Charticulator: Interactive Construction of Bespoke Chart Layouts

Donghao Ren[1,2], Bongshin Lee[2], and Matthew Brehmer[2]

[1] University of California, Santa Barbara
[2] Microsoft Research

Boston Weather

World Greenhouse Gas Emissions in 2005
Total: 44,153 MtCO2

Polio Incidence Rates
United States, 1950s

2017

Scaly
Smooth
Fibrous

None   Foul   Creosote

China
United States
India
Russian Fede
Japan
Korea, Repub
Iran (Islami
Indonesia
Saudi Arabi
Fr

iOS
kBerry OS
Unknown
Android
Playstation
Other

1951   1952
1957   1958
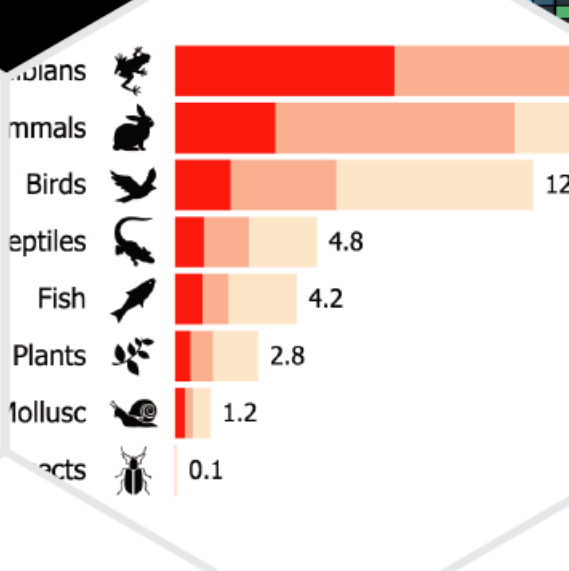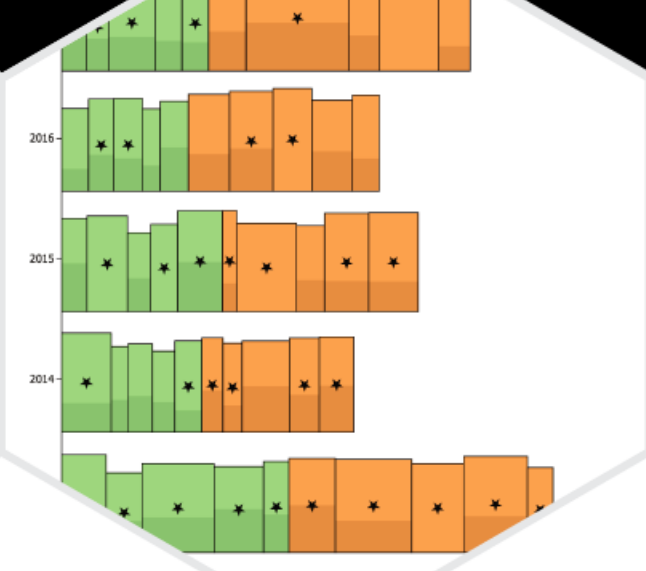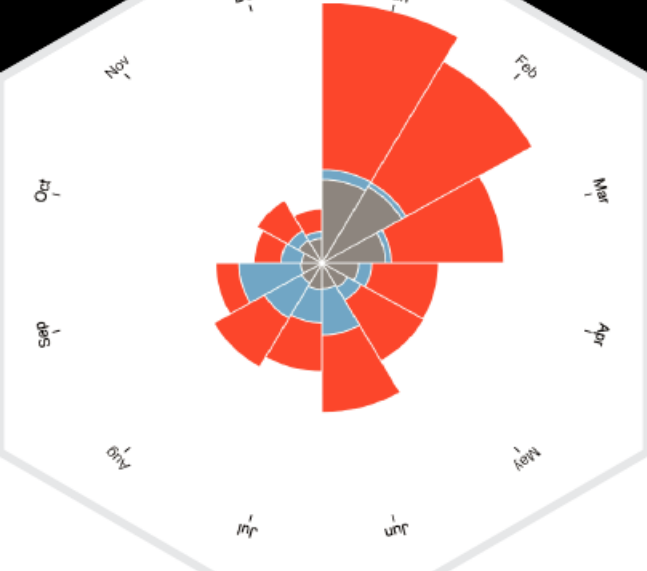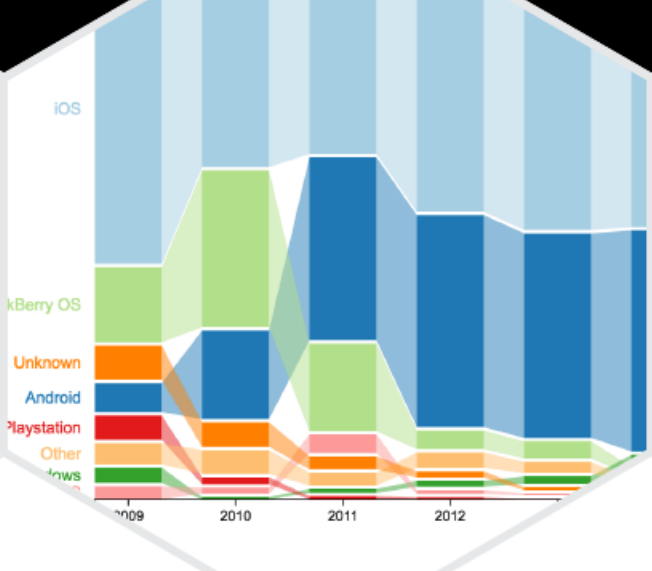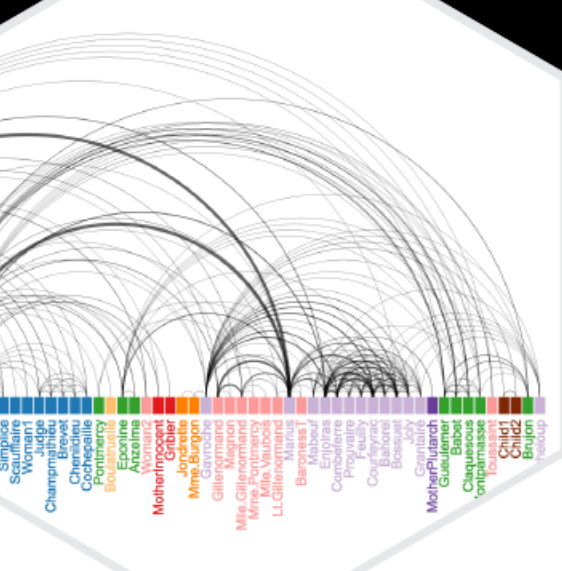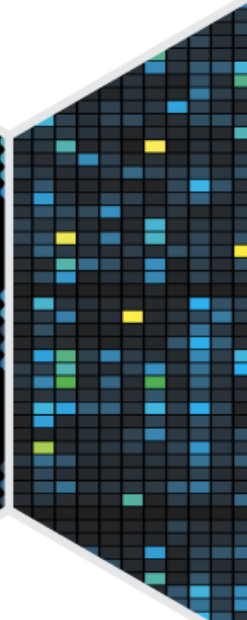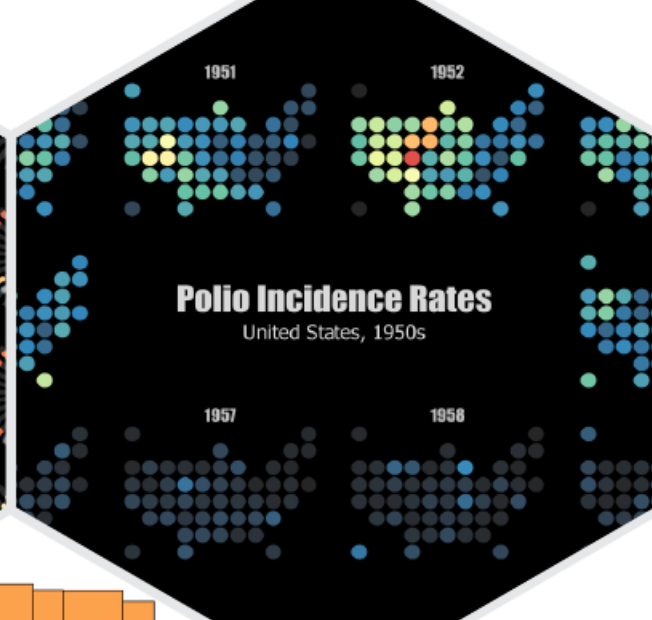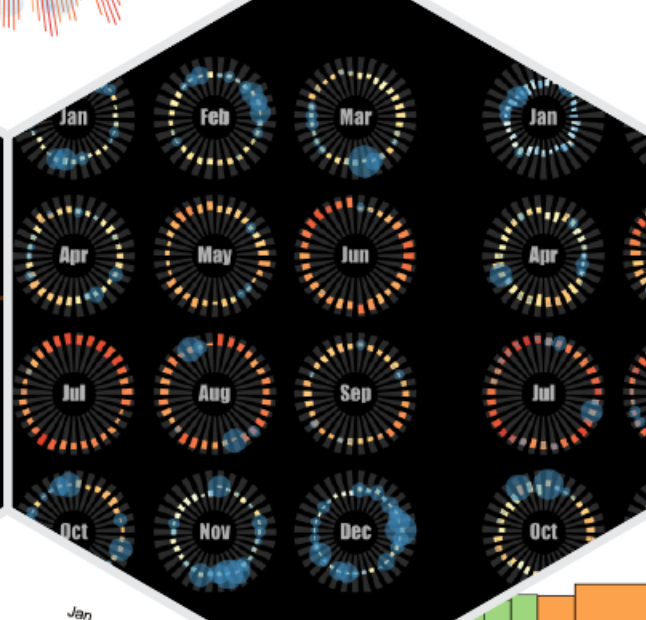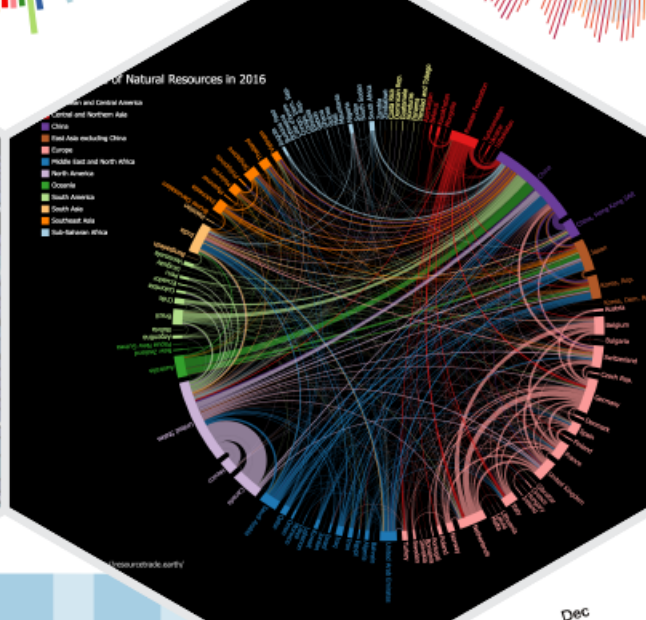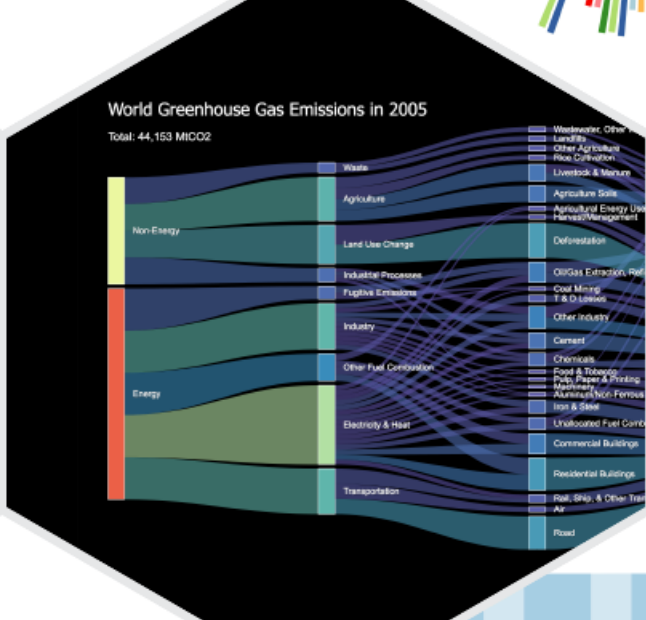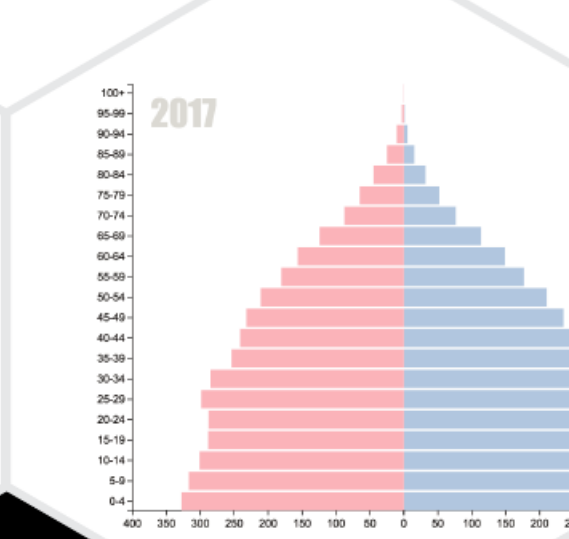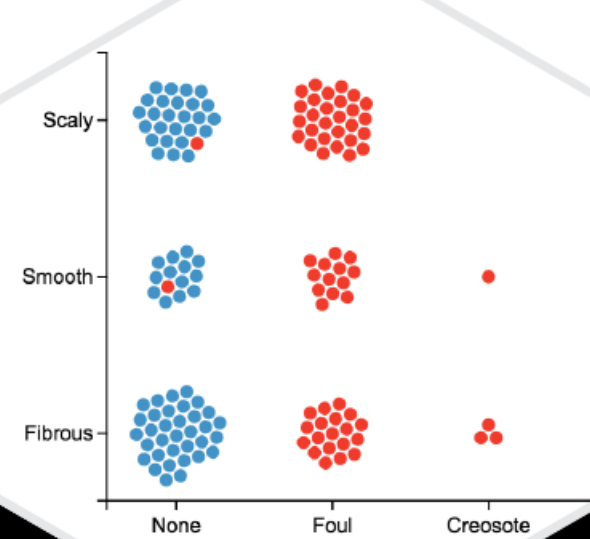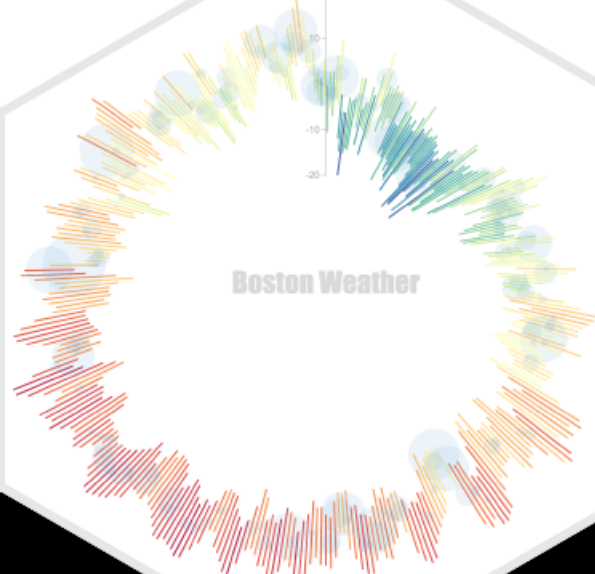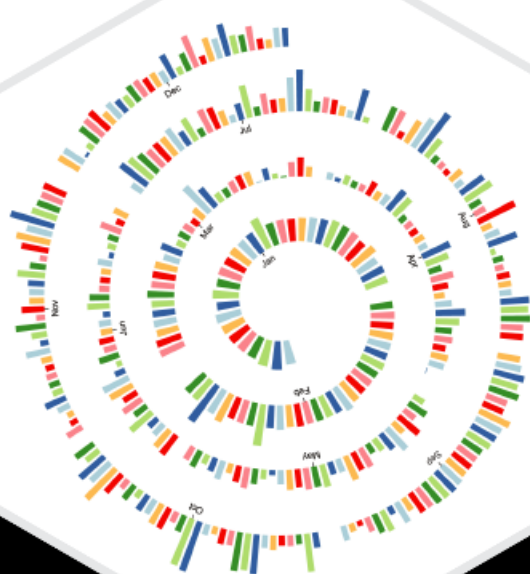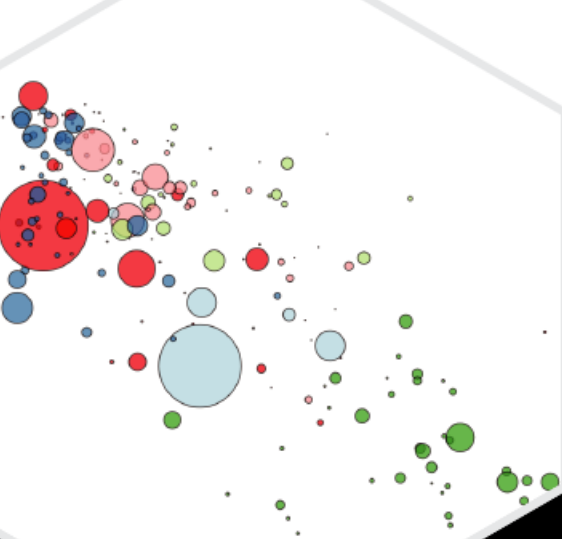
Brief overview of current tools

Template Editors
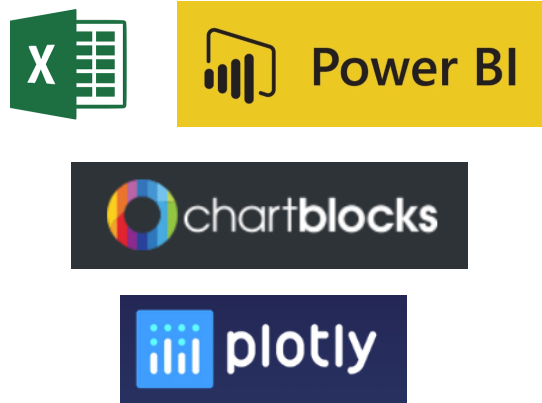
Shelf Configuration

Interactive Authoring

Data Illustrator
Lyra
iVisDesigner

Custom Code

Less expressive

Very expressive

Expressivity

Interactive authoring – Data Mapping
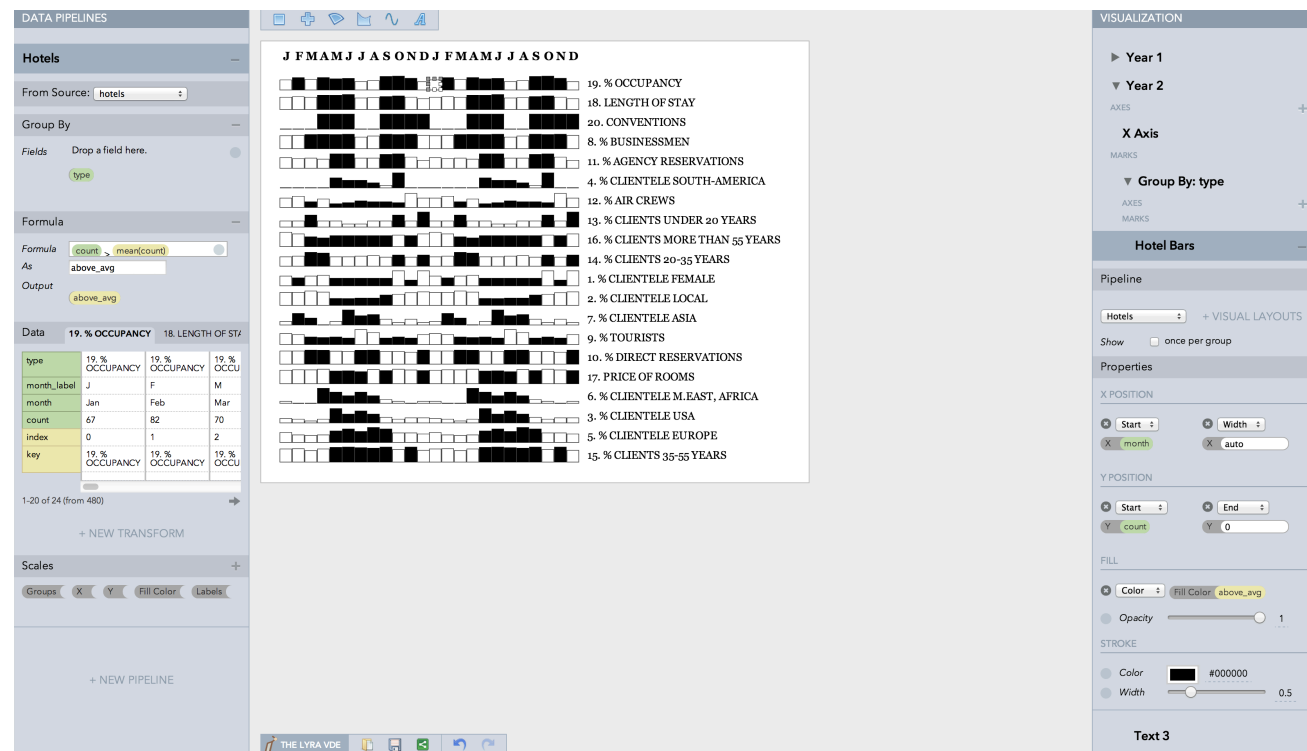
Lyra [A. Satyanarayan, J. Heer, 2014]
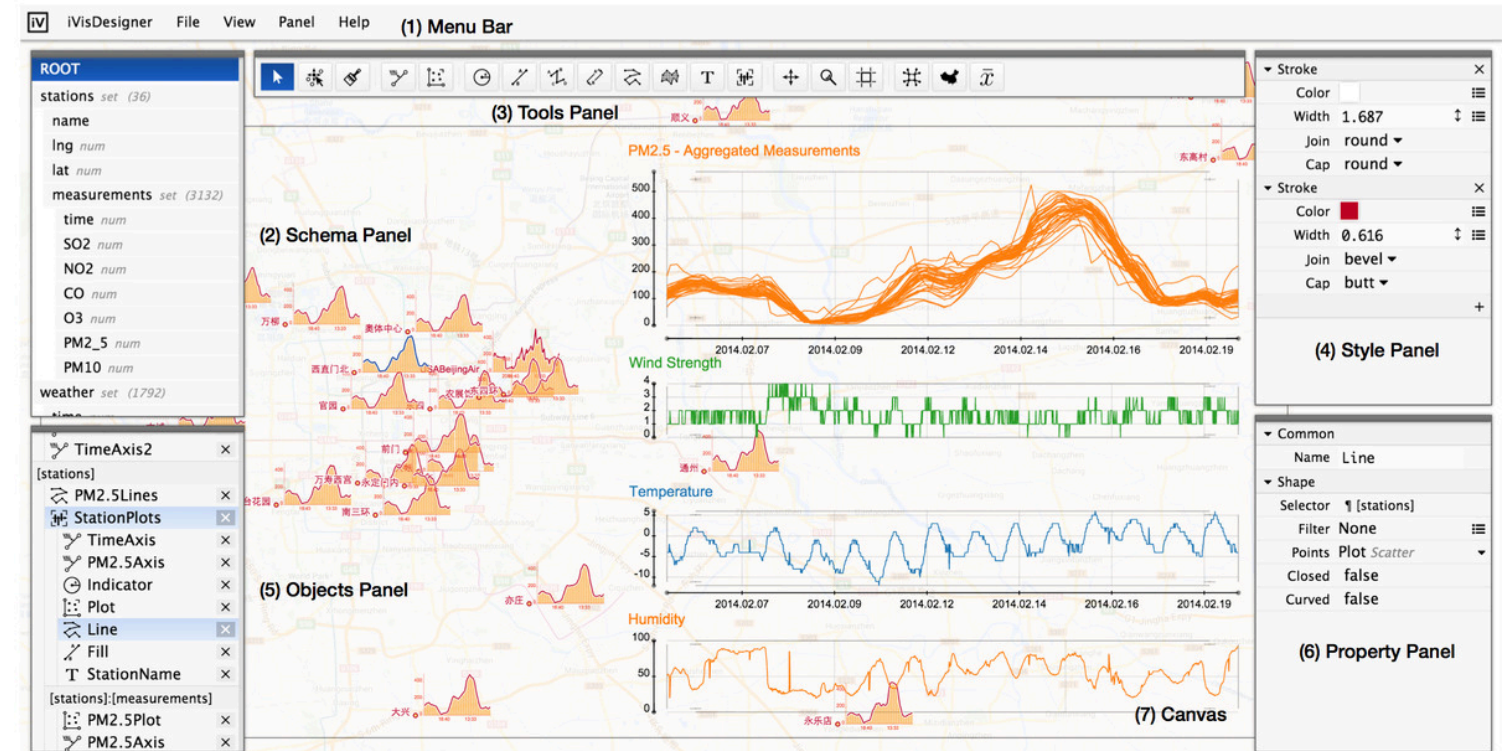
iVisDesigner [D. Ren et al., 2014]

Interactive authoring – Expressive Glyphs

Data-Driven Guides [N. W. Kim et al., 2016]
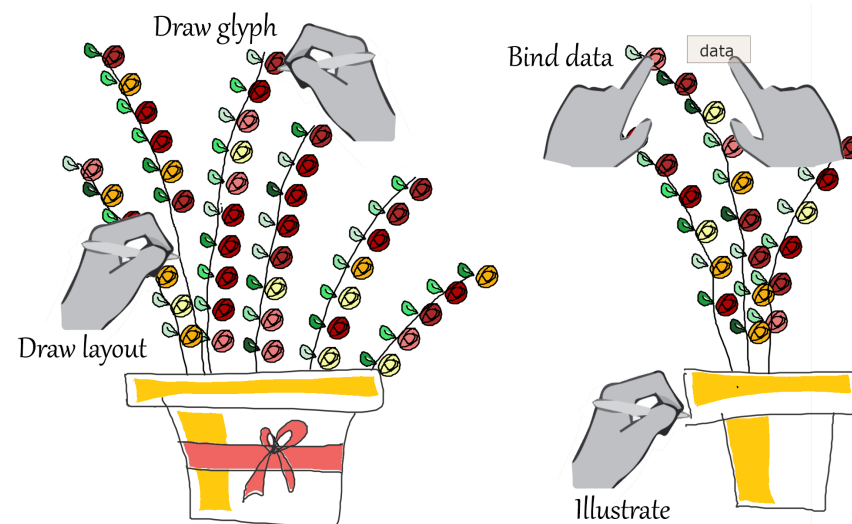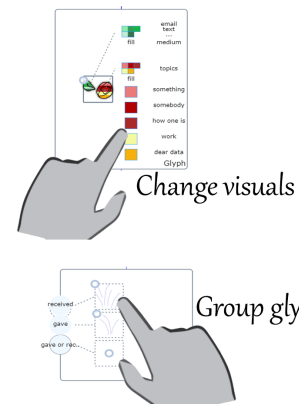
Data Ink [H. Xia et al., 2018]
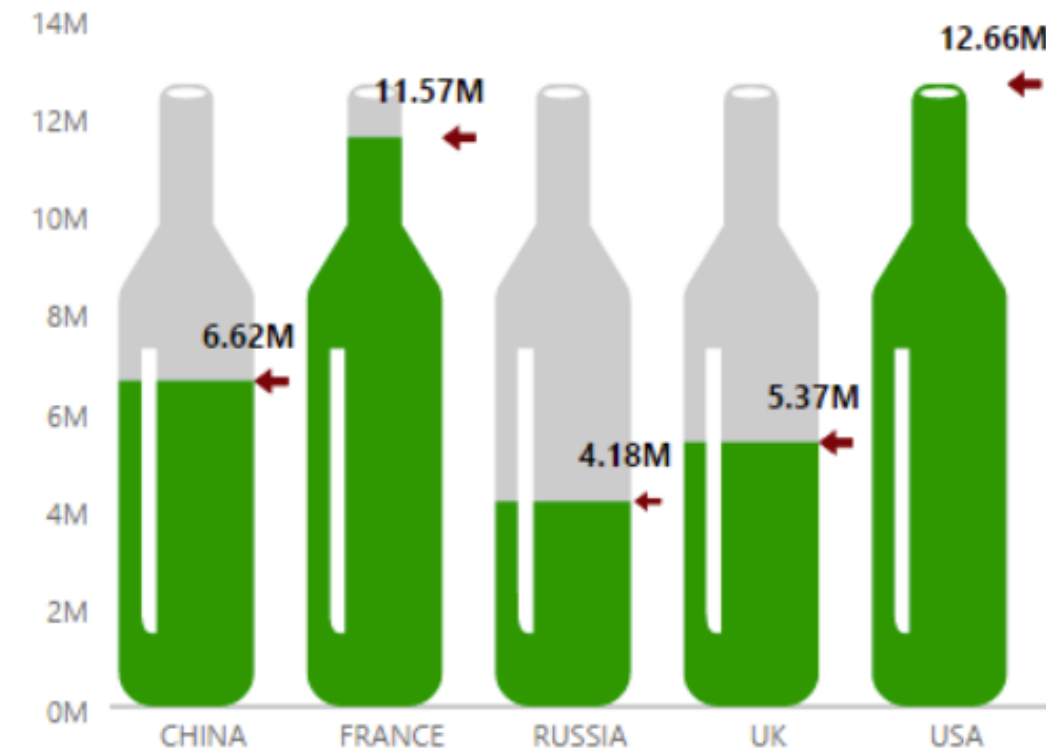
InfoNice [Y. Wang et al., 2018]

Interactive authoring – Partition & Repetition
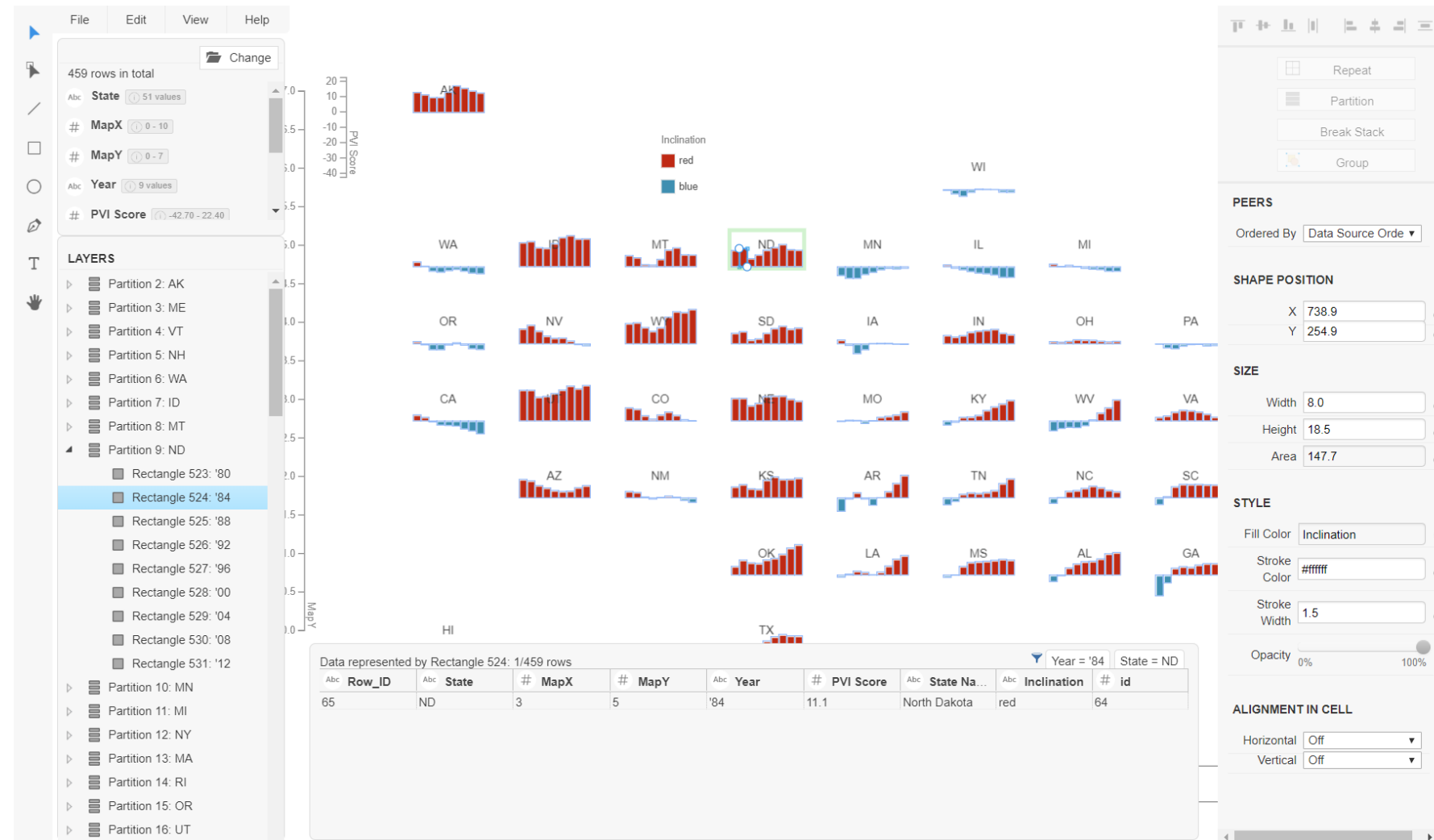
Data Illustrator [Z. Liu et al., 2018]

Layout and **Linking** are underexplored
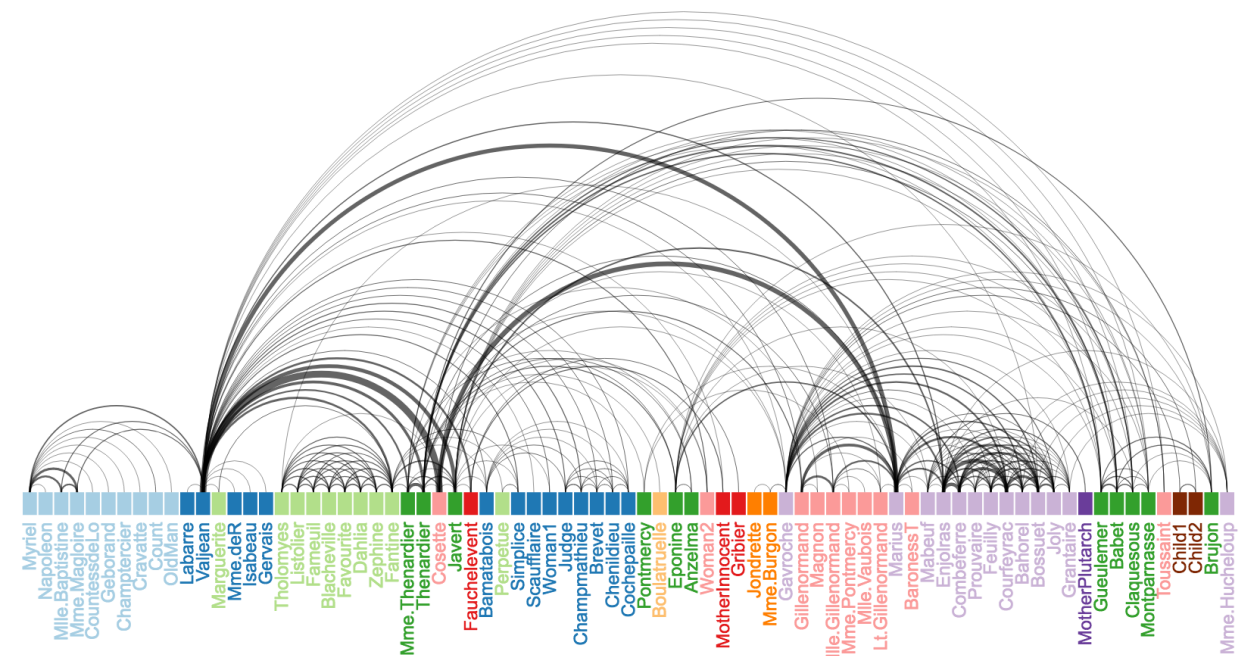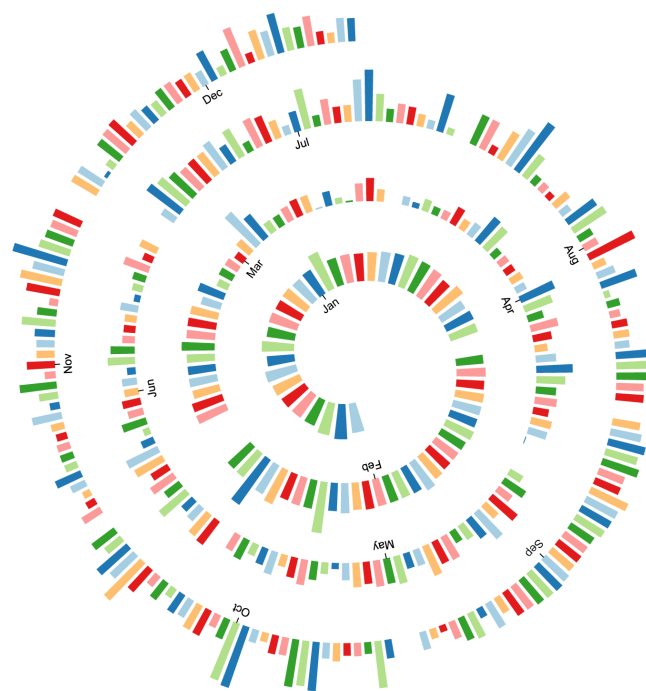
Charticulator = Chart + Articulator

- "Articulate" as Layout
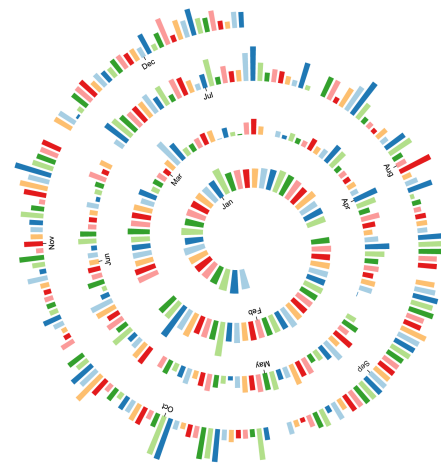
Spiral Layout

Nested Layout
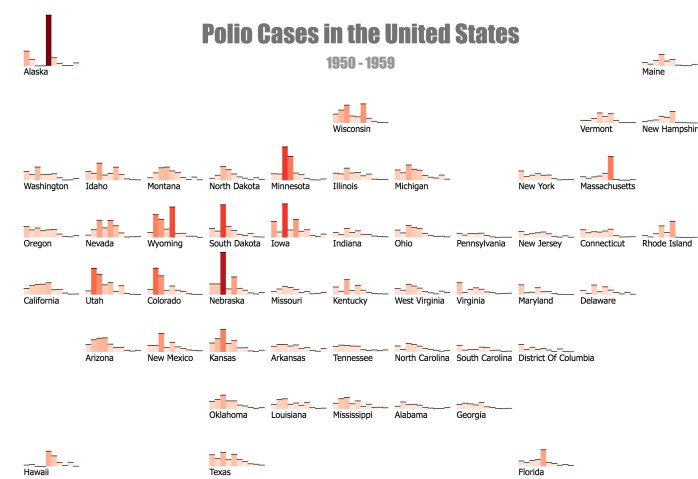
Circle Packing

Stacking

- "Articulate" as Linking

Connecting by Series
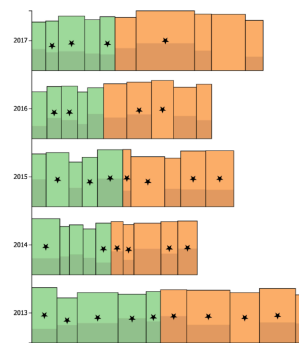
Connect with Link Data

Demo
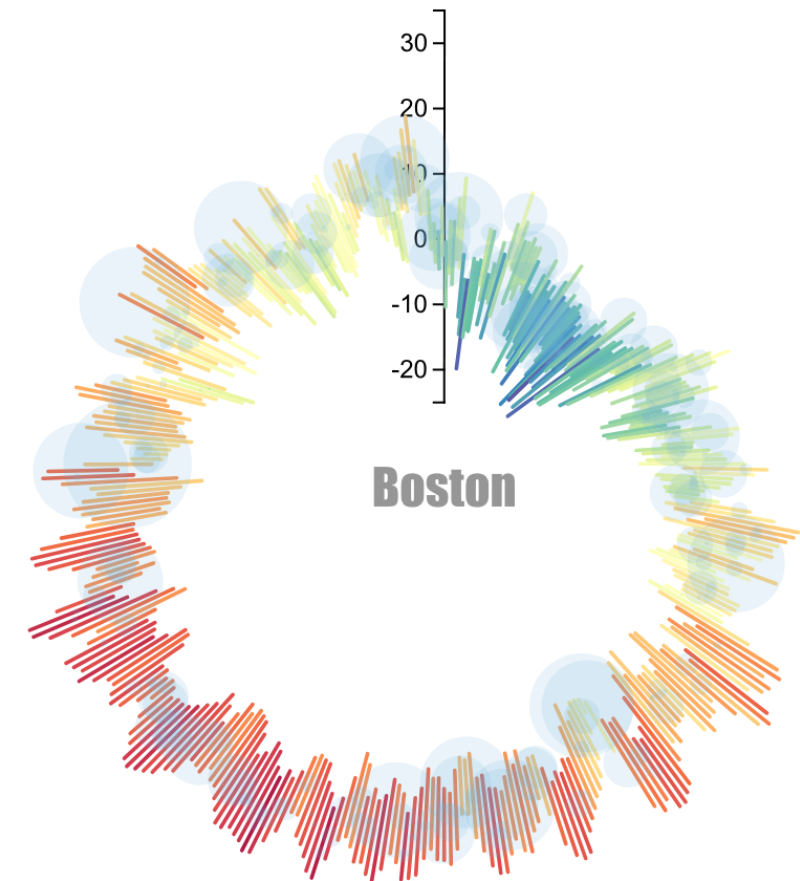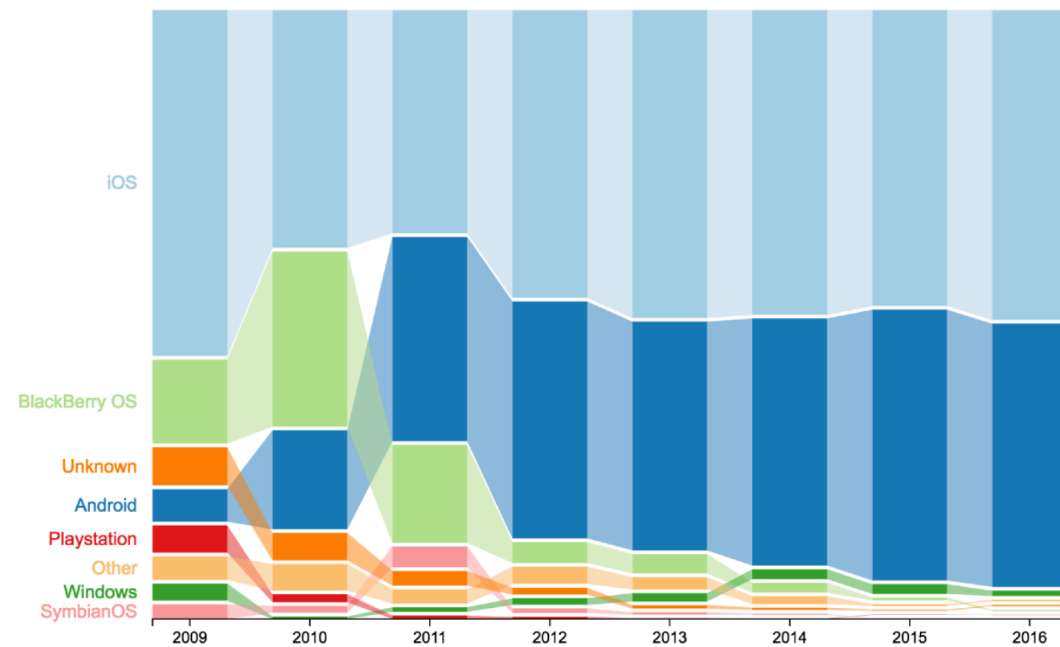
* Original Design: Weather Radials Poster by Raureif

Existing approaches for specifying layouts

- Layouts as special data transforms

```
┌──────────┐        ┌────────────────────┐        ┌──────────┐
│   Data   │───────▶│  Layout Transform  │───────▶│ Graphics │
└──────────┘        └────────────────────┘        └──────────┘
         data attributes              layout attributes
```
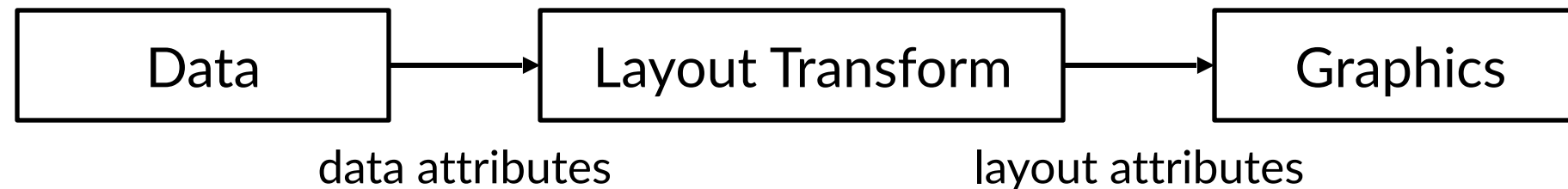
- Implications:
  - Hard to switch layout once graphics has been created
  - Difficult to combine layout transforms

- Most other approaches focus on specific visualizations
  - Graph, tree, word cloud, etc.

Compose complex layouts using partial specifications
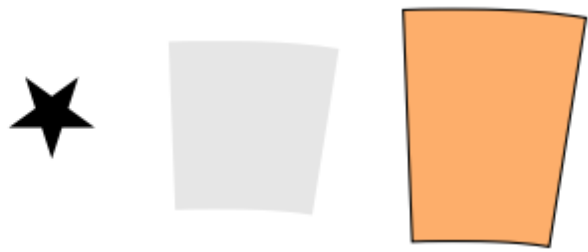
Design individual *marks*
e.g.,
Map data to width & height

Combine marks into a *Glyph*
e.g.,
Put the star at the center

Layout the glyphs
e.g.,
Stack the glyphs along a custom curve

* Design inspired by Best Bookshelf [Tanyoung Kim]

Compose complex layouts using partial specifications

Design individual *marks*
e.g.,
Map data to width & height

Combine marks into a *Glyph*
e.g.,
Put the star at the center

2013

2014

2015

2016

2017

Layout the glyphs
e.g.,
Group the glyphs by "Year",
within each year, stack horizontally

* Design inspired by Best Bookshelf [Tanyoung Kim]

Compose complex layouts using partial specifications

Design individual *marks*

e.g.,

~~Map data to width & height~~

Combine marks into a *Glyph*

e.g.,
Put the star at the center

Layout the glyphs

e.g.,
Group the glyphs by "Year",
within each year, stack horizontally

* Design inspired by Best Bookshelf [Tanyoung Kim]
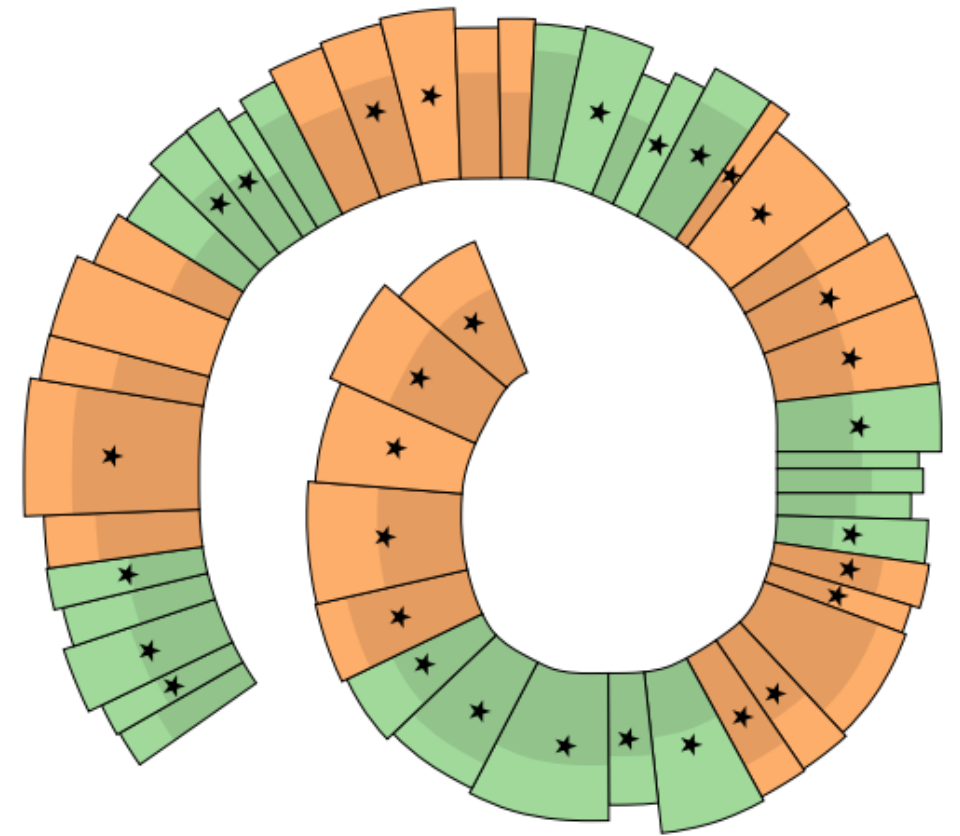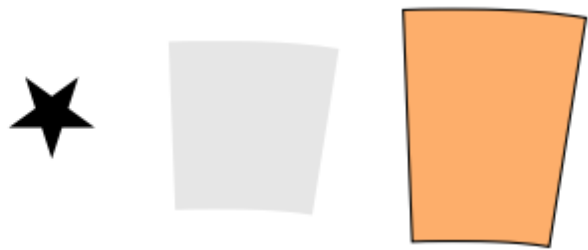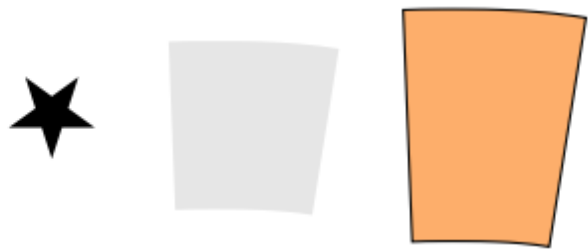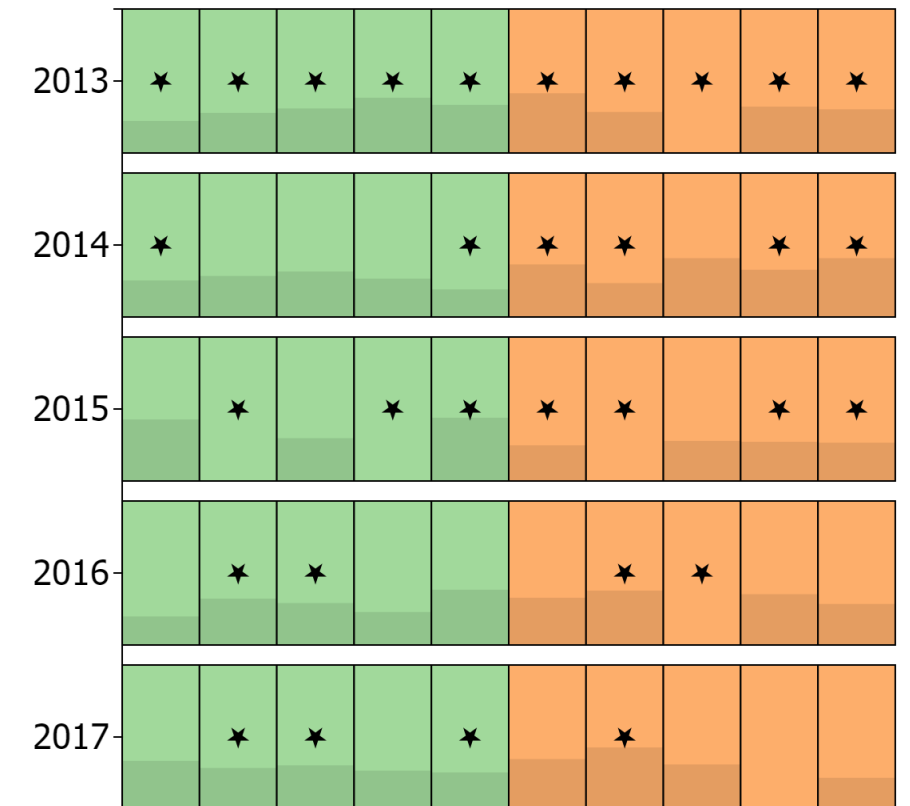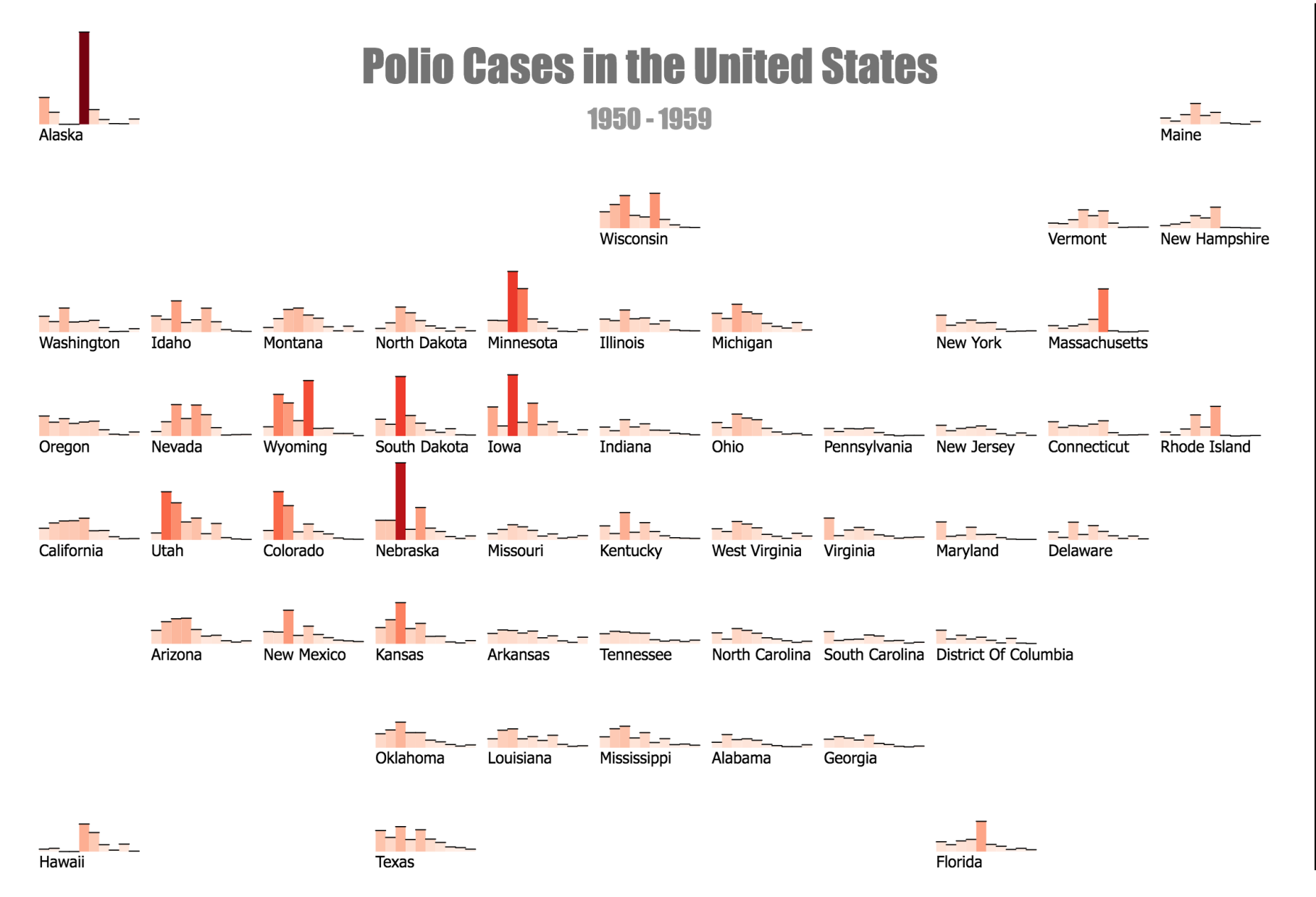
Compose complex layouts using partial specifications

Polio Cases in the United States

1950 - 1959

Alaska, Maine, Wisconsin, Vermont, New Hampshire, Washington, Idaho, Montana, North Dakota, Minnesota, Illinois, Michigan, New York, Massachusetts, Oregon, Nevada, Wyoming, South Dakota, Iowa, Indiana, Ohio, Pennsylvania, New Jersey, Connecticut, Rhode Island, California, Utah, Colorado, Nebraska, Missouri, Kentucky, West Virginia, Virginia, Maryland, Delaware, Arizona, New Mexico, Kansas, Arkansas, Tennessee, North Carolina, South Carolina, District Of Columbia, Oklahoma, Louisiana, Mississippi, Alabama, Georgia, Hawaii, Texas, Florida

Horizontal: Group by "MapX"

Vertical: Group by "MapY"

Massachusetts
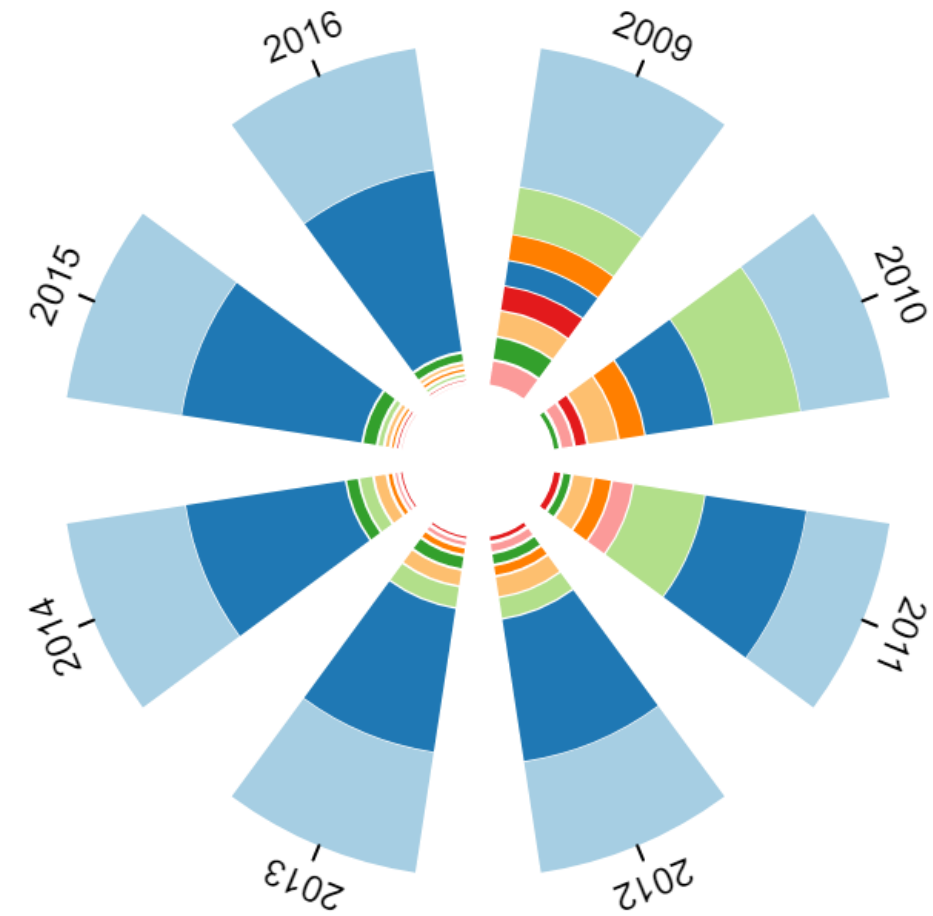
Horizontal layout
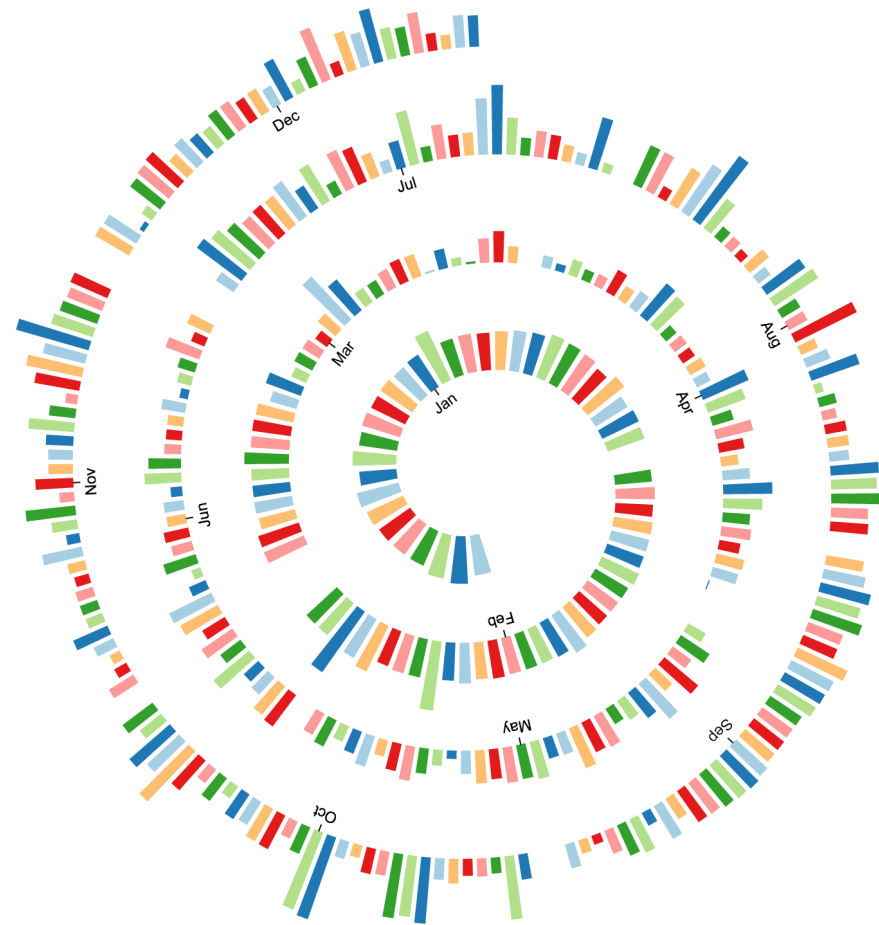for each cell

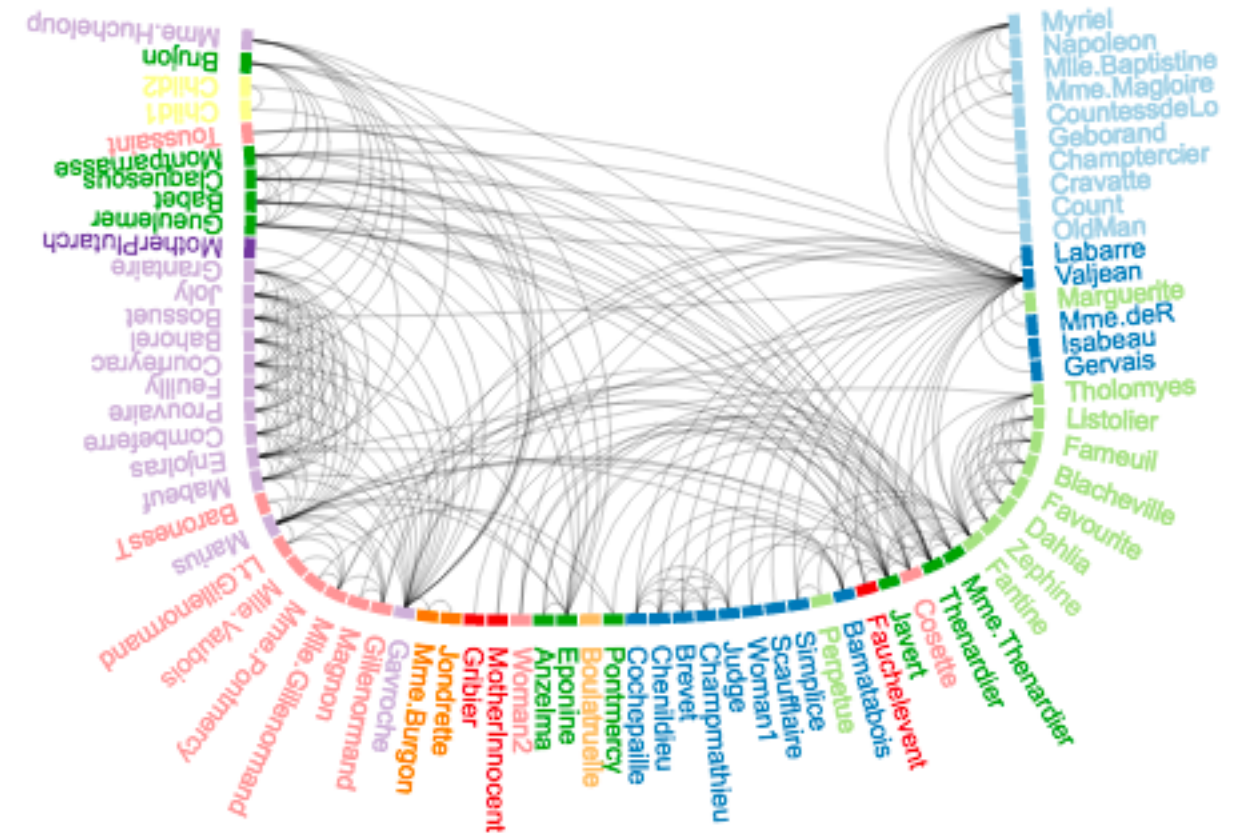# Coordinate Systems
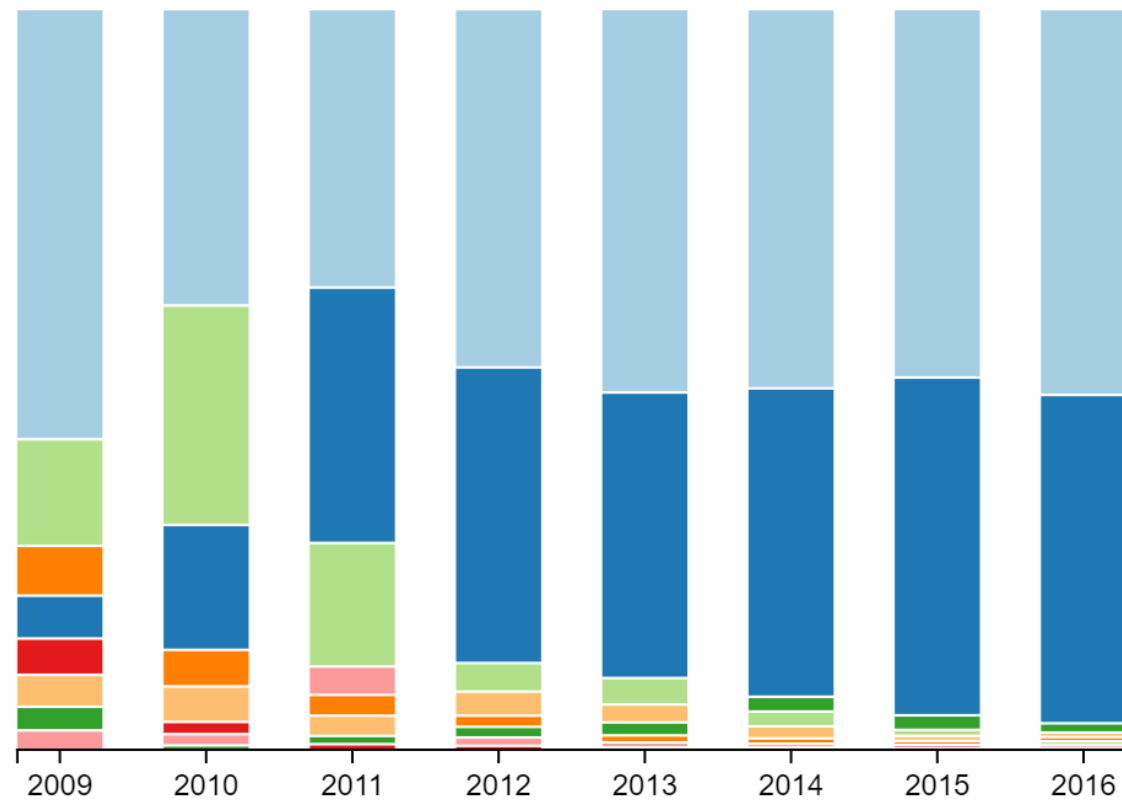
Cartesian Coordinates

Polar Coordinates

Coordinate Systems

Curve Coordinates (Spiral)

Character Co-occurrence in Les Miserables
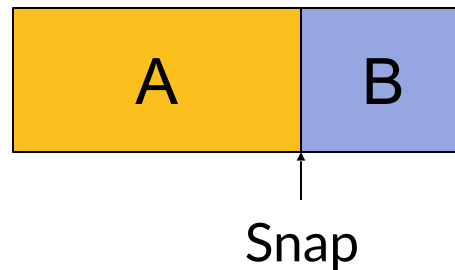
Curve Coordinates (Hand-drawn)

Implementation

- **Problem:** there is no direct way of computing the layout given a set of partial specifications

- **Our solution:** constraint-based layout algorithm

Constraint-based layout algorithm

Partial Layout Specification

Mathematical Constraints

Snap

$$A.x_2 = B.x_1$$

$$G1.x_2 + Gap = G2.x_1$$
$$G2.x_2 + Gap = G3.x_1$$
$$G3.x_2 + Gap = G4.x_1$$

Stack Horizontally

Constraint solver

- Sparse conjugate gradient algorithm

$$\min_{x} ||Ax - b||^2 + \lambda||x - x_0||^2$$

Constraints   Keep the current
positions

- Implemented using Eigen[*], compiled into WebAssembly for performance
- https://github.com/donghaoren/lscg-solver

[*] Eigen: http://eigen.tuxfamily.org/index.php?title=Main_Page

Creating reusable templates

- Constraint-based layout specification generates reusable designs
  - Adapt to different canvas sizes
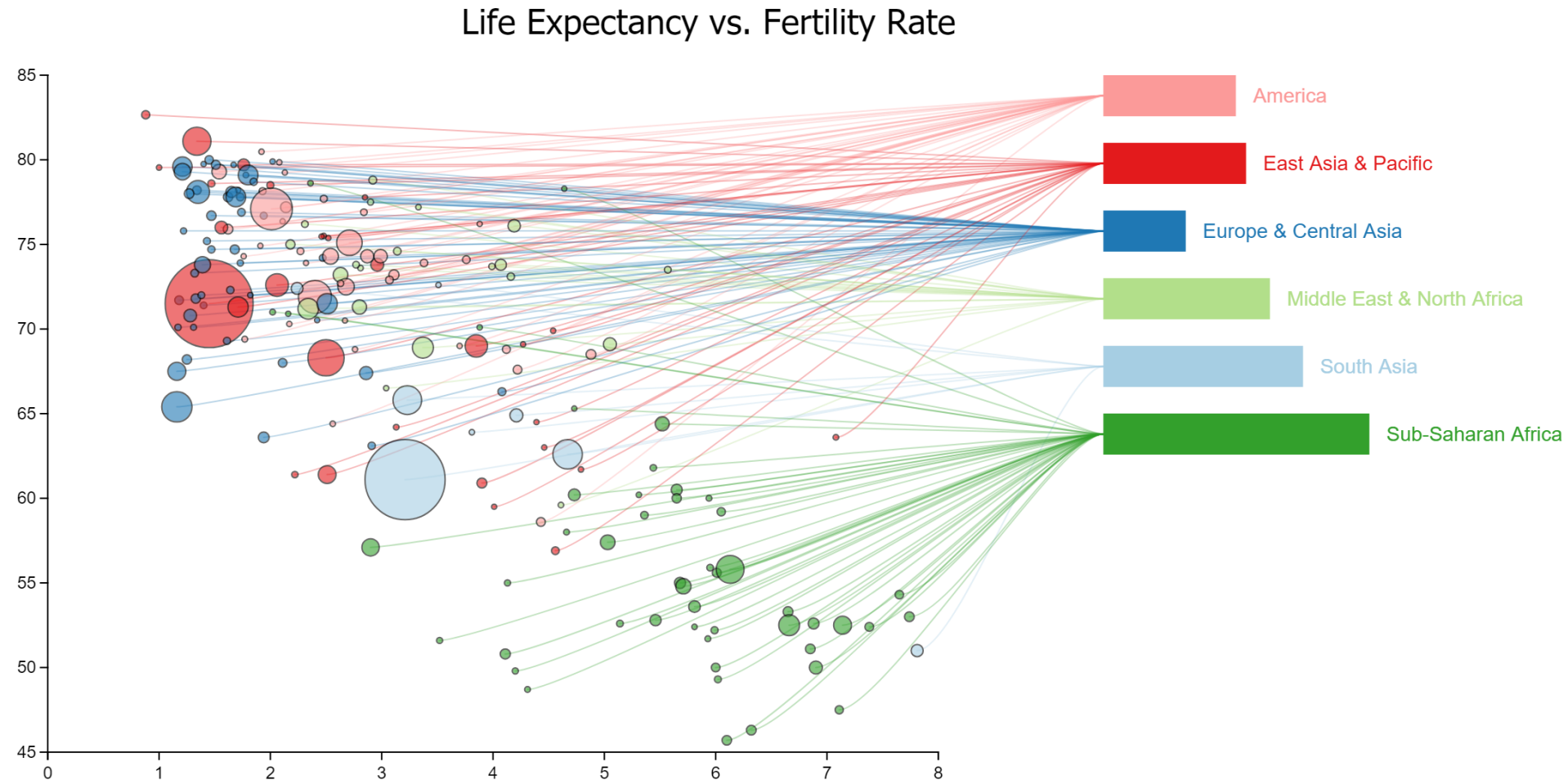 - Adapt to new (compatible) datasets
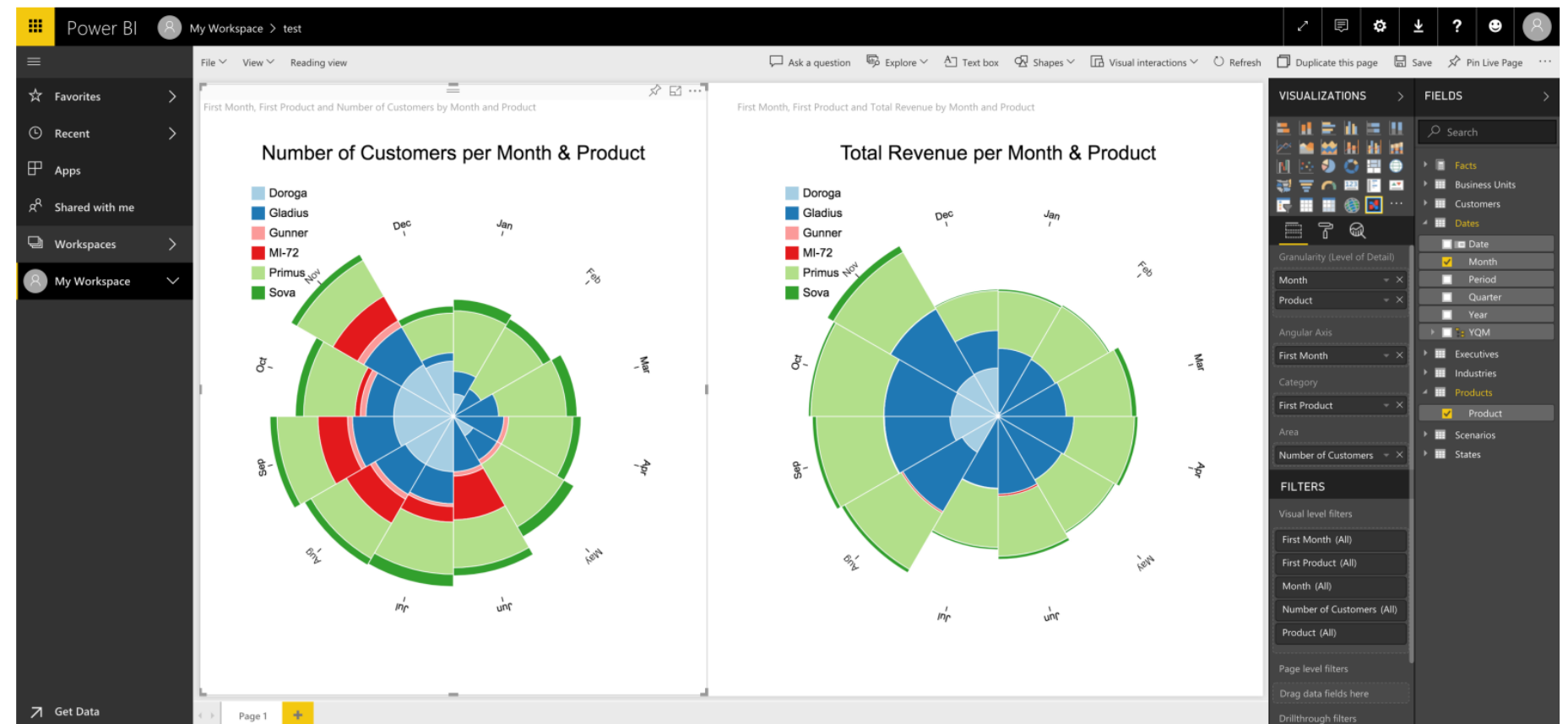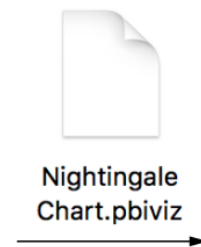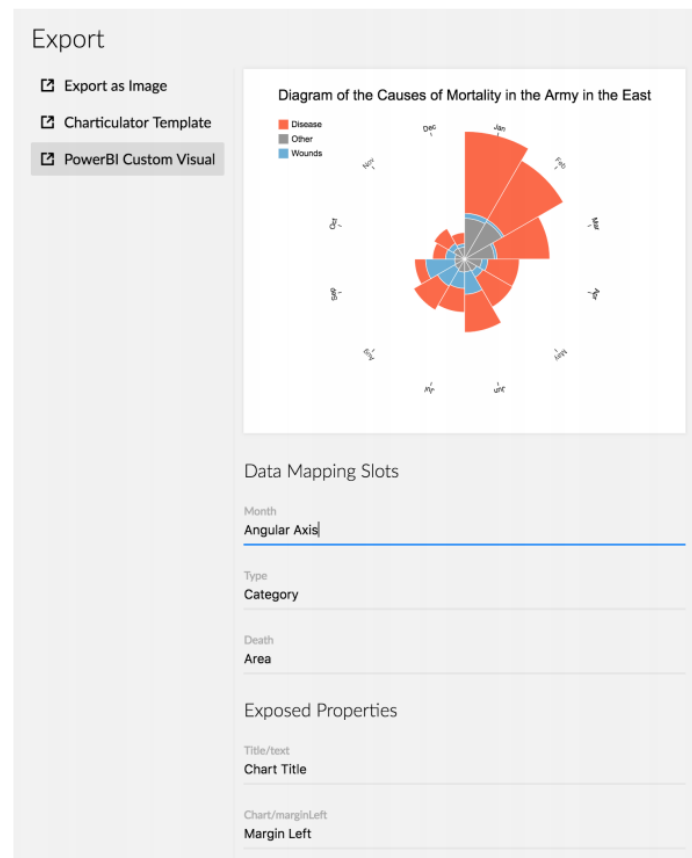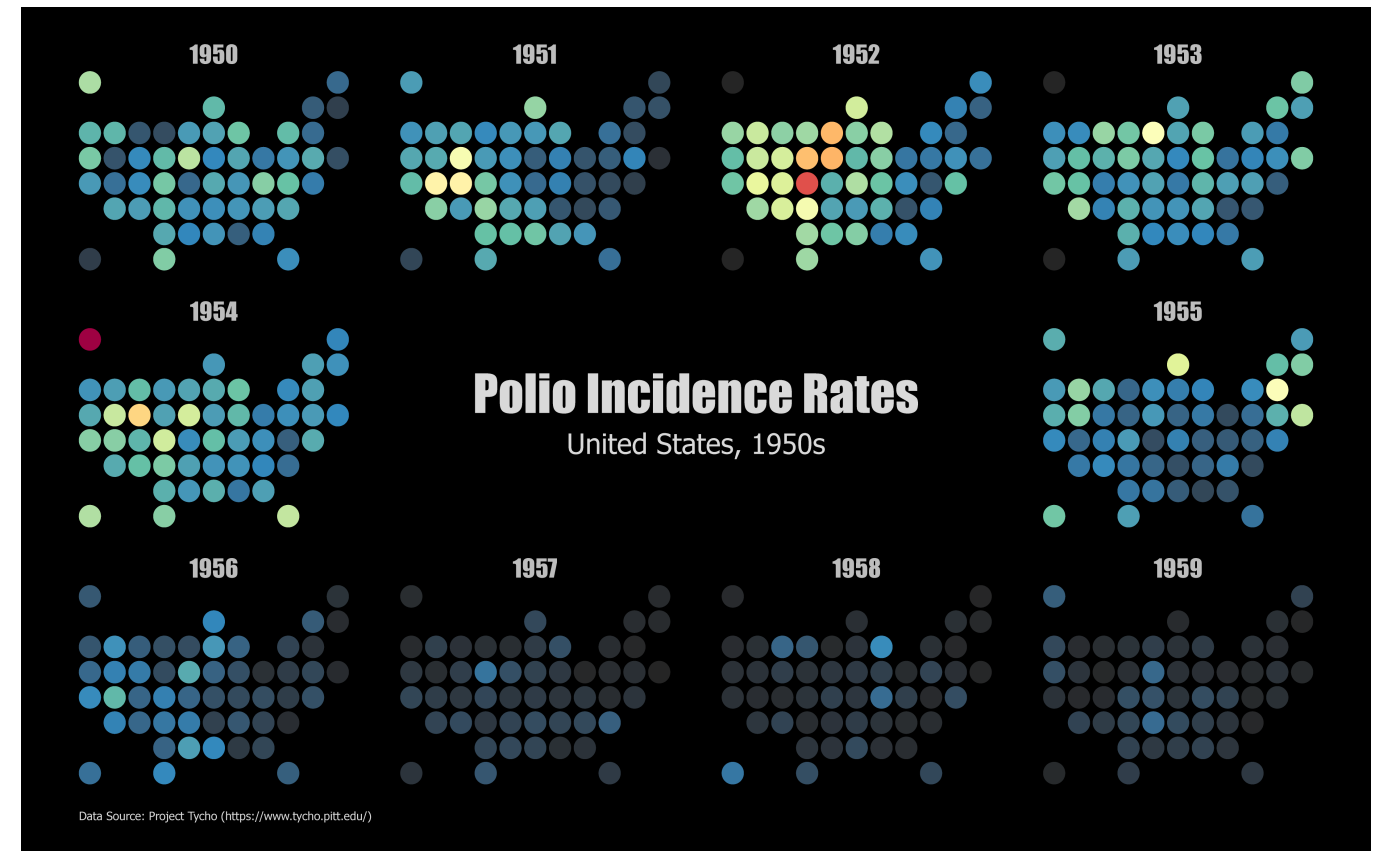

Charticulator

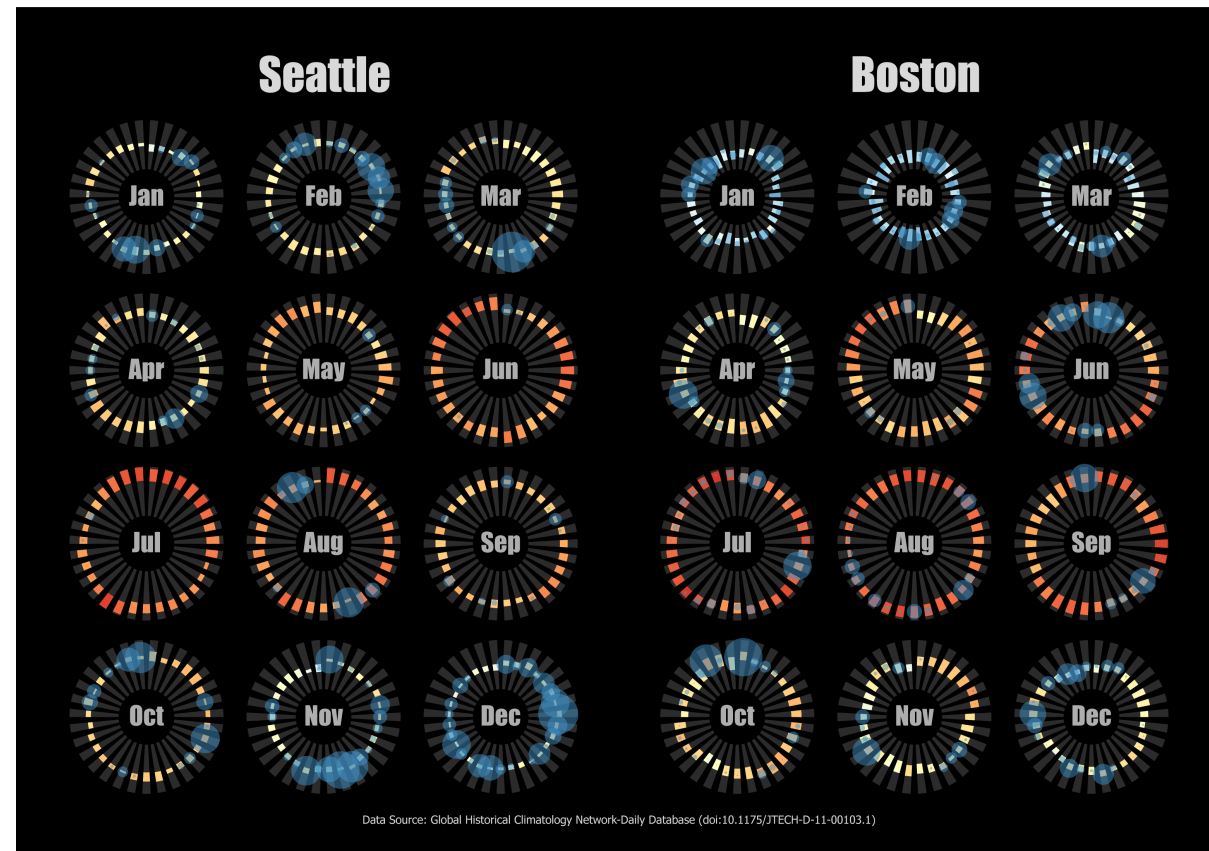Microsoft Power BI

Beyond the paper

Filter & Grouping & Multiple Glyphs

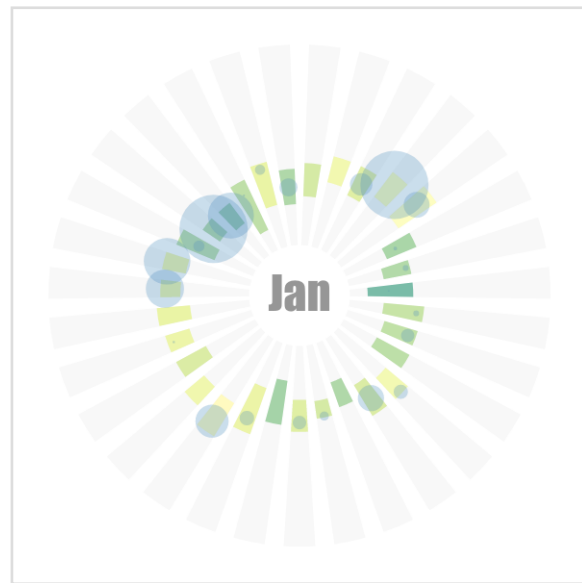- Support composite charts

Life Expectancy vs. Fertility Rate
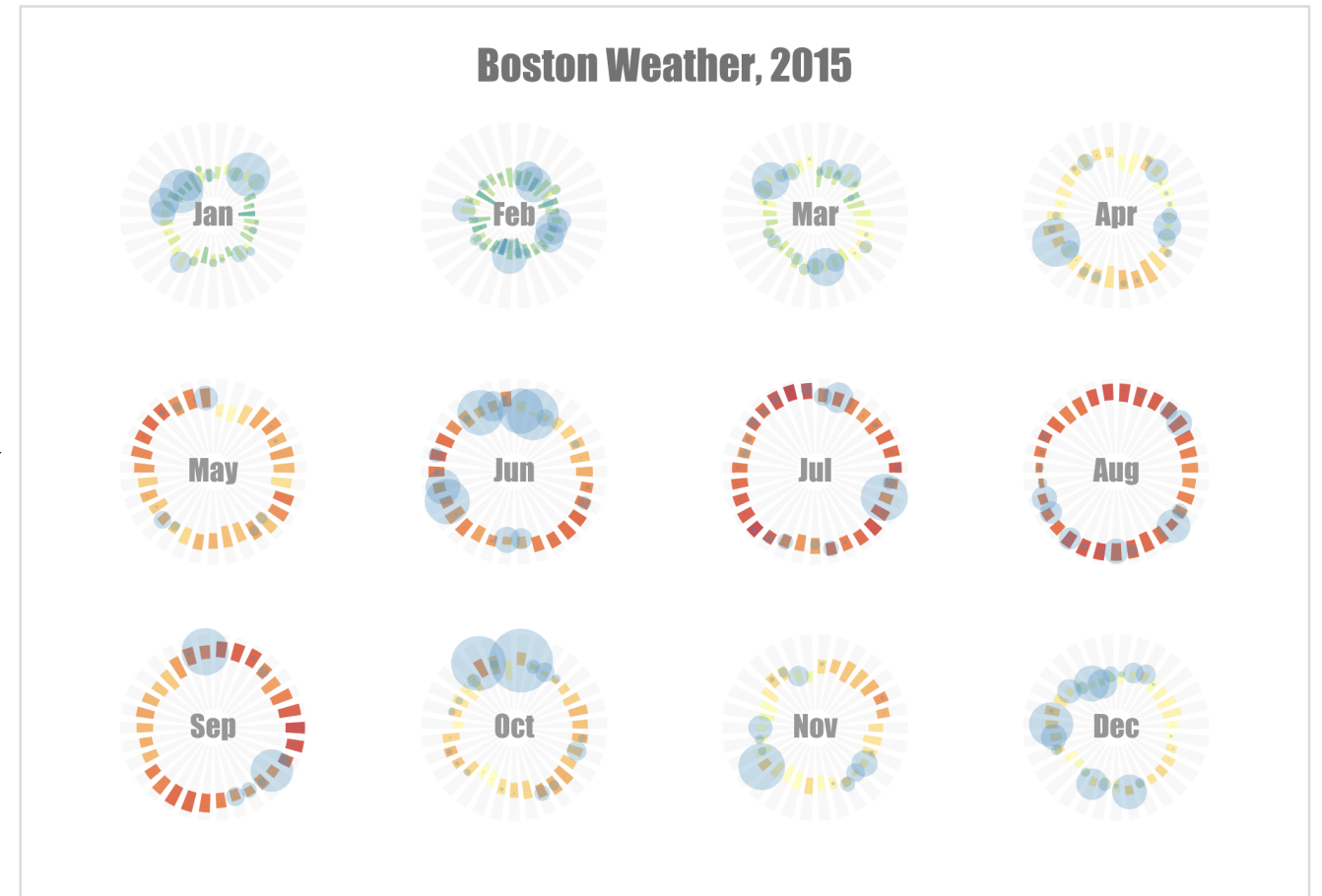
Nested Visualizations

- Support small multiples
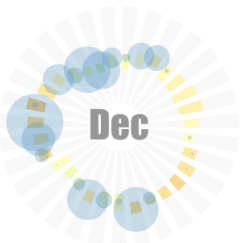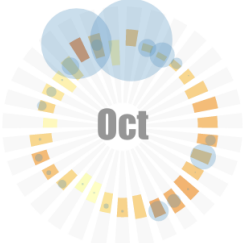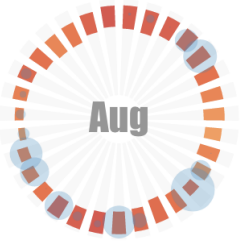
Nested Visualizations

- Use chart templates as marks

Create a single instance

Export & Reuse

Boston Weather, 2015

Demo

Boston Weather, 2015
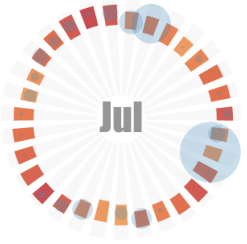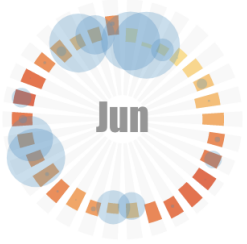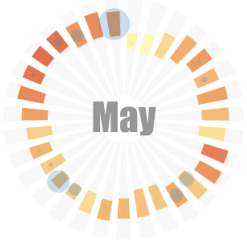
Deployment

- Available online:
  - https://charticulator.com/

- Open Source (MIT license):
  - https://github.com/Microsoft/charticulator

Deployment

- Number of users per day (average: 108)
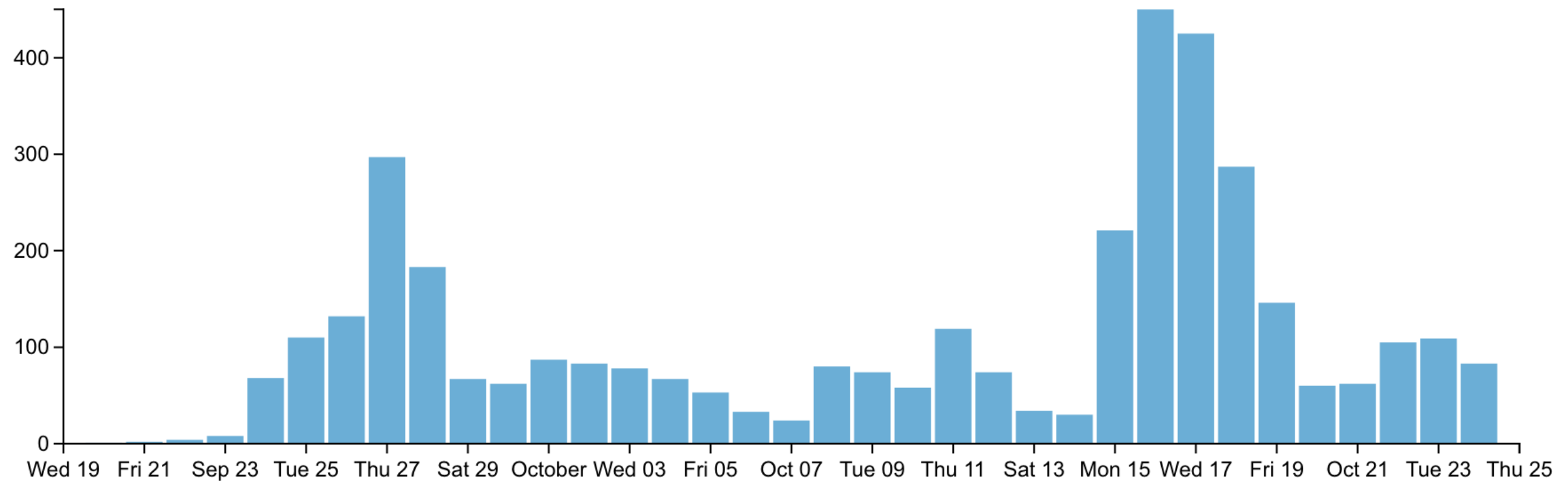
Future directions

- Incorporate multi-modal interactions
  - Pen & touch
  - Speech

- Support more sophisticated layout techniques, e.g.,
  - Edge bundling
  - Treemap
  - Force-directed layout algorithm

Thanks! Questions?

Donghao Ren
University of California, Santa Barbara

Bongshin Lee
Microsoft Research

Matthew Brehmer
Microsoft Research

Nathan Evans
Microsoft Research

Kate Lytvynets
Microsoft Research

David Tittsworth
Microsoft Research

Chris Trevino
Microsoft Research

- Links to Charticulator
  - https://charticulator.com/
  - https://github.com/Microsoft/charticulator