Vega

Visualization Grammar

About

- Visualization Grammar
- Built on top of D3.js
- Rapid generation without limiting the types of visualization possible

Purpose

- Enable fast, customizable design
- Make reusable, sharable visualizations
- Enable programmatic generation of visualization
- Improved performance

Getting Started

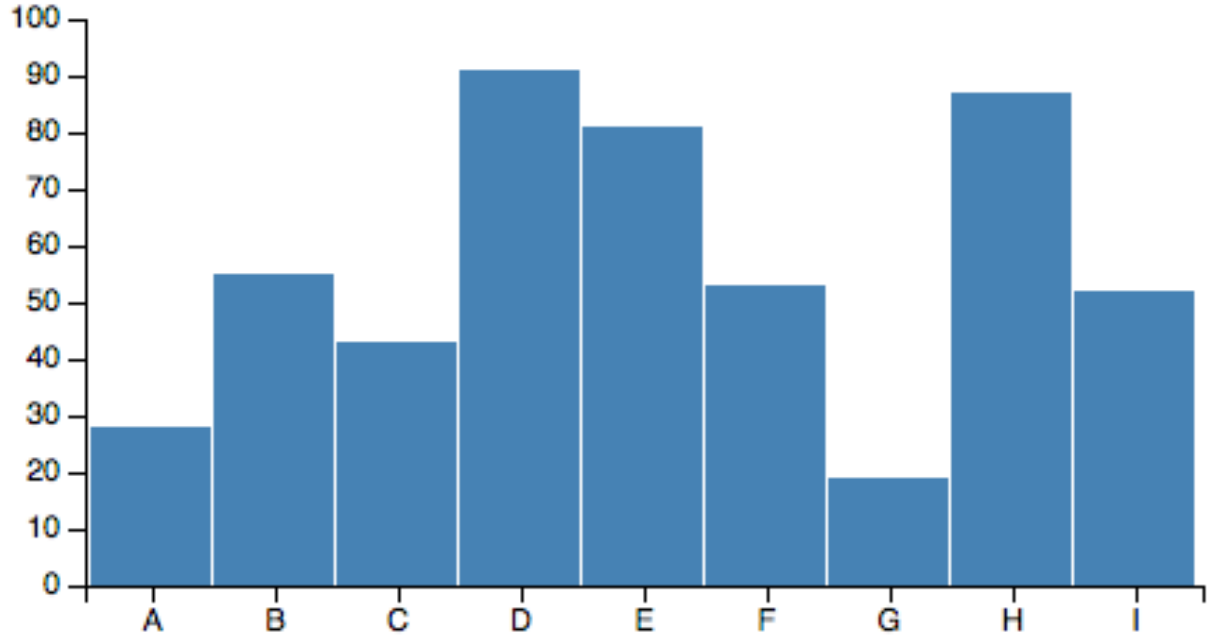
- HTML Document:

```
<html>
  <head>
    <title>Vega Scaffold</title>
    <script src="http://trifacta.github.io/vega/lib/d3.v3.min.js"></script>
    <script src="http://trifacta.github.io/vega/vega.js"></script>
  </head>
  <body>
    <div id="vis"></div>
  </body>
<script type="text/javascript">
// parse a spec and create a visualization view
function parse(spec) {
  vg.parse.spec(spec, function(chart) { chart({el:"#vis"}).update(); });
}
parse("uri/to/your/vega/spec.json");
</script>
</html>
```

- Node: npm install vega

Vega Specification

- Simply a JSON object



Vega Specification Properties

- name
- width
- height
- viewport
- padding
- data
- scales
- axes
- legends
- marks

Specification: Data

- Tabular data model

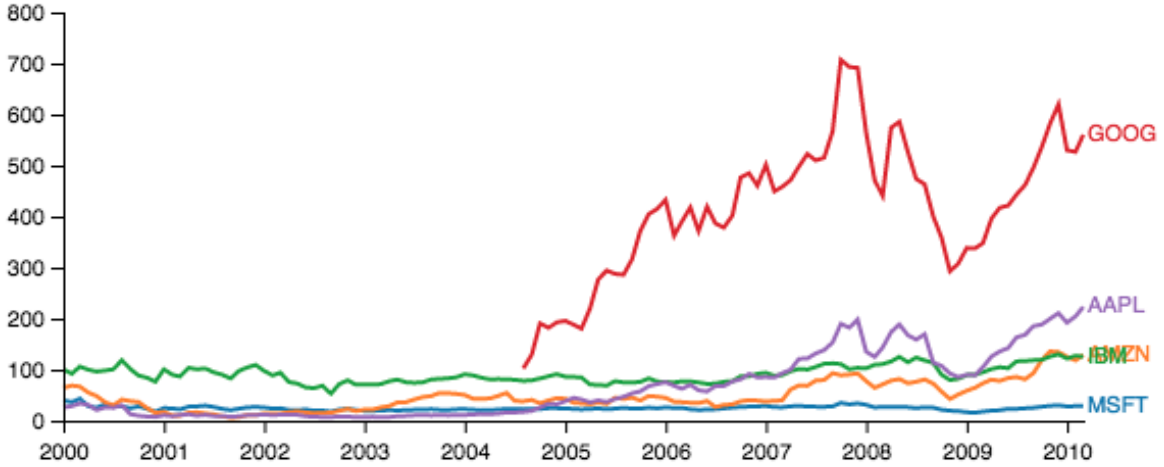
```
JSON:
[{"x":0, "y":3}, {"x":1, "y":5}]

Data Object:
[{"index":0, "data":{"x":0, "y":3}}, {"index":1, "data":{"x":1, "y":5}}]
```

- format
  - json, csv, tsv, topojson, treejson
- values
  - Allows data to be inlined directly in the specification

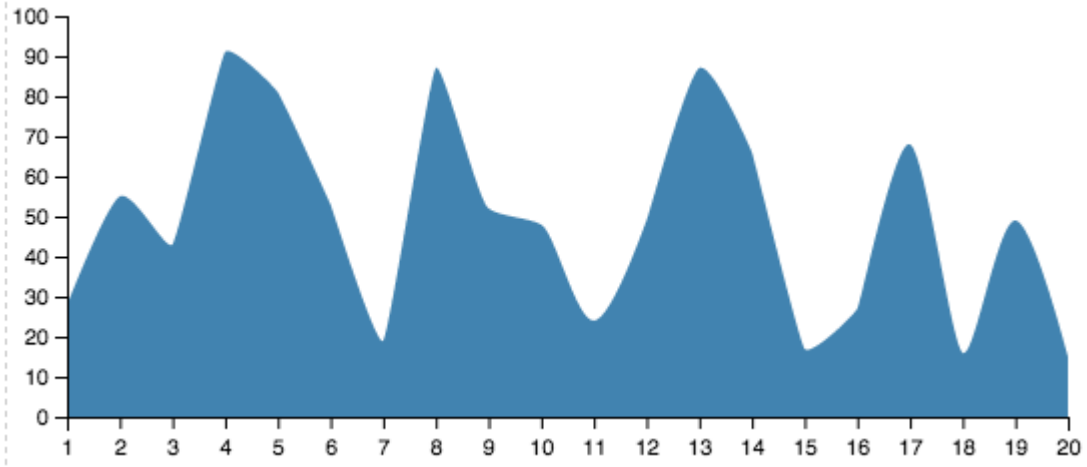
Specification: Scales

- Transform data values to visual values



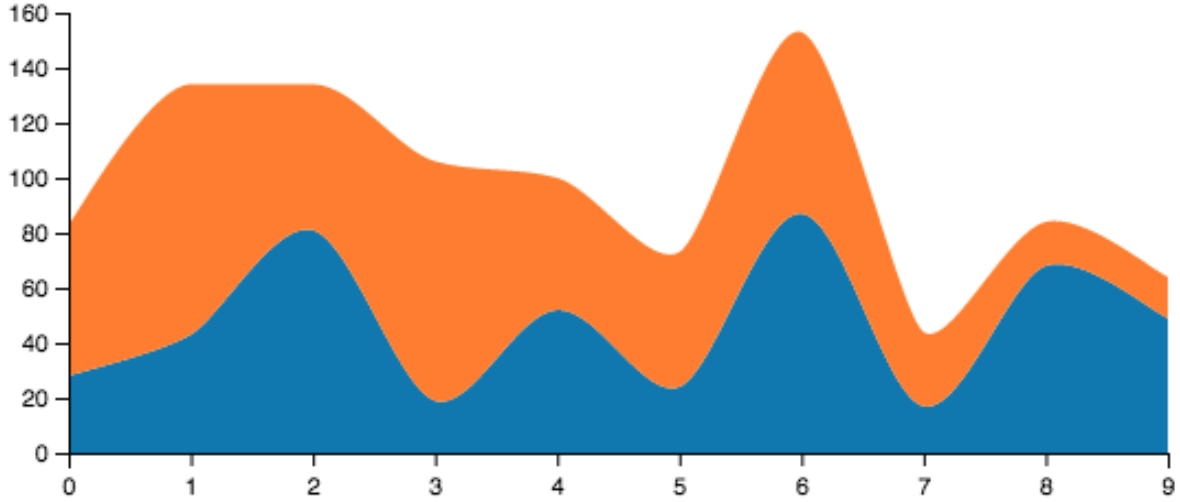
Axes and Legends

- Axes provide lines, ticks, and labels
- Legends provide colors, shapes, and sizes



Marks

- Marks are the basic visual building block for a visualization



Examples

Resources

- Vega:
  - http://trifacta.github.io/vega/