Quantitative-Scales#quantile-scales>

- ◆ <http://www.jeromecukier.net/blog/2011/08/11/d3-scales-and-color/>

Lab2: Heatmap

♦ Step4: Legend 추가

```
legend.append("rect")
    .attr("x", function(d, i) {
        return legendElementWidth * i })
    .attr("y", height)
    .attr("width", legendElementWidth)
    .attr("height", gridSize / 2)
    .style("fill", function(d, i) {
        return colors[i] })
```

Lab2: Heatmap

✦ Step5: 데이터 용 버튼 추가

```
var datasetpicker = d3.select("#dataset-  
picker").selectAll(".dataset-button")  
  .data(datasets)
```

```
datasetpicker.enter()  
  .append("input")  
  .attr("value", function(d){  
    return "Dataset " + d })  
  .attr("type", "button")  
  .attr("class", "dataset-button")  
  .on("click", function(d) {  
    heatmapChart(d)  
  })
```

Lab2: Heatmap

♦ Step6: Styling

```
rect.bordered {
  stroke: #E6E6E6;
  stroke-width:2px;
}
text.mono {
  font-size: 9pt;
  font-family: Consolas, courier;
  fill: #aaa;
}
text.axis-workweek {
  fill: #000;
}
text.axis-worktime {
  fill: #000;
}
```

Assignment 6: Treemap and/or Heatmap

- 제출: 11/15 (자정)

Assignment 6: Treemap and/or Heatmap

- ♦ 데이터: 자유롭게 선택
- ♦ Treemap and/or Heatmap 를 만들어 보자
- ♦ 제출: 다음주 일요일 (11/15) 자정
- ♦ 파일 제출 방법
 - ♦ 본인의 학번 뒷자리로 폴더를 만들고
 - ♦ html 파일은 반드시 index.html 로 이름을 부여한 후, 관련 파일을 모두 같은 폴더에 넣고 (주의: 한글 파일 이름 x)
 - ♦ zip 으로 압축해서 제출

Assignment 6: Treemap and/or Heatmap

♦ 참고 사이트

- ♦ Zoomable Treemap: <http://mbostock.github.io/d3/talk/20111018/treemap.html>
- ♦ Calendar View: <http://bl.ocks.org/mbostock/4063318>
- ♦ Journals: <http://neuralengr.com/asifr/journals/>
- ♦ Treemap with Headers: <http://www.billdwhite.com/wordpress/2012/12/16/d3-treemap-with-title-headers/>
- ♦ Visualization of Beijing Air Pollution: <http://scottcheng.github.io/bj-air-vis/>
- ♦ Circular Heat Chart: <http://prcweb.co.uk/lab/circularheat/>

Questions...?
