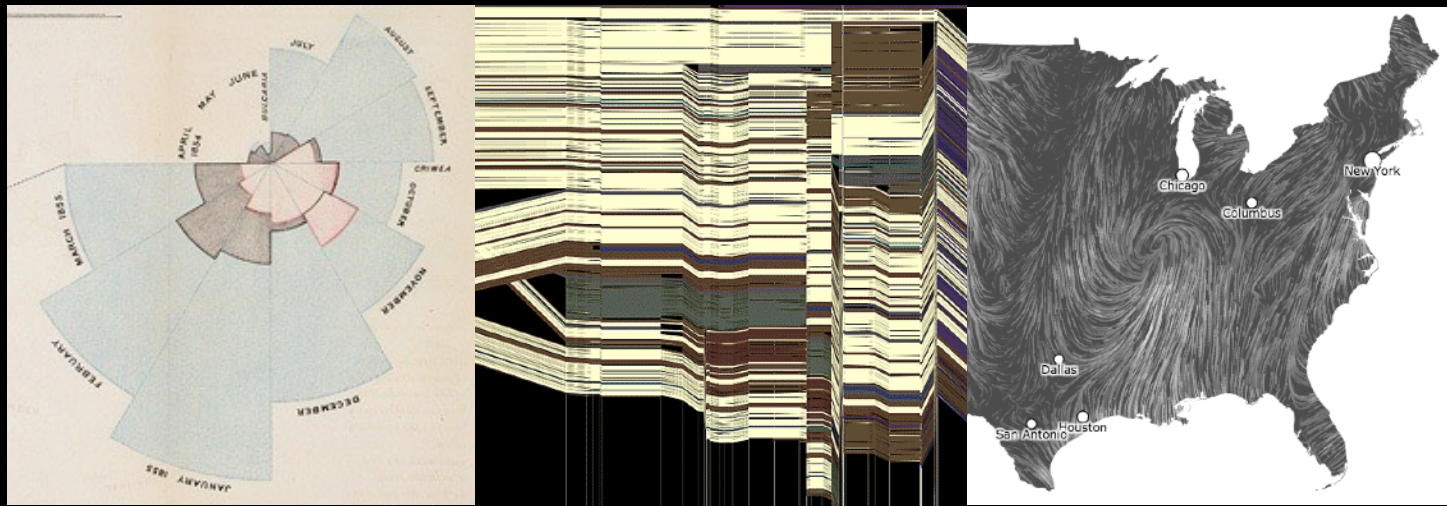


Visualization Tools, Part 1



Jeffrey Heer University of Washington

Tools & Trade-Offs

Ease-of-Use

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D



Offer **fine-grained control** for composing interactive graphics.

But require **verbose** specifications and technical expertise.

Ease-of-Use

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness

Grammar Building Blocks

Data

Input data to visualize

Transforms

Filter, aggregate, stats, layout

Scales

Map data values to visual values

Guides

Axes & legends to visualize scales

Marks

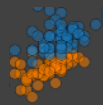
Data-representative graphics



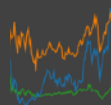
Area



Rect



Symbol



Line



Arc

Text

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

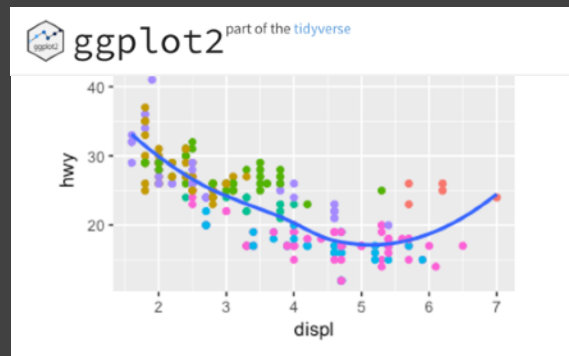
Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D



Facilitate **rapid exploration** with **concise** specifications by omitting low-level details.

Infer **sensible defaults** and customize by overriding defaults.

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

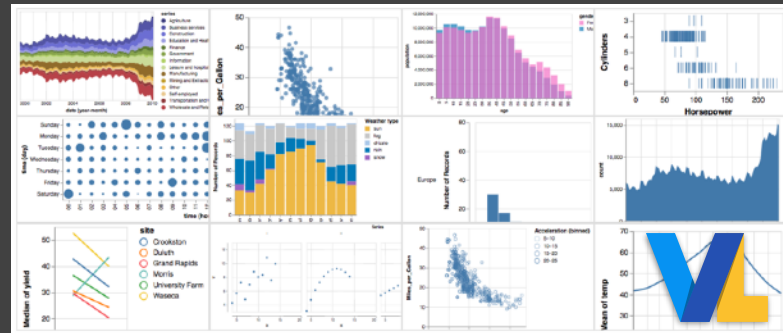
Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness



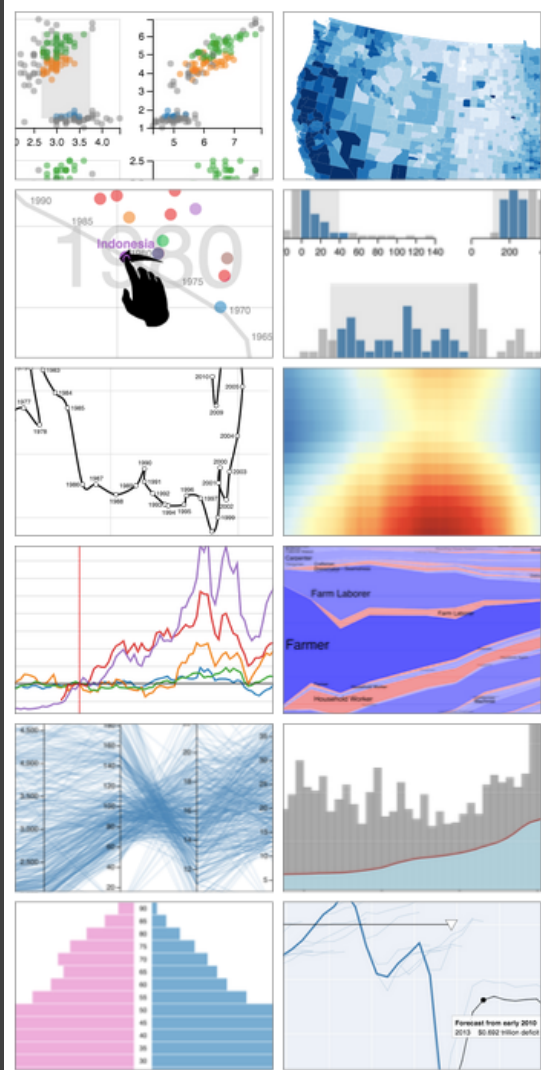
Facilitate **rapid exploration** with **concise** specifications by omitting low-level details.

Infer **sensible defaults** and customize by overriding defaults.

A Dive Into Vega-Lite

Vega-Lite is a *Visual Analysis Grammar*

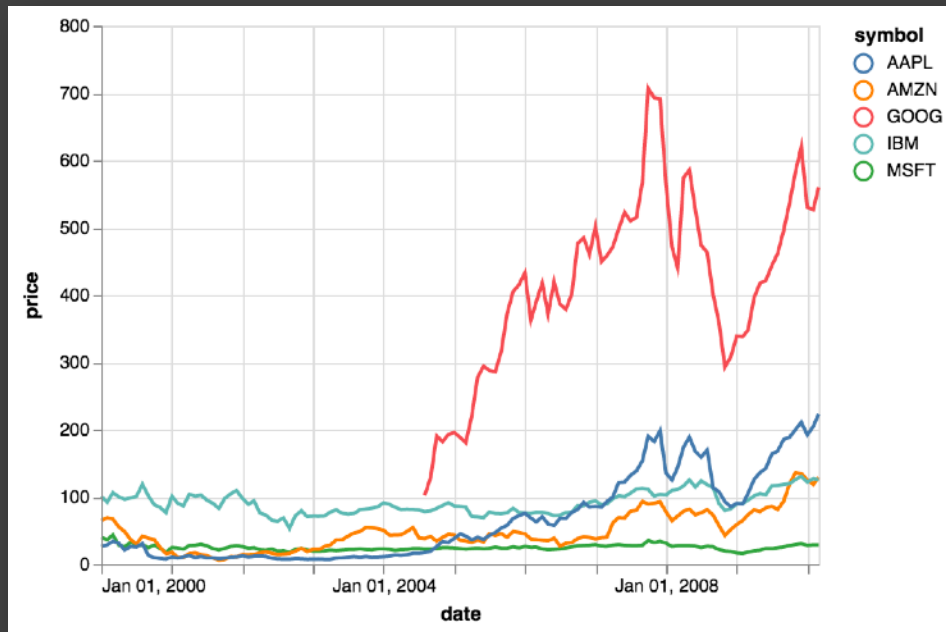
Similar in spirit to how SQL provides a language for expressing database queries, Vega-Lite is a high-level language for describing visualizations.



Vega-Lite is a *Visual Analysis Grammar*

Vega Altair Python API

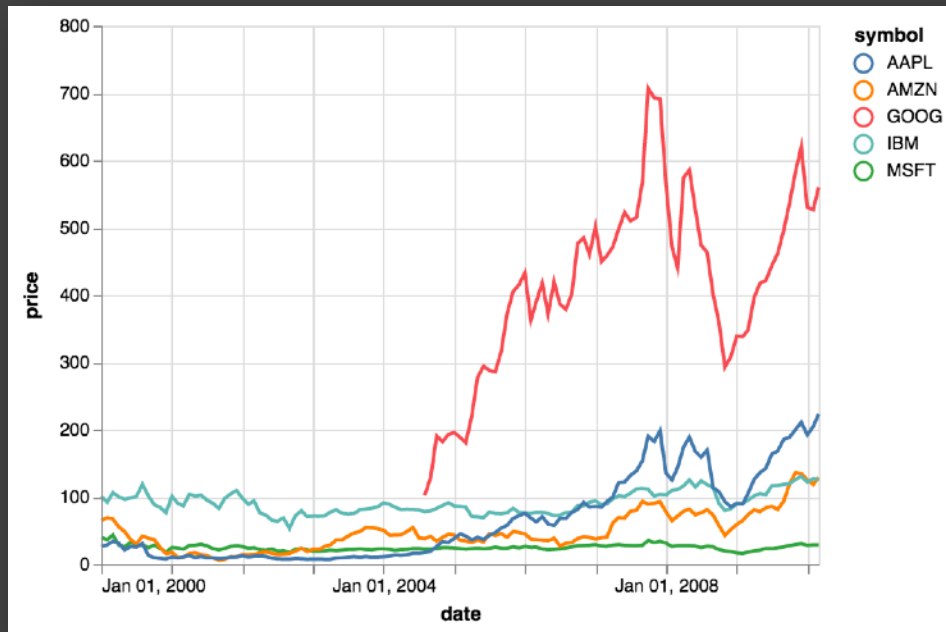
```
alt.Chart('stocks.csv')  
  .mark_line()  
  .encode(  
    x='date',  
    y='price',  
    color='symbol'  
  )
```



Vega-Lite is a *Visual Analysis Grammar*

Vega-Lite JavaScript API

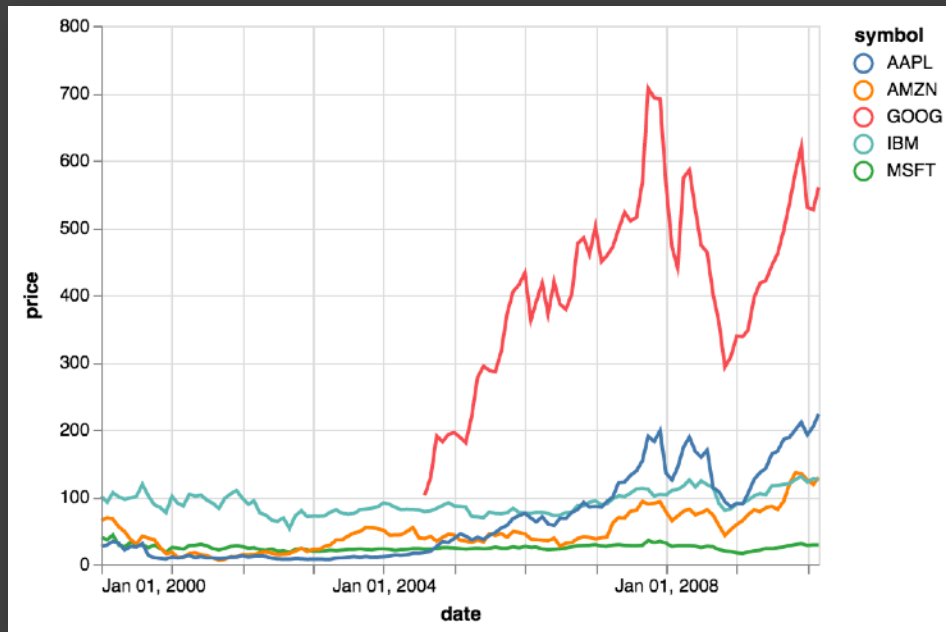
```
vl.data('stocks.csv')  
  .markLine()  
  .encode(  
    vl.x().fieldT('date'),  
    vl.y().fieldQ('price'),  
    vl.color().fieldN('symbol')  
  )
```



Vega-Lite is a *Visual Analysis Grammar*

Vega-Lite JSON Specification

```
{
  data: {url: "stocks.csv"},
  mark: "line",
  encoding: {
    x: {
      type: "temporal",
      field: "date"
    },
    y: {
      type: "quantitative",
      field: "price"
    },
    color: {
      type: "nominal",
      field: "symbol"
    }
  }
}
```



Specifying Single Views

Abstract Data



Visual Representation

Specifying Single Views

Abstract Data

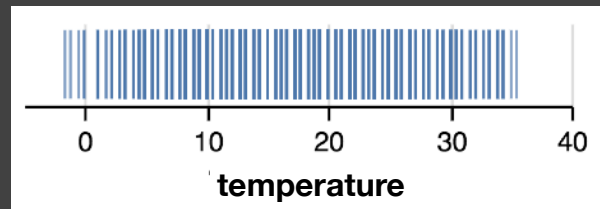


Visual Representation

Weather Data for Seattle

date	temperature	precipitation	weather
1/1	10.6	10.9	"rain"
1/2	11.7	0.8	"drizzle"
1/3	12.2	10.2	"rain"
...	...	...	...

Strip Plot of Temperature



Strip Plot = (Tick with x=field)

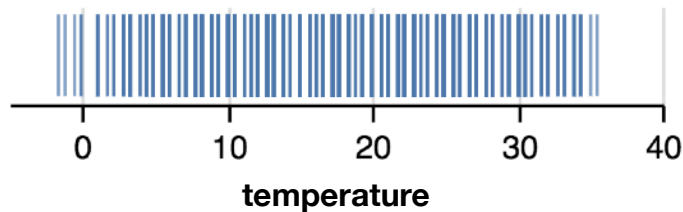
Tick Mark



Temperature
as x-position

(Quantitative)

```
{  
  data: {url: "weather-seattle.json"},  
  mark: "tick",  
  encoding: {  
    x: {  
      field: "temperature",  
      type: "quantitative"  
    }  
  }  
}
```



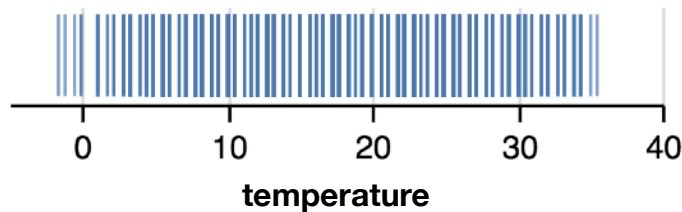
Strip Plot = (Tick with $x=field$)

Tick Mark



Temperature
as x-position
(Quantitative)

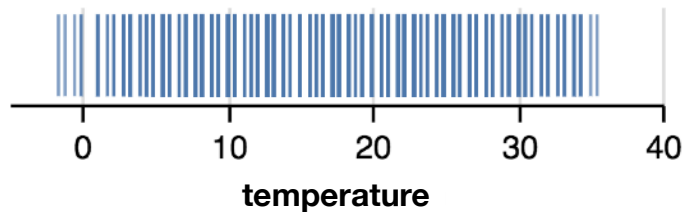
```
{  
  data: {url: "weather-seattle.json"},  
  mark: "tick",  
  encoding: {  
    x: {  
      field: "temperature",  
      type: "quantitative"  
    }  
  }  
}
```



Vega-Lite is portable JSON specification

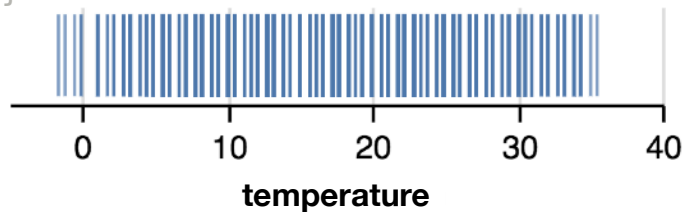
Strip Plot = (Tick with x=field)

```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      field: "temperature",
      type: "quantitative"
    }
  }
}
```



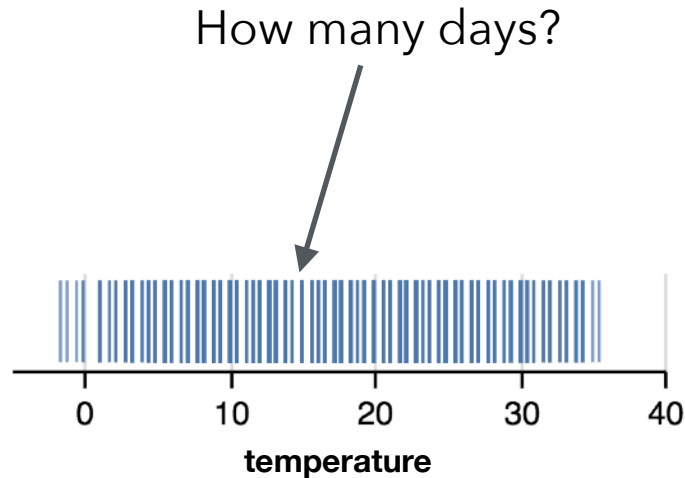
Strip Plot: Default Scales and Axes

```
{  
  data: {url: "weather-seattle.json"},  
  mark: "tick",  
  encoding: {  
    x: {  
      field: "temperature",  
      type: "quantitative",  
      scale: {type: "linear", domain: [-10, 40], ...}  
      axis: {title: "temperature", grid: true, ...}  
    }  
  }  
}
```



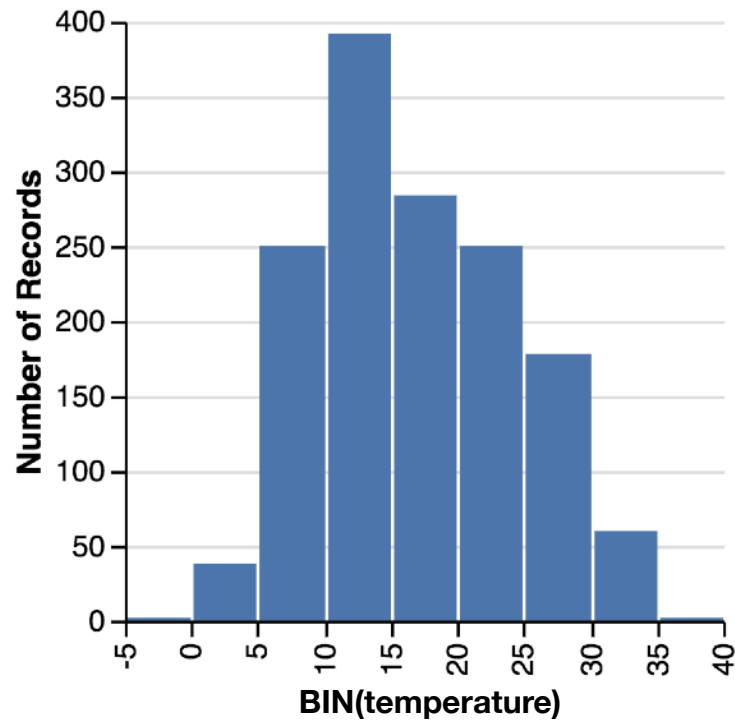
Strip Plot

```
{  
  data: {url: "weather-seattle.json"},  
  mark: "tick",  
  encoding: {  
    x: {  
      field: "temperature",  
      type: "quantitative"  
    }  
  }  
}
```



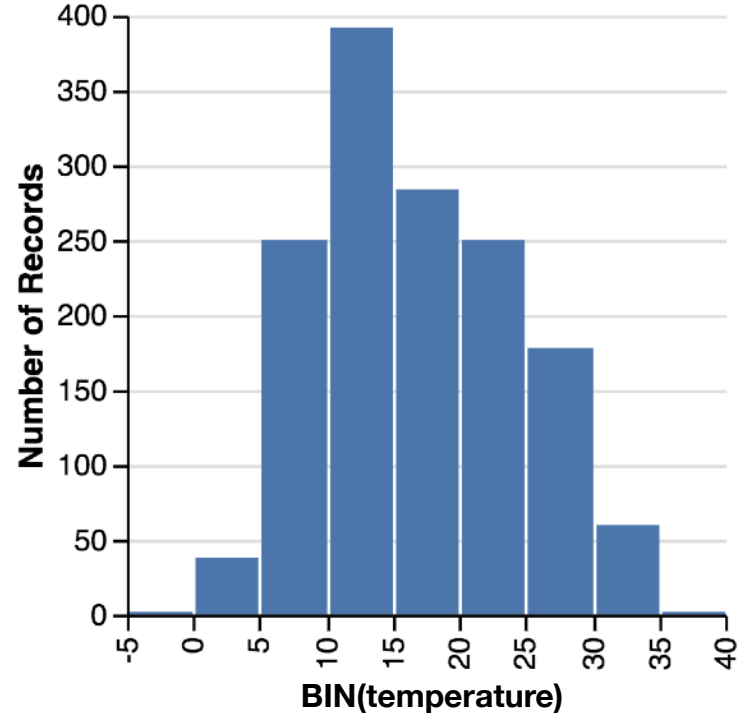
Histogram

Goal



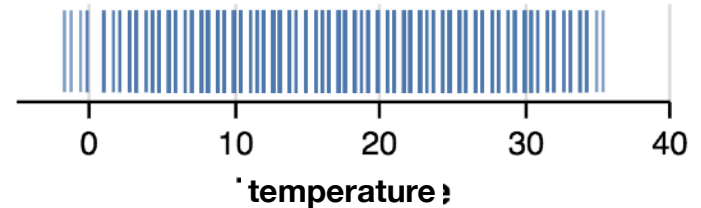
Histogram = (**Bar** with x =binned field, y =count)

Goal



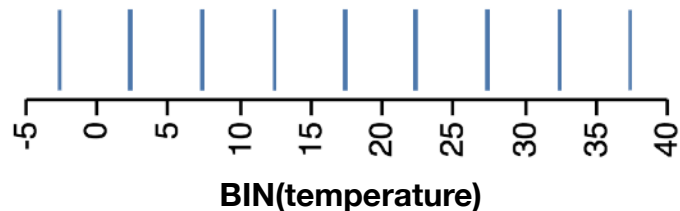
Histogram = (**Bar** with x =binned field, y =count)

```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      field: "temperature",
      type: "quantitative"
    }
  }
}
```



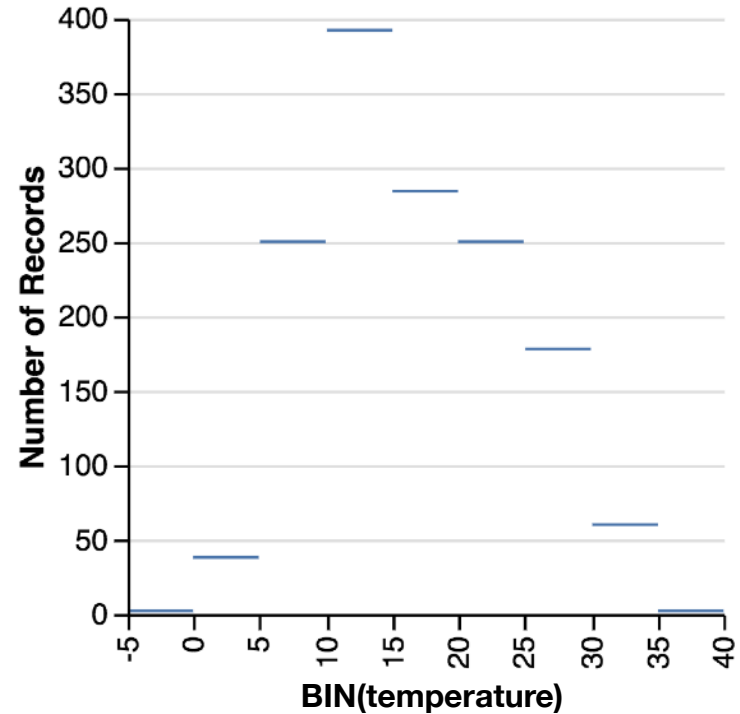
Histogram = (**Bar** with x=binned field, y=count)

```
{  
  data: {url: "weather-seattle.json"},  
  mark: "tick",  
  encoding: {  
    x: {  
      bin: true,  
      field: "temperature",  
      type: "quantitative"  
    }  
  }  
}
```



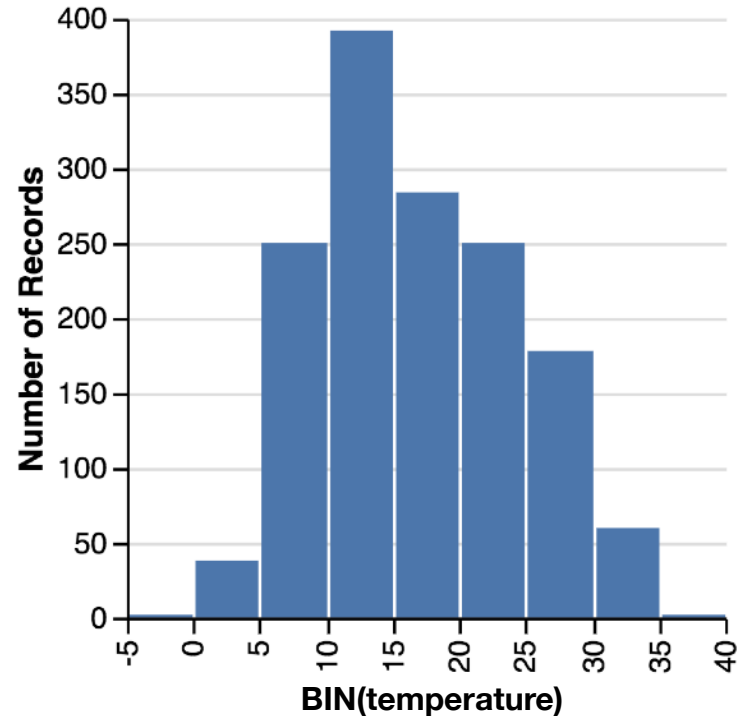
Histogram = (**Bar** with **x=bin** field, **y=count**)

```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    }
  }
}
```



Histogram = (Bar with x =binned field, y =count)

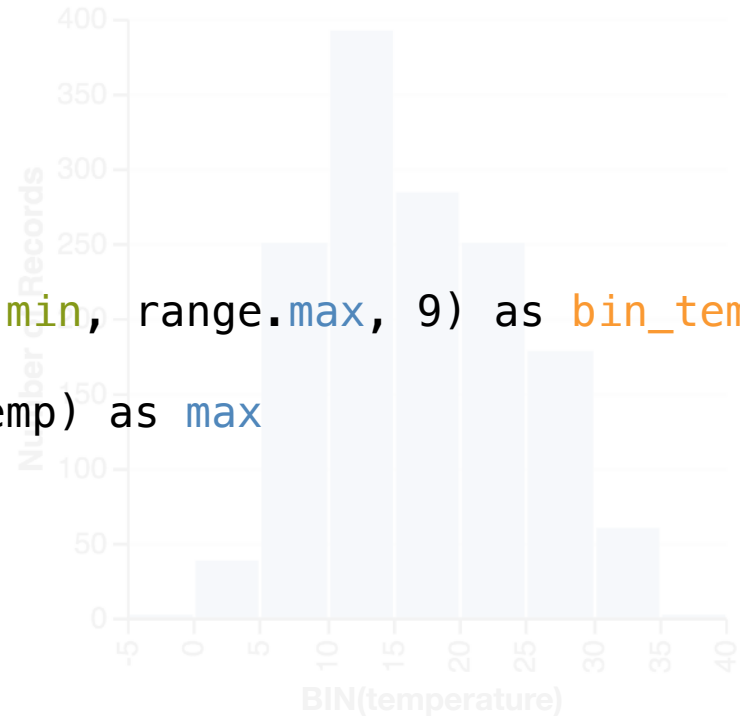
```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    }
  }
}
```



Histogram = (**Bar** with **x=binned field**, **y=count**)

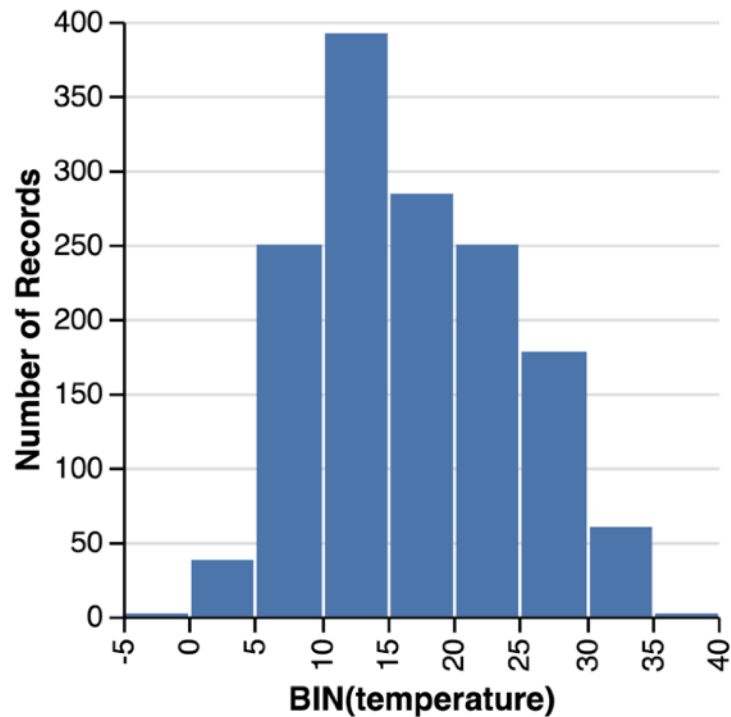
```
{  
  data: {url: "weather-seattle.json"},  
  mark: "bar",  
  encoding: {  
    x: {  
      bin: true,  
      type: "quantitative"  
    }  
  }  
}
```

```
SELECT bin_temp, count(*)  
FROM (  
  SELECT floor(weather.temp, range.min, range.max, 9) as bin_temp  
  FROM weather, (  
    SELECT min(temp) as min, max(temp) as max  
  ) range  
)
```



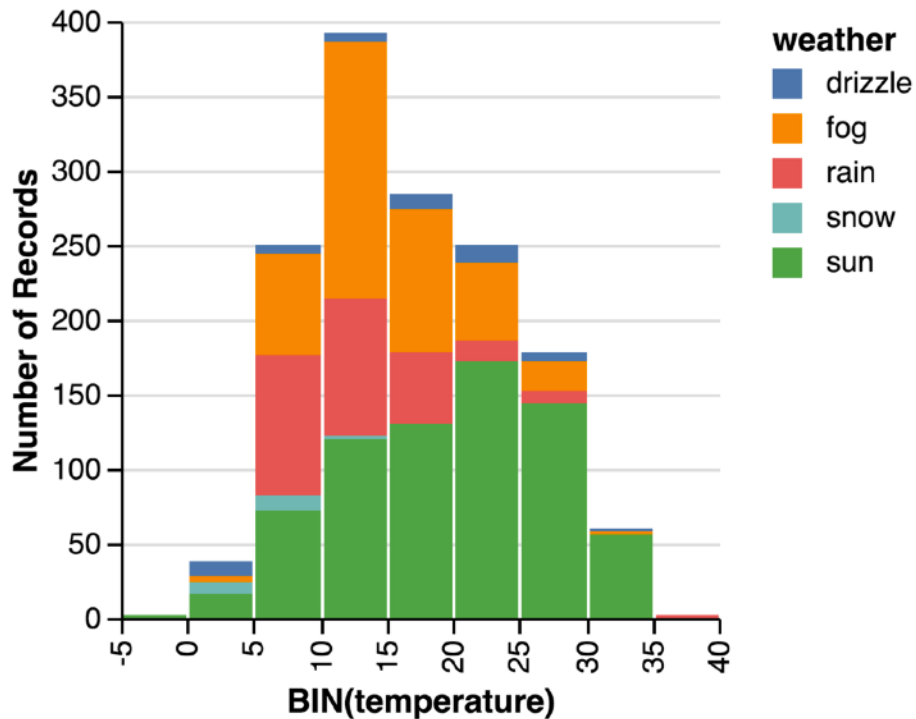
Histogram

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    }
  }
}
```



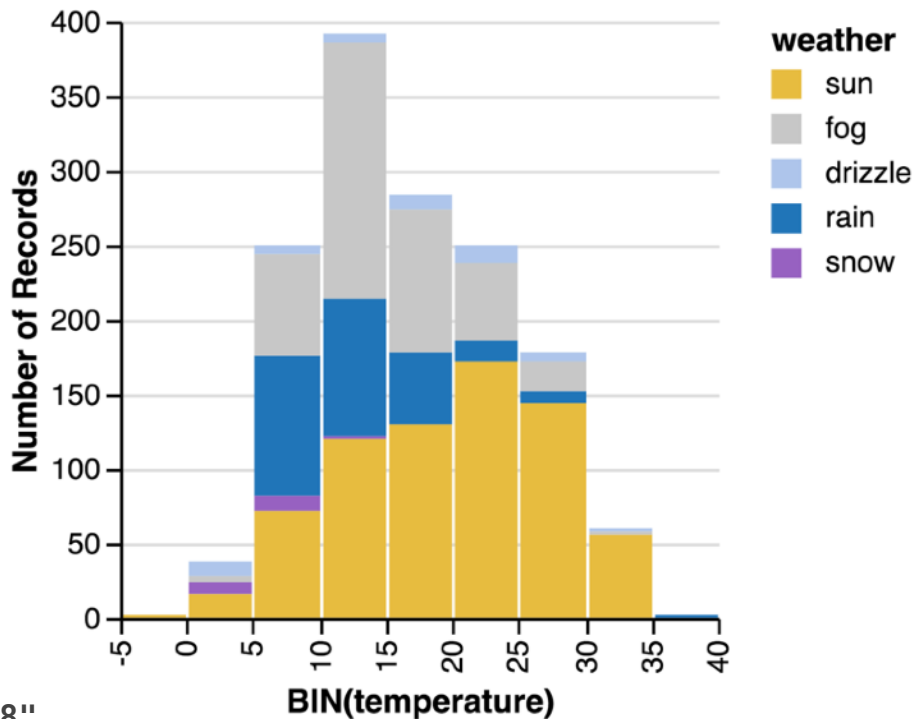
Histogram + Color

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    },
    color: {
      field: "weather",
      type: "nominal"
    }
  }
}
```



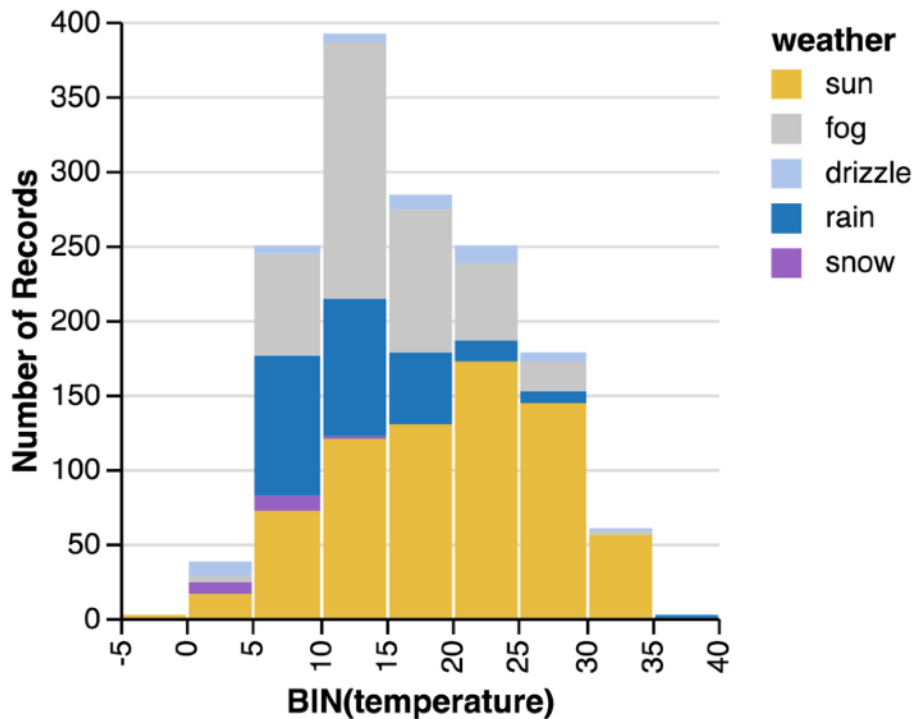
Histogram + Color

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    },
    color: {
      field: "weather",
      type: "nominal",
      "scale": {
        "domain": ["sun", "fog", "drizzle",
                  "rain", "snow"],
        "range": ["#e7ba52", "#c7c7c7", "#aec7e8",
                  "#1f77b4", "#9467bd"]
      }
    }
  }
}
```



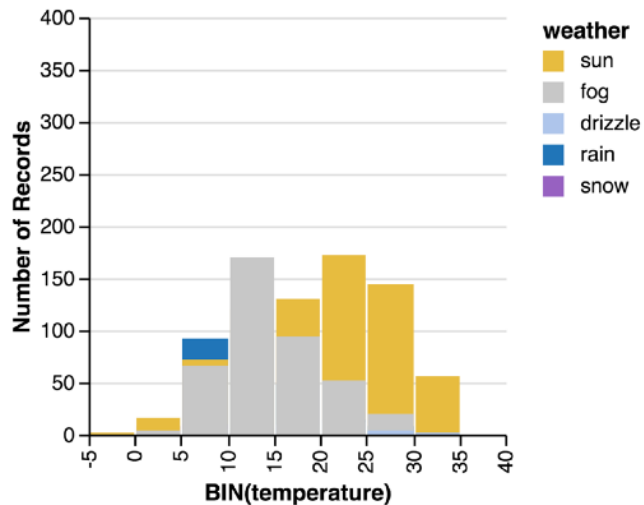
Histogram + Color = Stacked Histogram

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    },
    color: {
      field: "weather",
      type: "nominal",
      ...
    }
  }
}
```

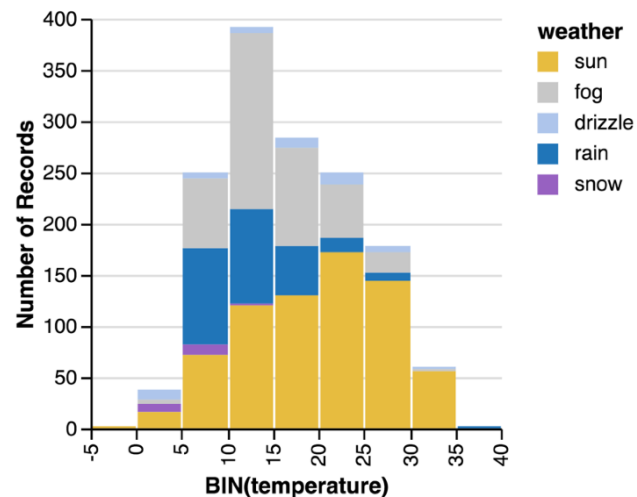


Stacked Histogram: **Sensible Defaults**

no stack



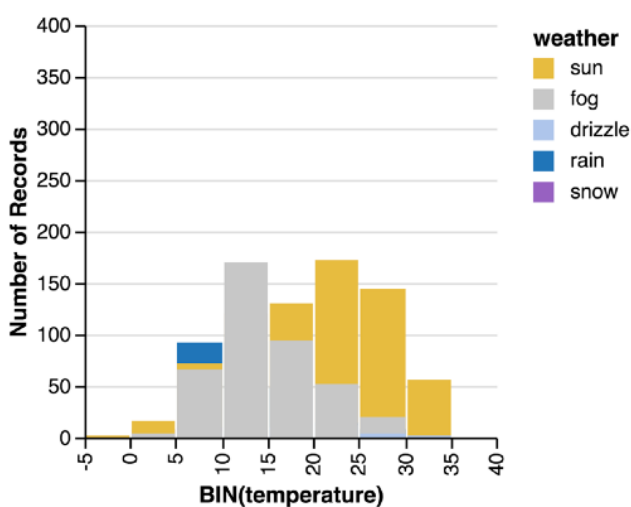
stack (default)



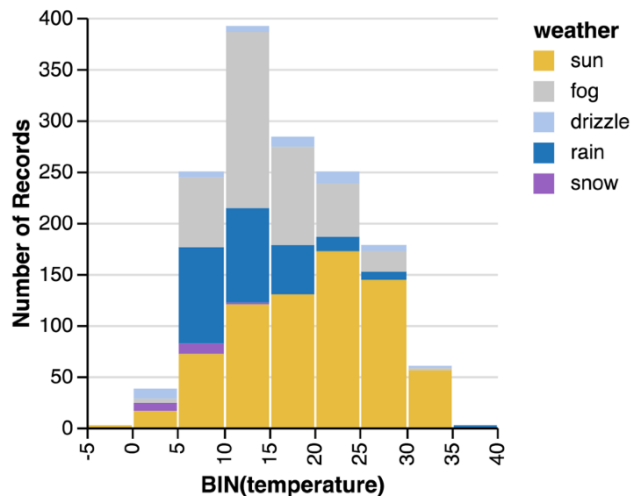
Stacked Histogram: **Sensible Defaults**

Channel (color) + Mark (bar) automatically enables stacking: a layout transform.

no stack



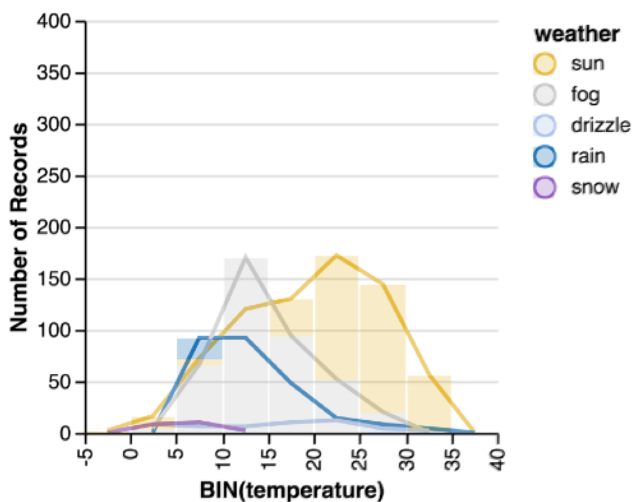
stack (default)



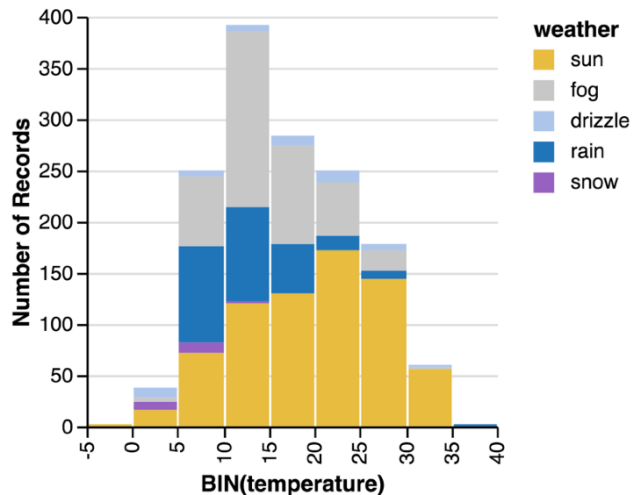
Stacked Histogram: **Sensible Defaults**

Channel (color) + Mark (bar) automatically enables stacking: a layout transform.

no stack → **overlap**

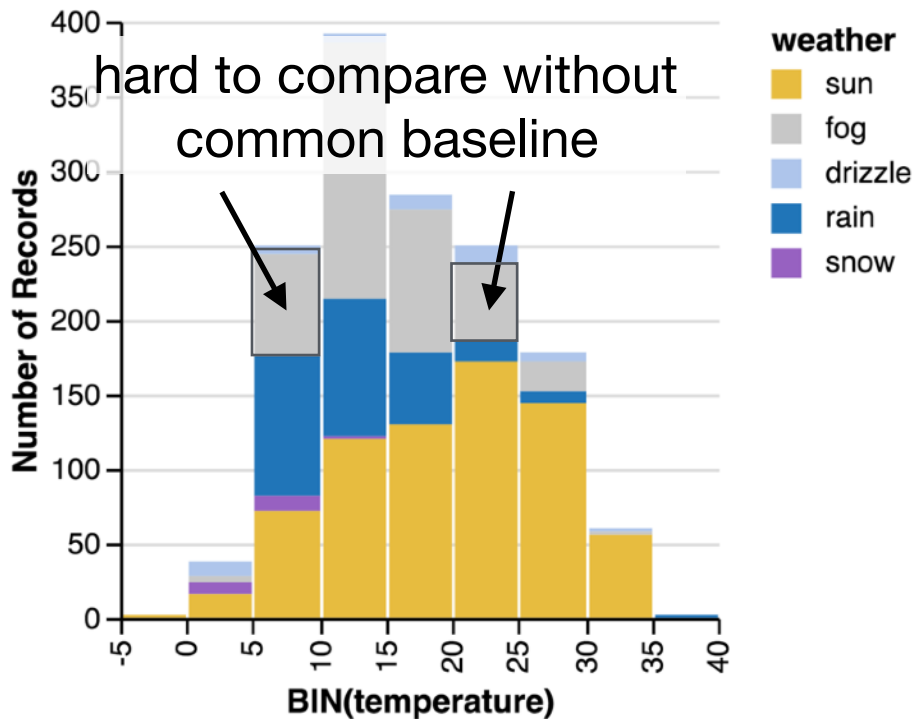


stack (default)



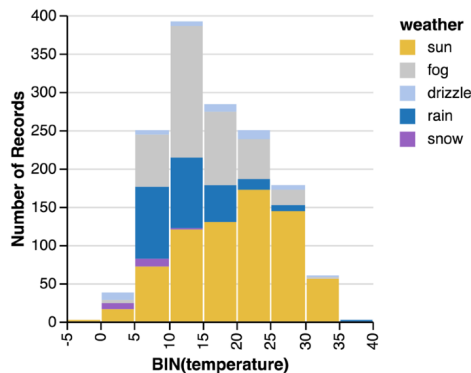
Histogram + Color = Stacked Histogram

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    },
    color: {
      field: "weather",
      type: "nominal"
    }
  }
}
```



Histogram + Color = Stacked Histogram

```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {bin: true, field: "temperature", type: "quantitative"},
    y: {aggregate: "count", type: "quantitative"},
    color: { field: "weather", type: "nominal"}
  }
}
```



Histogram + Column = Trellis Histogram

```
{  
  data: {url: "weather-seattle.json"},  
  mark: "bar",  
  encoding: {  
    x: {bin: true, field: "temperature", type: "quantitative"},  
    y: {aggregate: "count", type: "quantitative"},  
    column: {field: "weather", type: "nominal"}  
  }  
}
```

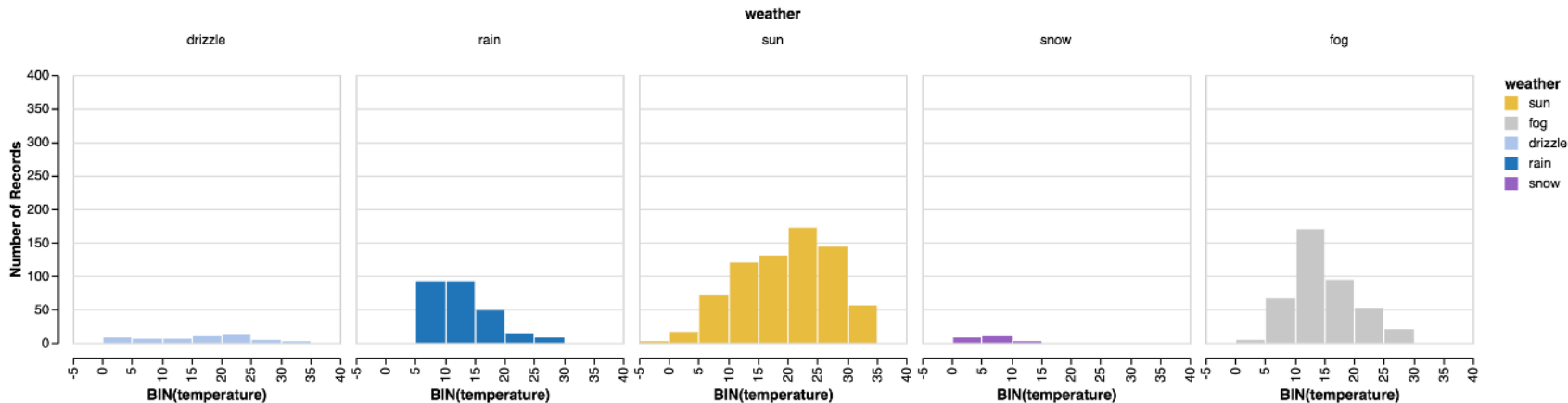


Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

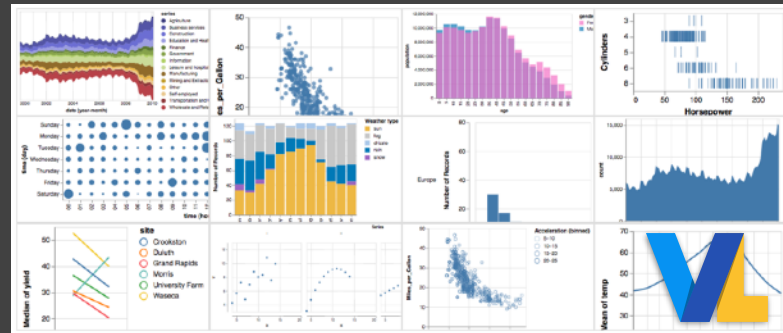
Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness



Facilitate **rapid exploration** with **concise** specifications by omitting low-level details.

Infer **sensible defaults** and customize by overriding defaults.

With **native** support for ***interaction!***

Ease-of-Use

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness

Ease-of-Use



Chart Typologies

Excel, Google Charts

Plotly

Bokeh

Seaborn

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness



Ease-of-Use

~~Chart Typologies~~

~~Excel, Google Charts~~

Visual Analysis Grammars

ggplot2, Observable Plot, Vega-Lite

Visualization Libraries

Matplotlib, D3, Vega

Component Architectures

VTK, Prefuse

Graphics & Event APIs

Processing, OpenGL, Java2D

Expressiveness



Visualization Guidance
Tableau, Voyager, Lux, LLMs, ...

Visual Analysis Grammars
 ggplot2, Observable Plot, Vega-Lite

Visualization Libraries
 Matplotlib, D3, Vega

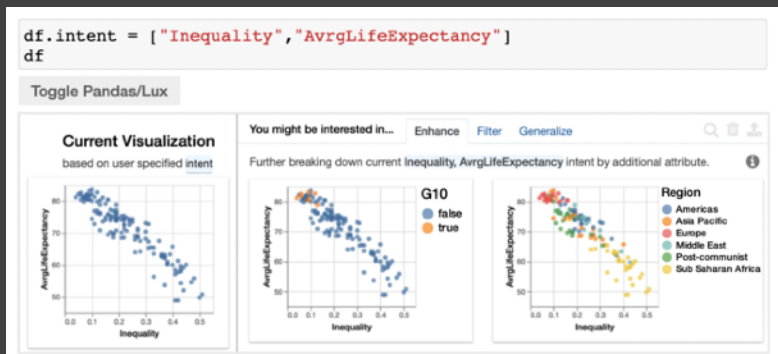
Component Architectures
 VTK, Prefuse

Graphics & Event APIs
 Processing, OpenGL, Java2D

Expressiveness



Voyager, Wongsuphasawat et al. 2017



Lux, Lee et al. 2021

Exercise & Assignment

Hours of Sunshine

The climate of a place can have a tremendous impact on people's lived experience. You will examine average monthly climate measurements for six major U.S. cities, roughly covering the edges of the continental United States.

Our in-class **exercise** is to get hands-on experience creating and publishing visualizations. You will create and revise a line chart of average monthly sunshine hours for six U.S. cities.

The **assignment** is to then design your own graphic.

Hours of Sunshine Exercise

Complete the exercises in the activity document.
Submit your results using the provided form link.

You may collaborate in groups of 1-3 people.

The course staff is here to help! Don't hesitate to ask us questions spanning design, tech, or more.

A1: Expository Visualization

Using the given climate data set...

Pick a **guiding question**, use it to title your vis.

Design a **static visualization** for that question.

You are free to **use any visualization tool**.

Deliverables via Gradescope

Image of your visualization (PNG or JPG format)

Short description + design rationale (≤ 4 paragraphs)

Due by **EOD, Tue January 13**.