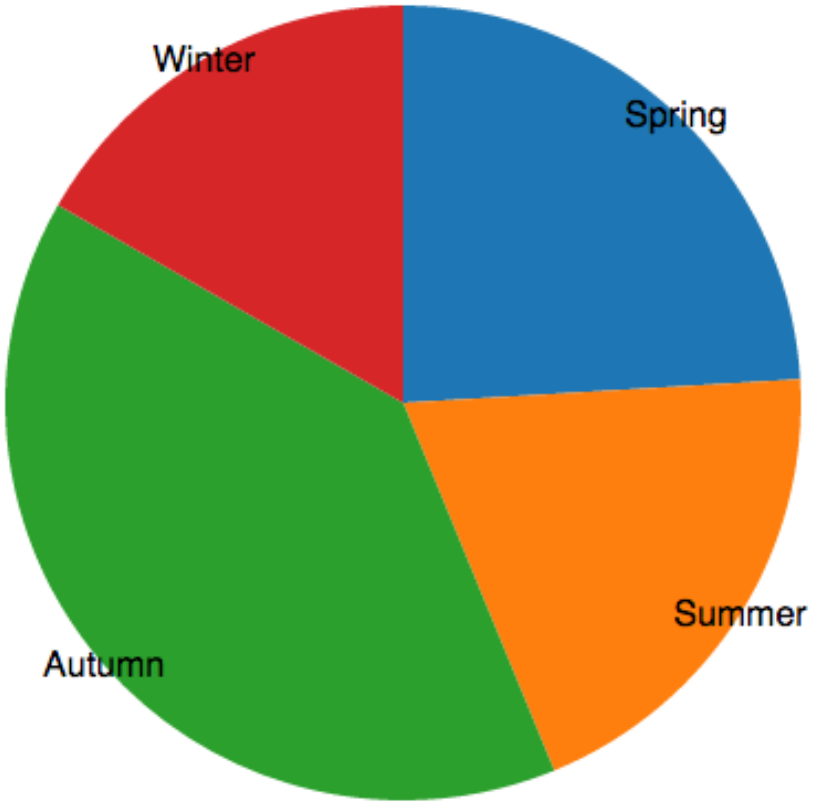
Data Visualisation

Week 6 – Pie Charts, Multiple Charts and Interactions

Pie Chart



Data (as JavaScript objects)

```
var dataset = [
    { label: 'Spring', count: 95, profit: 2000 },
    { label: 'Summer', count: 78, profit: 2900 },
    { label: 'Autumn', count: 156, profit: 1350 },
    { label: 'Winter', count: 66, profit: 3000 }
];
```

Setup

```
var width = 800;
var height = 800;
var radius = 180;
var centerX = 300;
var centerY = 200;
var color = d3.scale.category10();
```

Create SVG and Append a g

```
var svg = d3.select('body')
    .append('svg')
    .attr('width', width)
    .attr('height', height);

var g = svg.append('g')
    .attr('transform', 'translate(' + centerX +
',' + centerY + ')');
```

Arc

```
var arc = d3.svg.arc().outerRadius(radius);
```

- Also think about `innerRadius(…)`

Using Pie Layout

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

var data = pie(dataset);
```

Draw Pie Slices

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

Note: This “data” is processed by “pie”.

In order to get the original data, use “d.data” here.

Draw Text Labels

```
function midAngle(slice) {
    return slice.startAngle + (slice.endAngle - slice.startAngle) / 2;
}

var text = g.selectAll('text')
    .data(data)
    .enter()
    .append('text')
    .attr("x", function(d) { return Math.sin(midAngle(d)) * radius; })
    .attr("y", function(d) { return -1 * Math.cos(midAngle(d)) * radius; })
    .attr("dy", ".35em")
    .attr("text-anchor", "middle")
    .text(function(d, i) { return d.data.label; });
```

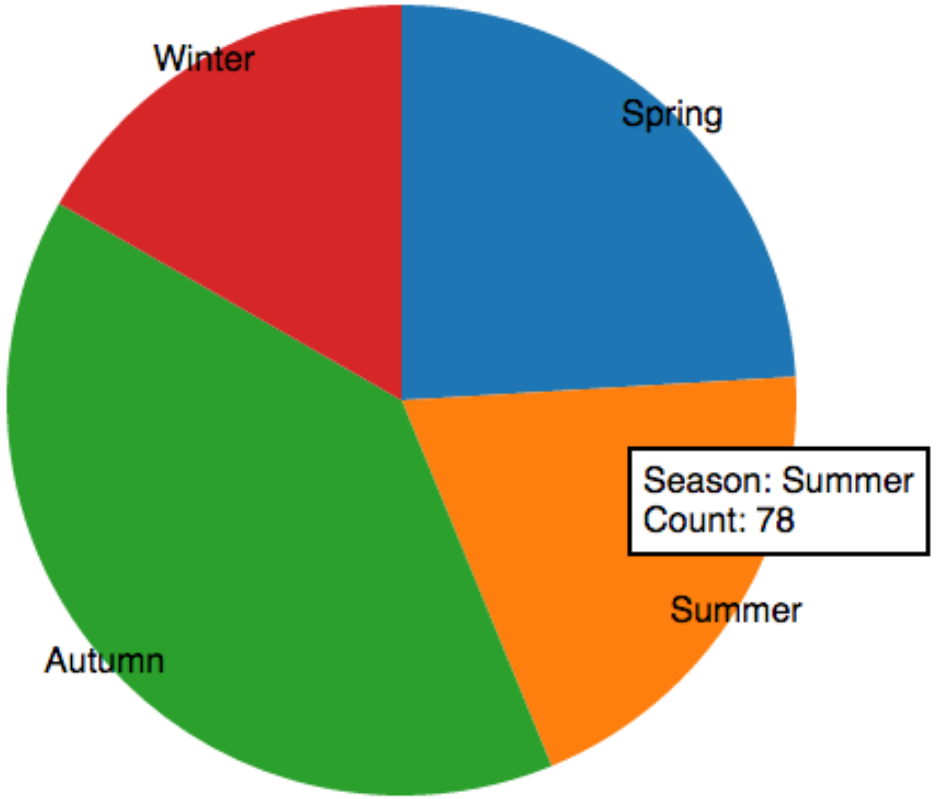
Interactions

- Handles certain events happening in the chart, e.g. mouse move, click…
- Use `on(…)` to declare what event to handle, and how to handle.

Tooltip When Mouse Is On

Idea:

- Create a tooltip box
- Handle `mousemove` event:
  - Populate text in the box
  - Set position
  - Show the box
- Handle `mouseout` event:
  - Hide the box



Create Tooltip Box (Using HTML & CSS)

HTML:

```
<div id="tooltip">
    Season: <span id="tooltip-season"></span><br>
    Count: <span id="tooltip-count"></span>
</div>
```

CSS:

```
#tooltip {
    position: absolute;
    background-color: white;
    border: 2px solid black;
    padding: 5px;
    z-index: 1000;
    display: none;
}
```

Positioning the tooltip box with the X, Y coordinate.

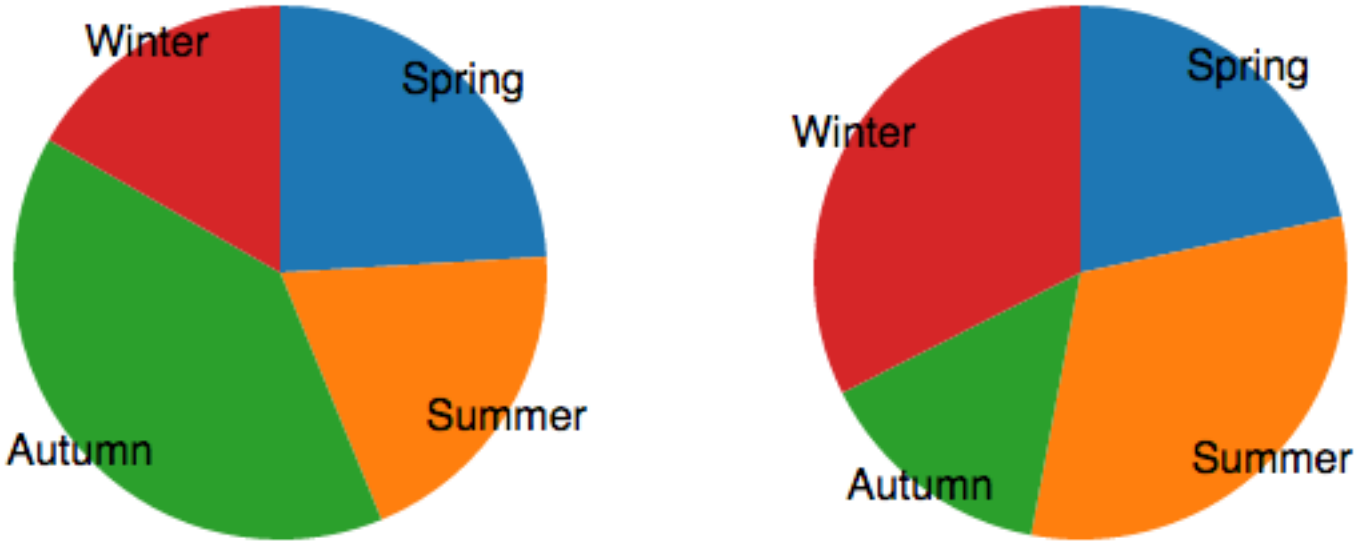
"Bring to Front"

Hide the box initially.

Handle Events

```
var path = g.selectAll('path')
    ……
    .on("mousemove", function(d) {
        d3.select("#tooltip-season").text(d.data.label);
        d3.select("#tooltip-count").text(d.data.count);
        d3.select("#tooltip")
            .style("left", d3.event.pageX + 10 + "px")
            .style("top", d3.event.pageY + 10 + "px")
            .style("display", "block");
    })
    .on("mouseout", function(d) {
        d3.select("#tooltip").style("display", "none");
    });
```

Multiple Charts



In this example, we draw one chart for “count” and one chart for “profit”.

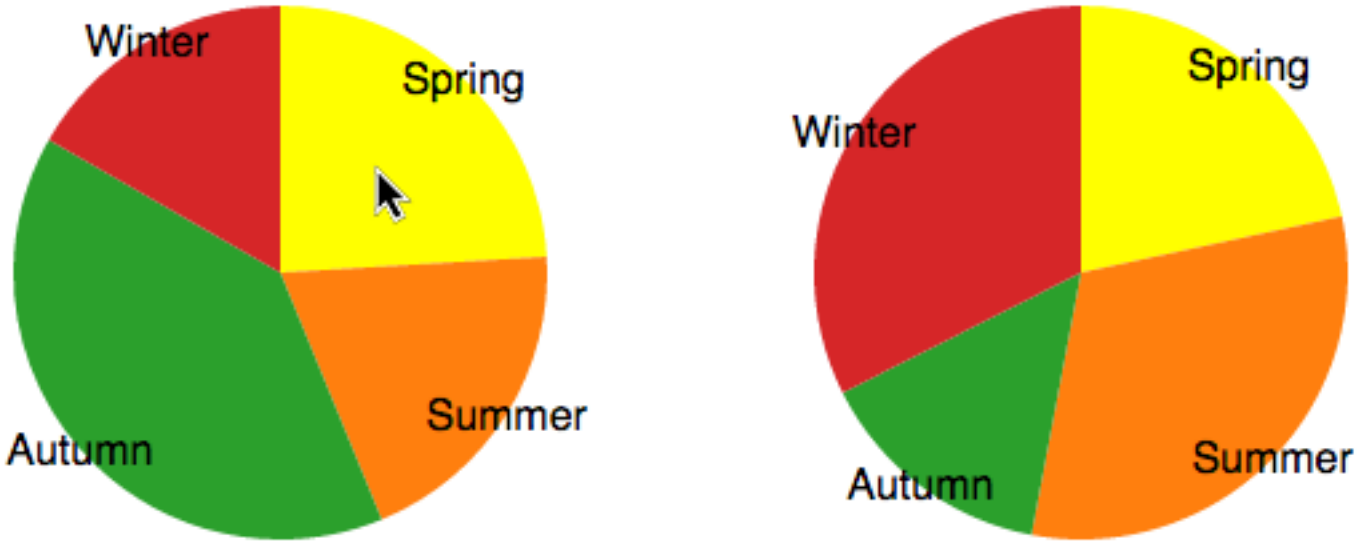
Encapsulate Drawing Code

```
function drawPie(pieNumber, attribute) {
    var g = svg.append('g')
        .attr('transform', 'translate(' + (pieNumber * centerX) +
',' + centerY + ')');
…….
    var pie = d3.layout.pie()
        .value(function(d) { return d[attribute]; })
        .sort(null);
.......
}
```

Draw Multiple Charts

```
drawPie(1, "count");
drawPie(2, "profit");
```

Interactions: Highlighting Related Items



Interactions: Highlighting Related Items

- First, add a common class name on items of same season:

```
var path = g.selectAll('path')
    .data(data)
    .enter()
    .append('path')
    .attr('d', arc)
    .attr('fill', function(d) { return color(d.data.label);
})
    .attr("class", function(d, i) { return "slice" + i})
    .on("mousemove", mouseMove)
    .on("mouseout", mouseOut);
```

Handle Events

```
function mouseMove(d, i) {
    d3.selectAll(".slice" + i).classed({"highlight": true});
}

function mouseOut(d, i) {
    d3.selectAll(".slice" + i).classed({"highlight": false});
}
```

Use CSS to Format Highlighting

```
<style type="text/css">
…
.highlight { fill: yellow; }
</style>
```