

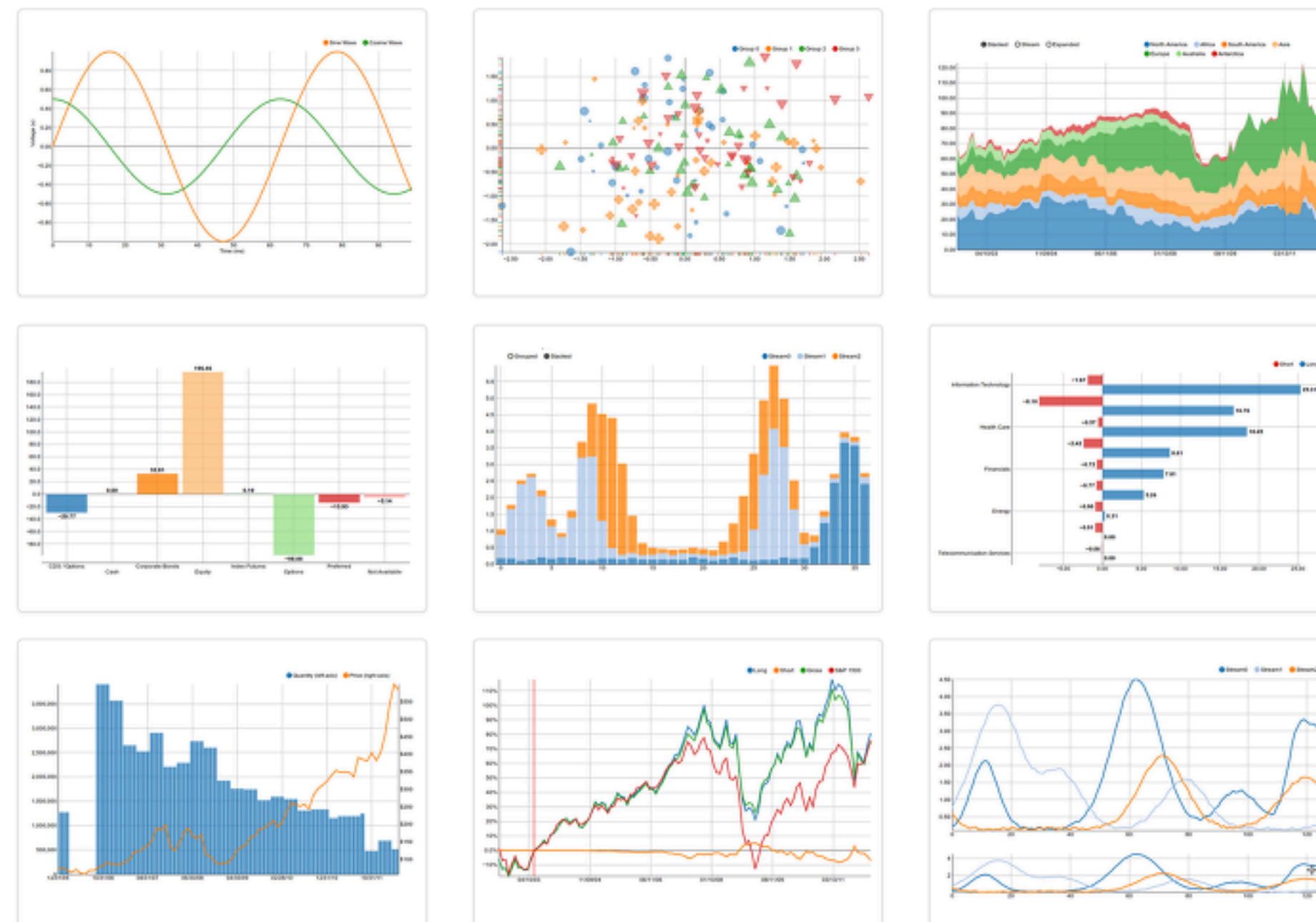
D3.js 스터디

2주차

이지호

D3.js란?

- D3.js는 data를 바탕으로 document를 조작하는 javascript library
- 다양한 차트 제작이나 데이터 시각화에 많이 사용



D3.js 시작하기

- Python3의 기본 웹 서버(http.server) 위에 구동

```
File Edit View Search Terminal Help
marble@marble:~/dev/DB_study/s2$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
█
```

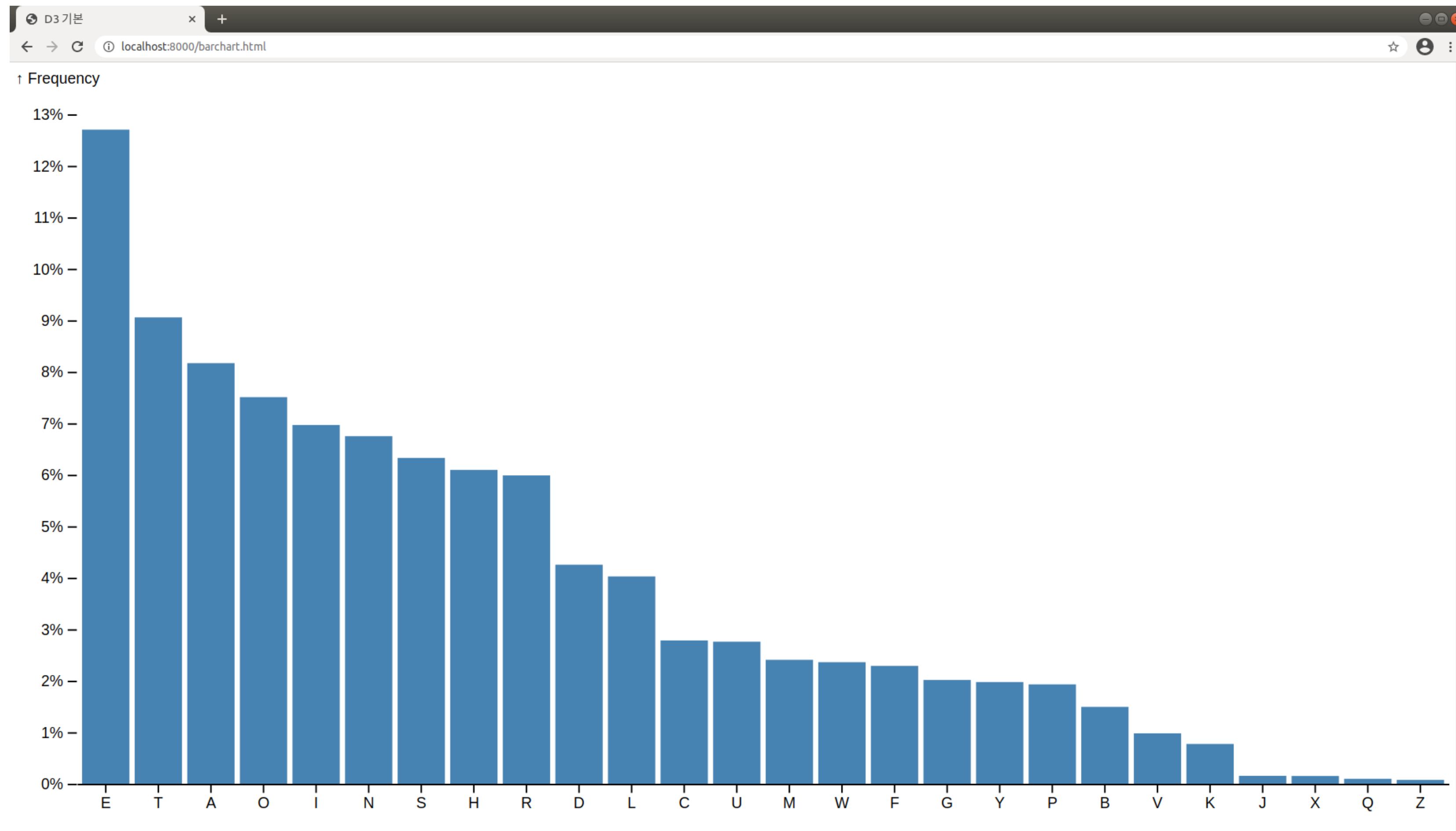
- CDN 방식으로 최신 버전 사용(v6.3.1)

```
<script src="https://d3js.org/d3.v6.min.js"></script>
```

- <https://observablehq.com/@d3/gallery> 사이트 참고하면서 예제 구현하는 방식으로 진행

Barchart

막대 그래프 그리기(<https://observablehq.com/@d3/bar-chart>)



Barchart

데이터(alphabet.csv)

```
letter,frequency
A,0.08167
B,0.01492
C,0.02782
D,0.04253
E,0.12702
F,0.02288
G,0.02015
H,0.06094
I,0.06966
J,0.00153
K,0.00772
L,0.04025
M,0.02406
N,0.06749
O,0.07507
P,0.01929
Q,0.00095
R,0.05987
S,0.06327
T,0.09056
U,0.02758
V,0.00978
W,0.0236
X,0.0015
Y,0.01974
Z,0.00074
```

Barchart

소스코드(1)

```
d3.csv("alphabet.csv", ({letter, frequency}) => ({name: letter, value: +frequency}))
  .then((d) => {
    const color = "steelblue";
    const margin = {
      top: 30,
      right: 0,
      bottom: 30,
      left: 40
    };
    const width = 942;
    const height = 500;

    const sorted_data = d.sort((a, b) => d3.descending(a.value, b.value));
    const data = Object.assign(sorted_data, {format: "%", y: "↑ Frequency"});
    console.log(data);

    x = d3.scaleBand()
      .domain(d3.range(data.length))
      .range([margin.left, width - margin.right])
      .padding(0.1);

    y = d3.scaleLinear()
      .domain([0, d3.max(data, d => d.value)]).nice()
      .range([height - margin.bottom, margin.top]);

    xAxis = g => g
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x).tickFormat(i => data[i].name).tickSizeOuter(0));

    yAxis = g => g
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y).ticks(null, data.format))
      .call(g => g.select(".domain").remove())
      .call(g => g.append("text")
        .attr("x", -margin.left)
        .attr("y", 10)
        .attr("fill", "currentColor")
        .attr("text-anchor", "start")
        .text(data.y));
```

Barchart

소스코드(2)

```
const svg = d3.select("svg")
  .attr("viewBox", [0, 0, width, height]);

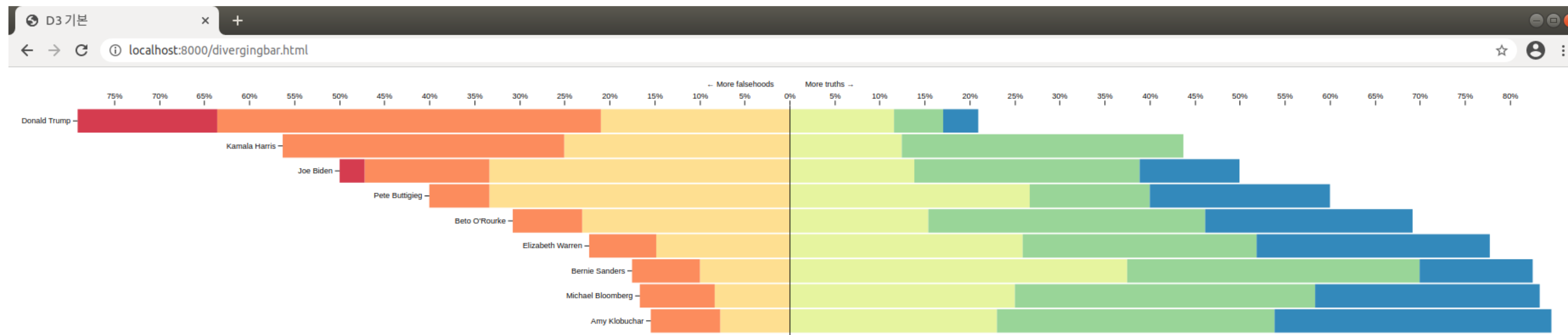
svg.append("g")
  .attr("fill", color)
  .selectAll("rect")
  .data(data)
  .join("rect")
  .attr("x", (d, i) => x(i))
  .attr("y", d => y(d.value))
  .attr("height", d => y(0) - y(d.value))
  .attr("width", x.bandwidth());

svg.append("g")
  .call(xAxis);

svg.append("g")
  .call(yAxis);
})
```

Diverging Stacked Bar Chart

막대 그래프 그리기(<https://observablehq.com/@d3/diverging-stacked-bar-chart>)



Diverging Stacked Bar Chart

데이터(politifact.csv)

```
speaker,ruling,count
Donald Trump,pants-fire,20
Donald Trump,half-true,15
Donald Trump,barely-true,27
Donald Trump,false,55
Donald Trump,mostly-true,7
Donald Trump,true,5
Bernie Sanders,mostly-true,13
Bernie Sanders,half-true,15
Bernie Sanders,barely-true,4
Bernie Sanders,true,5
Bernie Sanders,false,3
Joe Biden,mostly-true,9
Joe Biden,true,4
Joe Biden,pants-fire,1
Joe Biden,barely-true,12
Joe Biden,false,5
Joe Biden,half-true,5
Elizabeth Warren,false,2
Elizabeth Warren,barely-true,4
Elizabeth Warren,mostly-true,7
Elizabeth Warren,true,7
Elizabeth Warren,half-true,7
Kamala Harris,barely-true,4
Kamala Harris,false,5
Kamala Harris,mostly-true,5
Kamala Harris,half-true,2
Kamala Harris,full-flop,1
Pete Buttigieg,barely-true,5
Pete Buttigieg,half-true,4
Pete Buttigieg,true,3
Pete Buttigieg,false,1
Pete Buttigieg,mostly-true,2
Amy Klobuchar,barely-true,1
Amy Klobuchar,true,4
Amy Klobuchar,half-true,3
Amy Klobuchar,mostly-true,4
Amy Klobuchar,false,1
Beto O'Rourke,false,1
Beto O'Rourke,true,3
Beto O'Rourke,barely-true,3
Beto O'Rourke,mostly-true,4
Beto O'Rourke,half-true,2
Michael Bloomberg,half-true,3
Michael Bloomberg,true,3
Michael Bloomberg,mostly-true,4
Michael Bloomberg,barely-true,1
Michael Bloomberg,false,1
```

Diverging Stacked Bar Chart

소스코드(1)

```
const margin = ({top: 40, right: 30, bottom: 0, left: 80});
const height = 337;
const width = 2000;

const categories = {
  "pants-fire": "Pants on fire!",
  "false": "False",
  "mostly-false": "Mostly false",
  "barely-true": "Mostly false", // recategorized
  "half-true": "Half true",
  "mostly-true": "Mostly true",
  "true": "True"
};

d3.csv("politifact.csv", ({speaker: name, ruling: category, count: value}) => categories[category] ? {name, category: categories[category], value: +value} : null)
  .then((org_data) => {
    d3.rollup(org_data, group => {
      const sum = d3.sum(group, d => d.value);
      for (const d of group) d.value /= sum;
    }, d => d.name);

    const data = Object.assign(org_data, {
      format: ".0%",
      negative: "← More falsehoods",
      positive: "→ More truths",
      negatives: ["Pants on fire!", "False", "Mostly false"],
      positives: ["Half true", "Mostly true", "True"]
    });

    const signs = new Map([].concat(
      data.negatives.map(d => [d, -1]),
      data.positives.map(d => [d, +1])
    ));

    const bias = d3.rollups(data, v => d3.sum(v, d => d.value * Math.min(0, signs.get(d.category))), d => d.name)
      .sort(([, a], [, b]) => d3.ascending(a, b));

    const series = d3.stack()
      .keys([].concat(data.negatives.slice().reverse(), data.positives))
      .value([, value], category) => signs.get(category) * (value.get(category) || 0)
      .offset(d3.stackOffsetDiverging)
      (d3.rollups(data, data => d3.rollup(data, ([d]) => d.value, d => d.category), d => d.name));
```

Diverging Stacked Bar Chart

소스코드(2)

```
const x = d3.scaleLinear()
  .domain(d3.extent(series.flat(2)))
  .rangeRound([margin.left, width - margin.right]);

const y = d3.scaleBand()
  .domain(bias.map([name] => name))
  .rangeRound([margin.top, height - margin.bottom])
  .padding(2 / 33);

const color = d3.scaleOrdinal()
  .domain([].concat(data.negatives, data.positives))
  .range(d3.schemeSpectral[data.negatives.length + data.positives.length]);

const xAxis = g => g
  .attr("transform", `translate(0,${margin.top})`)
  .call(d3.axisTop(x)
    .ticks(width / 80)
    .tickFormat(formatValue)
    .tickSizeOuter(0))
  .call(g => g.select(".domain").remove())
  .call(g => g.append("text")
    .attr("x", x(0) + 20)
    .attr("y", -24)
    .attr("fill", "currentColor")
    .attr("text-anchor", "start")
    .text(data.positive))
  .call(g => g.append("text")
    .attr("x", x(0) - 20)
    .attr("y", -24)
    .attr("fill", "currentColor")
    .attr("text-anchor", "end")
    .text(data.negative));

const yAxis = g => g
  .call(d3.axisLeft(y).tickSizeOuter(0))
  .call(g => g.selectAll(".tick").data(bias).attr("transform", ([name, min]) => `translate(${x(min)},${y(name) + y.bandwidth() / 2})`))
  .call(g => g.select(".domain").attr("transform", `translate(${x(0)},0)`));

const formatValue = x => {
  const format = d3.format(data.format || "");
  return format(Math.abs(x));
}
```

Diverging Stacked Bar Chart

소스코드(3)

```
const svg = d3.select("svg")
  .attr("viewBox", [0, 0, width, height]);

svg.append("g")
  .selectAll("g")
  .data(series)
  .join("g")
  .attr("fill", d => color(d.key))
  .selectAll("rect")
  .data(d => d.map(v => Object.assign(v, {key: d.key})))
  .join("rect")
  .attr("x", d => x(d[0]))
  .attr("y", ({data: [name]} => y(name))
  .attr("width", d => x(d[1]) - x(d[0]))
  .attr("height", y.bandwidth())
  .append("title")
  .text(({key, data: [name, value]}) => `${name}
    ${formatValue(value.get(key))} ${key}`);

svg.append("g")
  .call(xAxis);

svg.append("g")
  .call(yAxis);

})
```

The Impact of Vaccines

막대 그래프 응용(<https://observablehq.com/@mbostock/the-impact-of-vaccines>)



The Impact of Vaccines

데이터(vaccines.json)

```
[{"id": "measles",
  "title": "Measles",
  "note": "Note: CDC data from 2003-2012 comes from its Summary of Notifiable Diseases, which publishes yearly rather than weekly and counts confirmed cases as opposed to provisional ones.",
  "data": [{"chart_options": {"vaccine_year": 1963},
    "values": [{"data": [{"year": 1928,
      "cases": 0,
      "label": null},
      {"year": 1928,
      "cases": 1,
      "label": "334.9621212"},
      {"year": 1928,
      "cases": 2,
      "label": "481.8083378"},
      {"year": 1928,
      "cases": 3,
      "label": "200.7109005"},
      {"year": 1928,
      "cases": 4,
      "label": "69.18974551"},
      {"year": 1928,
      "cases": 5,
      "label": "206.9526627"},
      {"year": 1928,
      "cases": 6,
      "label": "634.9714648"},
      {"year": 1928,
      "cases": 7,
      "label": "535.5949896"}]}]}]
```


The Impact of Vaccines

소스코드(1)

```
const margin = ({top: 20, right: 1, bottom: 40, left: 40});
const width = 2000;
const height = 16;
const innerHeight = 816;

const names = ["Alaska", "Ala.", "Ark.", "Ariz.", "Calif.", "Colo.", "Conn.", "D.C.", "Del.", "Fla.", "Ga.", "Hawaii", "Iowa", "Idaho", "Ill.", "Ind.", "Kan.", "Ky.", "La.", "Mass.", "Md.", "Maine", "Mich.", "Minn.", "Mo.", "Miss.", "Mont.", "N.C.", "N.D.", "Neb.", "N.H.", "N.J.", "N.M.", "Nev.", "N.Y.", "Ohio", "Okla.", "Ore.", "Pa.", "R.I.", "S.C.", "S.D.", "Tenn.", "Texas", "Utah", "Va.", "Vt.", "Wash.", "Wis.", "W.Va.", "Wyo."];

d3.json("vaccines.json")
  .then((data) => {
    const values = [];
    const year0 = d3.min(data[0].data.values.data, d => d[0]);
    const year1 = d3.max(data[0].data.values.data, d => d[0]);
    const years = d3.range(year0, year1 + 1);
    for (const [year, i, value] of data[0].data.values.data) {
      if (value == null) continue;
      (values[i] || (values[i] = []))[year - year0] = value;
    }

    data = {
      values,
      names,
      years,
      year: data[0].data.chart_options.vaccine_year
    };

    x = d3.scaleLinear()
      .domain([d3.min(data.years), d3.max(data.years) + 1])
      .rangeRound([margin.left, width - margin.right]);

    y = d3.scaleBand()
      .domain(data.names)
      .rangeRound([margin.top, margin.top + innerHeight]);

    color = d3.scaleSequentialSqrt([0, d3.max(data.values, d => d3.max(d))], d3.interpolatePuRd);
```

The Impact of Vaccines

소스코드(2)

```
xAxis = g => g
  .call(g => g.append("g")
    .attr("transform", `translate(0,${margin.top})`)
    .call(d3.axisTop(x).ticks(null, "d"))
    .call(g => g.select(".domain").remove()))
  .call(g => g.append("g")
    .attr("transform", `translate(0,${innerHeight + margin.top + 4})`)
    .call(d3.axisBottom(x)
      .tickValues([data.year])
      .tickFormat(x => x)
      .tickSize(-innerHeight - 10))
    .call(g => g.select(".tick text")
      .clone()
      .attr("dy", "2em")
      .style("font-weight", "bold")
      .text("Measles vaccine introduced"))
    .call(g => g.select(".domain").remove()));

yAxis = g => g
  .attr("transform", `translate(${margin.left},0)`)
  .call(d3.axisLeft(y).tickSize(0))
  .call(g => g.select(".domain").remove());

format = d => {
  const f = d3.format(",d");
  return isNaN(d) ? "N/A cases"
  : d === 0 ? "0 cases"
  : d < 1 ? "<1 case"
  : d < 1.5 ? "1 case"
  : `${f(d)} cases`;
};
```


The Impact of Vaccines

소스코드(3)

```
const svg = d3.select("svg")
  .attr("viewBox", [0, 0, width, innerHeight + margin.top + margin.bottom])
  .attr("font-family", "sans-serif")
  .attr("font-size", 10);

svg.append("g")
  .call(xAxis);

svg.append("g")
  .call(yAxis);

svg.append("g")
  .selectAll("g")
  .data(data.values)
  .join("g")
  .attr("transform", (d, i) => `translate(0,${y(data.names[i])})`)
  .selectAll("rect")
  .data(d => d)
  .join("rect")
  .attr("x", (d, i) => x(data.years[i]) + 1)
  .attr("width", (d, i) => x(data.years[i] + 1) - x(data.years[i]) - 1)
  .attr("height", y.bandwidth() - 1)
  .attr("fill", d => isNaN(d) ? "#eee" : d === 0 ? "#fff" : color(d))
  .append("title")
  .text((d, i) => `${format(d)} per 100,000 people in ${data.years[i]}`);
})
```