Information Visualization

W08: Creating Data Plot – Bar/Pie/Line/Area charts

Graduate School of System Informatics

Department of Computational Science

Naohisa Sakamoto Akira Kageyama

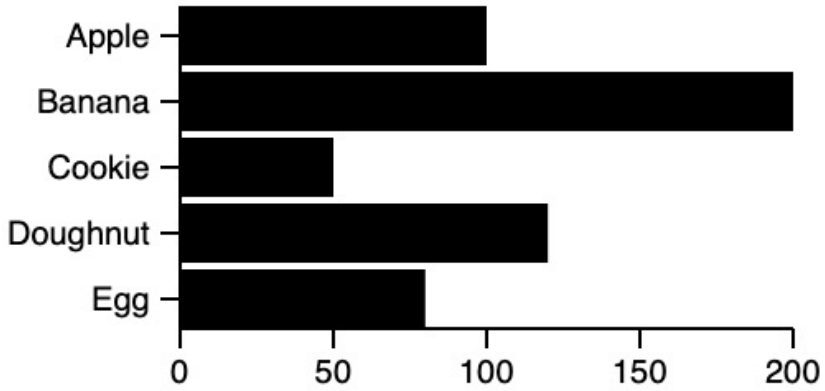
May 11, 2022

Schedule

- W01 4/12 Guidance
- W02 4/13 JavaScript Programming
- W03 4/19 Data and Tasks
- W04 4/20 Reading Data
- W05 4/26 Marks and Channels
- W06 4/27 Creating Data Plot - Scatter plot
- W07 5/10 Visualization Idioms
- W08 5/11 Creating Data Plot - Bar/Pie/Line/Area chars

Bar Chart

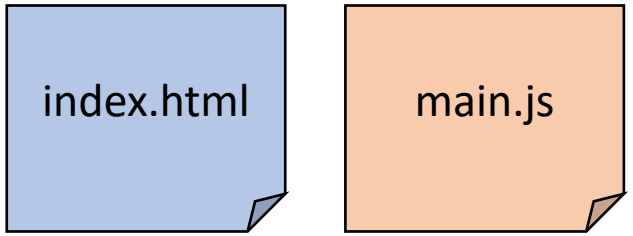
- **Drawing Bars**
  - Data
    - Label: categ. attrib.
    - Value: quant. attrib.




```
var data = [
    {label:'Apple', value:100},
    {label:'Banana', value:200},
    {label:'Cookie', value:50},
    {label:'Doughnut', value:120},
    {label:'Egg', value:80}
];
```

Example 01

- **Drawing Bars**
  - index.html
  - main.js

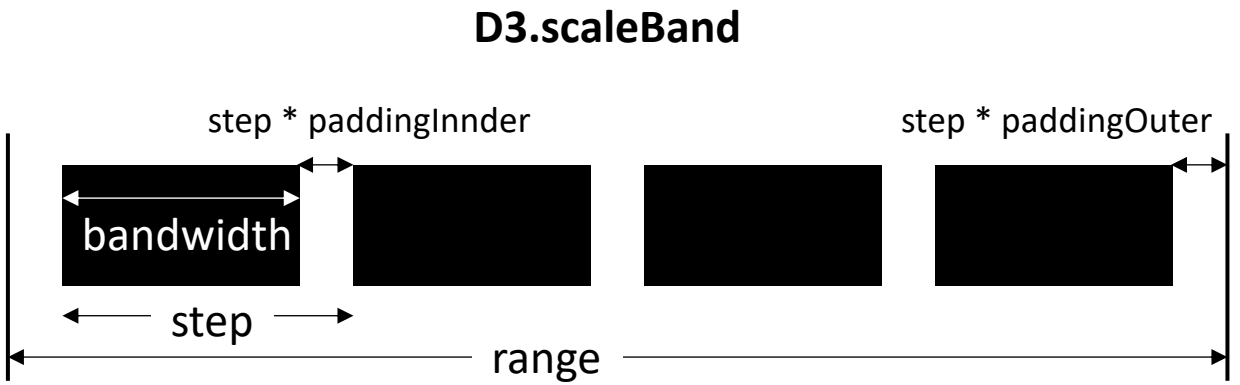


w08_ex01_index.html

```
<html>
        <head>
                <title>W08: Example 01</title>
        </head>
        <body>
                <svg id="drawing_region"></svg>
                <script src="https://d3js.org/d3.v6.min.js"></script>
                <script src="w08_ex01_main.js"></script>
        </body>
</html>
```

Example 01

- **Axis Scales**
  - X-axis (quant. attrib.)
    - D3. scaleLinear()
  - Y-axis (categ. attrib.)
    - D3.scaleBand()

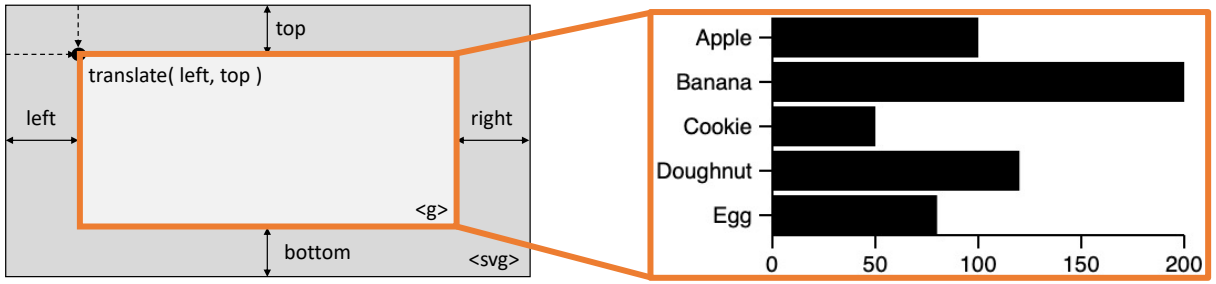


w08_ex01_main.js

```
const xscale = d3.scaleLinear()
      .domain([0, d3.max(data, d => d.value)])
      .range([0, inner_width]);

const yscale = d3.scaleBand()
      .domain(data.map(d => d.label))
      .range([0, inner_height])
      .paddingInner(0.1);
```

Example 01

- **Draw Bars**
  - Setting margins
    - {top, right, bottom, left}

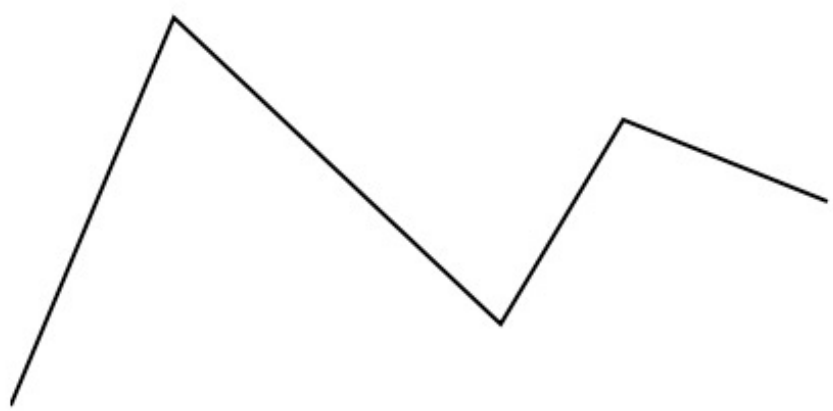


w08_ex01_main.js

```
var chart = svg.append('g')
    .attr('transform', `translate(${margin.left}, ${margin.top})`);
...

chart.selectAll("rect").data(data).enter()
    .append("rect")
    .attr("x", 0)
    .attr("y", d => yscale(d.label))
    .attr("width", d => xscale(d.value))
    .attr("height", yscale.bandwidth());
```

Line Chart

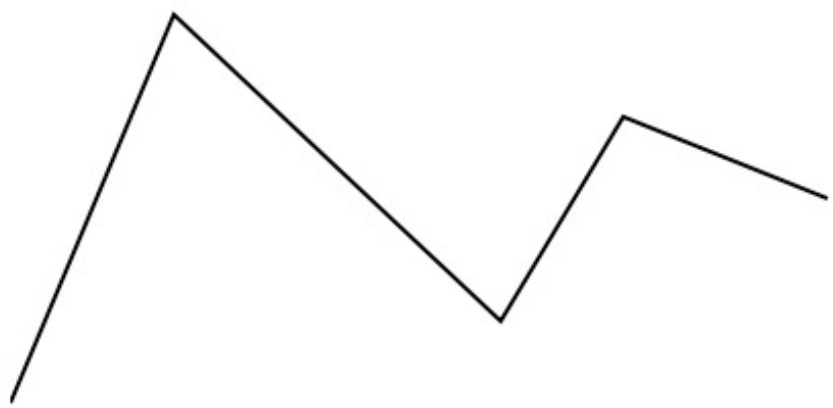
- **Drawing Line**
  - Data
    - x: quant. attrib.
    - y: quant. attrib.



```
var data = [
    {x:0, y:100},
    {x:40, y:5},
    {x:120, y:80},
    {x:150, y:30},
    {x:200, y:50}
];
```

Example 02

- **Drawing Line**
  - D3.line()

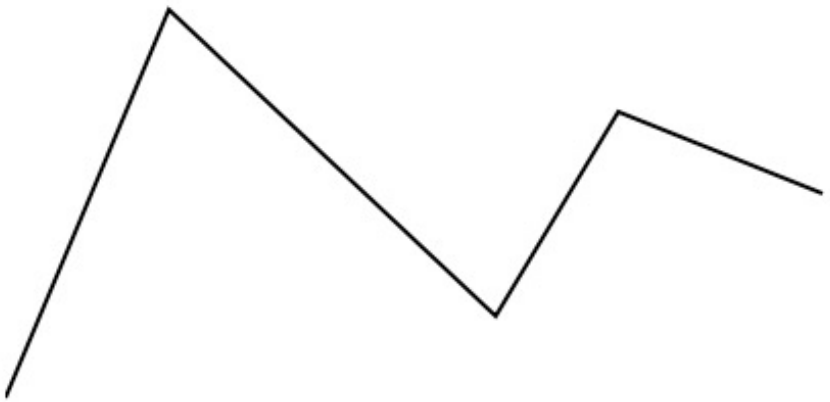


w08_ex02_main.js

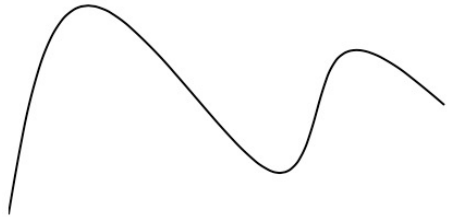
```
const line = d3.line()
      .x( d => d.x )
      .y( d => d.y );

svg.append('path')
    .attr('d', line(data))
    .attr('stroke', 'black')
    .attr('fill', 'none');
```

Curved Lines

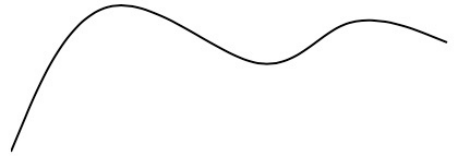
- **Drawing Line**
  - Curved lines



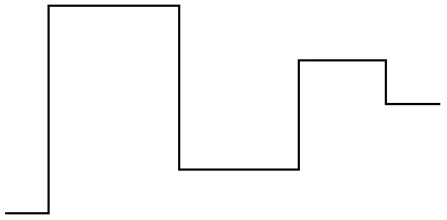
```
const line = d3.line()
      .x( d => d.x )
      .y( d => d.y )
      .curveLinear();
```



d3.curveNatural



d3.curveBasis



d3.curveStep

d3.curveBasisClosed

http://bl.ocks.org/d3indepth/b6d4845973089bc1012dec1674d3aff8

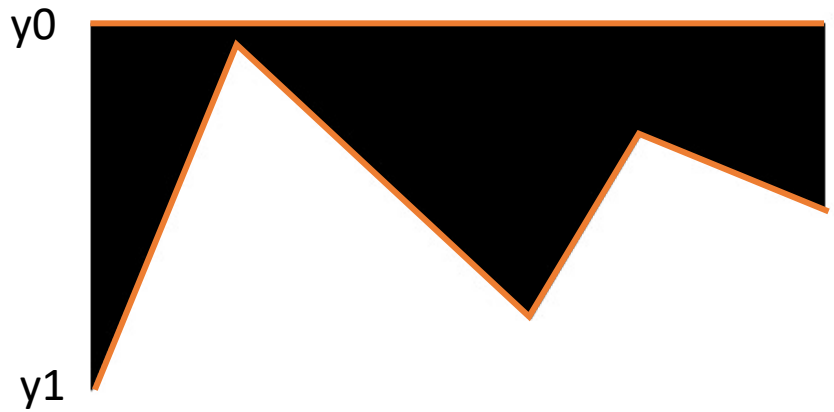
Area Chart

- **Drawing Area**
  - Data
    - x: quant. attrib.
    - y: quant. attrib.

```
var data = [
    {x:0, y:100},
    {x:40, y:5},
    {x:120, y:80},
    {x:150, y:30},
    {x:200, y:50}
];
```

Example 03

- **Drawing Area**
  - D3.area()
    - Defined by two polylines

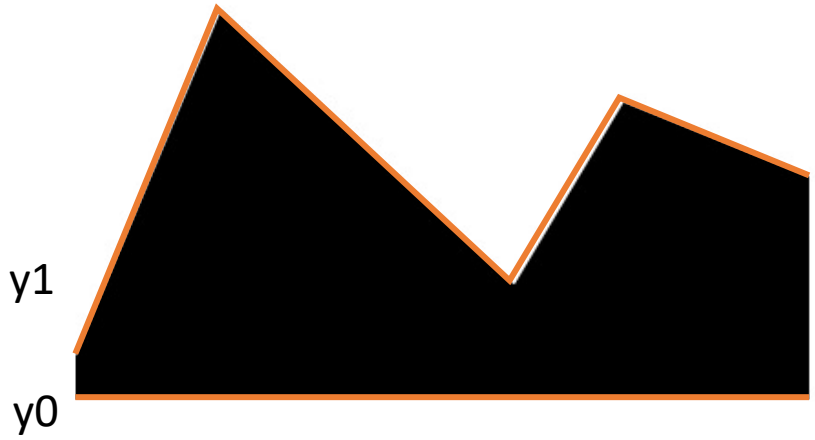


w08_ex02_main.js

```
const area = d3.area()
      .x( d => d.x )
      .y1( d => d.y )
      .y0( 0 );

svg.append('path')
    .attr('d', area(data))
    .attr('stroke', 'black')
    .attr('fill', 'black');
```

Example 03

- **Drawing Area**
  - Change baseline



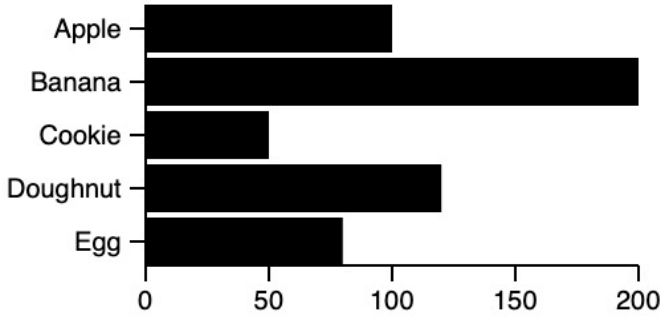
w08_ex02_main.js

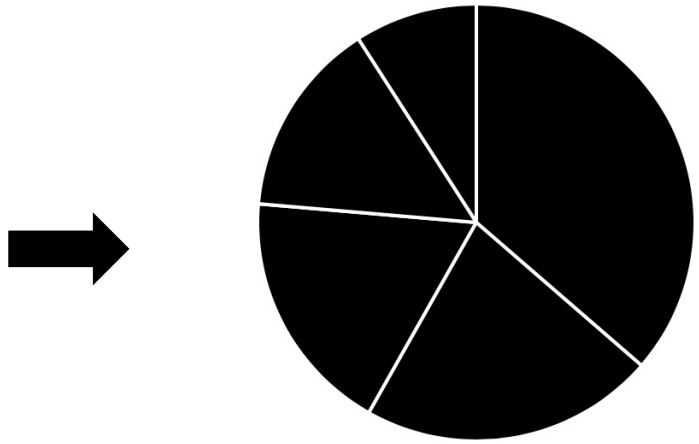
```
const area = d3.area()
      .x( d => d.x )
      .y1( d => d.y )
      .y0( d3.max(data, d => d.y ) + 10 );
```

Pie Chart

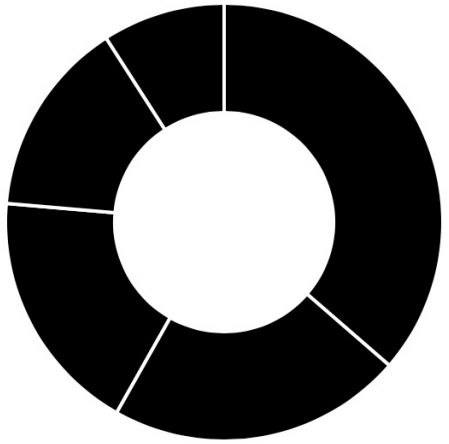
- **Drawing Pie Chart**
  - Data
    - Label: categ. attrib.
    - Value: quant. attrib.
  - Mark
    - Area
  - Channel
    - Angle

```
var data = [
    {label:'Apple', value:100},
    {label:'Banana', value:200},
    {label:'Cookie', value:50},
    {label:'Doughnut', value:120},
    {label:'Egg', value:80}
];
```





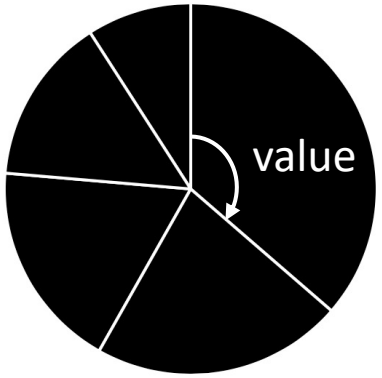
Pie chart

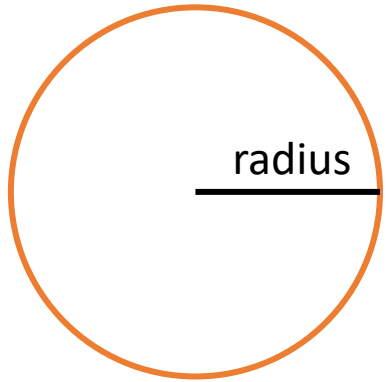


Donut chart

Example 04

- **Drawing Pie Chart**
  - D3.pie()
  - D3.arc()





w08_ex04_main.js

```
const pie = d3.pie()
      .value( d => d.value );

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

Example 04

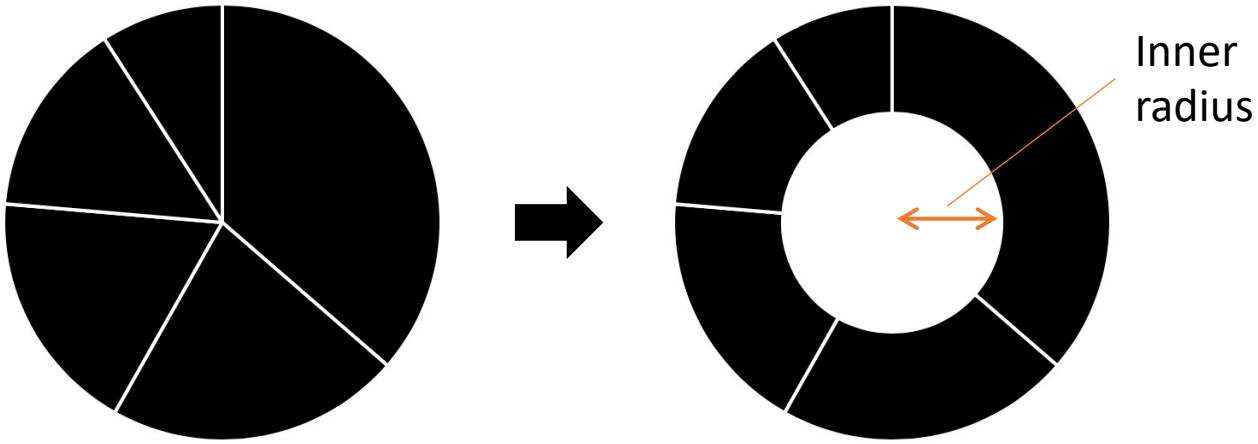
- **Drawing Pie Chart**
  - Construct pies and arcs

w08_ex04_main.js

```
var svg = d3.select('#drawing_region')
    .attr('width', width)
    .attr('height', height)
    .append('g')
    .attr('transform', `translate(${width/2}, ${height/2})`);
...
svg.selectAll('pie')
    .data( pie(data) )
    .enter()
    .append('path')
    .attr('d', arc)
    .attr('fill', 'black')
    .attr('stroke', 'white')
    .style('stroke-width', '2px');
```

Example 04

- **Drawing Donut Chart**
  - Change inner radius of arc



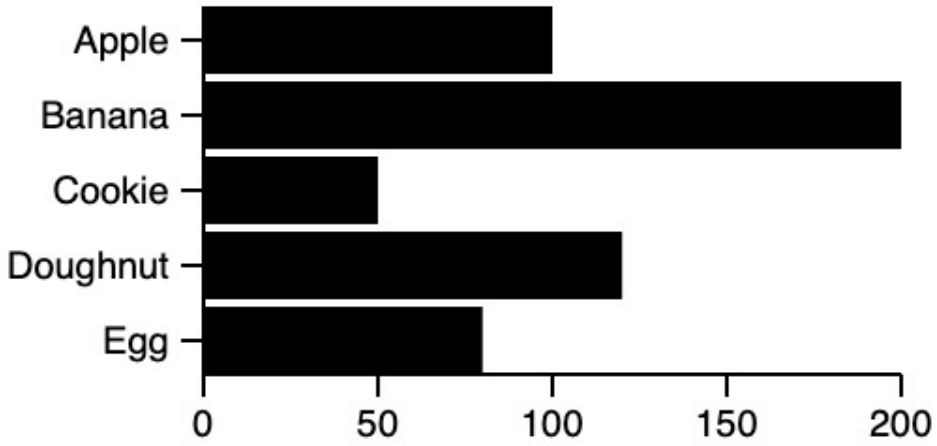
w08_ex04_main.js

```
const arc = d3.arc()
      .innerRadius(radius/2)
      .outerRadius(radius);
```

Task 1

- **Implement BarChart class**
  - Load an external data file
    - Same data as W04 - Task2
    - Other new data
    - …
  - Define
    - Constructor
    - init/update/render methods
  - Draw axes
  - Option
    - Draw chart tile and axis labels
    - Change bar orientation
    - …

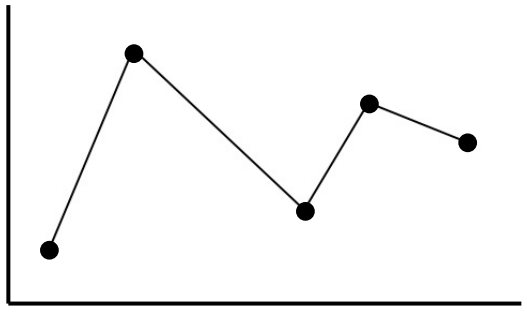
```
class BarChart {
    constructor() { ... }
    init() { ... }
    update() { ... }
    render() { ... }
    ...
}
```



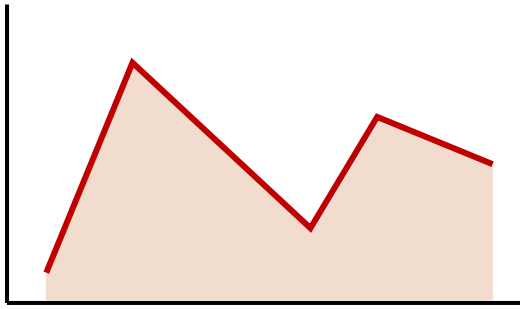
Task 2

- **Implement LineChart class**
  - Load an external data file
    - Same data as W08 - Example2
    - Other new data
    - …
  - Define
    - Constructor
    - init/update/render methods
  - Draw axes
  - Option
    - Draw dots
    - Draw area
    - Change bar orientation
    - …

```
class LineChart {
    constructor() { ... }
    init() { ... }
    update() { ... }
    render() { ... }
    ...
}
```



Draw line with dots

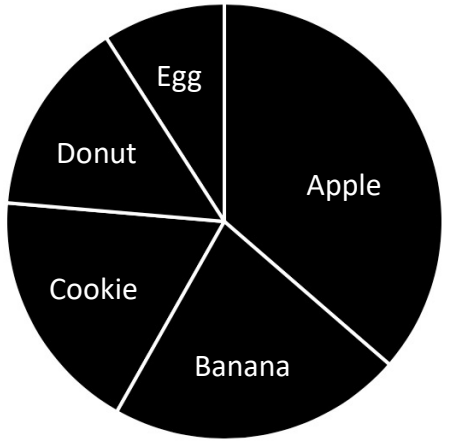


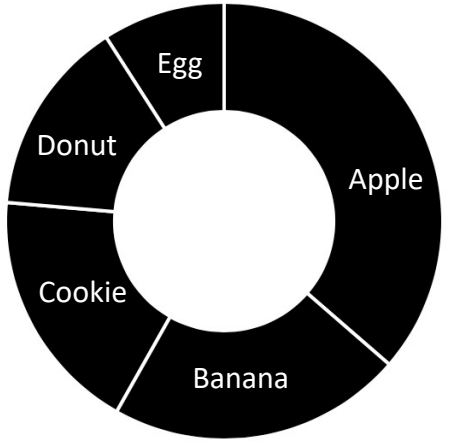
Draw line with area

Task 3

- **Implement PieChart class**
  - Load an external data file
    - Same data as W04 - Task2
    - Other new data
    - …
  - Define
    - Constructor
    - init/update/render methods
  - Draw labels
  - Option
    - Change inner radius of arc
    - Change colors of each pie
    - …

```
class PieChart {
    constructor() { ... }
    init() { ... }
    update() { ... }
    render() { ... }
    ...
}
```





Polling

- **Take the poll**
  - Student ID Number
  - Name
  - URL to Task 1
  - URL to Task 2
  - URL to Task 3
- **Submission deadline**
  - **May 16 (Mon), 2022 by 23:59 JST**