Grammar of Graphics

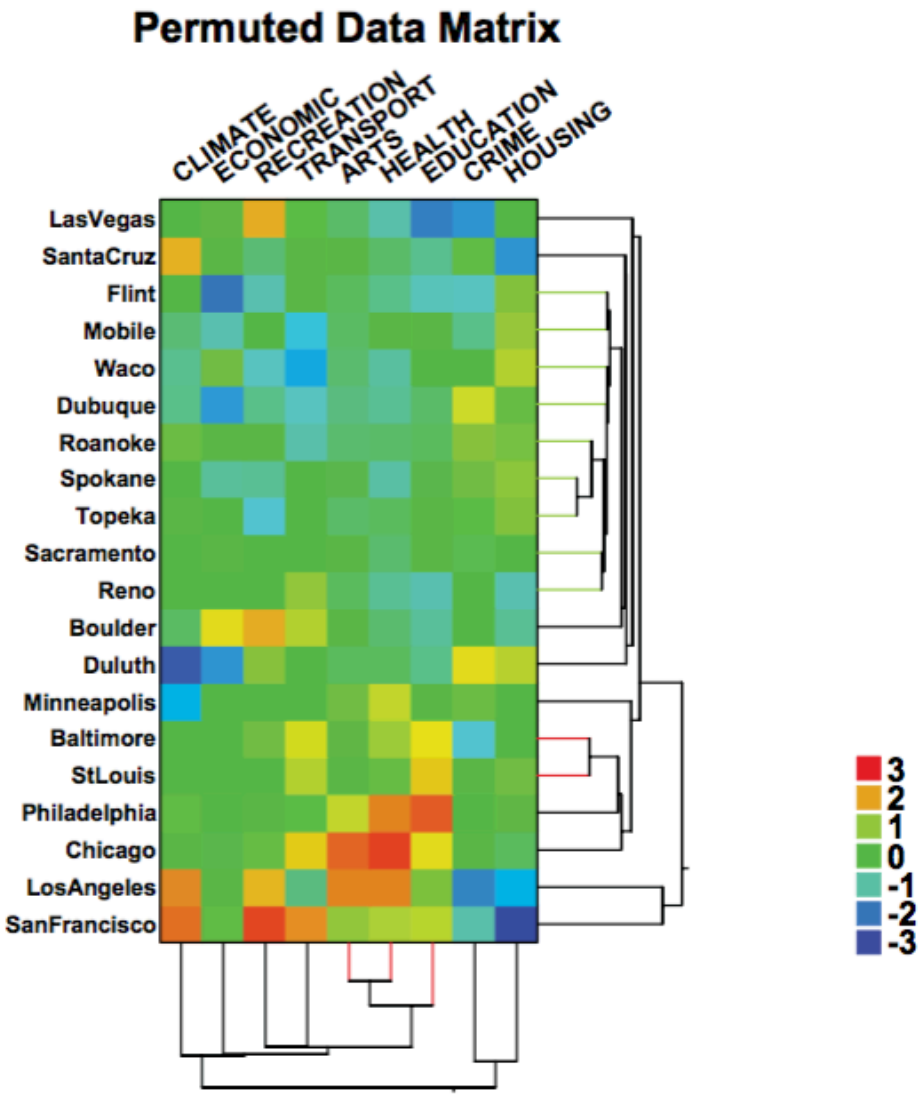
Graphics Production Language (GPL)

```
PAGE: begin
  SCALE: time(dim(1))
  SCALE: interval(dim(2), min(0), max(1.0d8), unit(currency.dollar))
  SCALE: cat(dim(3))
  SCALE: cat(dim(4))
  COORD: rect(dim(3,4), rect(dim(1,2)))
  GUIDE: axis(dim(1), label("Date"), format("mm/dd/yy"))
  GUIDE: axis(dim(2), label("Revenue"))
  GUIDE: axis(dim(3), label("Company"))
  GUIDE: axis(dim(4), label("State"))
  GUIDE: legend(aesthetic(color.blue), dim(1))
  ELEMENT: point(position(d*r*c*s), color(pe))
PAGE: end
```

Graphics Production Language (GPL)

```
DATA: unode = col(source("col tree"))
DATA: vnode = col(source("col tree"))
DATA: xnode = col(source("row tree"))
DATA: ynode = col(source("row tree"))
DATA: x = reshape.rect(climate, economic, recreation, transport,
          arts, health, education, crime, housing, "colname")
DATA: y = reshape.rect(climate, economic, recreation, transport, arts,
          health, education, crime, housing, "rowname")
DATA: d = reshape.rect(climate, economic, recreation, transport,
          arts, health, education, crime, housing, "value")
GRAPH: begin(origin(0, 0), scale(4cm, 8cm))
  ELEMENT: polygon(position(bin.rect(x*y)), color.hue(d))
GRAPH: end
GRAPH: begin(origin(0, -2cm), scale(4cm, 2cm))
  COORD: transpose(dim(1, 2), reflect(dim(2)))
  ELEMENT: edge(position(link.join(unode*vnode)))
GRAPH: end
GRAPH: begin(origin(4cm, 0cm), scale(2cm, 8cm))
  ELEMENT: edge(position(link.join(xnode*ynode)))
GRAPH: end
```

Graphics Production Language (GPL)

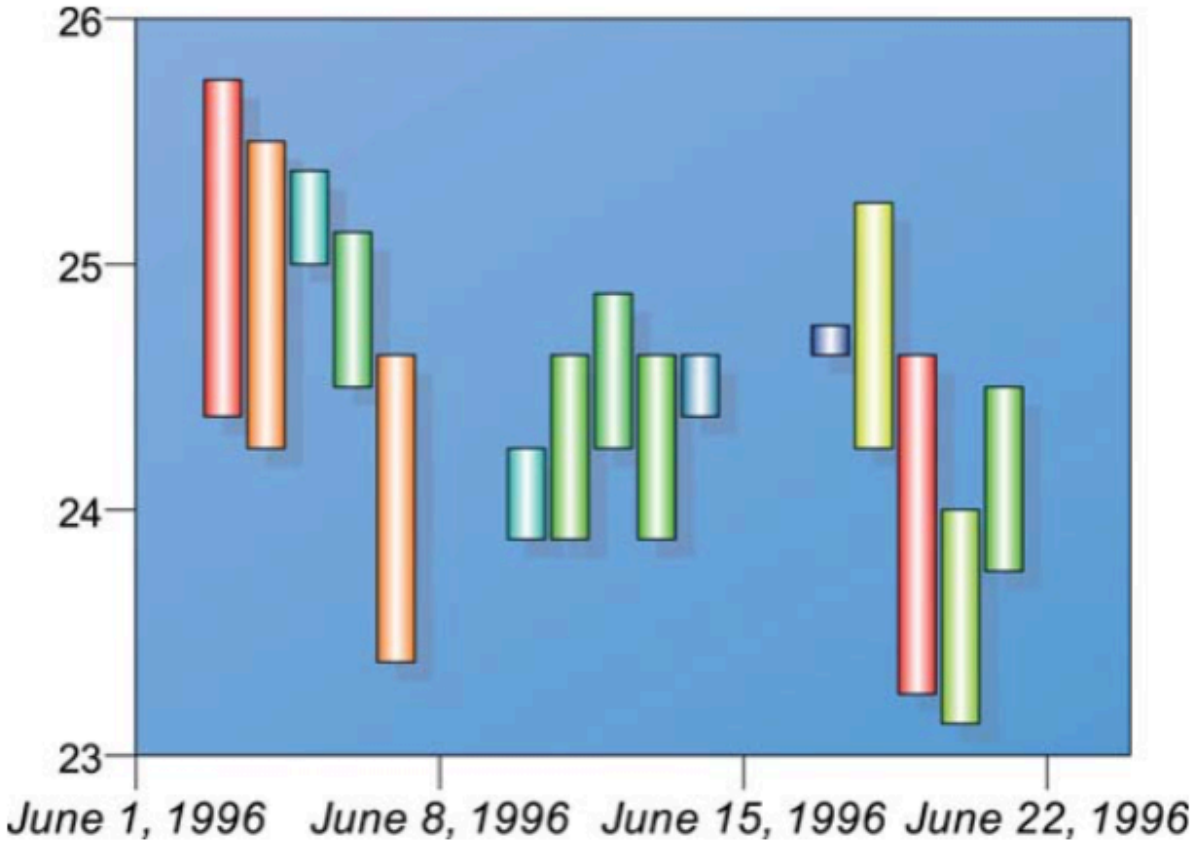


Visualization Markup Language (VizML)

XML-based (like SVG)

Visualization Markup Language (VizML)

XML-based (like SVG)



Visualization Markup Language (VizML)

```
<?xml version="1.0"?>
<graph>
   <source id="data">
      <fileAccess fileName="stockValues.csv" separator="," 
header="true"/>
   </source>
   <variable id="date" sourceName="date" source="data" cate-
gorical="false">
      <meta>
         <dateTimeFormat>
            <dateFormat>
               <dayOfMonth/>
               <monthName long="true"/>
               <year showCentury="true"/>
            </dateFormat>
         </dateTimeFormat>
      </meta>
   </variable>
   <variable id="high" sourceName="high" source="data" cate-
gorical="false"/>
   <variable id="low" sourceName="low" source="data" cate-
gorical="false"/>
   <variable id="close" sourceName="close" source="data"
             categorical="false"/>
   <variable id="range" categorical="false" expres-
sion="high+low"
             source="data"/>

   <domain categorical="false" id="dateDomain">
      <interval min="1996-06-02" max="1996-06-23"/>
   </domain>
```

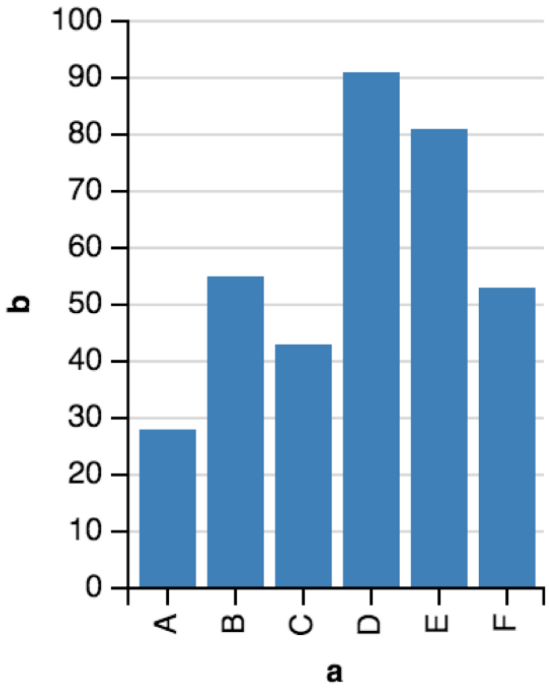
VizQL

Proprietary visualization grammar underlying Tableau

Evolved from Polaris system at Stanford (extends pivot tables)

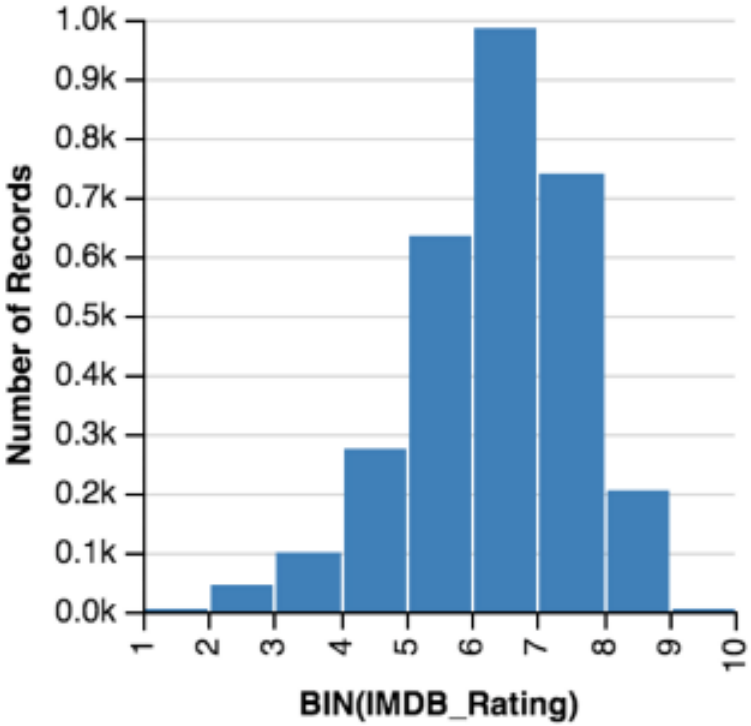
Vega and Vega-Lite

```
{
  "data": {
    "values": [
      {"a": "A","b": 28}, {"a": "B","b": 55}, {"a": "C","b": 43},
      {"a": "D","b": 91}, {"a": "E","b": 81}, {"a": "F","b": 53}
    ]
  },
  "mark": "bar",
  "encoding": {
    "x": {"field": "a", "type": "ordinal"},
    "y": {"field": "b", "type": "quantitative"}
  }
}
```



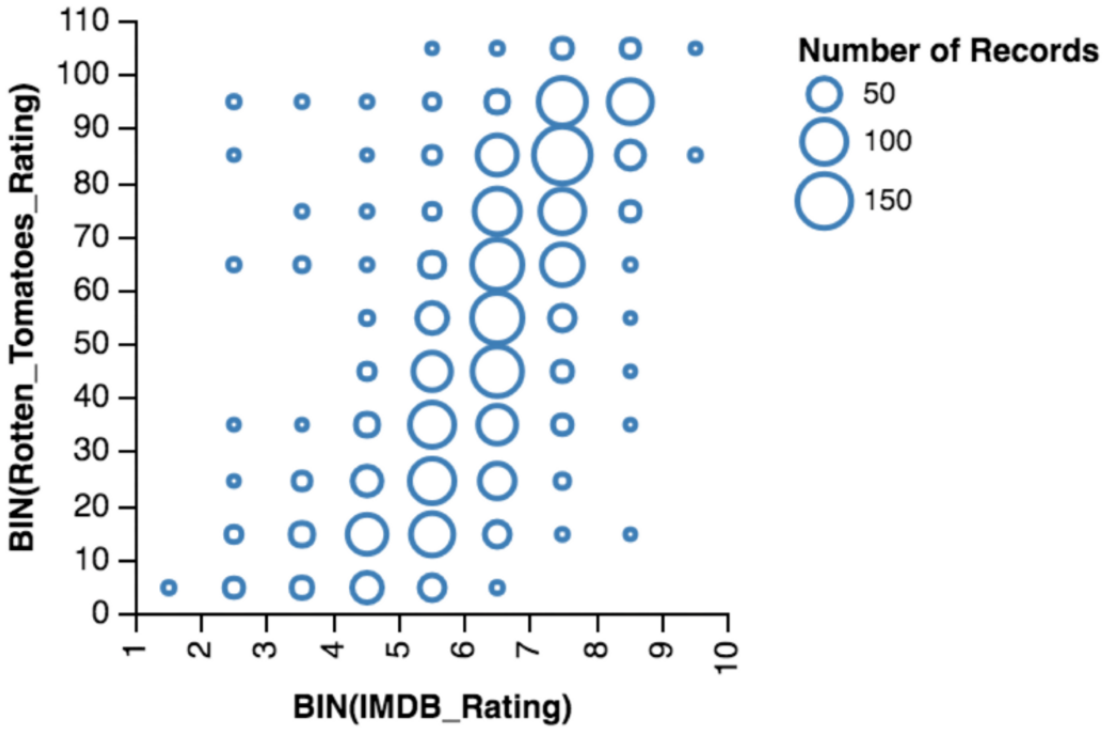
Vega-Lite

```
{
  "data": {"url": "data/movies.json"},
  "mark": "bar",
  "encoding": {
    "x": {
      "bin": {"maxbins": 10},
      "field": "IMDB_Rating",
      "type": "quantitative"
    },
    "y": {
      "aggregate": "count",
      "field": "*",
      "type": "quantitative"
    }
  }
}
```



Vega-Lite

```
{
  "data": {"url": "data/movies.json"},
  "mark": "point",
  "encoding": {
    "x": {
      "bin": {"maxbins": 10},
      "field": "IMDB_Rating",
      "type": "quantitative"
    },
    "y": {
      "bin": {"maxbins": 10},
      "field": "Rotten_Tomatoes_Rating",
      "type": "quantitative"
    },
    "size": {
      "aggregate": "count",
      "field": "*",
      "type": "quantitative"
    }
  }
}
```



Berkeley SCHOOL OF INFORMATION