

# 빅데이터 분석 결과 시각화

## 인터랙티브(상호작용) 시각화 만들기

## 학습내용

- 상호작용 개념 익히기
- 마우스 이벤트 시각화
- 인터랙티브(상호작용) 시각화

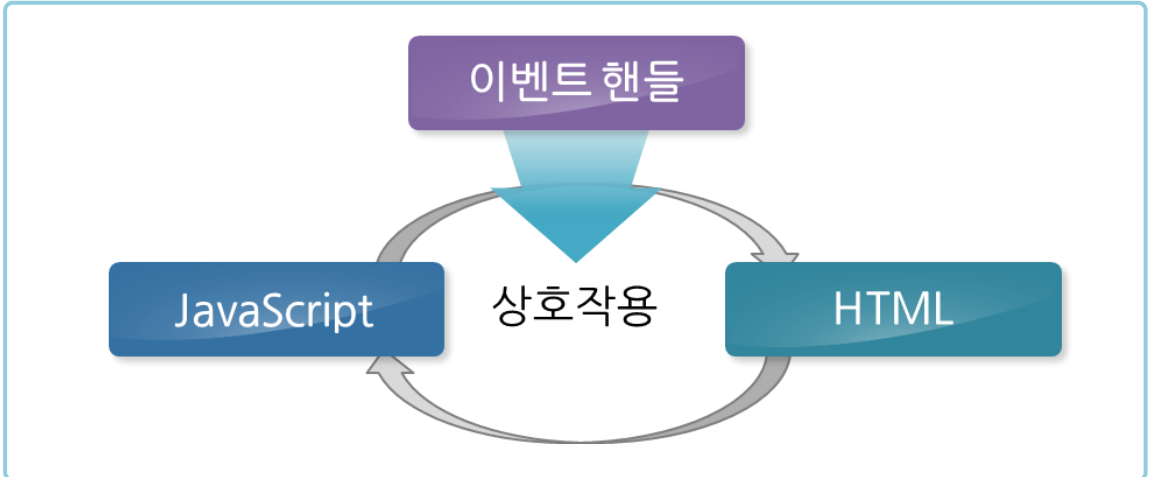
## 학습목표

- 상호작용 그래프를 시각화 하는 방법을 설명할 수 있다.
- 도넛 그래프에서 마우스 이벤트를 시각화할 수 있다.
- CSV 데이터를 활용하여 인터랙티브 시각화할 수 있다.

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ 기본 원형 차트 정의



- 이벤트를 일으키는 주체 : 사용자
- 이벤트 발생 조건
  - 브라우저에서 무언가를 클릭할 때
  - 페이지가 로딩될 때
  - 브라우저 화면이 닫힐 때
  - 브라우저 사이즈가 변할 때
- 이벤트의 발생
  - Document Object Model(DOM) Level 3
  - 모든 HTML 요소에서 발생함
- HTML DOM Event
  - 이벤트 핸들러로 HTML 요소를 제어 할 수 있음
  - 보통 일반적인 함수와 함께 사용함

## ● 상호작용 개념 익히기

---

### 1. 마우스 이벤트

#### ◆ 기본 원형 차트 정의

- Mouse Events
- Keyboard Events
- Frame/Object Events
- Form Events
- Event Object
- EventTarget Object
- EventListener Object
- DocumentEvent Object
- MouseEvent / KeyboardEvent Object

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ 기본 원형 차트 정의

- onclick 이벤트 타입
  - 마우스 왼쪽 버튼을 누르면 바로 onclick 인데 모든 HTML 태그에 포함되어 있음
  - ‘누르시오’라는 버튼을 클릭  
→pressBtn() 함수를 호출해서 함수 내에 있는 소스 실행

```
<html>
<head>
<script type="text/javascript">
<!--
function pressBtn() {
    alert("버튼을 누르셨군요.")
}
//-->

</script>
</head>
<body>
<input type="button" onclick="pressBtn()" value="누르시오" />
</body>
</html>
```

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ 기본 원형 차트 정의

- onmouseover, onmouseout 이벤트 타입

onmouseover	onmouseout
<ul style="list-style-type: none"><li>• 마우스를 가져갔을 때 이벤트가 실행됨</li></ul>	<ul style="list-style-type: none"><li>• 마우스가 태그 위를 빠져나갔을 때 이벤트가 실행됨</li></ul>

- div 태그 위에 마우스 오버 : over() 함수 실행
- 마우스가 태그 위를 빠져나감 : out()함수 실행

```
<html>
<head>
<script type="text/javascript">
<!--
function over() {
    alert("Mouse Over");
}
function out() {
    alert("Mouse Out");
}
//-->
</script>
</head>
<body>
<div onmouseover="over()" onmouseout="out()">
<h2> 이 곳에 마우스를 올려보세요.</h2>
</div>
```

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ 표준 HTML5 마우스 이벤트 자료

- onmouseover, onmouseout 이벤트 타입

Event	Value	Description
onchange	script	요소가 변경될 때 실행
onsubmit	script	양식이 제출될 때 실행
onreset	script	폼이 재설정될 때 실행
onselect	script	값이 선택되었을 때 실행
onblur	script	태그에 포커스를 잃었을 때 실행
onfocus	script	태그에 포커스가 갔을 때 실행
onkeydown	script	키를 눌렀을 때 실행
onkeypress	script	키를 눌렀다가 놓았을 때 실행
onkeyup	script	키를 놓았을 때 실행
onclick	script	마우스를 클릭했을 때 실행
ondblclick	script	마우스를 더블 클릭했을 때 실행
onmousedown	script	마우스 버튼을 누르고 있을 때 실행
onmousemove	script	마우스 포인트를 이동했을 때 실행
onmouseout	script	마우스 포인트가 요소의 바깥으로 이동했을 때 실행
onmouseover	script	마우스 포인트가 요소로 들어왔을 때 실행
onmouseup	script	마우스를 놓았을 때 실행

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ D3.js selection.on(type[,listener])

- 마우스 오버하면 사각형에 파란색이 표시되고 마우스를 밖으로 빼면 흰색으로 표시됨

```
.on("mouseover", function(){d3.select(this).style("fill", "blue");})  
.on("mouseout", function(){d3.select(this).style("fill", "white");});
```



mouseout



mouseover



## ● 상호작용 개념 익히기

---

### 1. 마우스 이벤트

#### ◆ D3.js 마우스 오버 시 이벤트

```
<div id="mouseEvent_sample"></div>
<script type="text/javascript">
  var sampleSVG = d3.select("#mouseEvent_sample")
  .append("svg:svg")
  .attr("width", 200)
  .attr("height", 200);
  sampleSVG.append("svg:rect")
  .style("stroke", "red")
  .style("fill", "white")
  .attr("x", 50) .attr("y", 50) .attr("width", 100) .attr("height", 100)
  .on("mouseover", function(){d3.select(this).style("fill", "blue");})
  .on("mouseout", function(){d3.select(this).style("fill", "white");});

</script>
```

## ● 상호작용 개념 익히기

### 1. 마우스 이벤트

#### ◆ D3.js 클릭 이벤트

- D3.js에서 사각형 이미지 클릭 시 7가지 컬러 중에 난수 값을 발생시켜 색상이 다르게 표현됨

```
var source = ["blue", "black", "orange", "yellow", "green", "violet", "pink"];  
var color = source[Math.floor(Math.random()*source.length)];
```

```
.on("click", function(){d3.select(this).style("fill", function() {return  
source[Math.floor(Math.random()*source.length)];});})  
.on("mouseout", function(){d3.select(this).style("fill", "white");});
```



mouseout



mouseclick

- 클릭할 때마다 7가지 컬러가 랜덤하게 나타남

## ● 상호작용 개념 익히기

---

### 2. 상호작용 실습

#### ◆ 상호작용 실습 순서

- ① HTML 소스 에서 D3.js 스크립트 불러오기
- ② HTML 소스에서 컬러지정하기
- ③ 마우스 오버 시 색상 값이 변하도록 설정하기
- ④ 마우스 클릭 시 색상 값이 변하도록 설정하기
- ⑤ 마우스 이벤트 결과 화면 확인하기

## ● 마우스 이벤트 시각화

### 1. 기본 원형 차트 정의

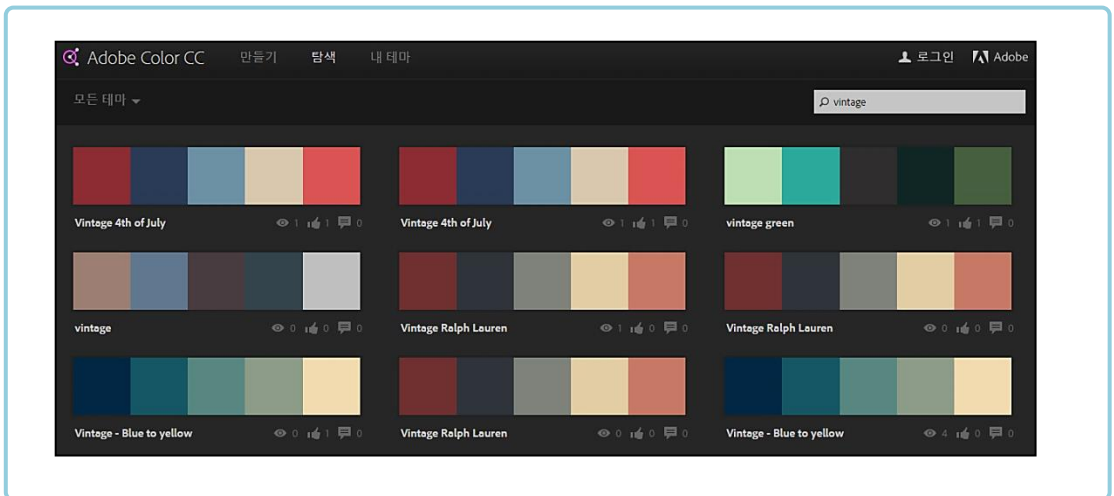
#### ◆ 기본 데이터 설정

- 데이터 셋에 데이터를 입력함

```
var dataset = [  
  { label: '초콜릿', count: 10 },  
  { label: '캔디', count: 20 },  
  { label: '과자', count: 30 },  
  { label: '아이스크림', count: 40 }  
];
```

#### ◆ 파이 차트 색상 코디

- 파이 차트에 들어가는 컬러를 지정하기 위해 어도비 사이트의 컬러 테마를 선택함



## ● 마우스 이벤트 시각화

### 1. 기본 원형 차트 정의

#### ◆ 파이 그리기

- 원의 크기를 결정하며 원의 반지름을 설정함

```
var width = 400;  
var height = 400;  
var radius = Math.min(width, height) / 2;
```

- D3에서 제공하는 컬러를 사용함

```
var color = d3.scaleOrdinal(d3.schemeCategory20b);  
  
#393b79#5254a3#6b6ecf#9c9ede#637939#8ca252#b5cf6b#cedb9c#  
8c6d31#bd9e39#e7ba52#e7cb94  
  
#843c39#ad494a#d6616b#e7969c#7b4173#a55194#ce6dbd#de9ed6  
#393b79#5254a3
```

- D3에서 4개의 내장 컬러 팔레트가 있음

- d3.scale.category10 ()
- d3.scale.category20 ()
- d3.scale.category20b ()
- d3.scale.category20c ()

- 컬러를 자신이 선택한 값으로 만들 수 있음

```
// 자신이 선택한 컬러의 크기값 설정  
var color = d3.scaleOrdinal()  
.range(['#F9EFD', '#242428', '#5E6050', '#A78849', '#6E553C']);
```

- 푸른색과 초록색, 갈색의 3가지 컬러로 20개 구성

## ● 마우스 이벤트 시각화

---

### 1. 기본 원형 차트 정의

#### ◆ SVG 차트 범위 만들기

- 전체 차트의 크기를 반경으로 나누어 중심점을 결정함

```
var svg = d3.select('#chart')
    .append('svg')
    .attr('width', width)
    .attr('height', height)
    .append('g')
    .attr('transform', 'translate(' + (width / 2) + ', ' + (height / 2) + ')');
```

#### ◆ 반지름 설정 및 파이 그리기

- 반지름 설정

```
var arc = d3.arc()
    .innerRadius(0)
    .outerRadius(radius);
```

- 파이 그래프 그리기

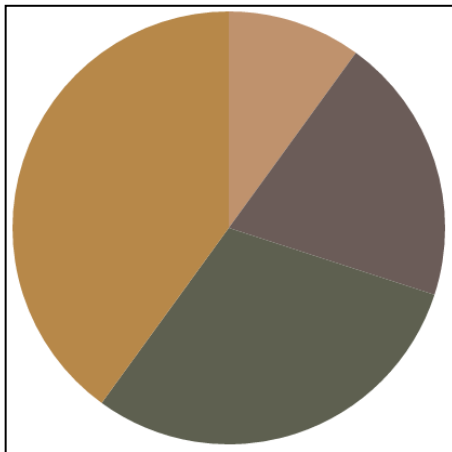
```
var pie = d3.pie()
    .value(function(d) { return d.count; })
    .sort(null);
```

## ● 마우스 이벤트 시각화

### 1. 기본 원형 차트 정의

#### ◆ 파이 그래프 데이터 세팅하기

- 파이 그래프에 색상 및 데이터를 셋팅함



```
var path = svg.selectAll('path')
.data(pie(dataset))
.enter()
.append('path')
.attr('d', arc)
.attr('fill', function(d, i) {
return color(d.data.label);
});
```

## ● 마우스 이벤트 시각화

---

### 2. 도넛 그래프 실습

#### ◆ 도넛 그래프 실습 순서

- ① 데이터 준비하기
- ② CSV 외부 데이터 불러오기
- ③ 도넛 그래프 그리기
- ④ mouseover, mouseout 스크립트 작성하기
- ⑤ 마우스 이벤트 결과 확인하기



## ● 인터랙티브(상호작용) 시각화

---

### 1. 애니메이션 상호작용

#### ◆ 애니메이션 상호작용 실습 순서

- ① CSS 차트 크기 및 사각 체크 설정하기
- ② CSV 파일 불러내기
- ③ 마우스 클릭 시 이벤트 발생
- ④ 파이 그래프 애니메이션 넣기
- ⑤ 결과 화면 확인하기

### 2. 인터랙티브(상호작용) 파이그래프 실습

#### ◆ 인터랙티브(상호작용) 파이그래프 실습 순서

- ① 그래프 변경하기
- ② 전체 그래프 사이즈 설정하기
- ③ 결과 화면 확인하기

### 1. 상호작용 개념 익히기

#### ■ 마우스 이벤트 정의

- JavaScript와 HTML의 상호작용은 이벤트 핸들에 의해 이루어짐
- 이벤트를 일으키는 주체 : 사용자
- 이벤트의 발생
  - ① 브라우저에서 무언가를 클릭하는 경우
  - ② 페이지가 로딩될 때
  - ③ 브라우저 화면이 닫힐 때
  - ④ 브라우저 사이즈가 변할 때
- 이벤트는 Document Object Model(DOM) Level 3와 모든 HTML 요소에서 가지고 있음
- HTML DOM Event는 이벤트 핸들러로 HTML 요소를 제어 할 수 있으며, 보통 일반적인 함수와 함께 사용함

### 1. 상호작용 개념 익히기

#### ■ 마우스 이벤트 종류

##### ■ onclick 이벤트 타입

- 마우스 왼쪽 버튼을 누르면 바로 onclick 인데 모든 HTML 태그에 포함되어 있음
- 다음 예제는 ‘누르시오’라는 버튼을 클릭하게 되면 pressBtn() 함수를 호출해서 함수 내에 있는 소스를 실행하게 됨

##### ■ onmouseover, onmouseout 이벤트 타입

- onmouseover : 마우스를 가져갔을 때 이벤트가 실행됨
- onmouseout : 마우스가 태그 위를 빠져나갔을 때 실행됨
- 다음 코드는 div 태그 위에 마우스가 가면 over() 함수가 실행되고 떠나게 되면 out()이 실행됨

```
<html>
<head>
<script type="text/javascript">
```

### 1. 상호작용 개념 익히기

#### ■ 표준 HTML5 마우스 이벤트 자료

- onchange : 요소가 변경될 때 실행
- onsubmit : 양식이 제출될 때 실행
- onreset : 폼이 재설정될 때 실행
- onselect : 값이 선택되었을 때 실행
- onblur : 태그에 포커스를 잃었을 때 실행
- onfocus : 태그에 포커스가 갔을 때 실행
- onkeydown : 키를 눌렀을 때 실행
- onkeypress : 키를 눌렀다가 놓았을 때 실행
- onkeyup : 키를 놓았을 때 실행
- onclick : 마우스를 클릭했을 때 실행
- ondblclick : 마우스를 더블 클릭했을 때 실행
- onmousedown : 마우스 버튼을 누르고 있을 때 실행
- onmousemove : 마우스 포인터를 이동했을 때 실행
- onmouseout : 마우스 포인트가 요소의 바깥으로 이동했을 때 실행
- onmouseover : 마우스 포인트가 요소로 들어왔을 때 실행
- onmouseup : 마우스를 놓았을 때 실행

### 1. 상호작용 개념 익히기

#### ■ D3.js selection.on(type[,listener])

- 마우스 오버하면 사각형에 파란색이 표시되고 마우스를 밖으로 빼면 흰색으로 표시됨

#### ■ D3.js 마우스 오버 시 이벤트

- D3.js에서 마우스 오버 시 이벤트가 발생하는 JavaScript 소스임

#### ■ D3.js 클릭 이벤트

- D3.js에서 사각형 이미지 클릭 시 7가지 컬러 중에 난수 값을 발생하여 색상이 다르게 표현됨

### 2. 마우스 이벤트 시각화

#### ■ 기본 데이터 설정

- 데이터 셋에 데이터를 입력함

#### ■ 파이 차트 색상 코디

- 파이 차트에 들어가는 컬러를 지정하기 위해 어도비 사이트의 컬러 테마를 선택함

#### ■ 파이 그리기

- 원의 크기를 결정하며 원의 반지름을 만듦
- D3에서 제공하는 컬러를 사용함
- 컬러를 자신이 선택한 값으로 만들 수 있음

#### ■ SVG 차트 범위 만들기

- 전체 차트의 크기를 반경으로 나누어 중심점을 결정함

#### ■ 파이 그래프 데이터 세팅하기

- 파이 그래프에 색상 및 데이터를 세팅함

### 3. CSV 그래프 그리기

#### ■ .rangeBands 다루기 실습 시 유의사항 작성

- 치역에는 반올림한 대역폭을 사용함
- 치역을 지정할 때, 모든 값을 직접 지정해야 하는 range()를 사용하거나  
균등하게 알아서 분할해 주는 rangeBand()를 사용할 수도 있음
- rangeBand()는 치역의 양 끝 점을 전달인자로 받음