VISUALIZATION ON THE WEB

Reusable modules
FROM JAVASCRIPT CODE TO MODULES

- D3.js provides a vast library of examples
- In many projects, an example is modified and adapted for a specific use
- However, the code is difficult to maintain and adapt to different scenarios
- Solution: encapsulate all the code within a module that is bound to data and a container
**JAVASCRIPT AND OBJECTS**

- We want to organize our visualization into components for
  - **Modularity**: separate the different parts of a complicated visualization
  - **Composability and reusability**: reuse smaller pieces in different visualization
  - **Simplification**: concentrate on smaller part of the main problem first

- To implement this approach we use objects, i.e. entities with properties and functions

- Objects are not fully supported in Javascript (prior to ES2016)
  - We exploit function closures
AN EXAMPLE FOR BARCHART

// Creates bar chart component and configures its margins
barChart = chart()
    .margin({top: 5, left: 10});

container = d3.select('.chart-container');

// Calls bar chart with the data-fed selector
container.datum(dataset).call(barChart);
**GENERAL SCHEMA FOR A CHART**

```javascript
function chart() {
  var width = 720, // default width
      height = 80; // default height

  function my(selection) {
    // generate chart here, using `width` and `height`
  }

  my.width = function(value) {
    if (!arguments.length) return width;
    width = value;
    return my;
  }

  my.height = function(value) {
    if (!arguments.length) return height;
    height = value;
    return my;
  }

  return my;
}
```

- **Internal properties of the object**: width and height
- **Constructor and preparation for the chart attached to the selection**
- **Getter and setter for width**
- **Getter and setter for height**
- **Export the internal function outside this scope**
BAR CHART AS A REUSABLE COMPONENT

• Specification
  • Input: the component takes in input an array of numbers
  • Visualization: each number is rendered as a bar whose length is proportional to its value; an axis provide reference for the values
EXAMPLE - LINES

• Repository on GitHub: https://github.com/va602aa-2021/d3_vue