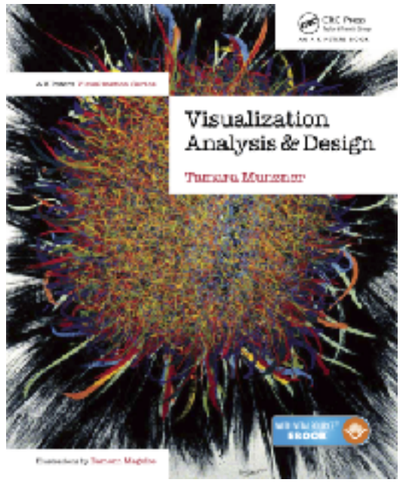
Visualization Analysis & Design

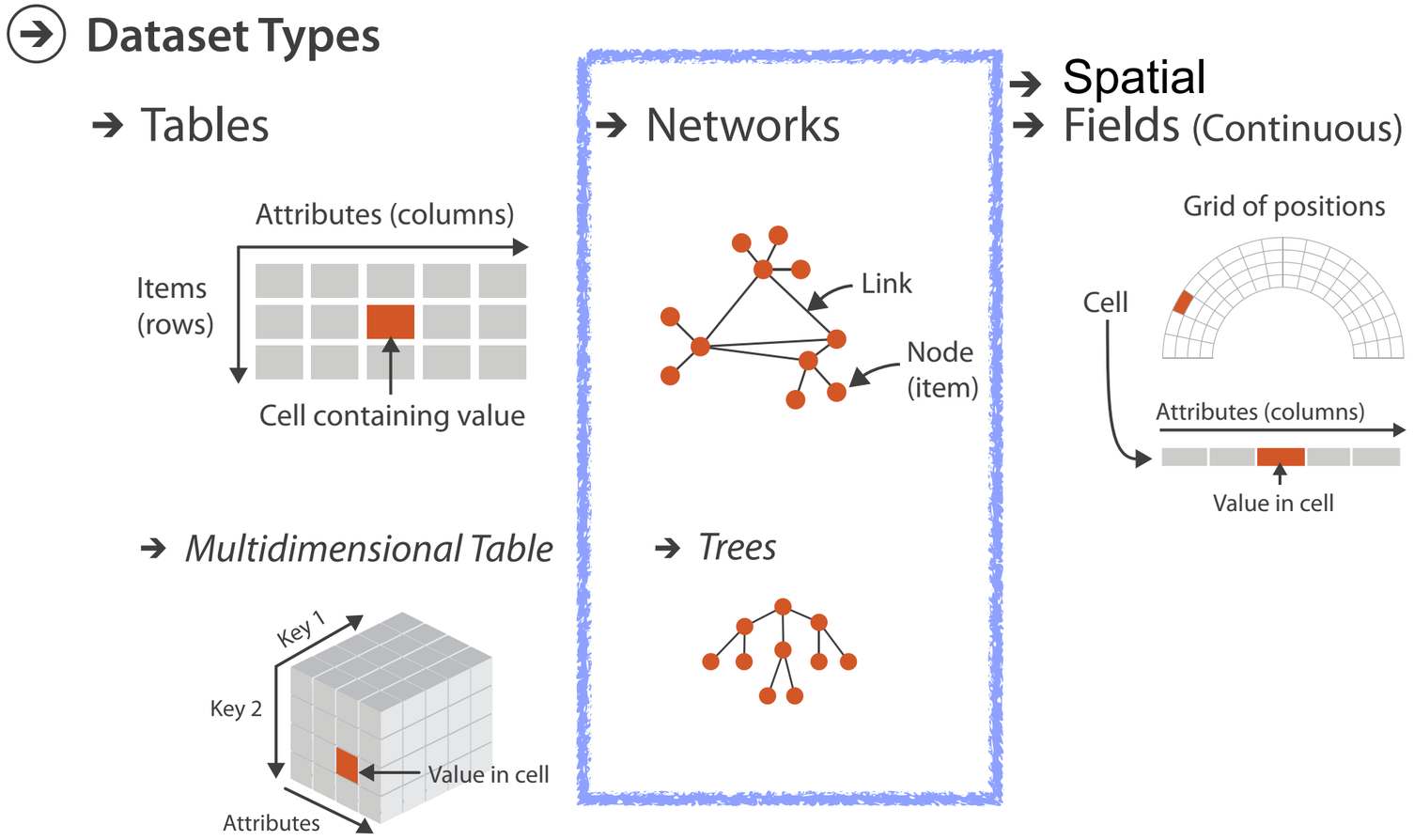


Network Data (Ch 9)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

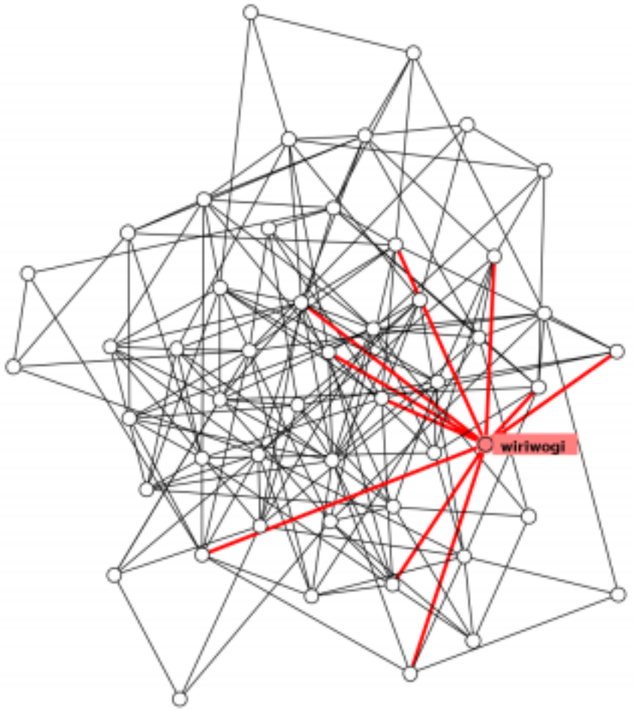
Network data

- networks
  - model relationships between things
    - aka graphs
  - two kinds of items, both can have attributes
    - nodes
    - links
- tree
  - special case
  - no cycles
    - one parent per node



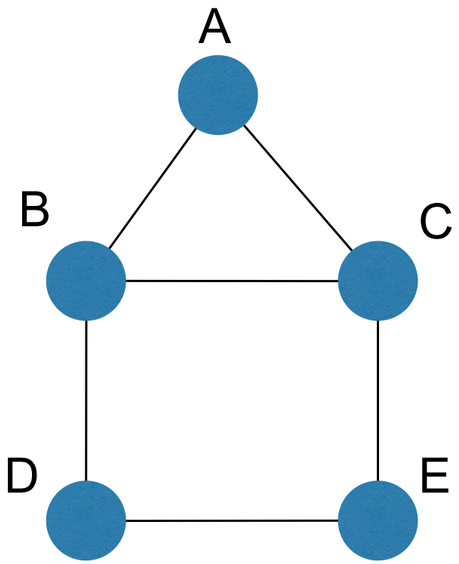
Network tasks: topology-based and attribute-based

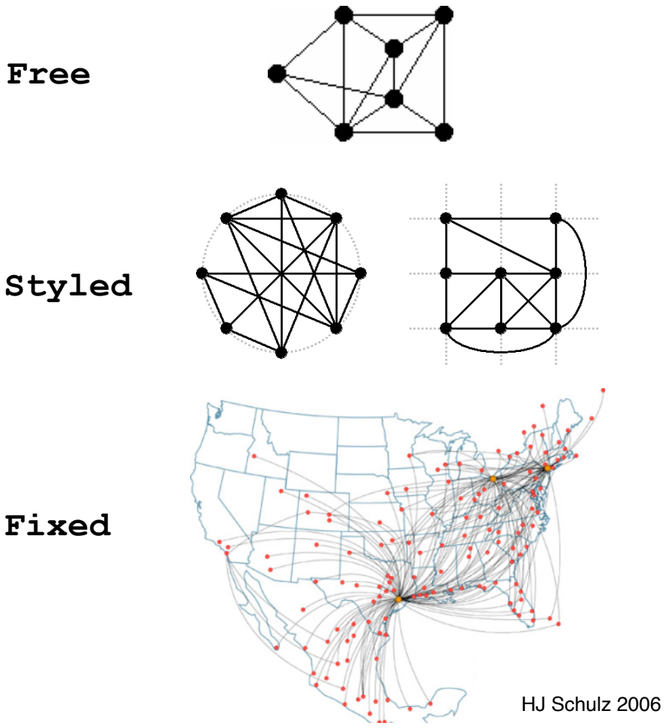
- topology based tasks
  - find paths
  - find (topological) neighbors
  - compare centrality/importance measures
  - identify clusters / communities
- attribute based tasks (similar to table data)
  - find distributions, ...
- combination tasks, incorporating both
  - example: find friends-of-friends who like cats
    - topology: find all adjacent nodes of given node
    - attributes: check if has-pet (node attribute) == cat



Node-link diagrams

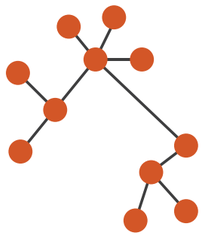
- nodes: point marks
- links: line marks
  - straight lines or arcs
  - connections between nodes
- intuitive & familiar
  - most common
  - many, many variants





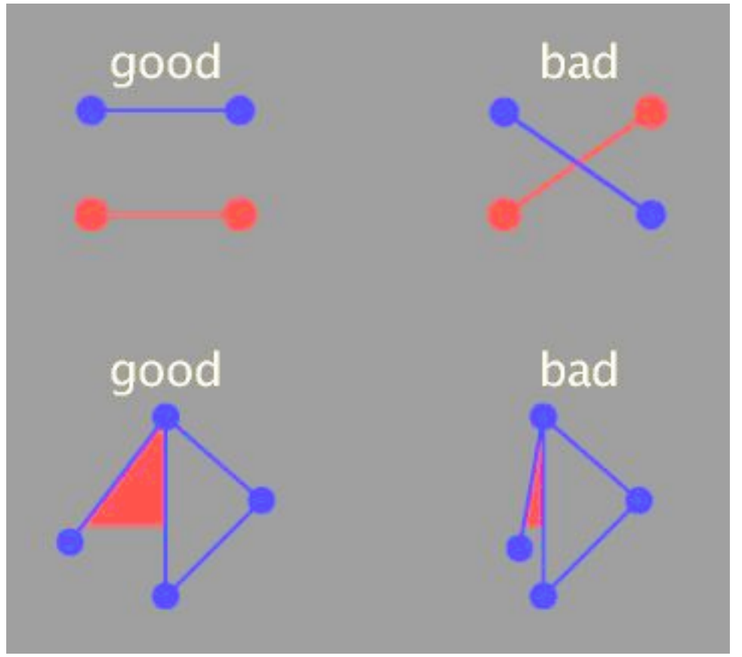
Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



Criteria for good node-link layouts

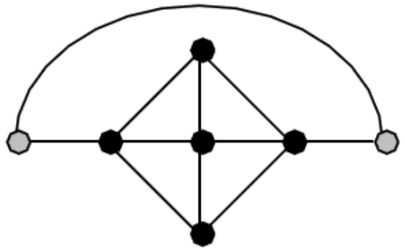
- minimize
  - edge crossings, node overlaps
  - distances between topological neighbor nodes
  - total drawing area
  - edge bends
- maximize
  - angular distance between different edges
  - aspect ratio disparities
- emphasize symmetry
  - similar graph structures should look similar in layout



Criteria conflict

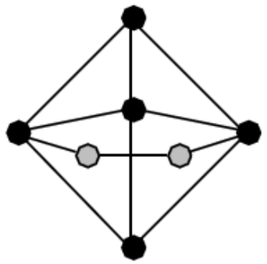
- most criteria NP-hard individually
- many criteria directly conflict with each other

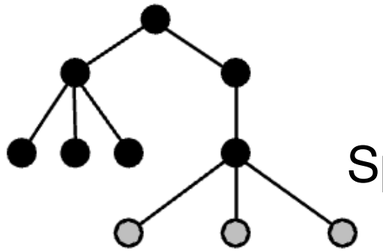
Minimum number of edge crossings



vs.

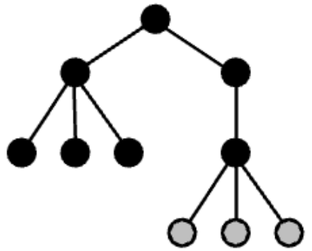
Uniform edge length





Space utilization

vs.



Symmetry

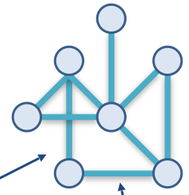
Schulz 2004

Optimization-based layouts

- formulate layout problem as optimization problem
- convert criteria into weighted cost function
  - F(layout) = a*[crossing counts] + b*[drawing space used]+...
- use known optimization techniques to find layout at minimal cost
  - energy-based physics models
  - force-directed placement
  - spring embedders

Force-directed placement

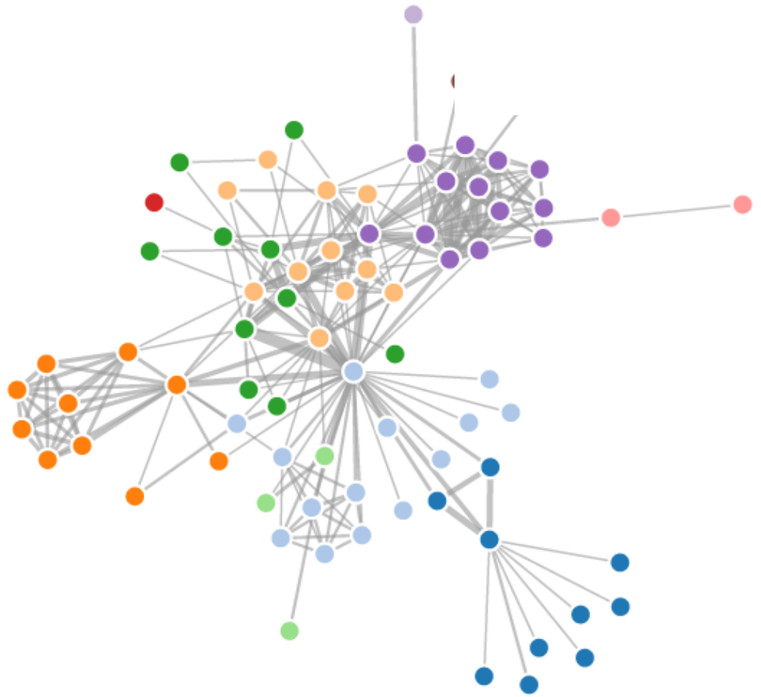
- physics model
  - links = springs pull together
  - nodes = magnets repulse apart



Expander
(pushing nodes apart)

Spring Coil
(pulling nodes together)

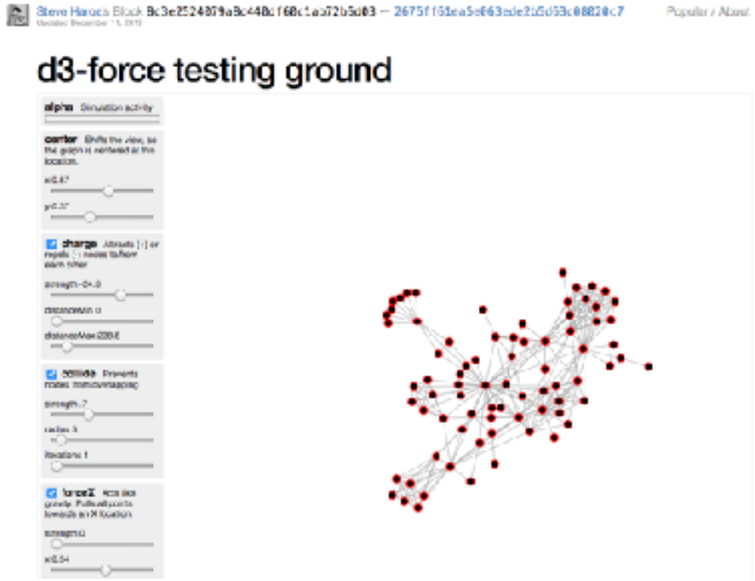
- algorithm
  - place vertices in random locations
  - while not equilibrium
    - calculate force on vertex
      - sum of
        - » pairwise repulsion of all nodes
        - » attraction between connected nodes
    - move vertex by c * vertex_force



http://mbostock.github.com/d3/ex/force.html

Force-directed placement properties

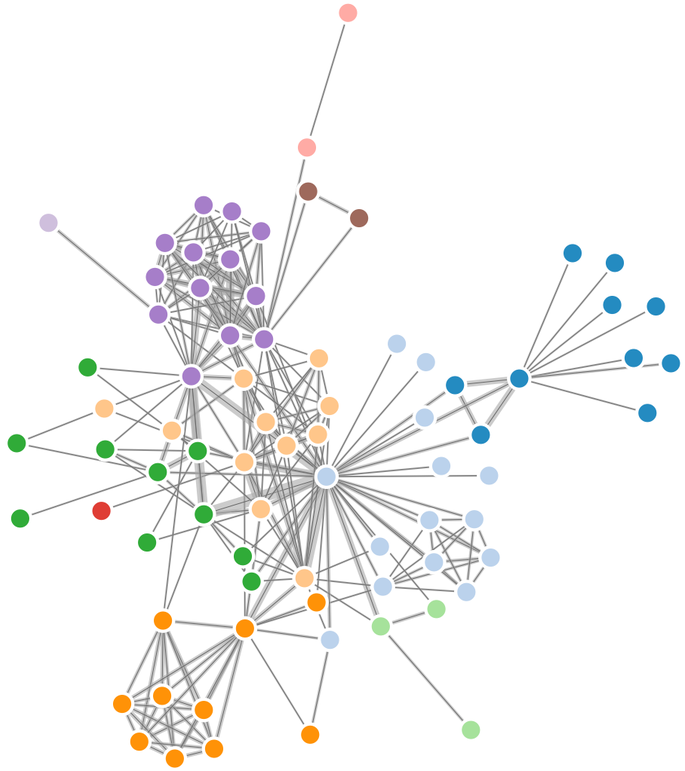
- strengths
  - reasonable layout for small, sparse graphs
  - clusters typically visible
  - edge length uniformity
- weaknesses
  - nondeterministic
  - computationally expensive: O(n^3) for n nodes
    - each step is n^2, takes ~n cycles to reach equilibrium
  - naive FD doesn't scale well beyond 1K nodes
  - iterative progress: engaging but distracting



https://bl.ocks.org/steveharoz/8c3e2524079a8c440df60c1ab72b5d03

Idiom: **force-directed placement**

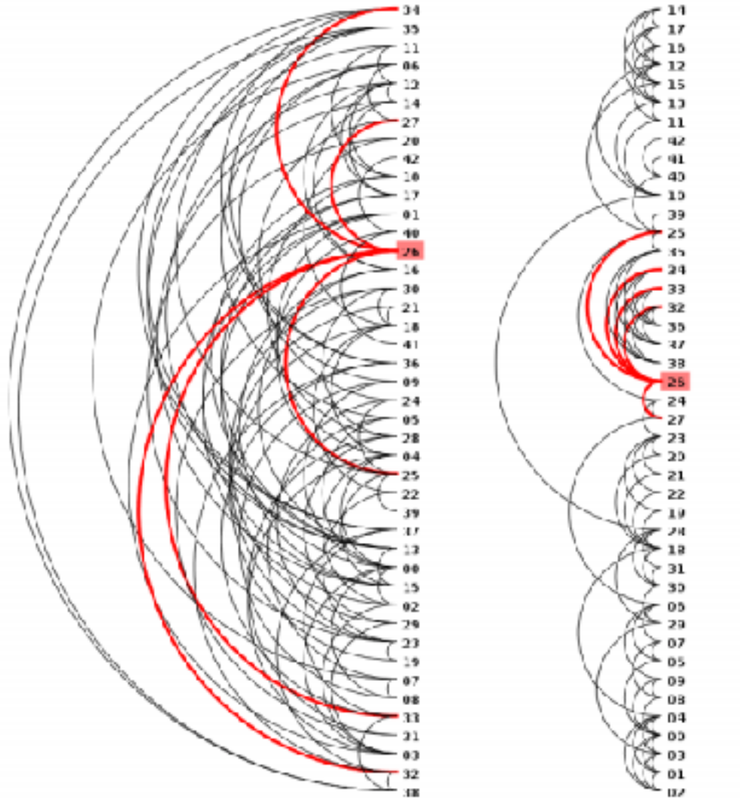
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short
- tasks
  - explore topology; locate paths, clusters
- scalability
  - node/edge density E < 4N

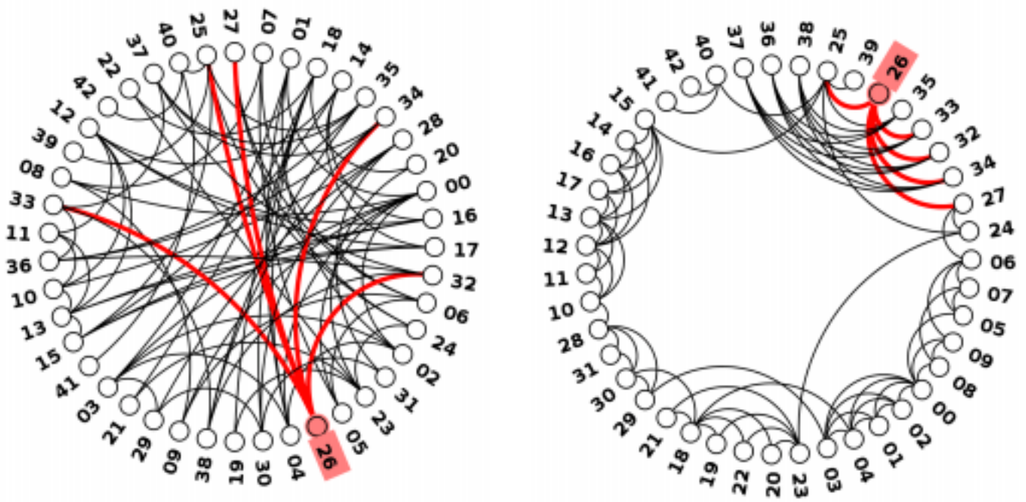


http://mbostock.github.com/d3/ex/force.html

Idiom: **circular layouts / arc diagrams (node-link)**

- restricted node-link layouts: lay out nodes around circle or along line
- data
  - original: network
  - derived: node ordering attribute (global computation)
- considerations: node ordering crucial to avoid excessive clutter from edge crossings
  - examples: before & after barycentric ordering

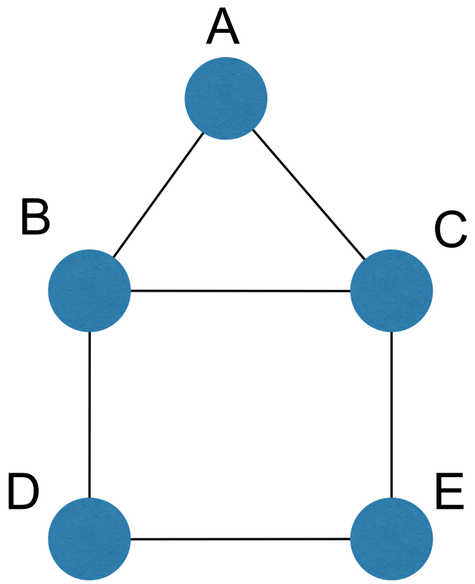


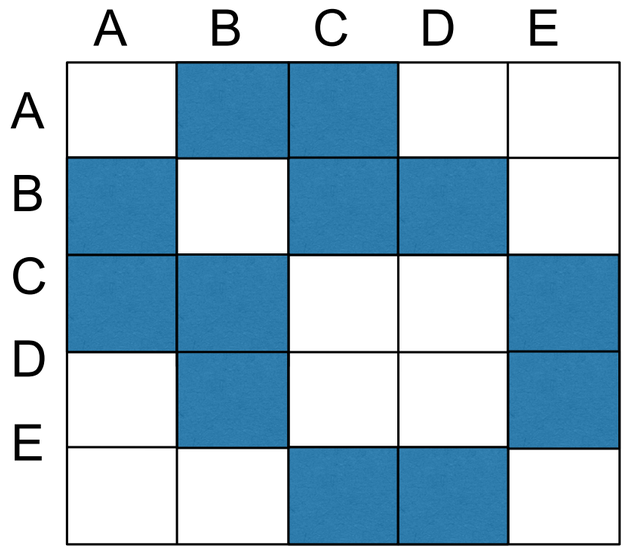


http://profs.etsmtl.ca/mmcguffin/research/2012-mcguffin-simpleNetVis/mcguffin-2012-simpleNetVis.pdf

Adjacency matrix representations

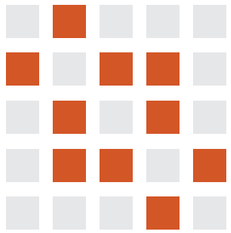
- derive adjacency matrix from network



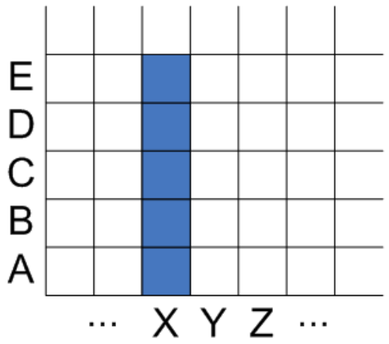


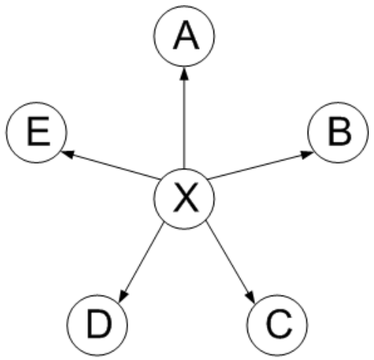
Adjacency Matrix
Derived Table

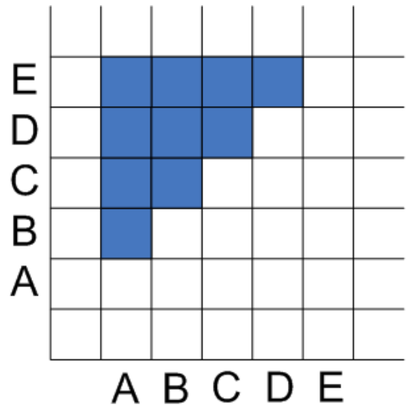
✔ NETWORKS ✔ TREES

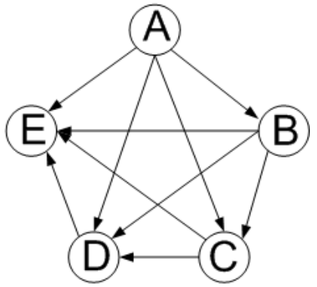


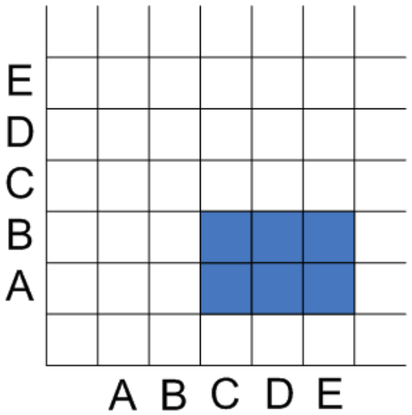
Adjacency matrix examples

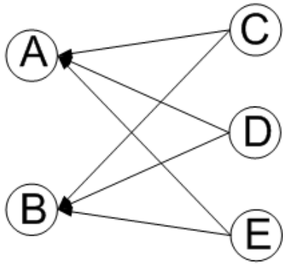


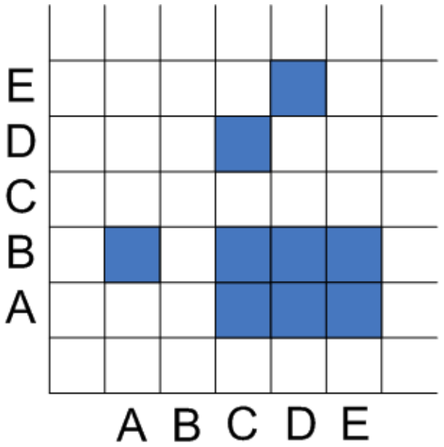


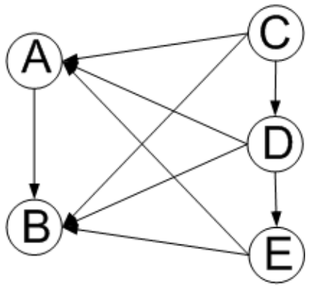






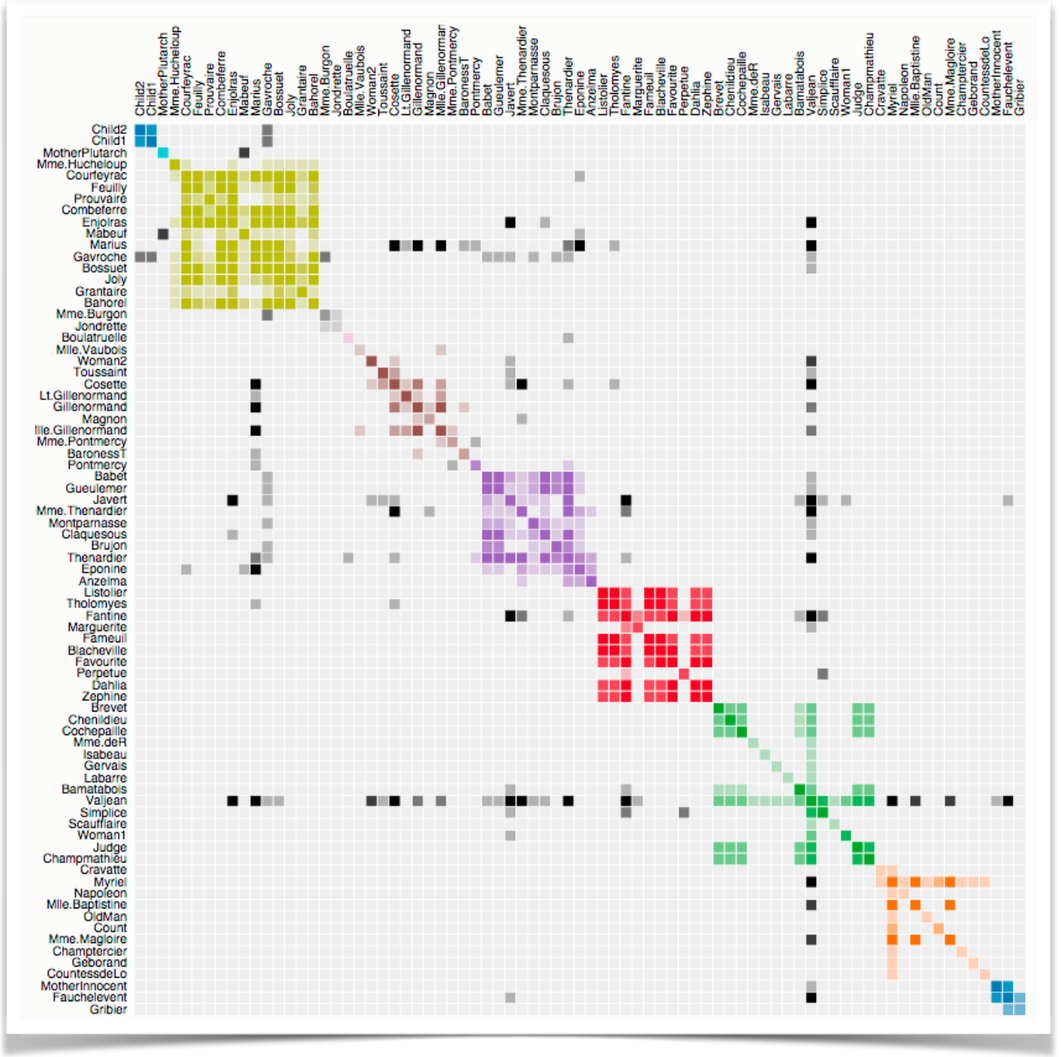






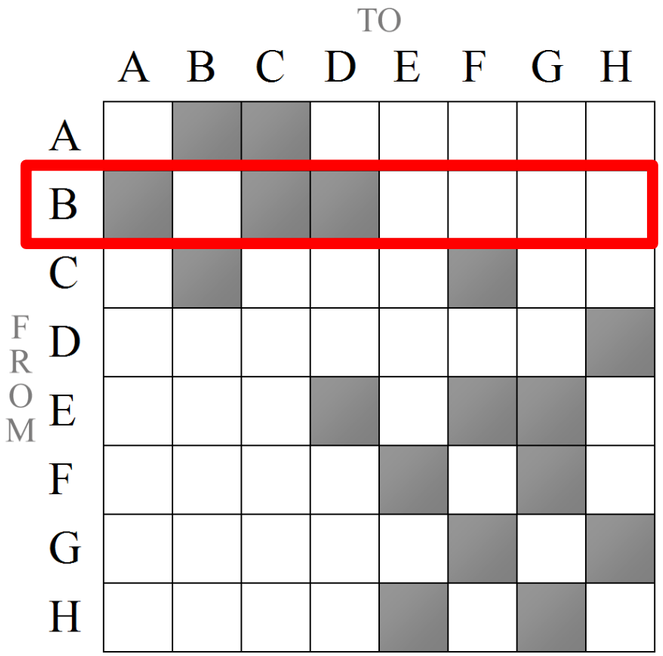
HJ Schulz 2007

Node order is crucial: Reordering

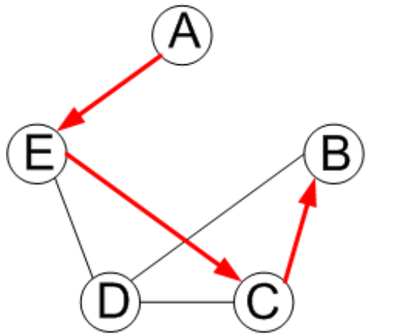


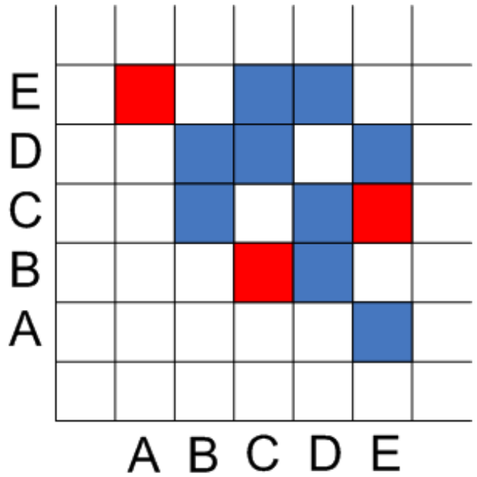
https://bost.ocks.org/mike/miserables/

Adjacency matrix



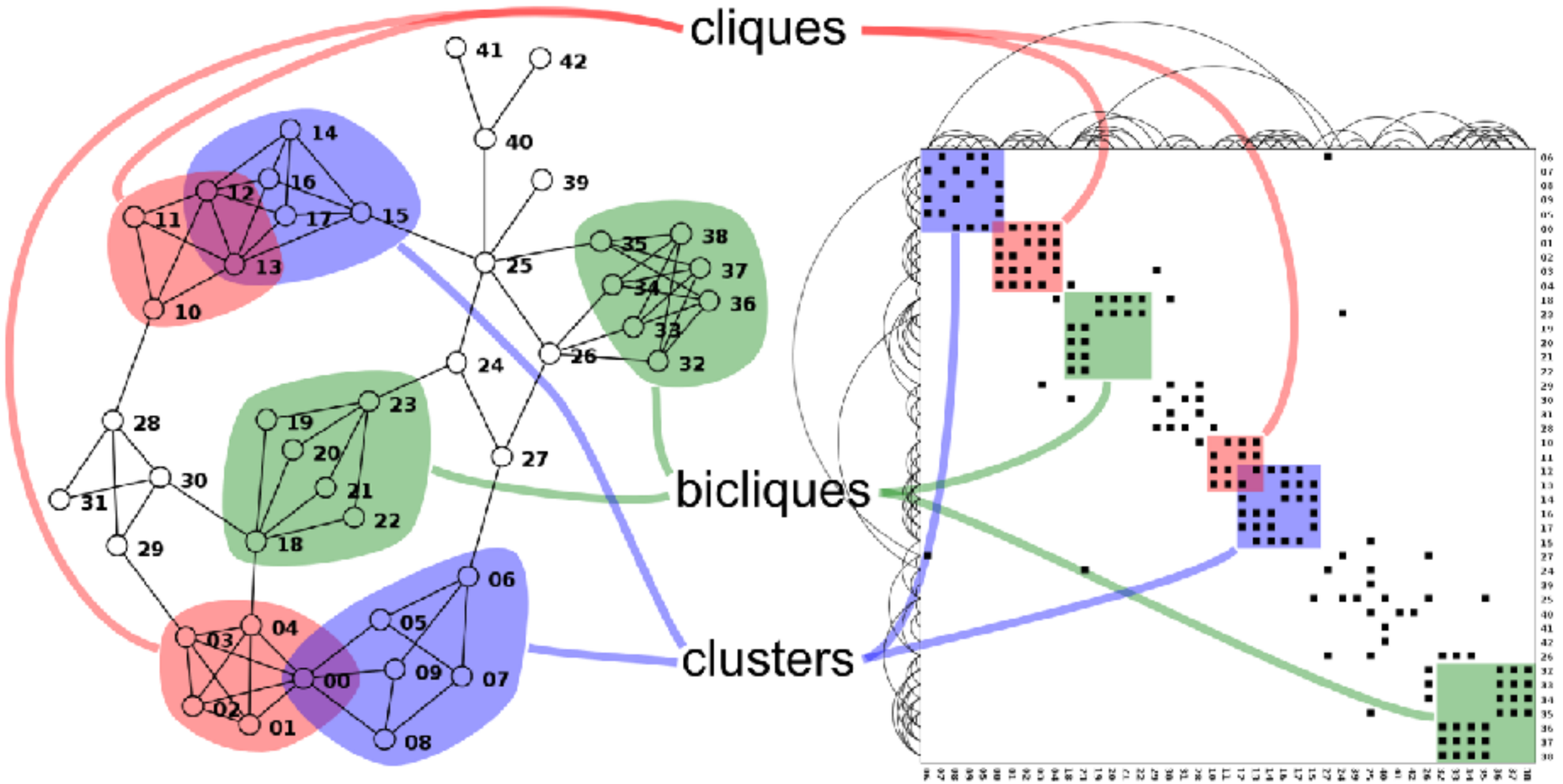
good for topology tasks
related to neighborhoods
(node 1-hop neighbors)





bad for topology tasks
related to paths

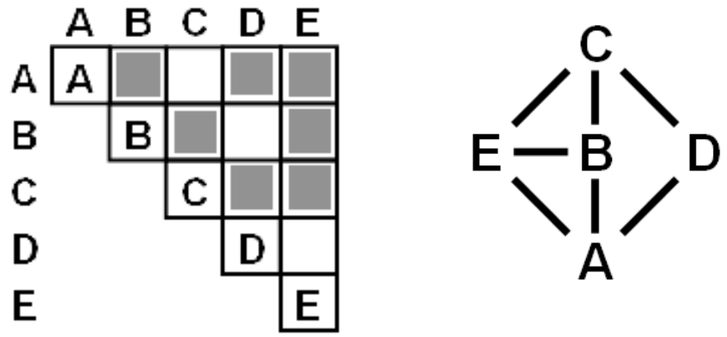
Structures visible in both



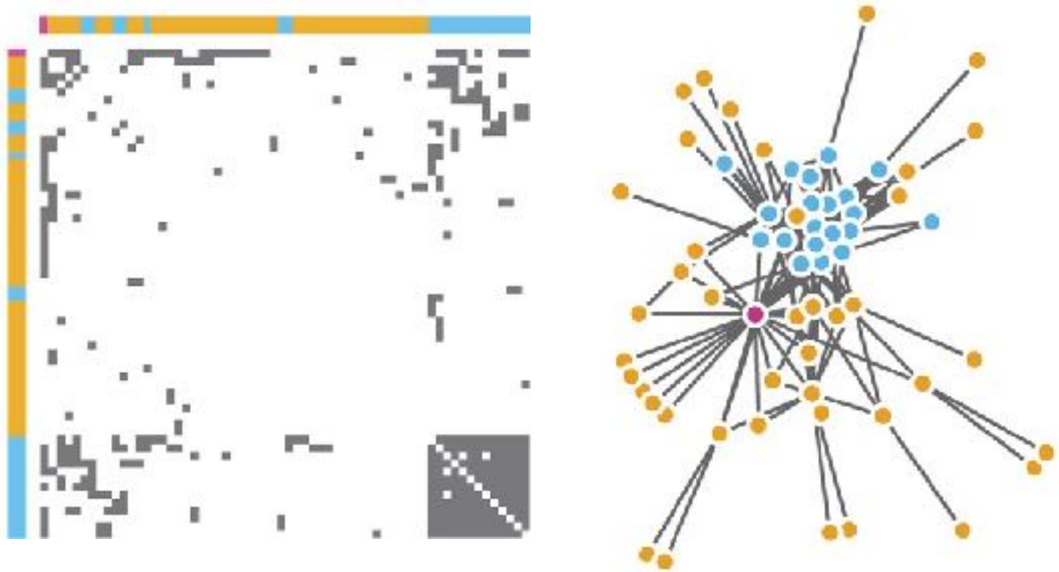
http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Idiom: **adjacency matrix view**

- data: network
  - transform into same data/encoding as heatmap
- derived data: table from network
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- visual encoding
  - cell shows presence/absence of edge
- scalability
  - 1K nodes, 1M edges



[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]

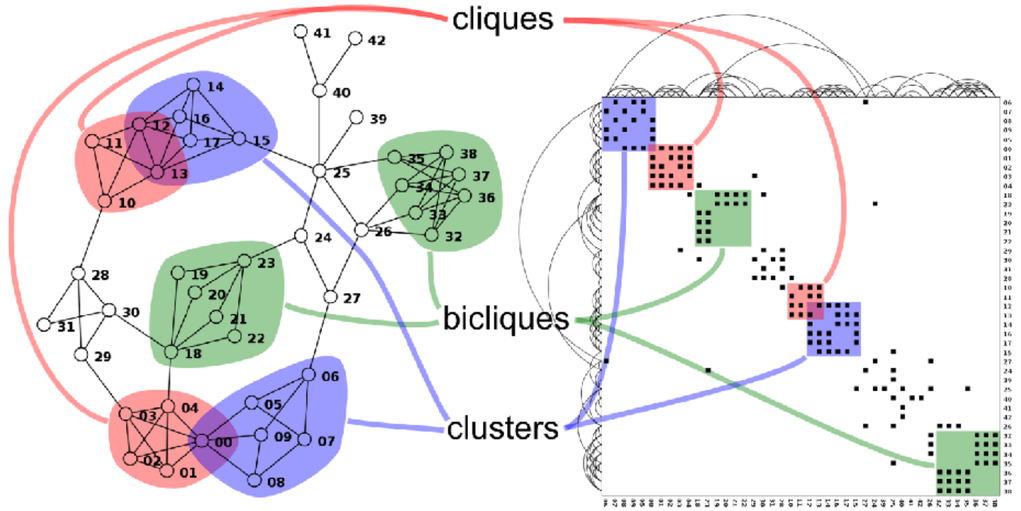


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

Node-link vs. matrix comparison

- node-link diagram strengths
  - topology understanding, path tracing
  - intuitive, flexible, no training needed
- adjacency matrix strengths
  - focus on edges rather than nodes
  - layout straightforward (reordering needed)
  - predictability, scalability
  - some topology tasks trainable
- empirical study
  - node-link best for small networks
  - matrix best for large networks
    - if tasks don't involve path tracing!

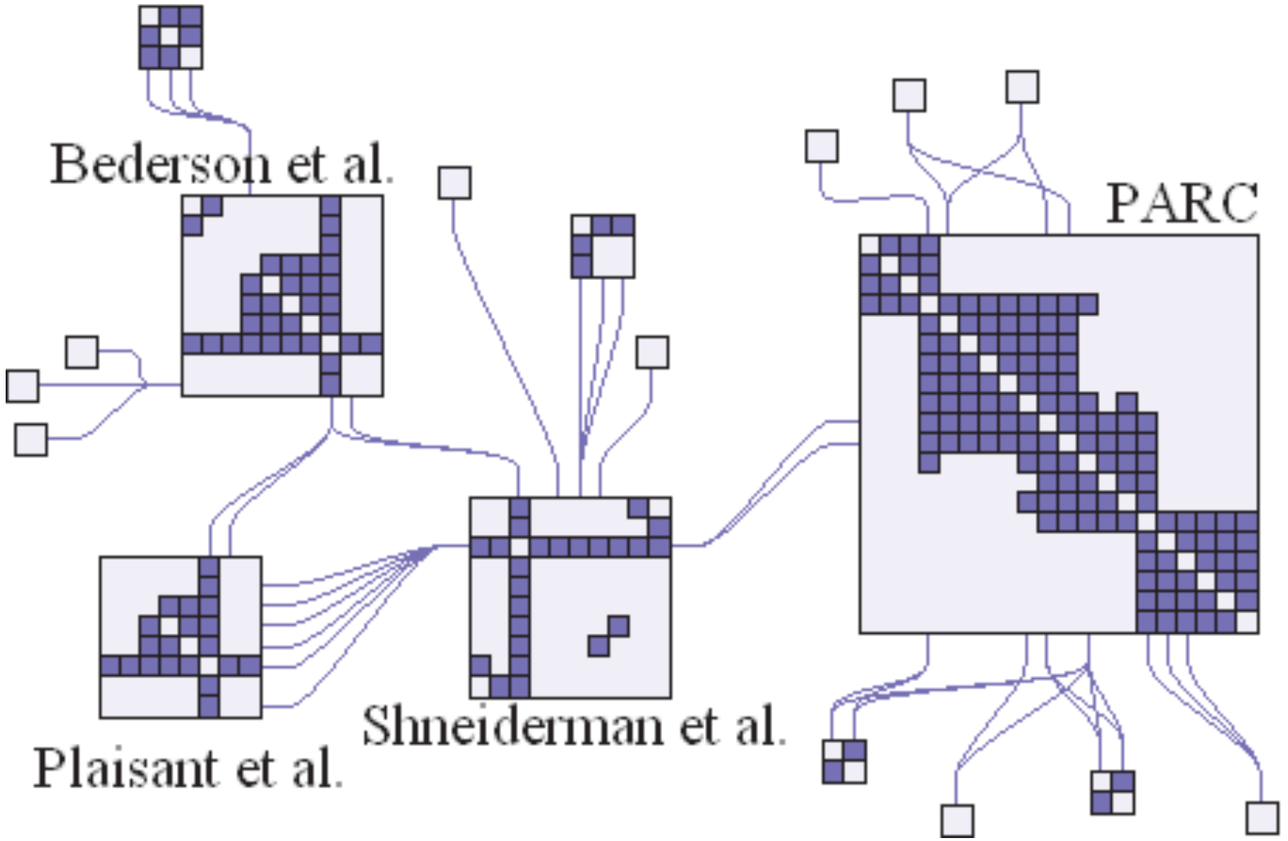
[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]



http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Idiom: **NodeTrix**

- hybrid nodelink/matrix
- capture strengths of both



[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]

Trees

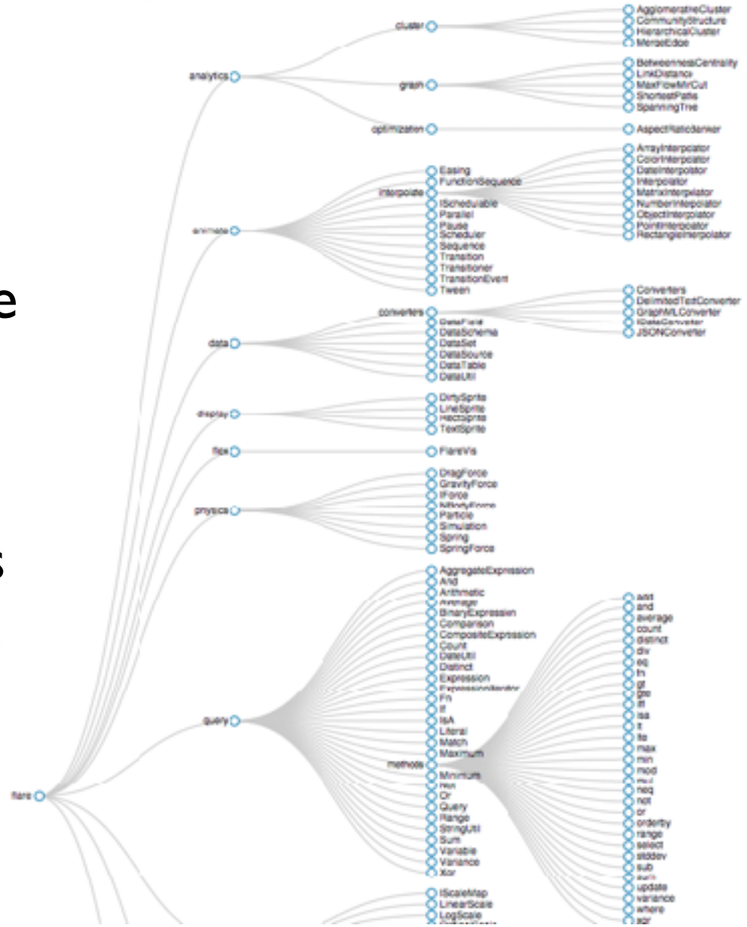
Node-link trees

- Reingold-Tilford
  - tidy drawings of trees
    - exploit parent/child structure
  - allocate space: compact but without overlap
    - rectilinear and radial variants

[Tidier drawing of trees. Reingold and Tilford. IEEE Trans. Software Eng., SE-7(2):223–228, 1981.]

  - nice algorithm writeup

http://billmill.org/pymag-trees/

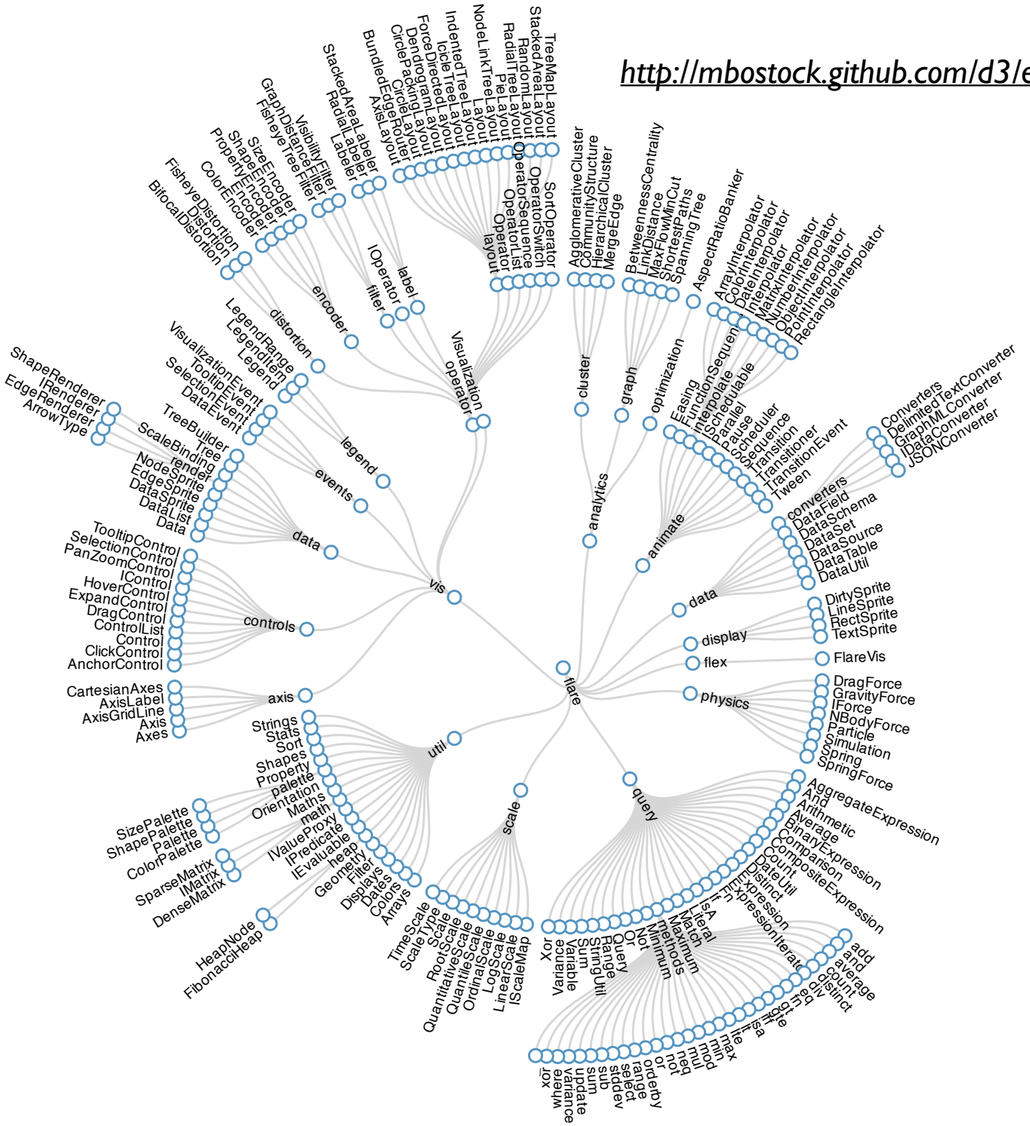


http://bl.ocks.org/mbostock/4339184

http://bl.ocks.org/mbostock/4063550

Idiom: **radial node-link tree**

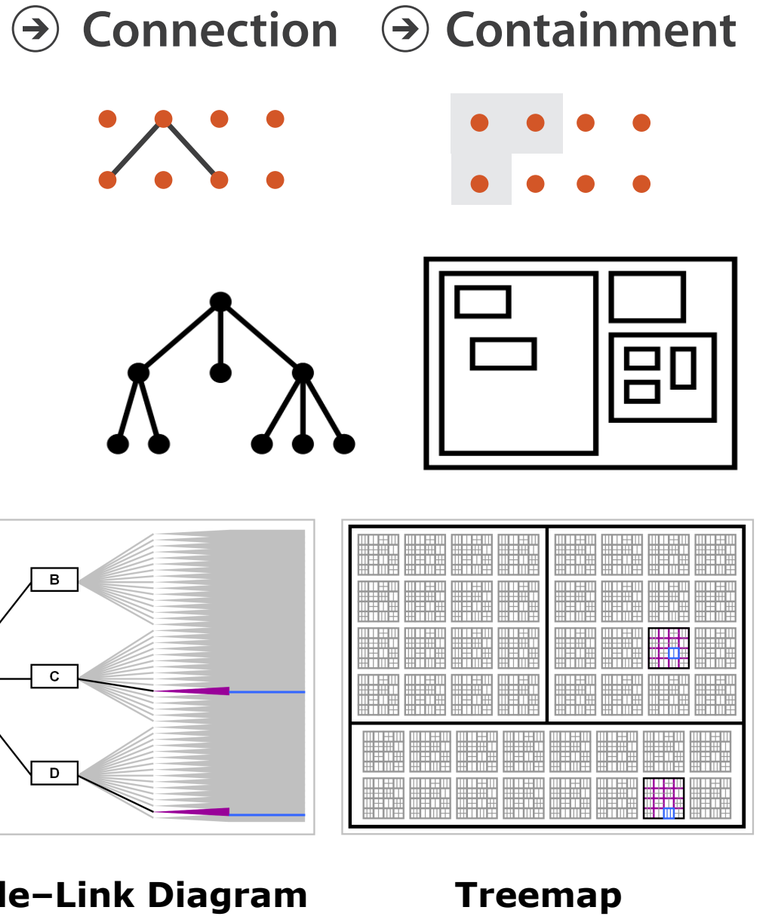
- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes (with/without labels)



http://mbostock.github.com/d3/ex/tree.html

Link marks: Connection and containment

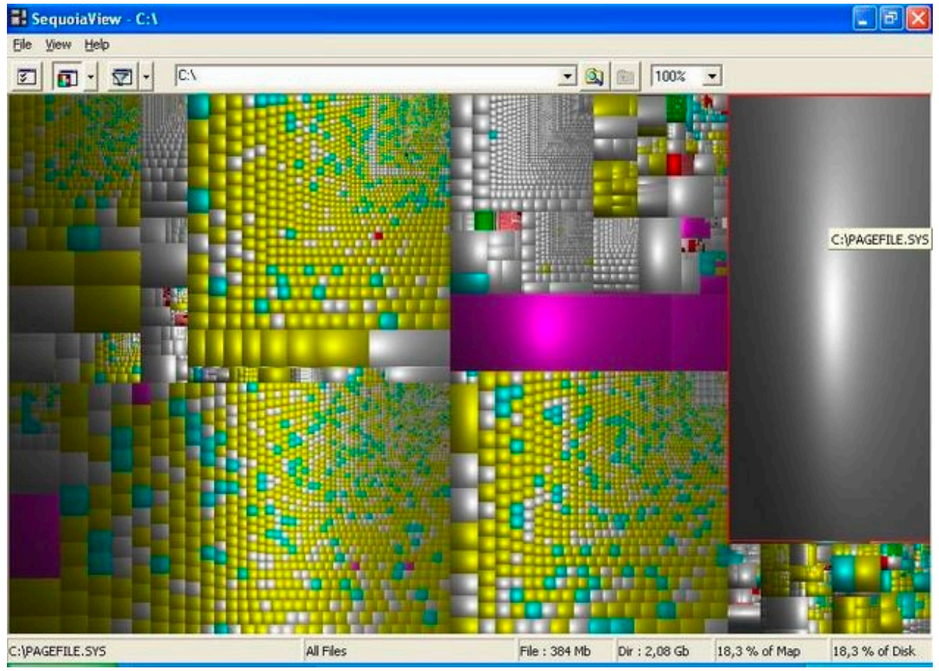
- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
    - only trees



[Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. Dong, McGuffin, and Chignell. Proc. InfoVis 2005, p. 57-64.]

Idiom: **treemap**

- data
  - tree
  - 1 quant attrib at leaf nodes
- encoding
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- tasks
  - query attribute at leaf nodes
  - ex: disk space usage within filesystem
- scalability
  - 1M leaf nodes



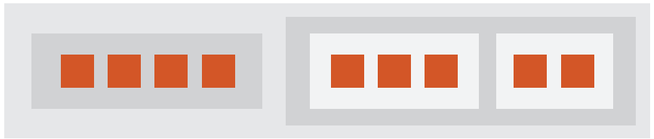
https://www.win.tue.nl/sequoiaview/

[Cushion Treemaps. van Wijk and van de Wetering. Proc. Symp. InfoVis 1999, 73-78.]

Enclosure

Containment Marks

✕ NETWORKS ✓ TREES

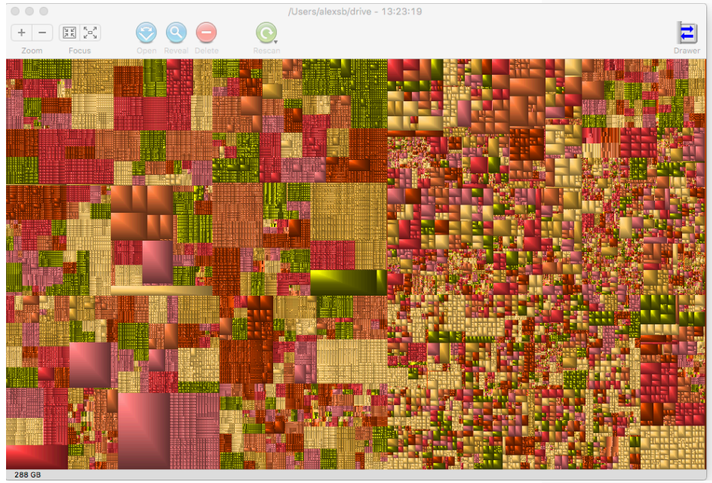


Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

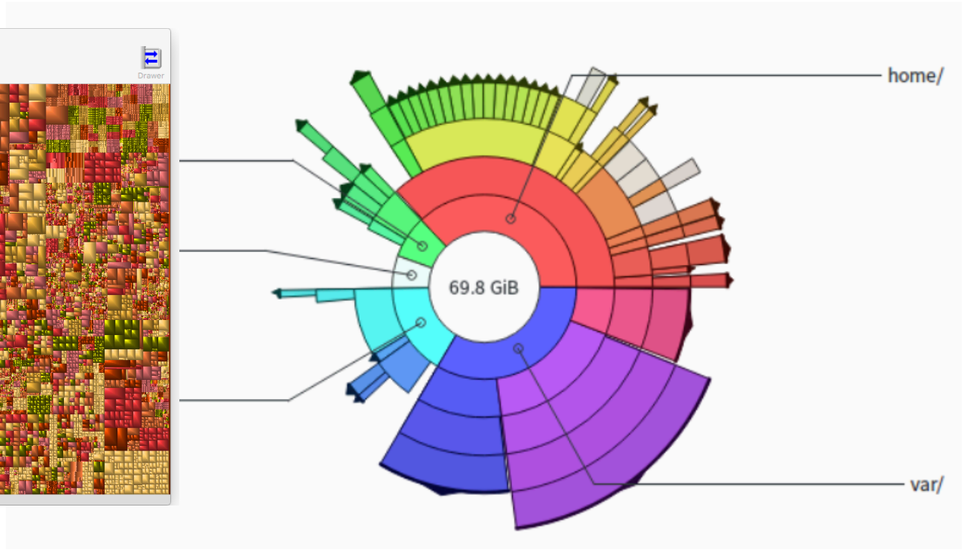
Treemap

containment



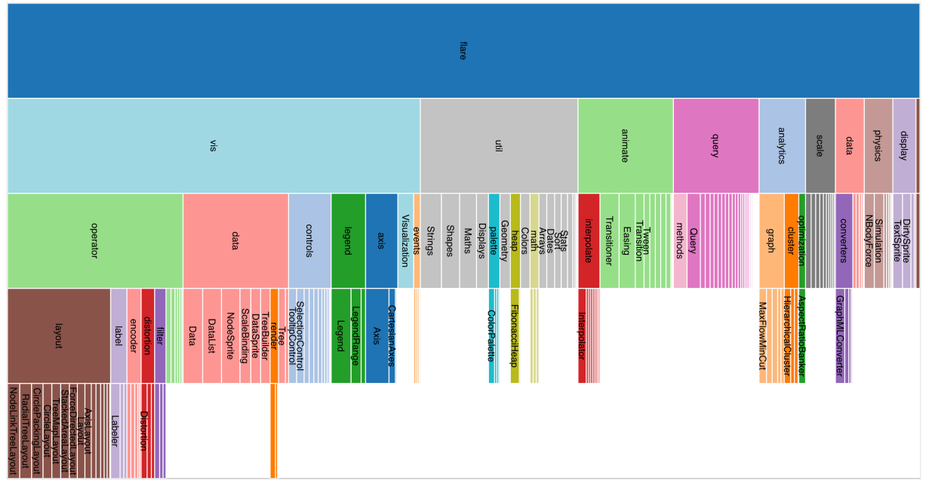
Sunburst

position (radial)



Icicle Plot

position (rectilinear)



Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

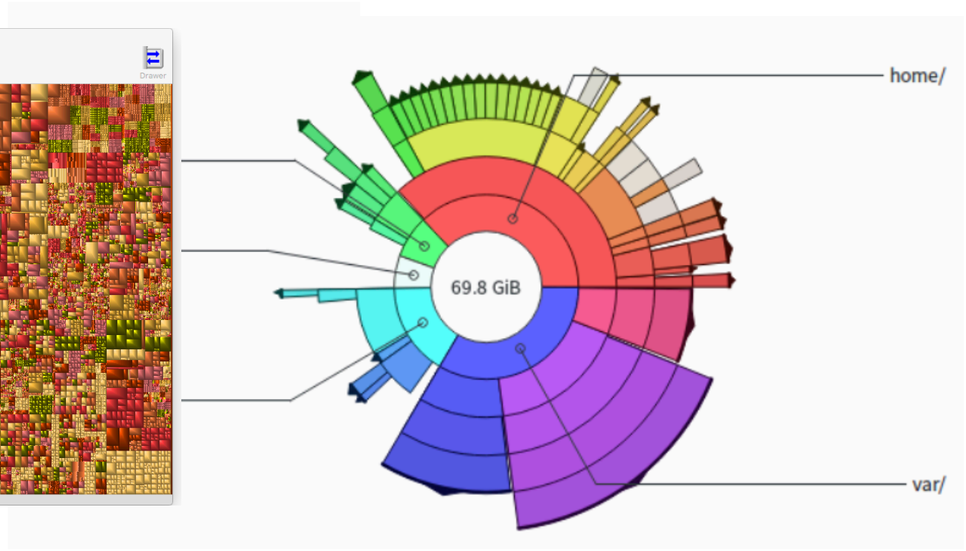
containment

only leaves visible

Sunburst

position (radial)

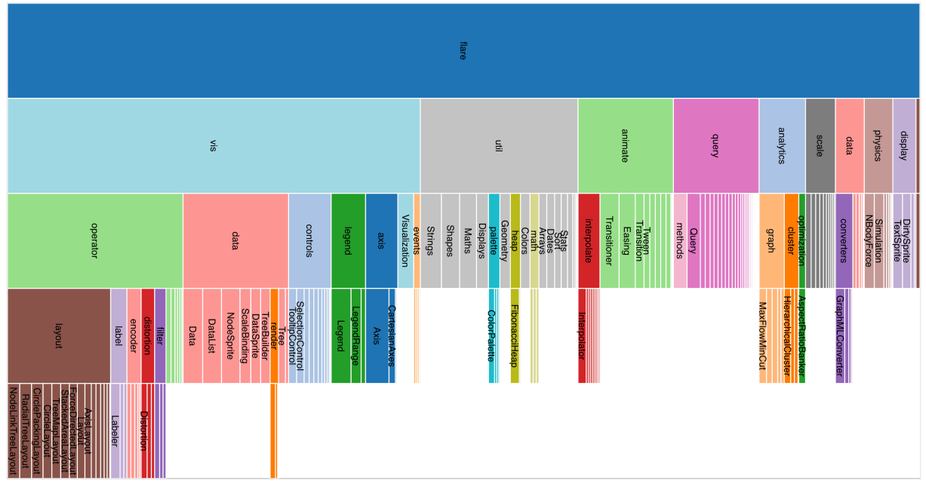
inner nodes & leaves visible



Icicle Plot

position (rectilinear)

inner nodes & leaves visible



Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

containment

only leaves visible

Sunburst

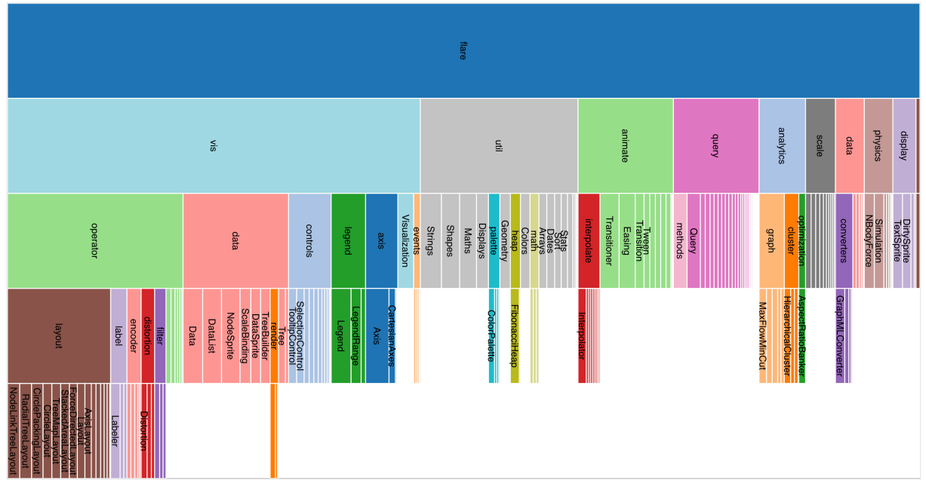
position (radial)

inner nodes & leaves visible

Icicle Plot

position (rectilinear)

inner nodes & leaves visible

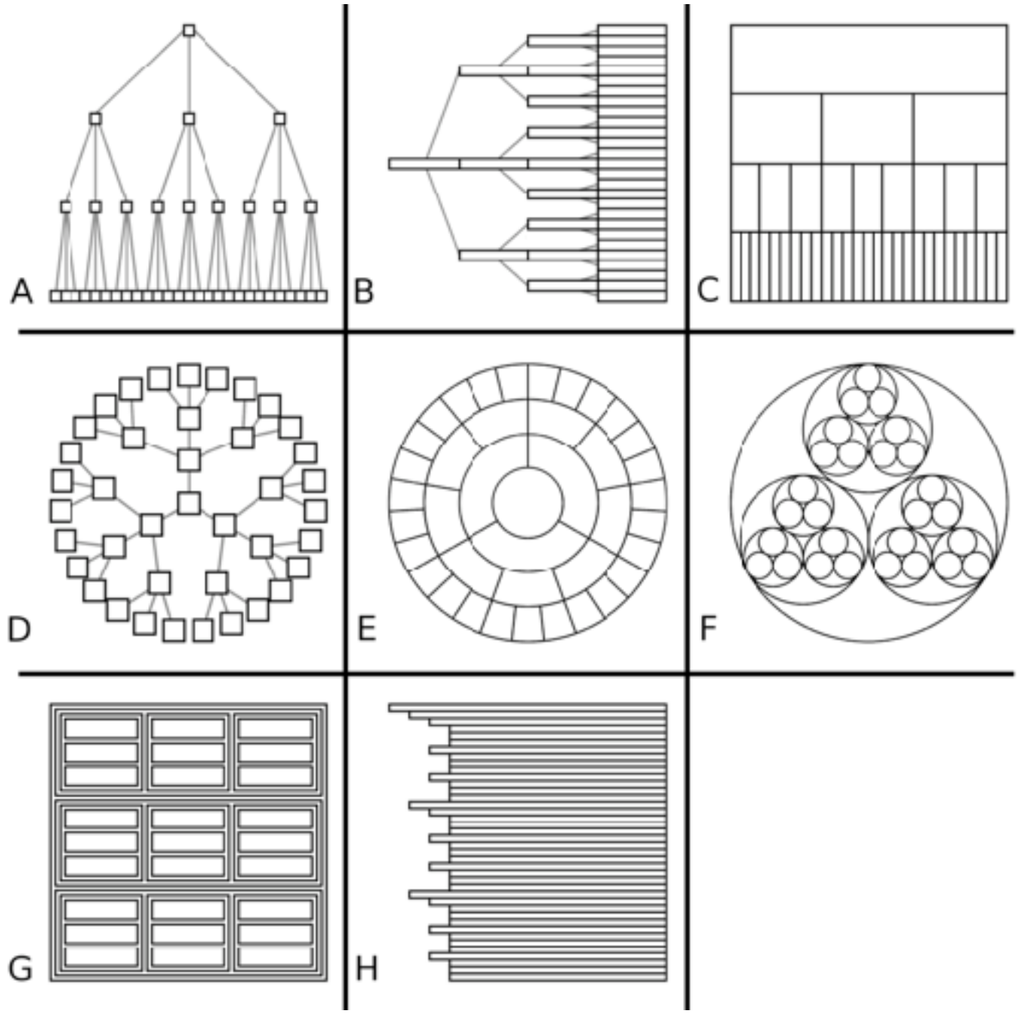


Implicit

Spatial Position

✕ NETWORKS ✓ TREES

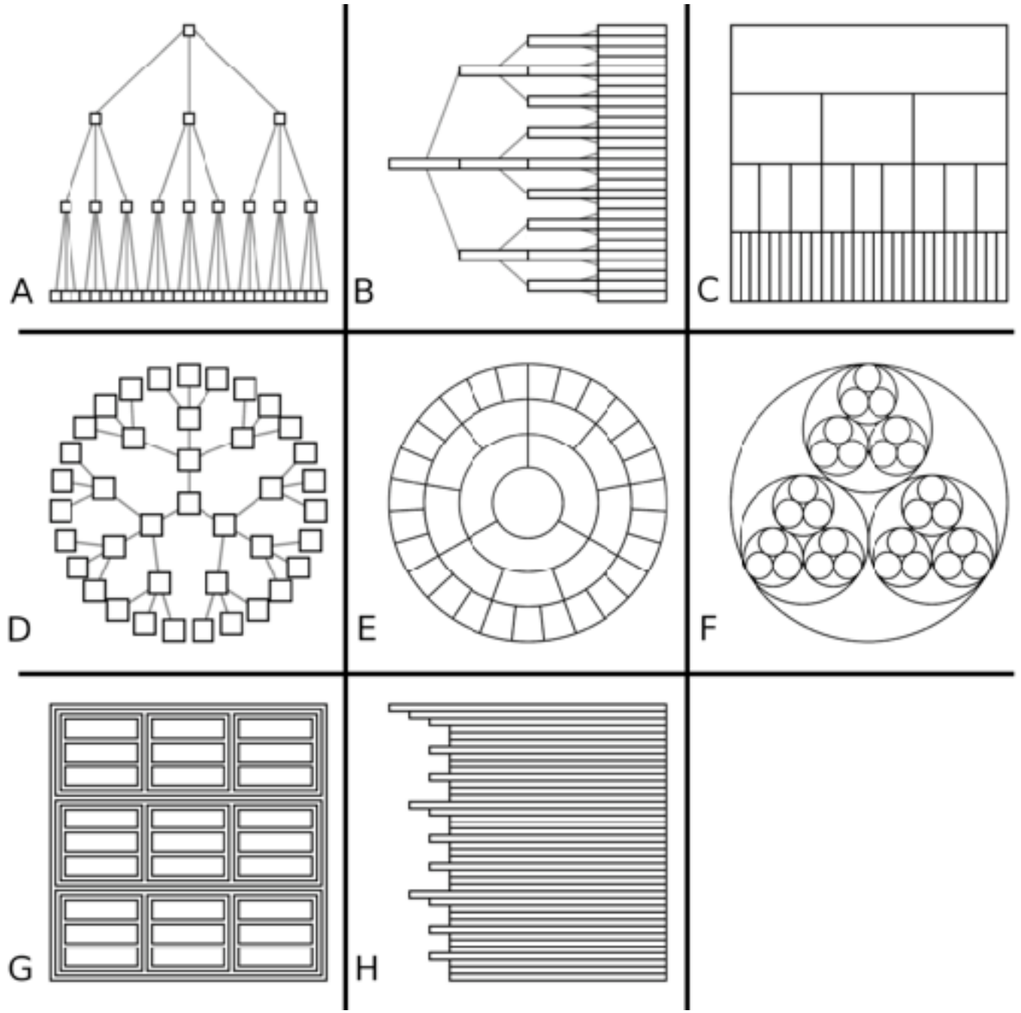
Tree drawing idioms comparison



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

Comparison: tree drawing idioms

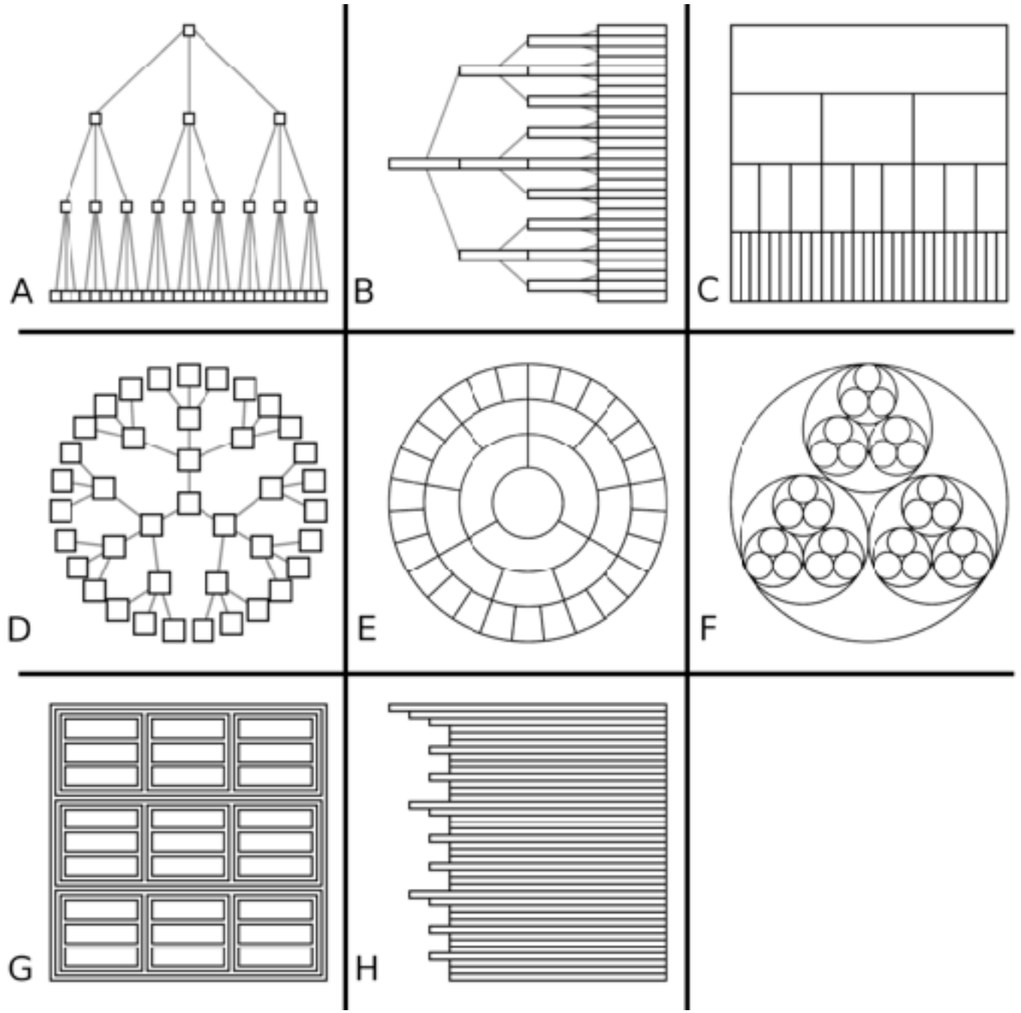
- data shown
  - link relationships
  - tree depth
  - sibling order



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

Comparison: tree drawing idioms

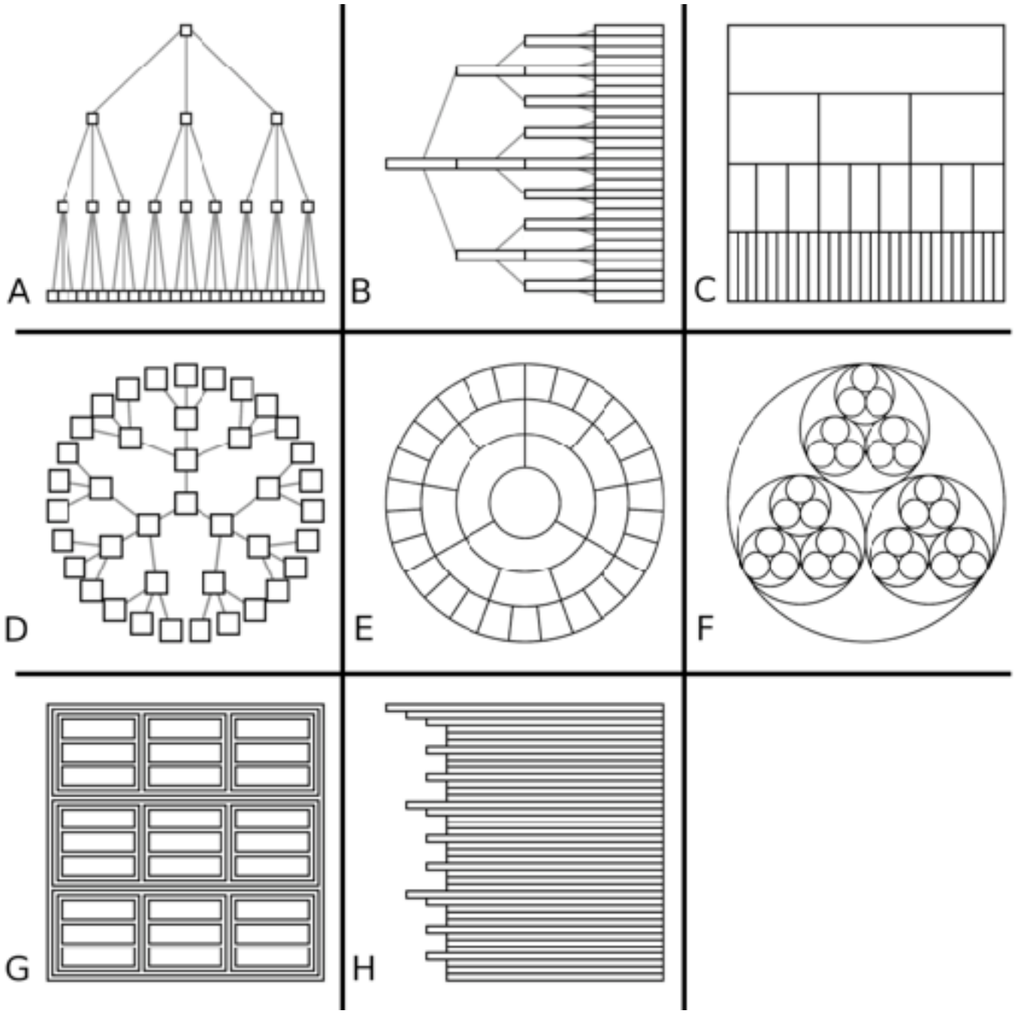
- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

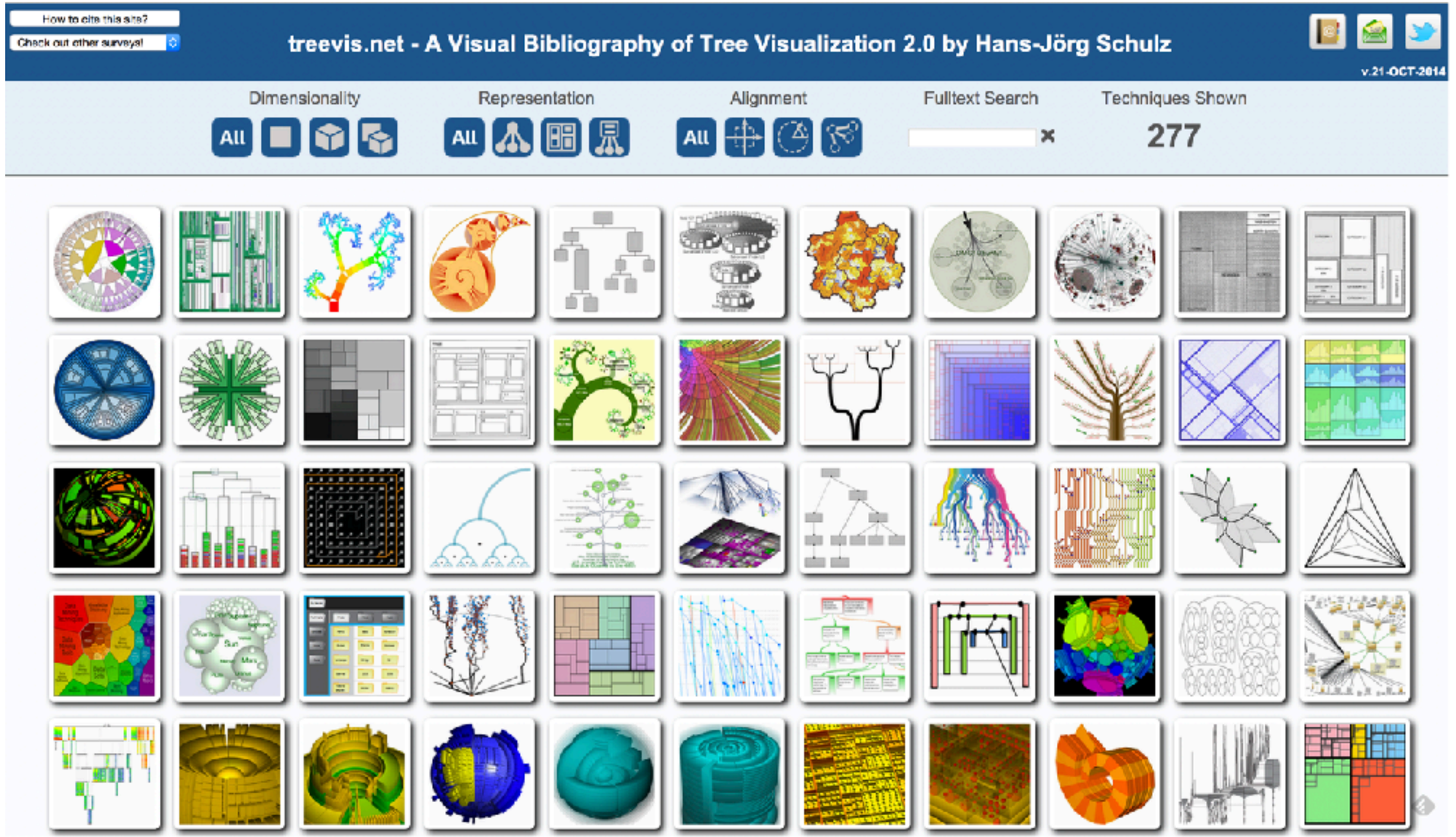
Comparison: tree drawing idioms

- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels
- considerations
  - redundant? arbitrary?
  - information density?
    - avoid wasting space
    - consider where to fit labels!



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

treevis.net: Many, many options!



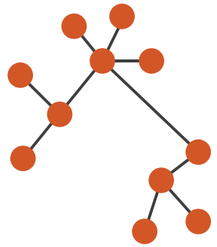
https://treevis.net/

Arrange networks and trees

Node–Link Diagrams

Connection Marks

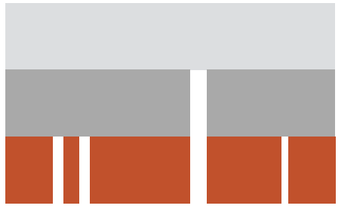
✔ NETWORKS ✔ TREES



Implicit

Spatial Position

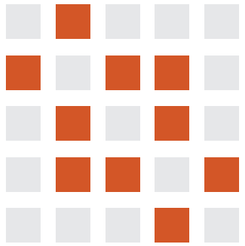
✘ NETWORKS ✔ TREES



Adjacency Matrix

Derived Table

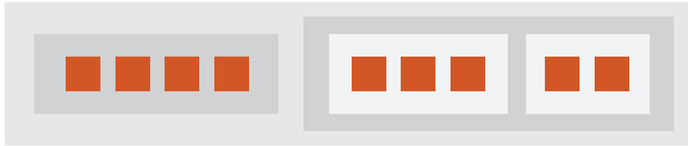
✔ NETWORKS ✔ TREES



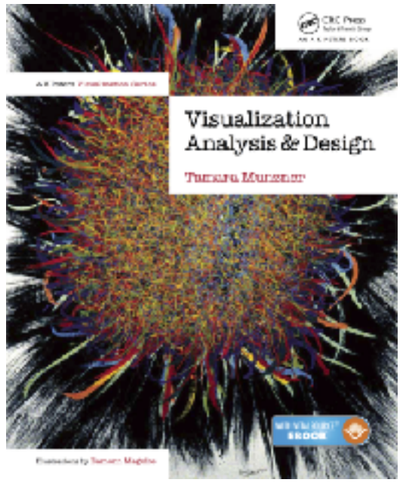
Enclosure

Containment Marks

✘ NETWORKS ✔ TREES



Visualization Analysis & Design

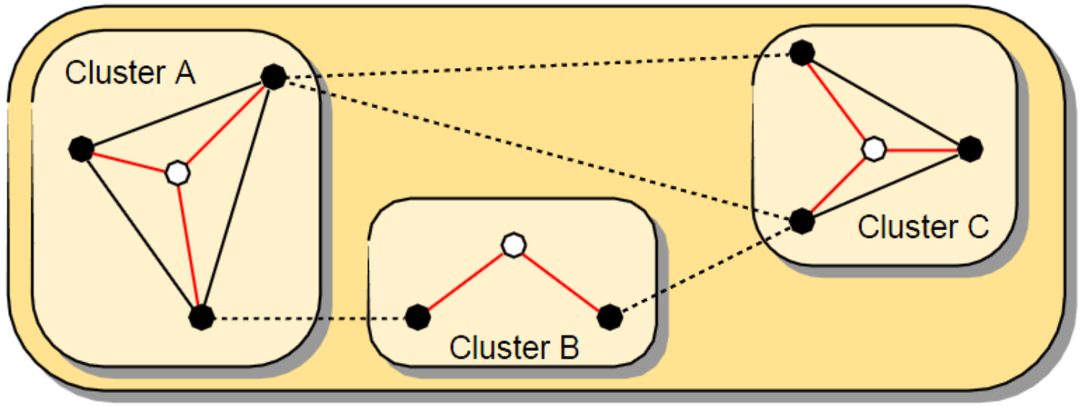


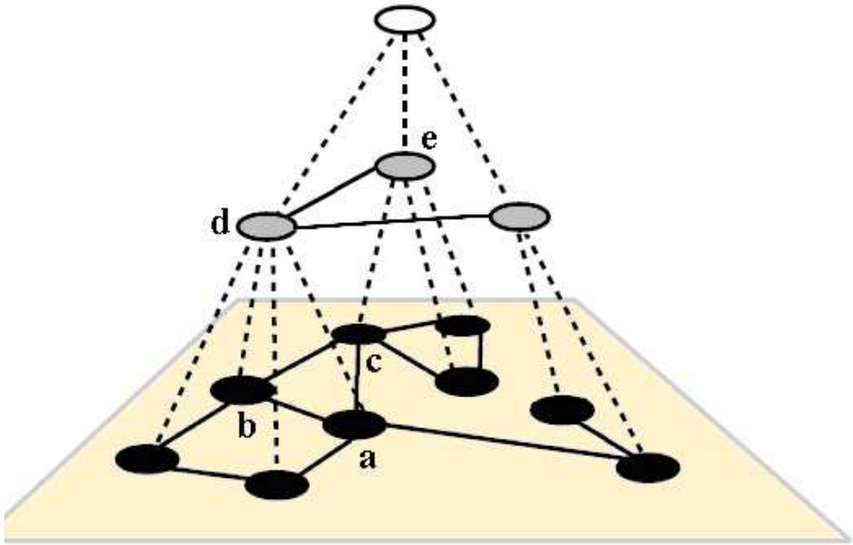
Network Data (Ch 9) II

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

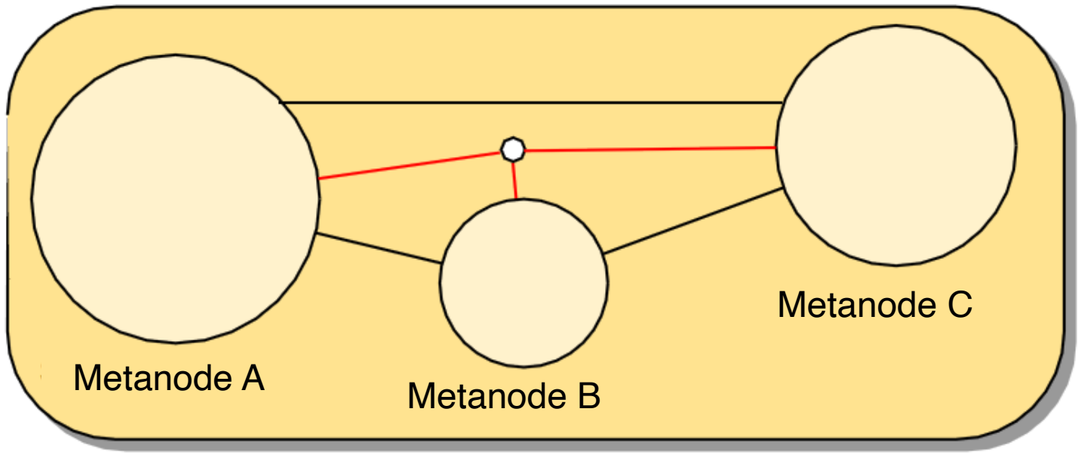
Multilevel networks

- derive cluster hierarchy of metanodes on top of original graph nodes





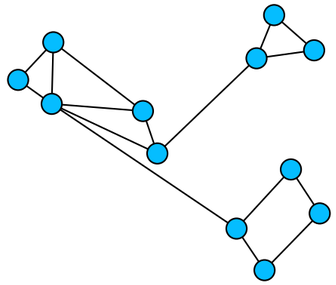
real vertex
virtual vertex
internal spring
virtual spring
external spring

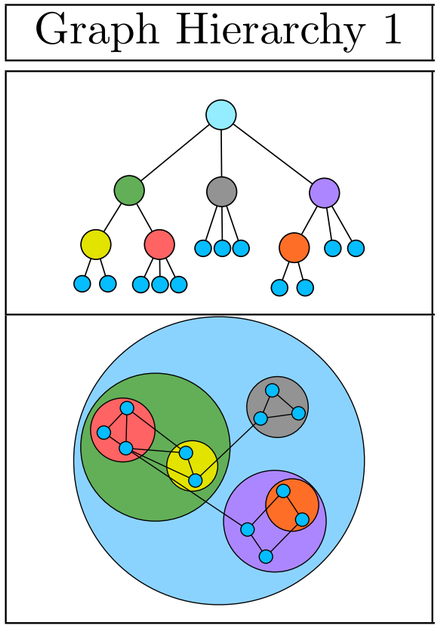


[Schulz 2004]

Idiom: **GrouseFlocks**

- data: compound network
  - network
  - cluster hierarchy atop it
    - derived or interactively chosen
- visual encoding
  - connection marks for network links
  - containment marks for hierarchy
  - point marks for nodes
- dynamic interaction
  - select individual metanodes in hierarchy to expand/ contract

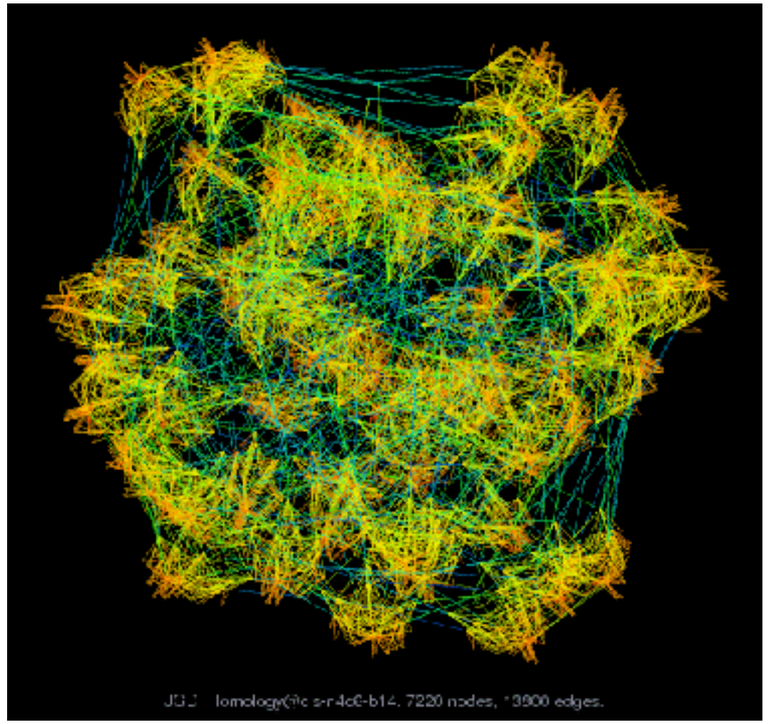




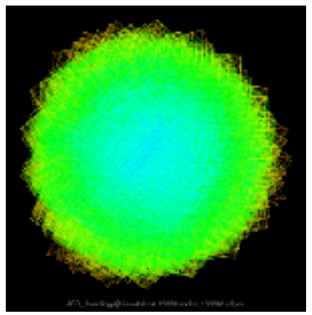
[GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. Archambault, Munzner, and Auber. IEEE TVCG 14(4):900-913, 2008.]

Idiom: **`sfdp`** (multi-level force-directed placement)

- data: compound graph
  - original: network
  - derived: cluster hierarchy atop it
- considerations
  - better algorithm for same encoding technique
    - same: fundamental use of space
    - hierarchy used for algorithm speed/quality but not shown explicitly
- scalability
  - nodes, edges: 1K-10K
  - hairball problem eventually hits



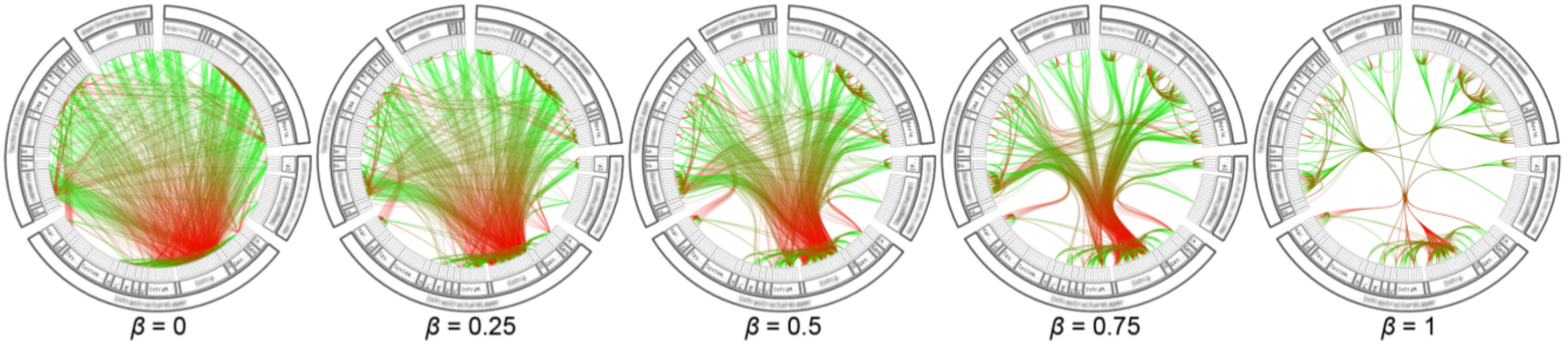
[Efficient and high quality force-directed graph drawing. Hu. The Mathematica Journal 10:37–71, 2005.]



http://www.research.att.com/yifanhu/GALLERY/GRAPHS/index1.html

Idiom: **hierarchical edge bundling**

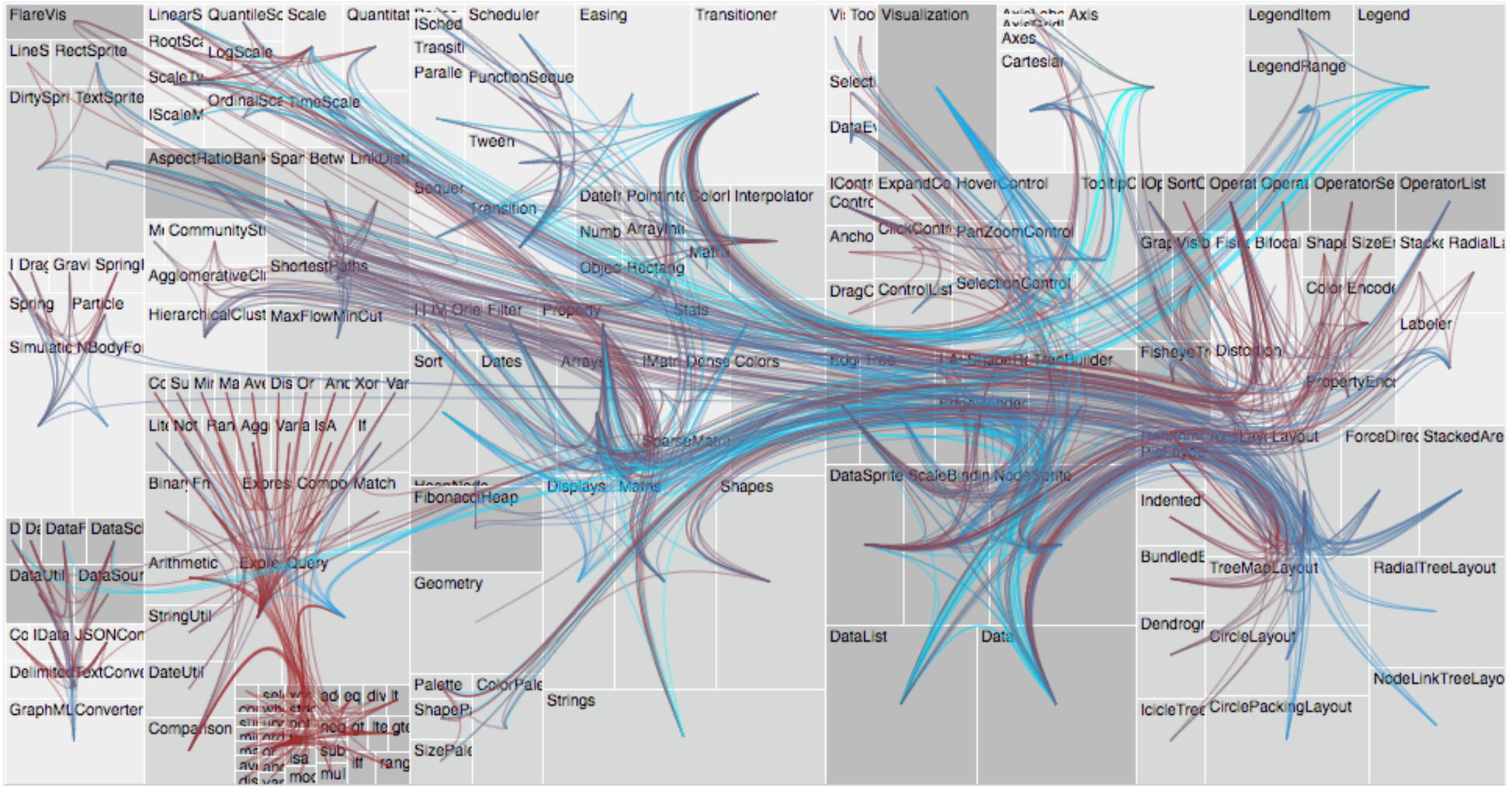
- data
  - any layout of compound network
    - network: software classes (nodes), import/export between classes (links)
    - cluster hierarchy: class package structure
  - derived: bundles of edges with same source/destination (multi-level)
- idiom: curve edge routes according to bundles
- task: edge clutter reduction

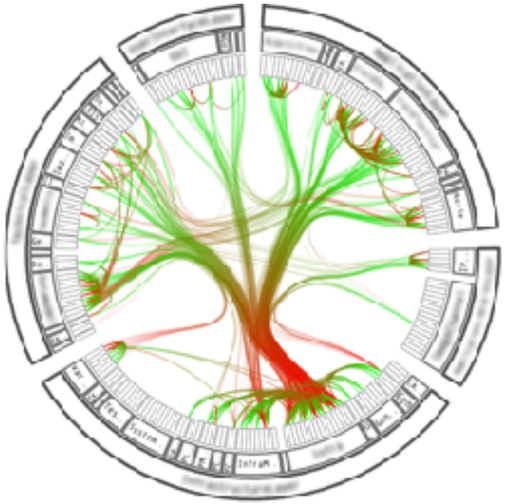


[Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. Danny Holten. TVCG 12(5):741-748 2006]

Hierarchical edge bundling

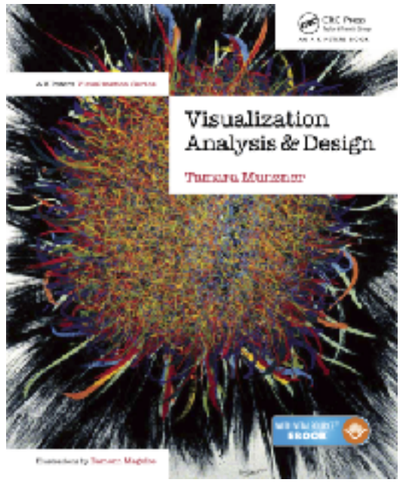
- works for any layout: treemap vs radial





[Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. Danny Holten. TVCG 12(5):741-748 2006]

Visualization Analysis & Design



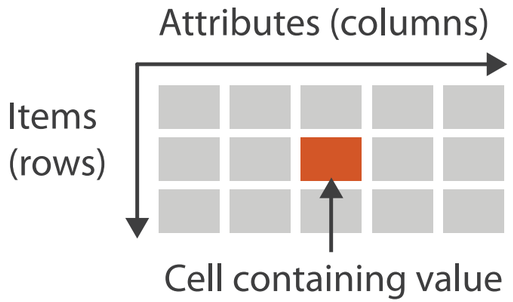
Spatial Data (Ch 9)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

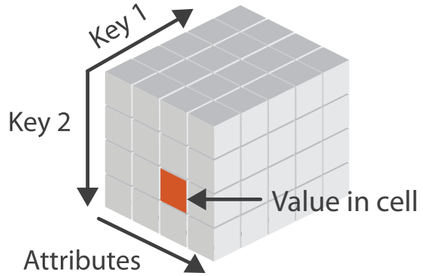
Focus on Spatial

Dataset Types

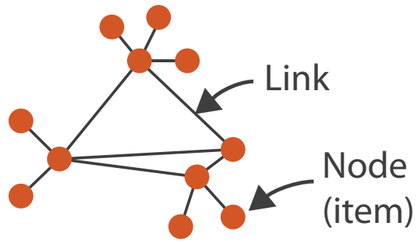
Tables



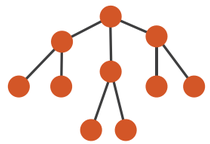
Multidimensional Table



Networks

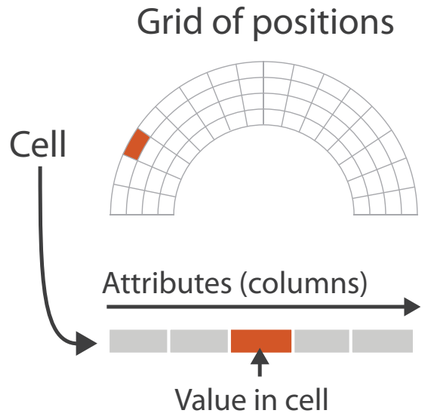


Trees

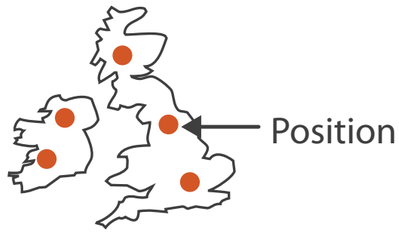


Spatial

Fields (Continuous)



Geometry (Spatial)



How?

Encode

Arrange

- Express
- Separate
- Order
- Align
- Use

Map

from categorical and ordered attributes

- Color
  - *Hue*
  - *Saturation*
  - *Luminance*
- Size, Angle, Curvature, ...
- Shape
- Motion

Direction, Rate, Frequency, ...

Manipulate

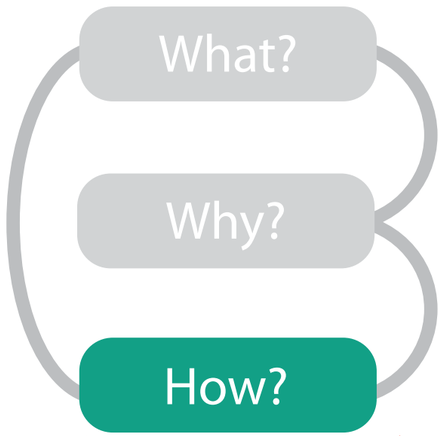
- Change
- Select
- Navigate

Facet

- Juxtapose
- Partition
- Superimpose

Reduce

- Filter
- Aggregate
- Embed



How?

Encode

Arrange

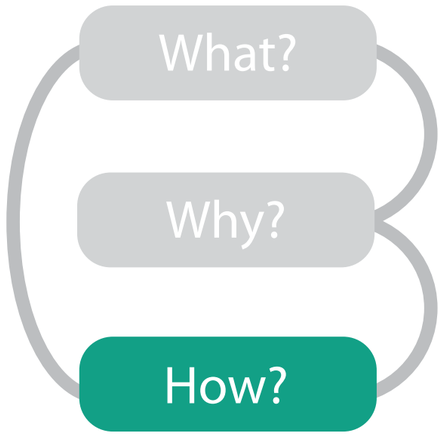
- Express
- Separate
- Order
- Align
- Use

Map

from categorical and ordered attributes

- Color
  - *Hue*
  - *Saturation*
  - *Luminance*
- Size, Angle, Curvature, ...
- Shape
- Motion

Direction, Rate, Frequency, ...



Manipulate

Change

Select

Navigate

Facet

Juxtapose

Partition

Superimpose

Reduce

Filter

Aggregate

Embed

Spatial data

- use given spatial position
- when?
  - dataset contains spatial attributes and they have primary importance
  - central tasks revolve around understanding spatial relationships

- examples
  - geographical/cartographic data
  - sensor/simulation data

Geographic Maps

Geographic Map

Interlocking marks

- **shape** coded
- **area** coded
- **position** coded

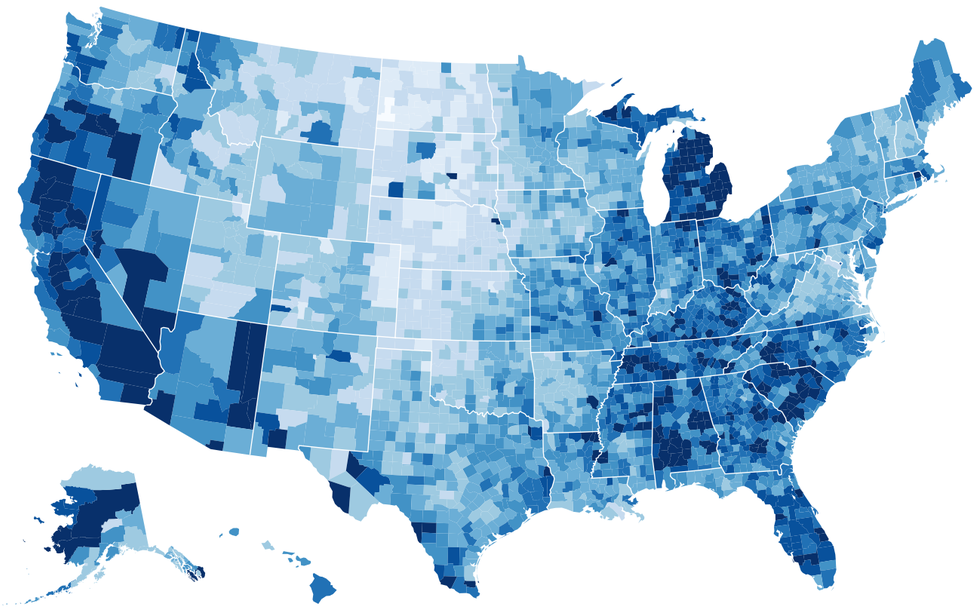
- cannot encode another attribute with these channels, they're "taken"

Thematic maps

- show spatial variability of attribute ("theme")
  - combine geographic / reference map with (simple, flat) tabular data
  - join together
    - region: interlocking area marks (provinces, countries with outline shapes)
      - also could have point marks (cities, locations with 2D lat/lon coords)
    - region: categorical key attribute in table
      - use to look up value attributes
- major idioms
  - choropleth
  - symbol maps
  - cartograms
  - dot density maps

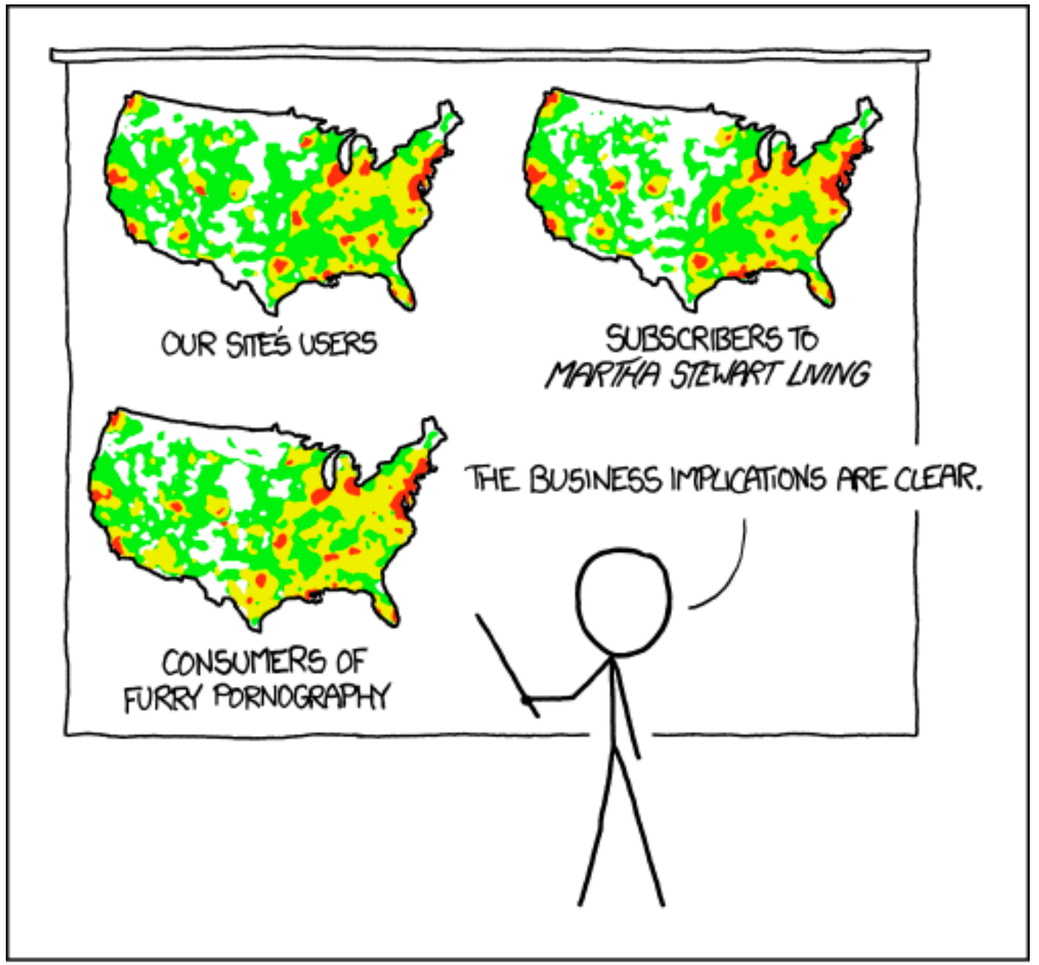
Idiom: **choropleth map**

- use given spatial data
  - when central task is understanding spatial relationships
- data
  - geographic geometry
  - table with 1 quant attribute per region
- encoding
  - position:
    use given geometry for area mark boundaries
  - color:
    sequential segmented colormap



http://bl.ocks.org/mbostock/4060606

Beware: Population maps trickiness!

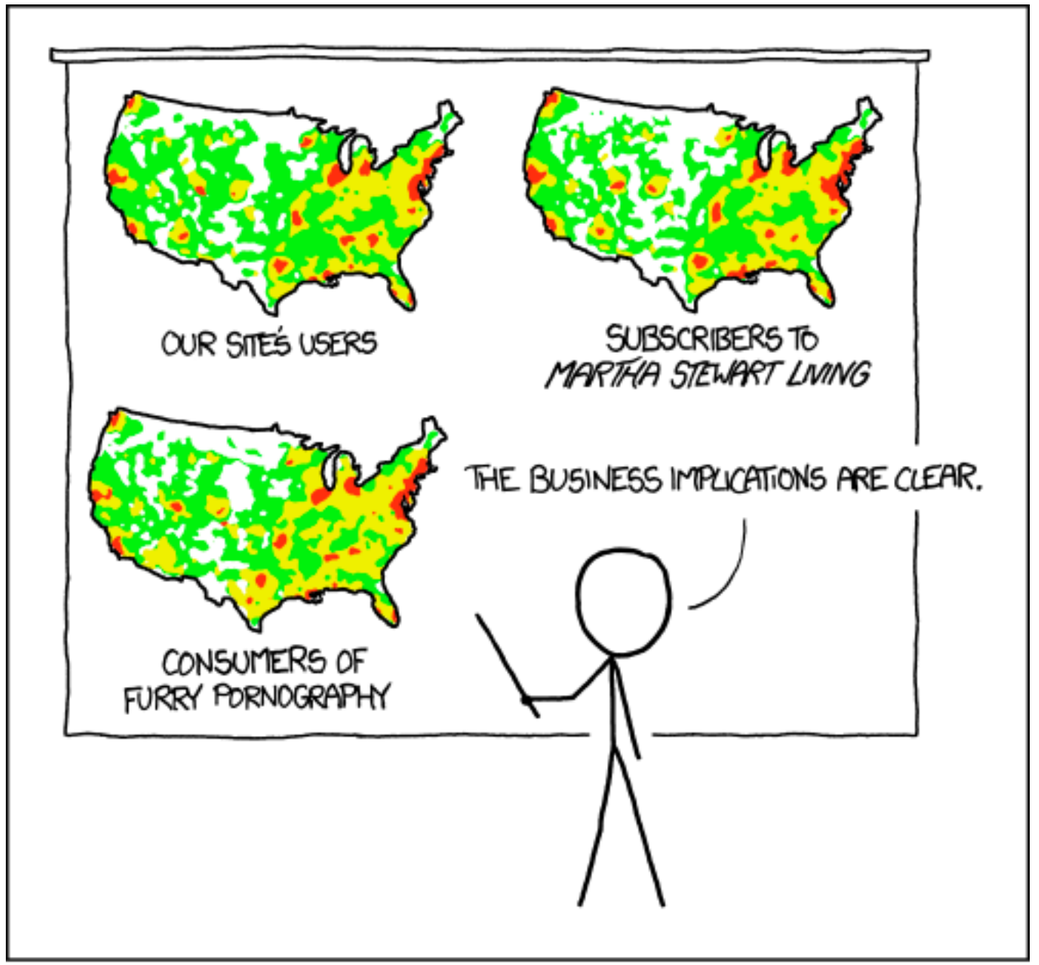


PET PEEVE #208:
GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS

[https://xkcd.com/1138]

Beware: Population maps trickiness!

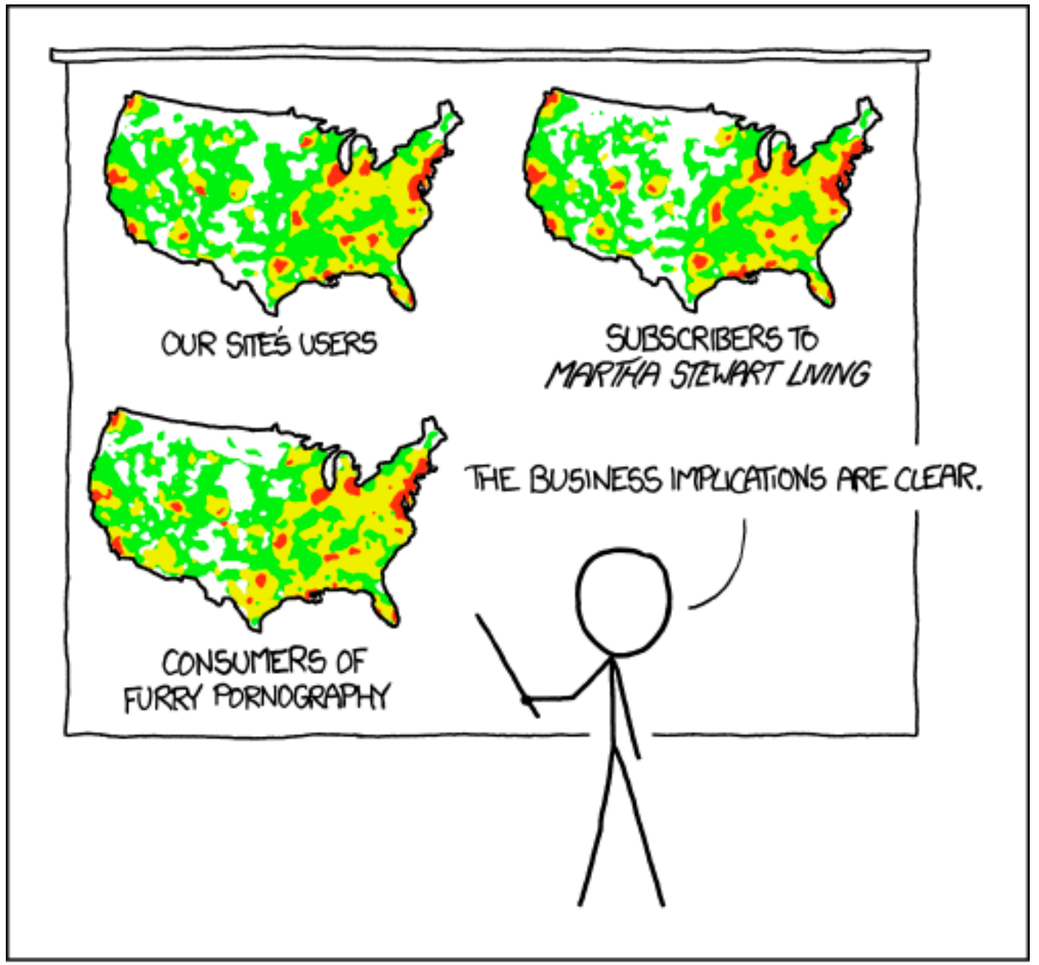
- spurious correlations: most attributes just show where people live



[https://xkcd.com/1138]

Beware: Population maps trickiness!

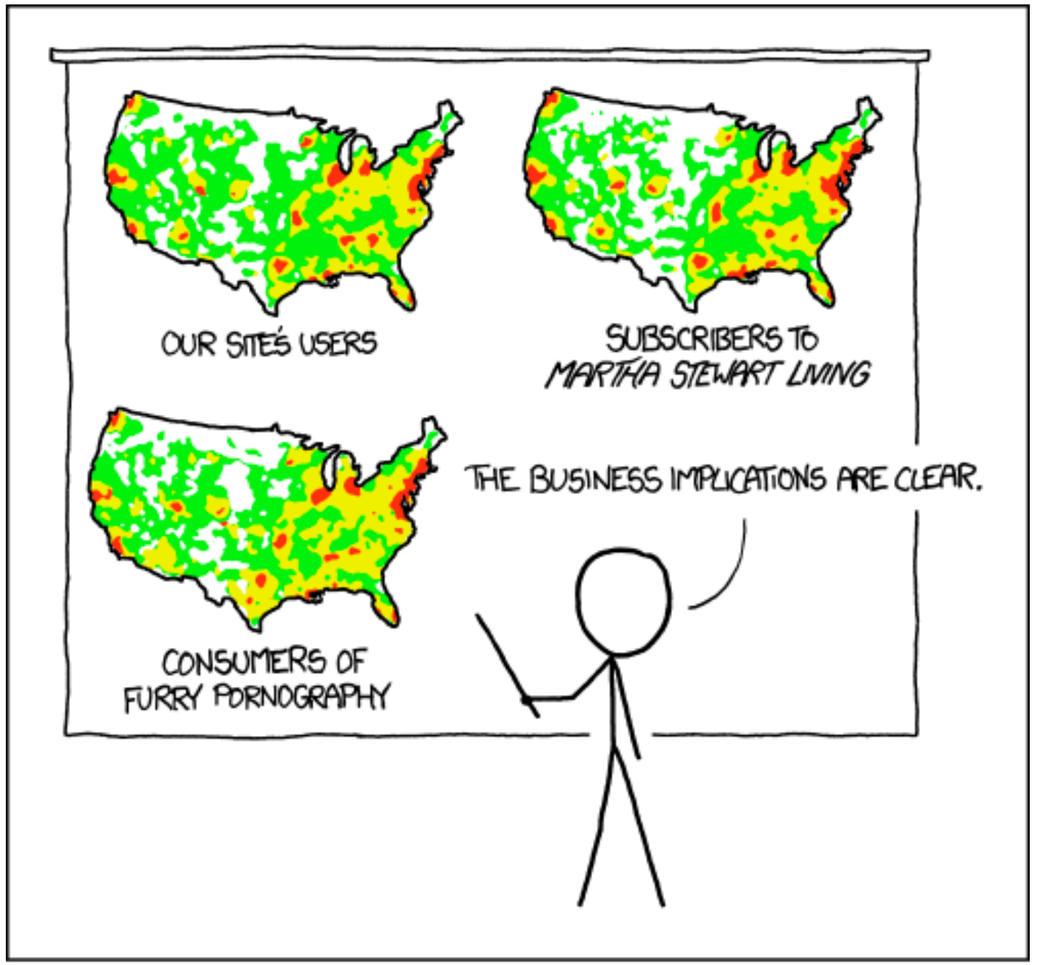
- spurious correlations: most attributes just show where people live
- consider when to normalize by population density
  - encode raw data values
    - tied to underlying population
  - but should use normalized values
    - unemployed people per 100 citizens, mean family income



[https://xkcd.com/1138]

Beware: Population maps trickiness!

- spurious correlations: most attributes just show where people live
- consider when to normalize by population density
  - encode raw data values
    - tied to underlying population
  - but should use normalized values
    - unemployed people per 100 citizens, mean family income
- general issue
  - absolute counts vs relative/normalized data
  - failure to normalize is common error



[https://xkcd.com/1138]

Choropleth maps: Recommendations

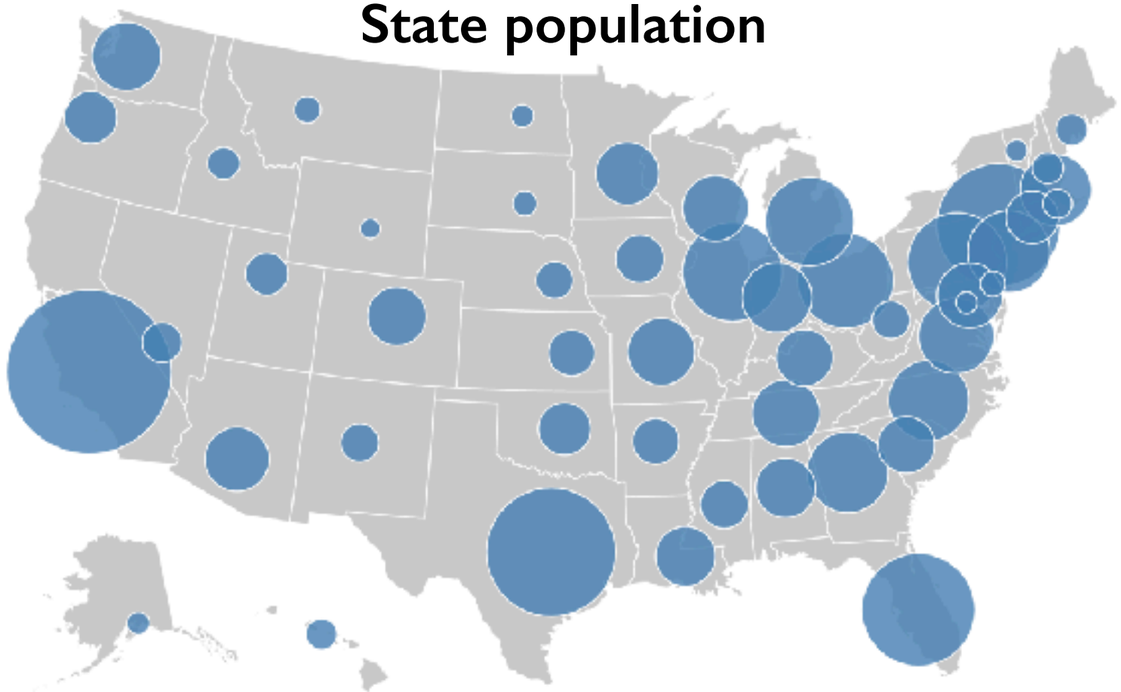
- only use when central task is understanding spatial relationships
- show only one variable at a time
- normalize when appropriate
- be careful when choosing colors & bins
- best case: regions are roughly equal sized

Choropleth map: Pros & cons

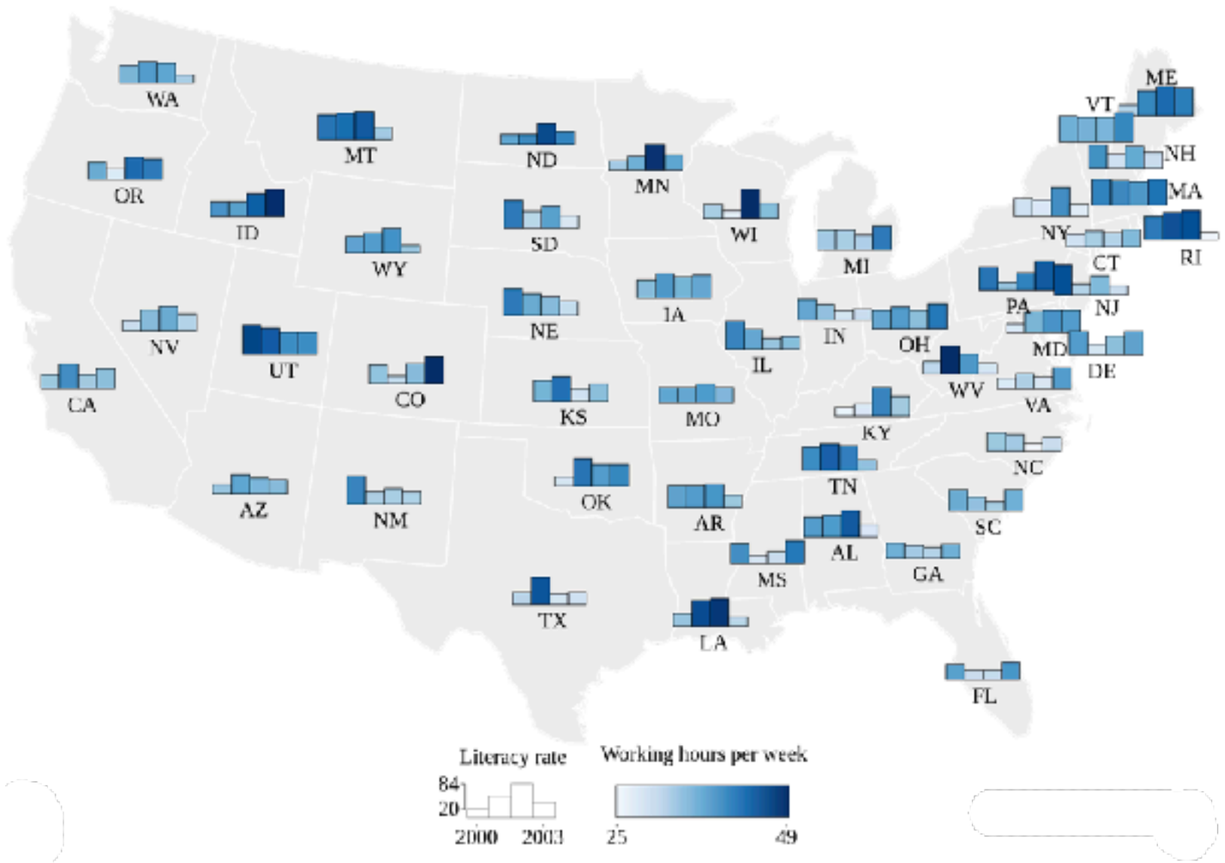
- pros
  - easy to read and understand
  - well established visualization (no learning curve)
  - data is often collected and aggregated by geographical regions
- cons
  - most effective visual variable used for geographic location
  - visual salience depends on region size, not true importance wrt attribute value
    - large regions appear more important than small ones
  - color palette choice has a huge influence on the result

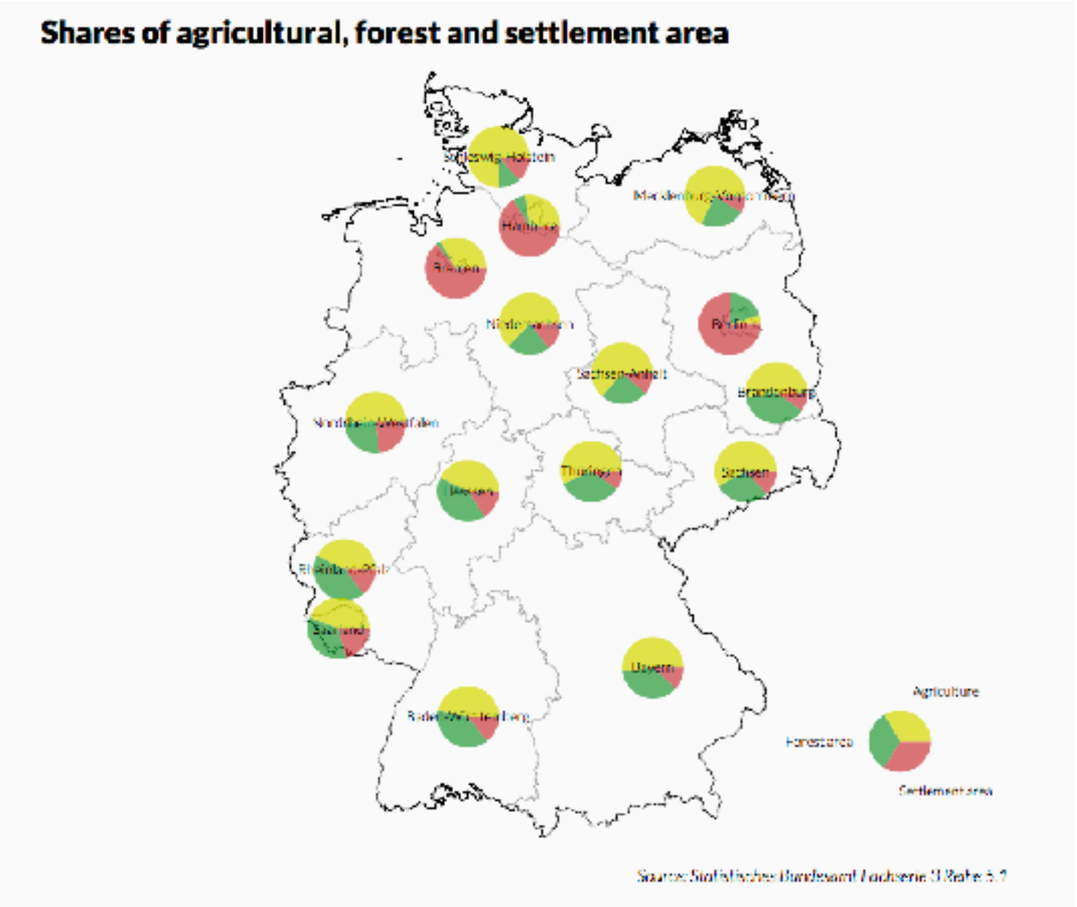
Idiom: **Symbol maps**

- symbol is used to represent aggregated data (mark or glyph)
  - allows use of size and shape and color channels
    - aka proportional symbol maps, graduated symbol maps
- keep original spatial geometry in the background
- often a good alternative to choropleth maps



Symbol maps with glyphs



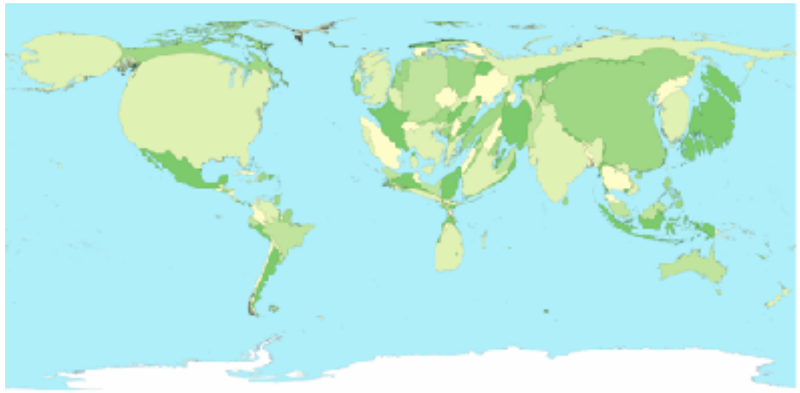


Symbol map: Pros & cons

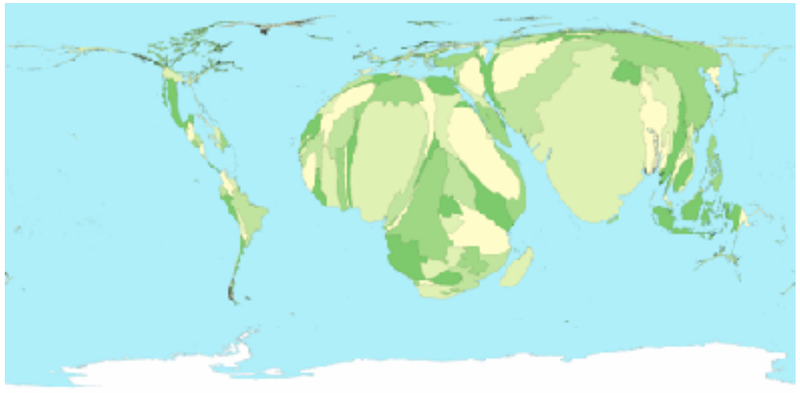
- pros
  - somewhat intuitive to read and understand
  - mitigate problems with region size vs data salience
    - marks: symbol size follows attribute value
    - glyphs: symbol size can be uniform
- cons
  - possible occlusion / overlap
    - symbols could overlap each other
    - symbols could occlude region boundaries
  - complex glyphs may require explanation / training

Idiom: **Contiguous cartogram**

- interlocking marks:
  shape, area, and position coded
- derive new interlocking marks
  - based on combination of original interlocking marks and new quantitative attribute
- algorithm to create new marks
  - input: target size
  - goal: shape as close to the original as possible
  - requirement: maintain constraints
    - relative position
    - contiguous boundaries with their neighbours

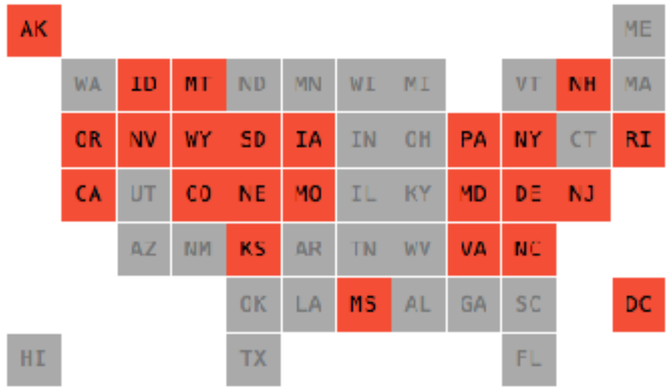


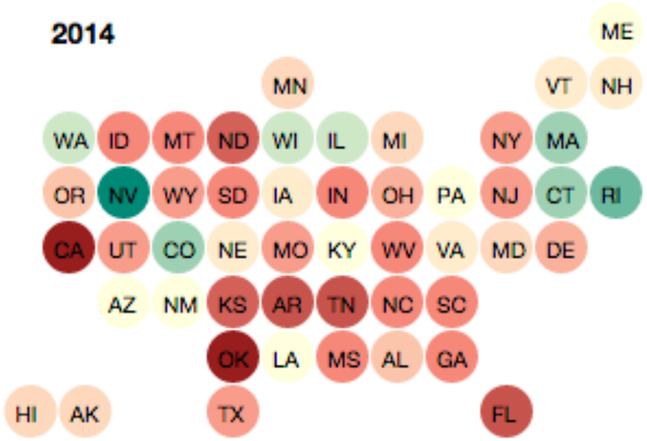
Greenhouse Emissions



Child Mortality

Idiom: **Grid Cartogram**





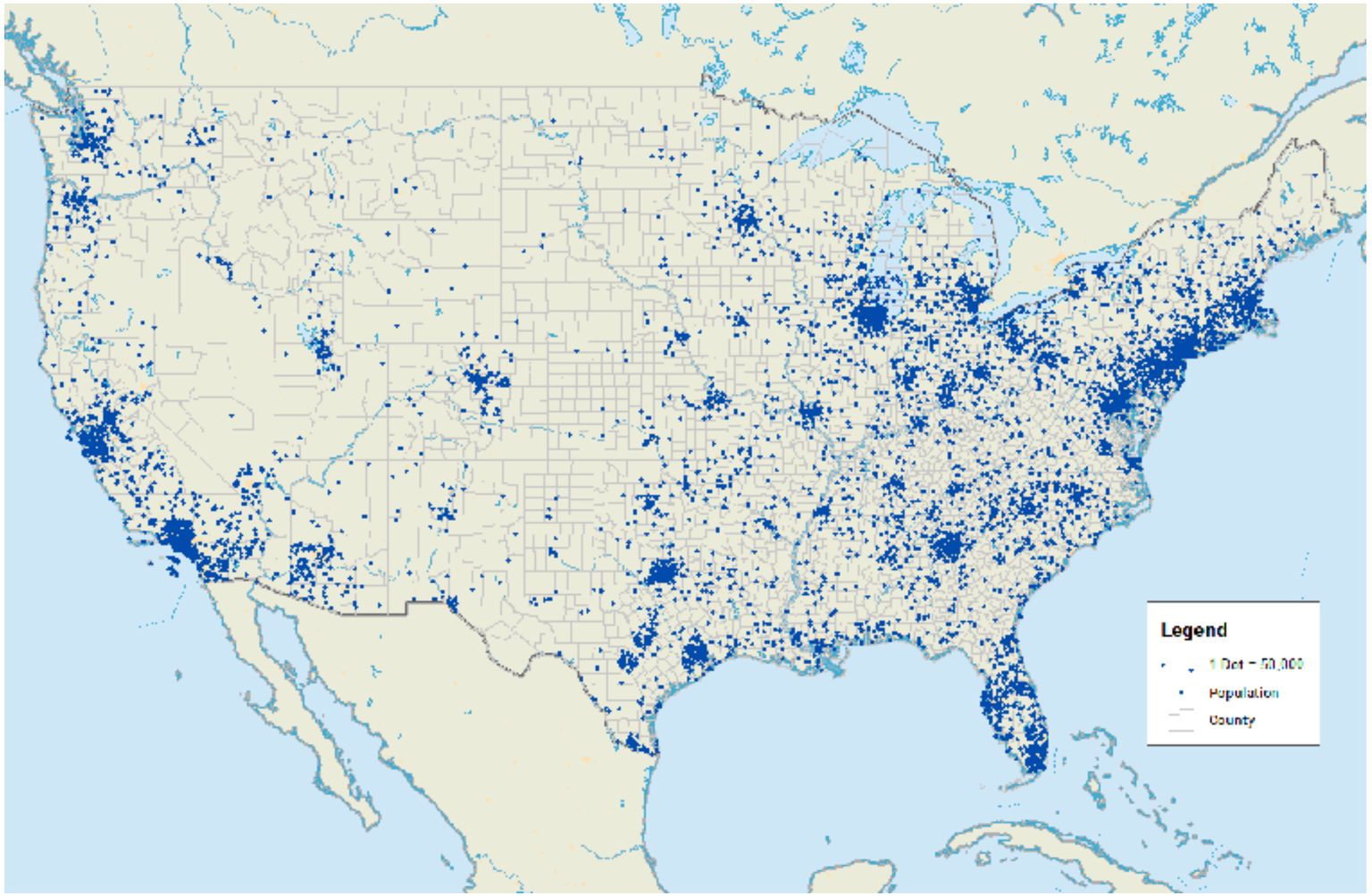
- uniform-sized shapes arranged in rectilinear grid
- maintain approximate spatial position and arrangement

Cartogram: Pros & cons

- pros
  - can be intriguing and engaging
  - best case: strong and surprising size disparities
  - non-contiguous cartograms often easier to understand
- cons
  - require substantial familiarity with original dataset & use of memory
    - compare distorted marks to memory of original marks
    - mitigation strategies: transitions or side by side views
  - major distortion is problematic
    - may be aesthetically displeasing
    - may result in unrecognizable marks
  - difficult to extract exact quantities

Idiom: **Dot density maps**

- visualize distribution of a phenomenon by placing dots
- one symbol represents a constant number of items
  - dots have uniform size & shape
  - allows use of color channel
- task:
  show spatial patterns, clusters

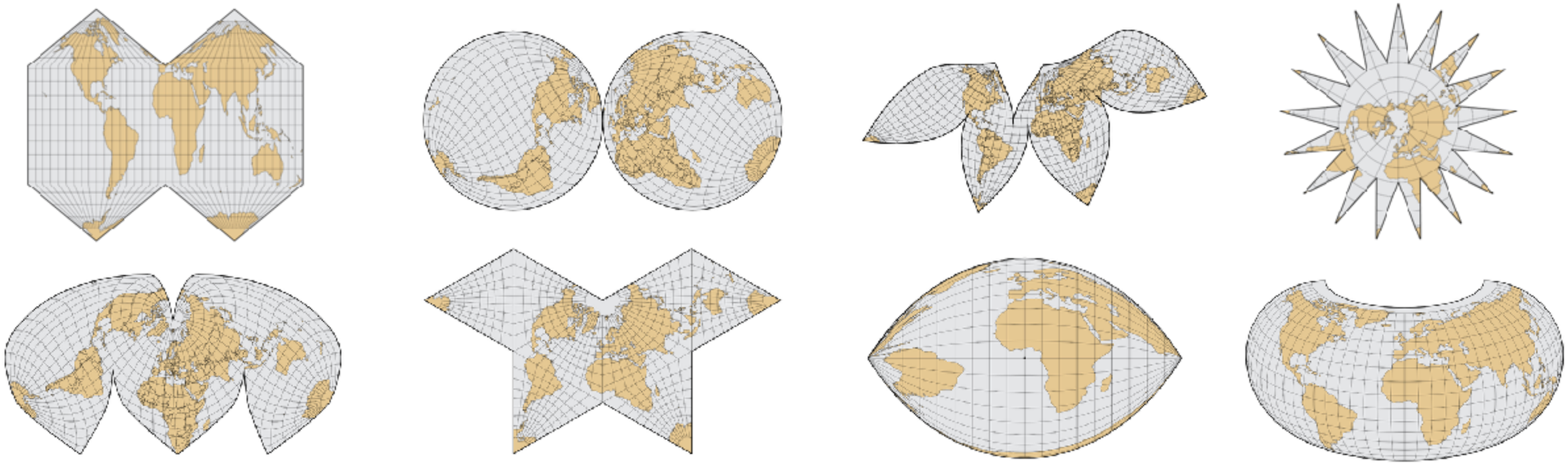


Dot density maps: Pros and cons

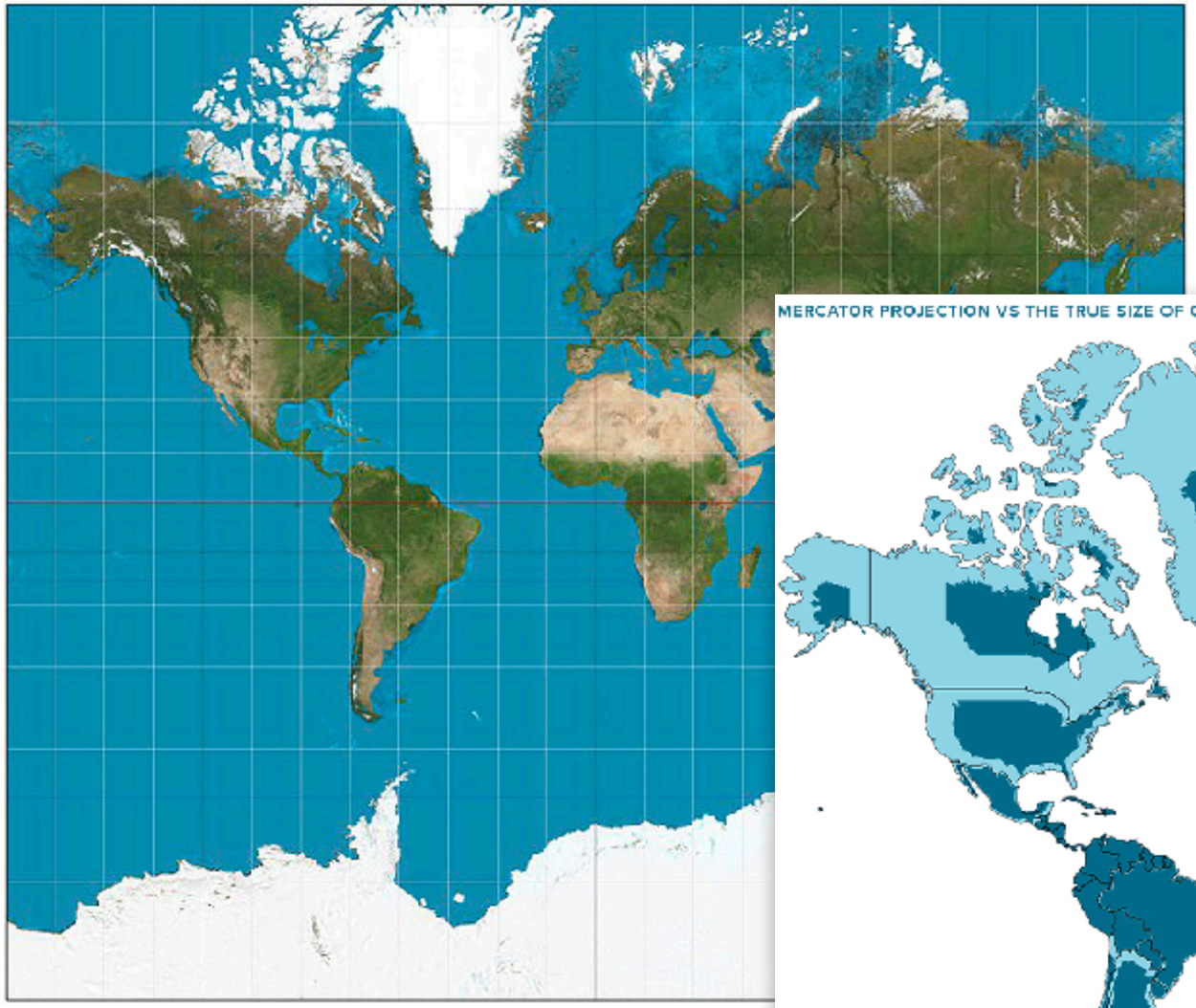
- pros
  - straightforward to understand
  - avoids choropleth non-uniform region size problems
- cons
  - challenge: normalization, just like choropleths
    - show population density (correlated with attribute), not effect of interest
  - perceptual disadvantage:
    difficult to extract quantities
  - performance disadvantage:
    rendering many dots can be slow

Map Projections

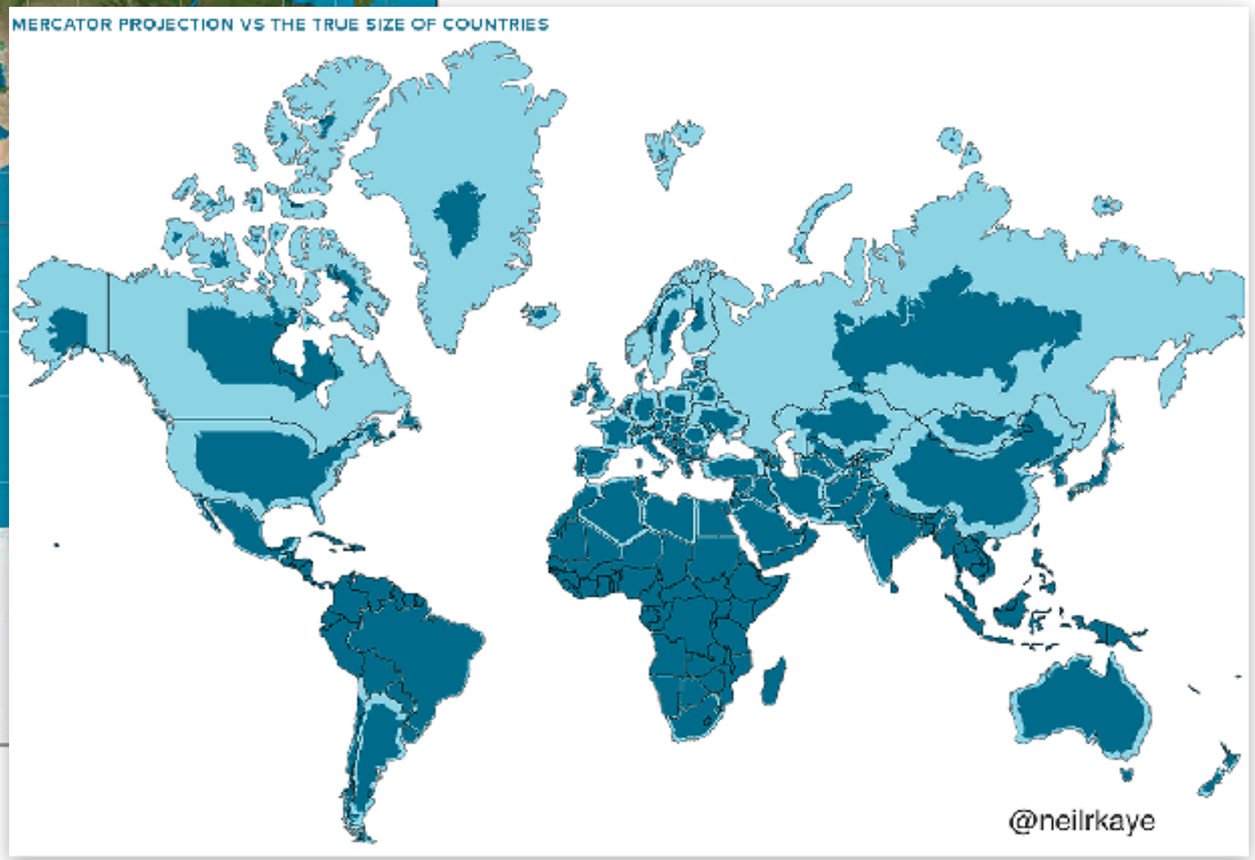
- mathematical functions that map 3D surface geometry of the Earth to 2D maps
- all projections of sphere on plane necessarily distort surface in some way
- interactive: philogb.github.io/page/myriahedral/ and jasondavies.com/maps/



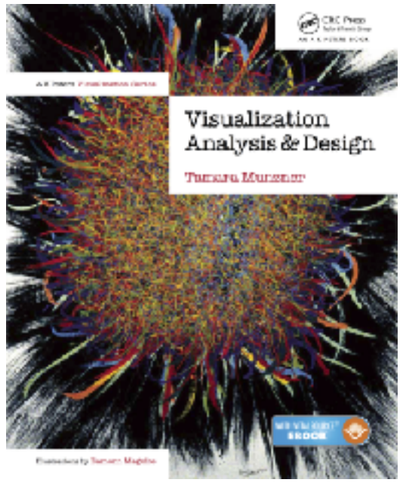
Mercator Projection



» Heavily distorts country sizes; particularly close to the poles.



Visualization Analysis & Design



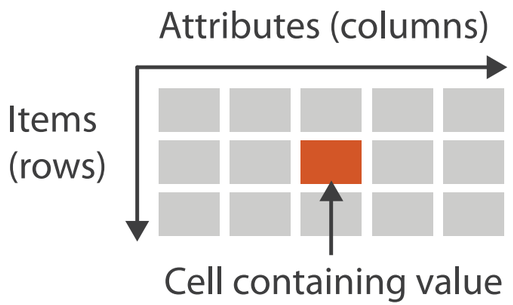
Spatial Data (Ch 9) II

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

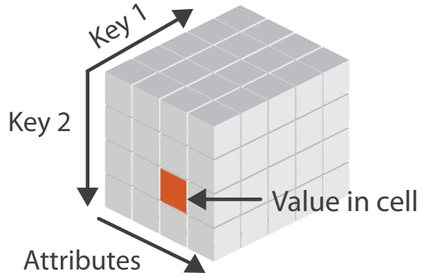
Focus on Spatial

Dataset Types

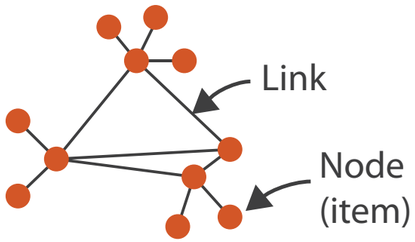
Tables



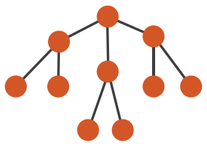
Multidimensional Table



Networks

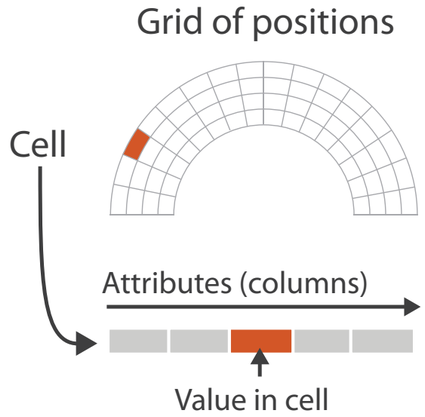


Trees

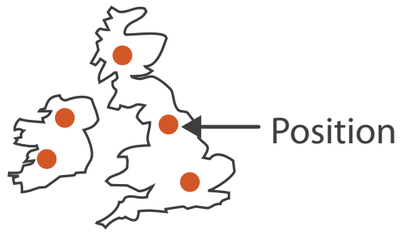


Spatial

Fields (Continuous)



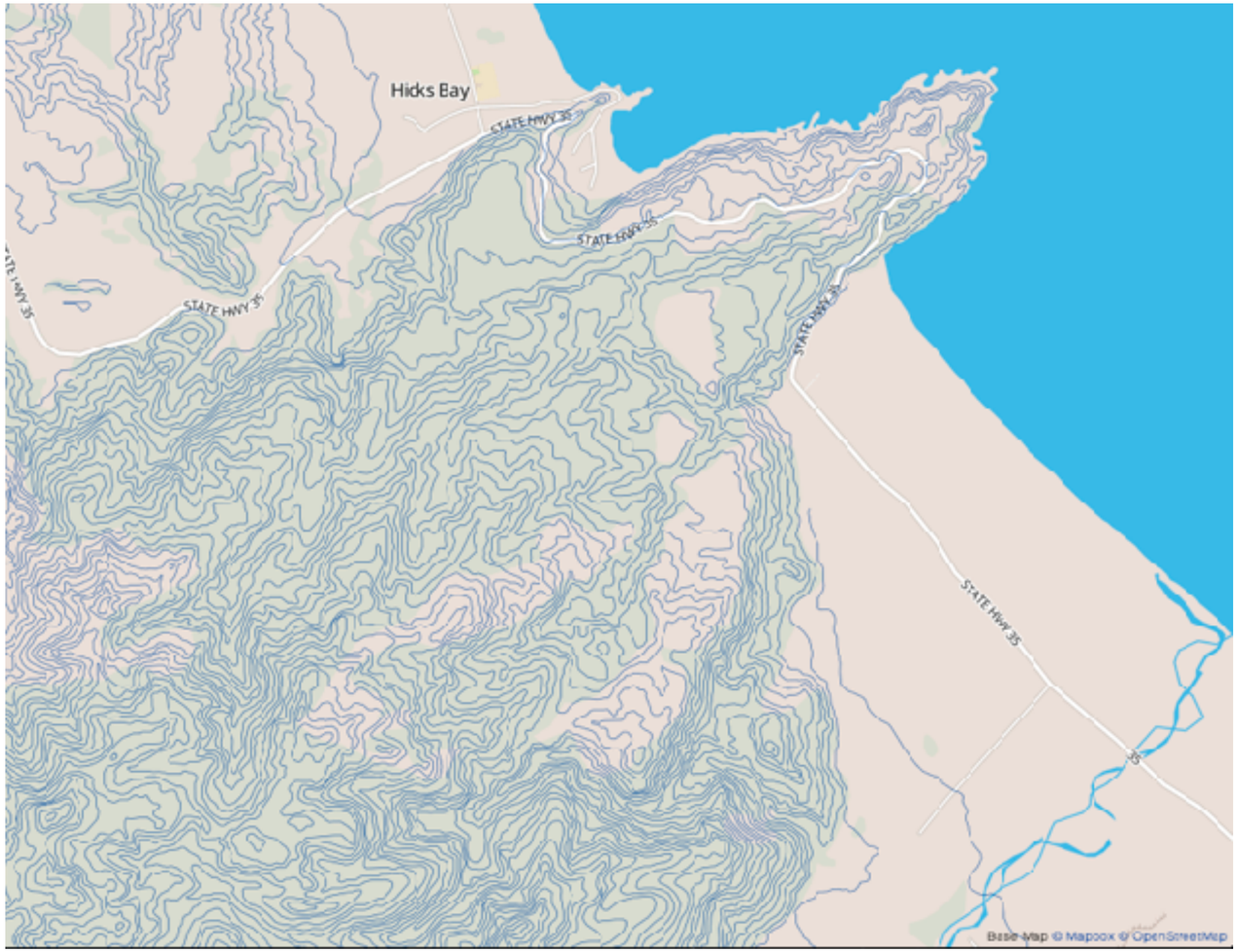
Geometry (Spatial)



Spatial Fields

Idiom: **topographic map**

- data
  - geographic geometry
  - scalar spatial field
    - 1 quant attribute per grid cell
- derived data
  - isoline geometry
    - isocontours computed for specific levels of scalar values
- task
  - understanding terrain shape
    - densely lined regions = steep
- pros
  - use only 2D position, avoid 3D challenges
  - color channel available for other attributes
- cons
  - significant clutter from additional lines



Land Information New Zealand Data Service

Idioms: **isosurfaces, direct volume rendering**

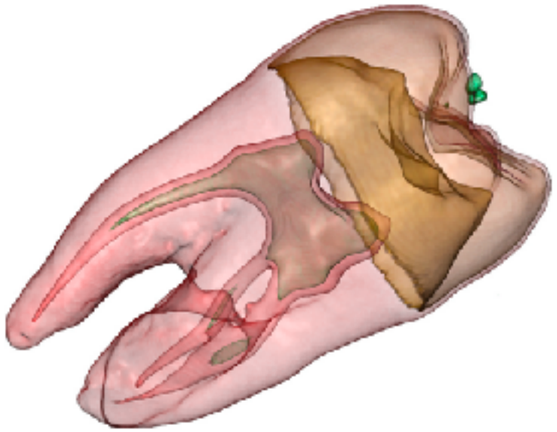
- data
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- task
  - shape understanding, spatial relationships

[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]

[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]

Idioms: **isosurfaces, direct volume rendering**

- data
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- task
  - shape understanding, spatial relationships
- isosurface
  - derived data: isocontours computed for specific levels of scalar values

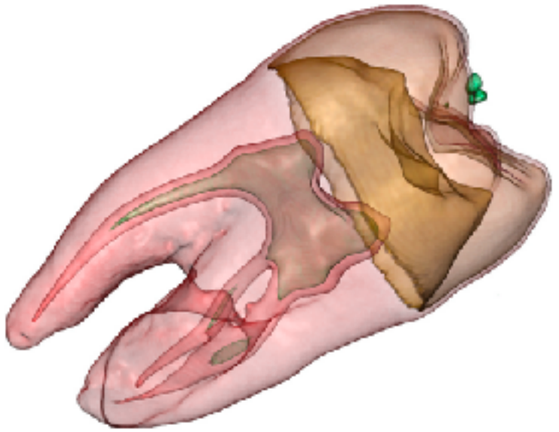


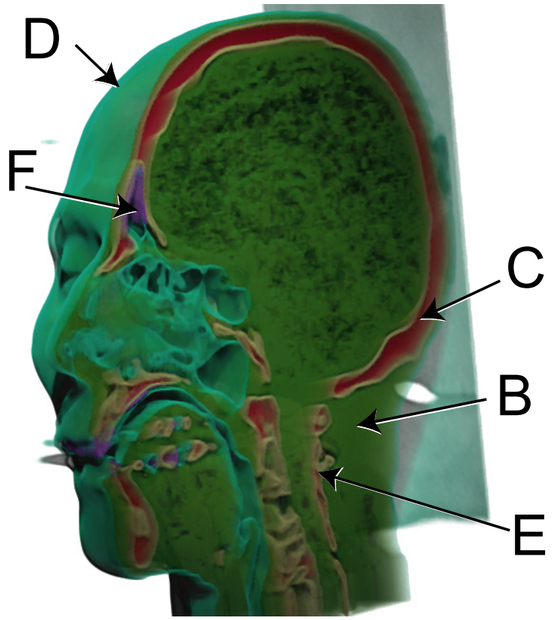
[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]

[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]

Idioms: **isosurfaces, direct volume rendering**

- data
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- task
  - shape understanding, spatial relationships
- isosurface
  - derived data: isocontours computed for specific levels of scalar values
- direct volume rendering
  - transfer function maps scalar values to color, opacity
    - no derived geometry



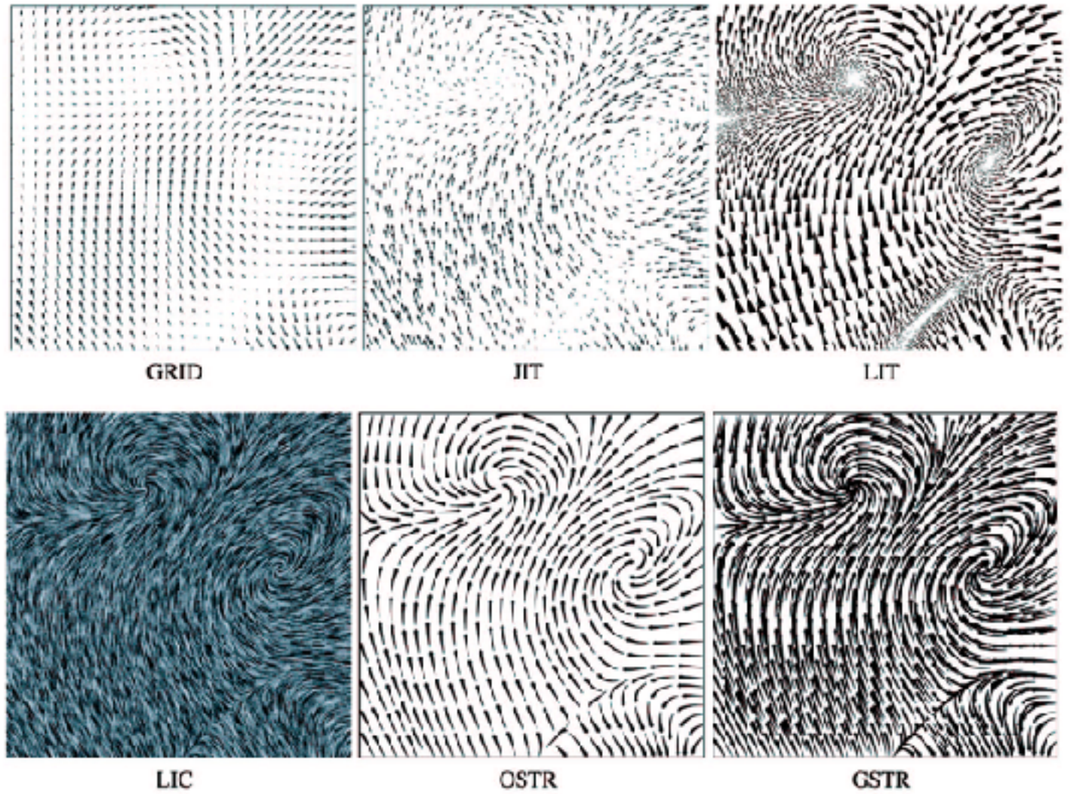


[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]

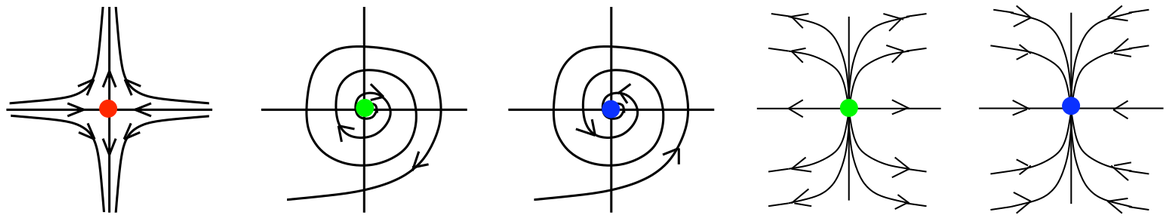
[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]

Vector and tensor fields

- data
  - multiple attribs per cell (vector: 2)
- idiom families
  - flow *glyphs*
    - purely local
  - *geometric* flow
    - derived data from tracing particle trajectories
    - sparse set of seed points
  - *texture* flow
    - derived data, dense seeds
  - *feature* flow
    - global computation to detect features



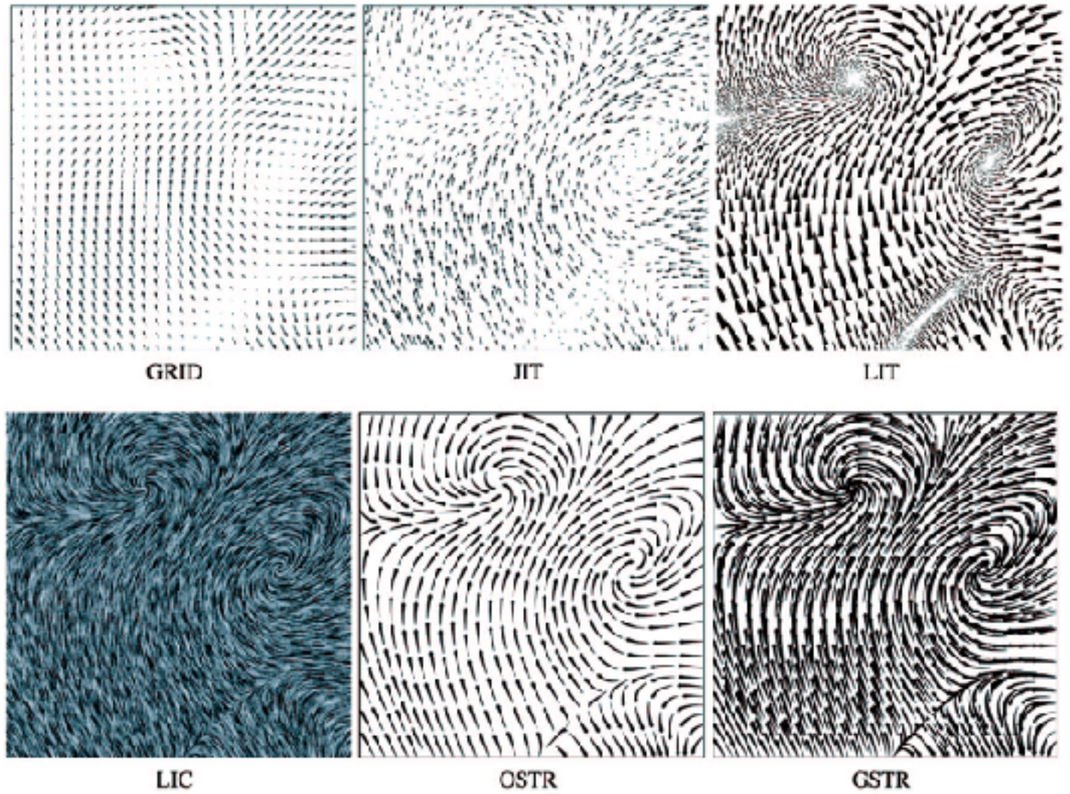
[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. IEEE Trans. Visualization and Computer Graphics (TVCG) 11:1 (2005), 59–70.]



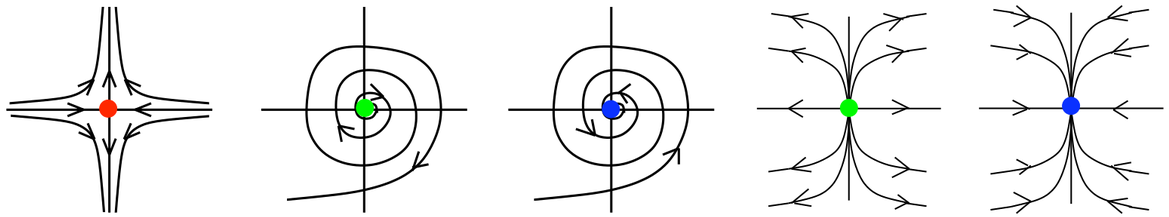
[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. Computers & Graphics 26:2 (2002), 249–257.]

Vector fields

- empirical study tasks
  - finding critical points, identifying their types
  - identifying what type of critical point is at a specific location
  - predicting where a particle starting at a specified point will end up (advection)



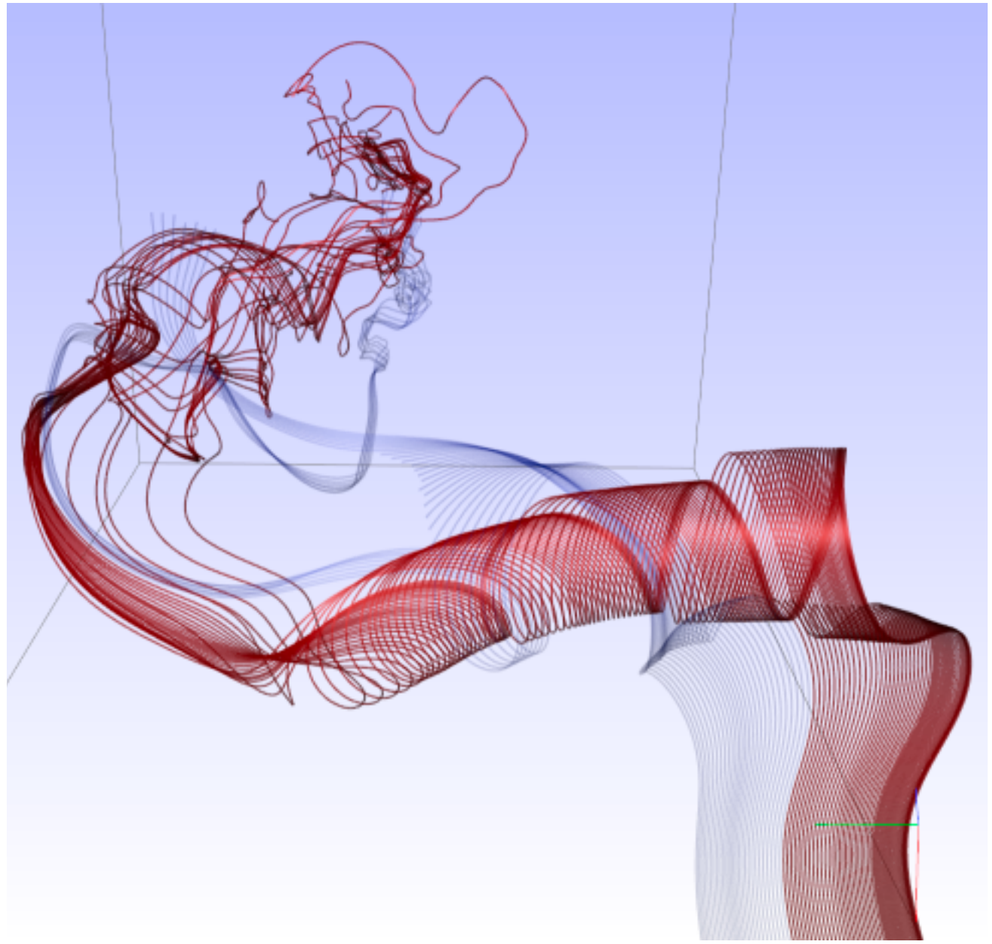
[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. IEEE Trans. Visualization and Computer Graphics (TVCG) 11:1 (2005), 59–70.]



[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. Computers & Graphics 26:2 (2002), 249–257.]

Idiom: **similarity-clustered streamlines**

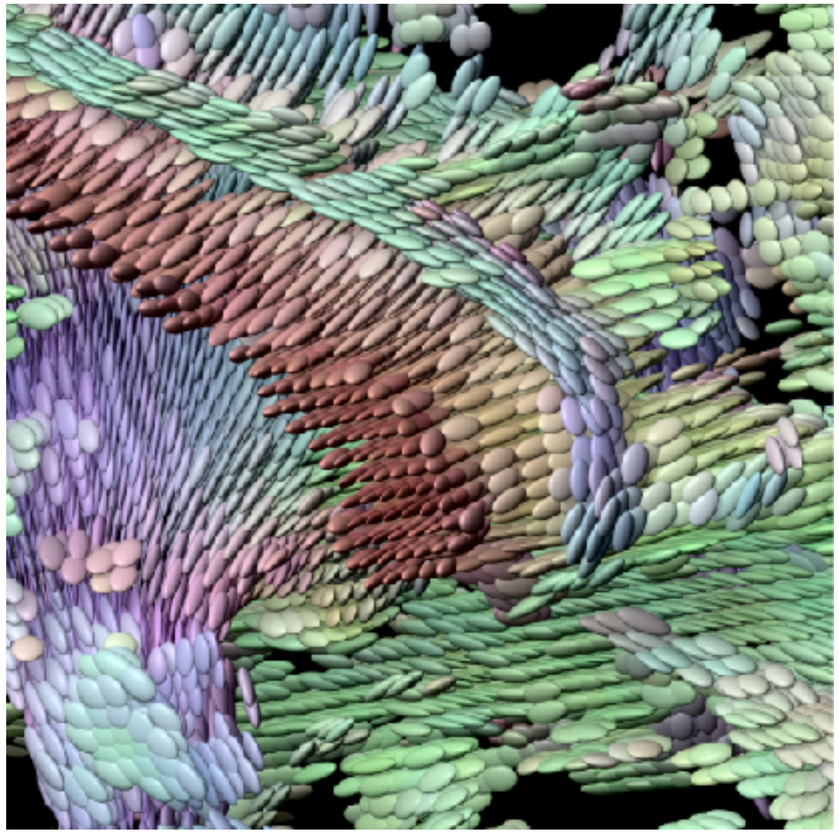
- data
  - 3D vector field
- derived data (from field)
  - streamlines: trajectory particle will follow
- derived data (per streamline)
  - curvature, torsion, tortuosity
  - signature: complex weighted combination
  - compute cluster hierarchy across all signatures
  - encode: color and opacity by cluster
- tasks
  - find features, query shape
- scalability
  - millions of samples, hundreds of streamlines



[Similarity Measures for Enhancing Interactive Streamline Seeding. McLoughlin,. Jones, Laramee, Malki, Masters, and. Hansen. IEEE Trans. Visualization and Computer Graphics 19:8 (2013), 1342–1353.]

Idiom: Ellipsoid Tensor Glyphs

- data
  - tensor field: multiple attributes at each cell (entire matrix)
    - stress, conductivity, curvature, diffusivity...
  - derived data:
    - shape (eigenvalues)
    - orientation (eigenvectors)
- visual encoding
  - glyph: 3D ellipsoid

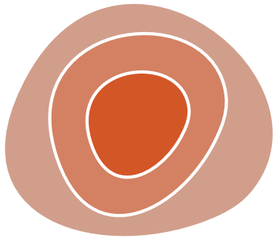


[Superquadric Tensor Glyphs. Kindlmann. Proc. VisSym04, p147-154, 2004.]

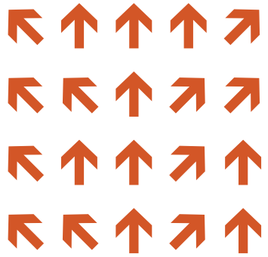
Arrange spatial data

- **Use Given**
  - Geometry
    - *Geographic*

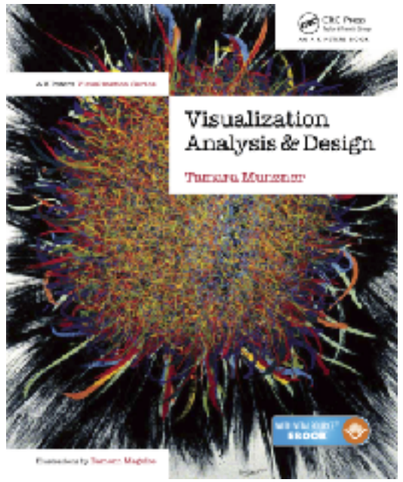
  - Spatial Fields
    - *Scalar Fields (one value per cell)*
      - *Isocontours*
      - *Direct Volume Rendering*



    - *Vector and Tensor Fields (many values per cell)*
      - *Flow Glyphs (local)*
      - *Geometric (sparse seeds)*
      - *Textures (dense seeds)*
      - *Features (globally derived)*



Visualization Analysis & Design



Interactive Views (Ch 11/12)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

How to handle complexity: 1 previous strategy

➔ *Derive*

- derive new data to show within view

How to handle complexity: 1 previous strategy + 2 more

→ *Derive*

- derive new data to show within view
- change view over time
- facet across multiple views

Manipulate

→ Change

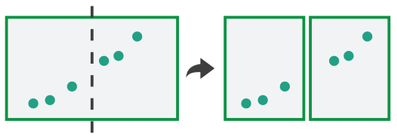
→ Select

→ Navigate

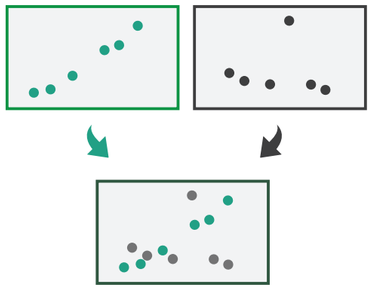
Facet

→ Juxtapose

→ Partition



→ Superimpose



Manipulate View

Manipulate

Change over Time

Change over time

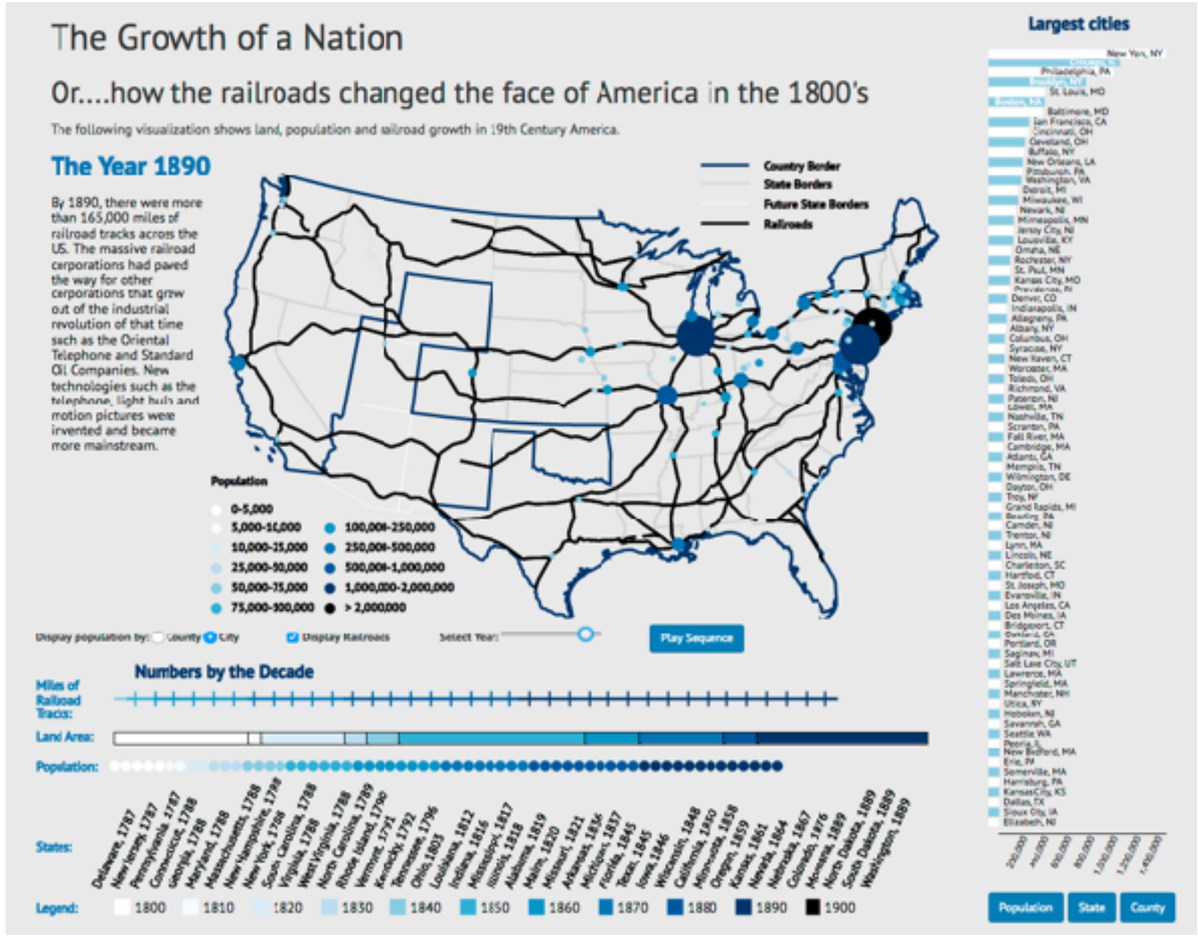
- change any of the other choices
  - encoding itself
  - parameters
  - arrange: rearrange, reorder
  - aggregation level, what is filtered...
  - interaction entails change
- powerful & flexible

Idiom: **Re-encode**

made with Tableau, http://tableausoftware.com

Idiom: **Change parameters**

- widgets and controls
  - sliders, buttons, radio buttons, checkboxes, dropdowns/comboboxes
- pros
  - clear affordances, self-documenting (with labels)
- cons
  - uses screen space
- design choices
  - separated vs interleaved
    - controls & canvas

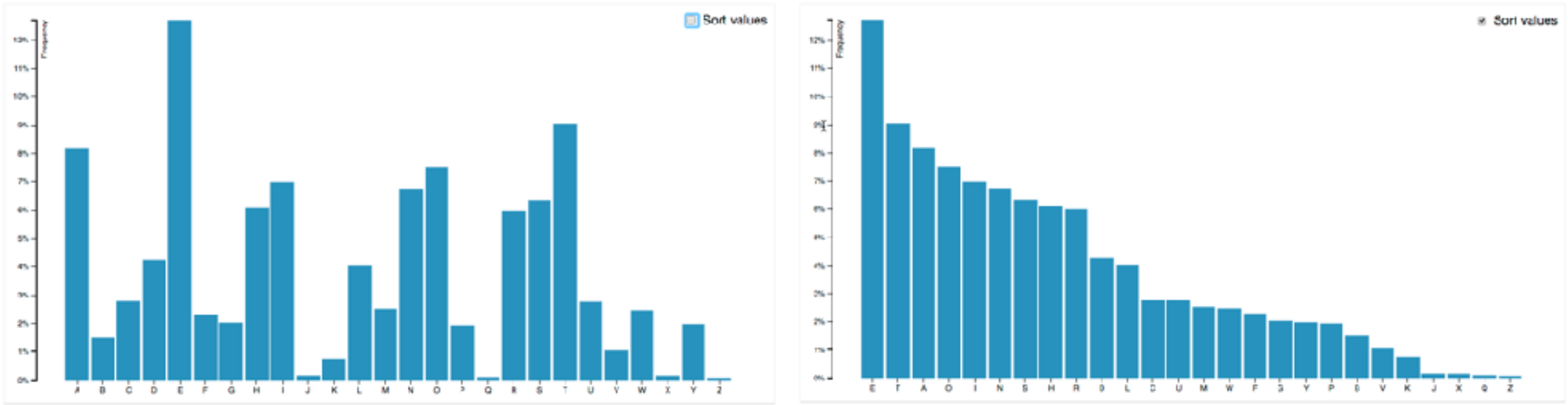


[Growth of a Nation](http://laurenwood.github.io/)

made with D3

Idiom: **Change order/arrangement**

- what: simple table
- how: data-driven reordering
- why: find extreme values, trends



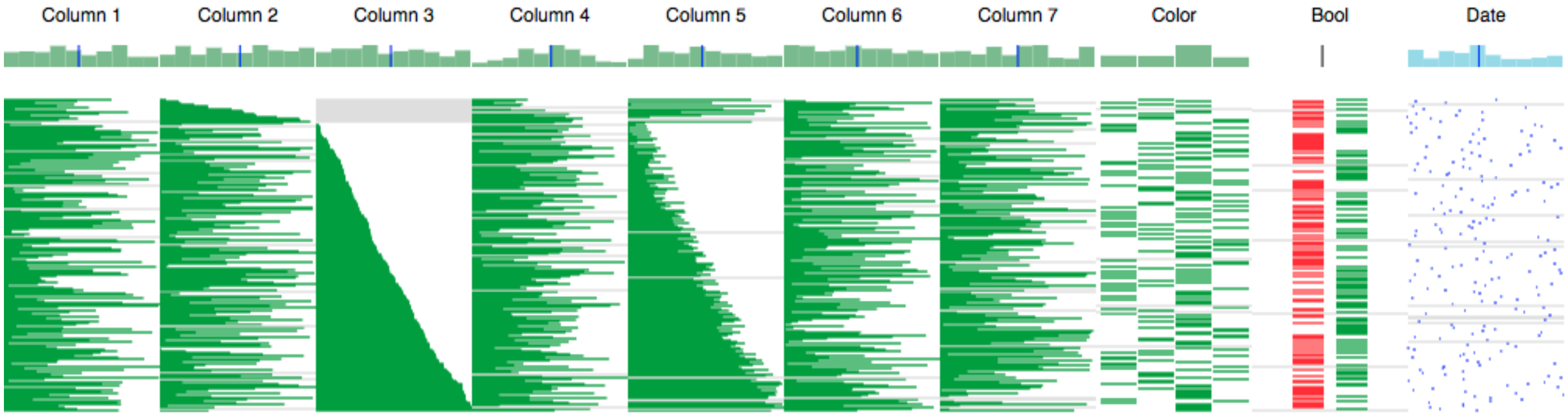
[Sortable Bar Chart] https://observablehq.com/@d3/sortable-bar-chart

made with D3

Idiom: **Reorder**

System: **DataStripes**

- what: table with many attributes
- how: data-driven reordering by selecting column
- why: find correlations between attributes



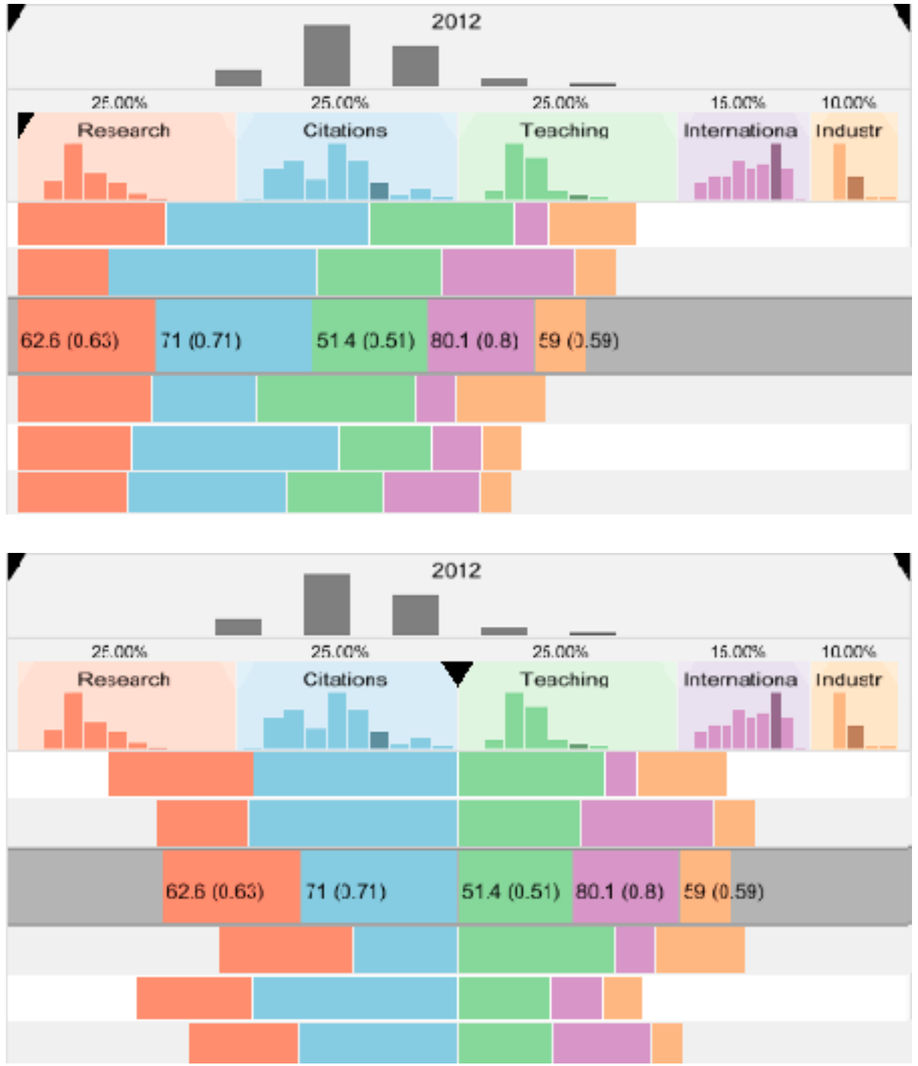
[http://carlmanaster.github.io/datastripes/]

made with D3

Idiom: **Change alignment**

System: **LineUp**

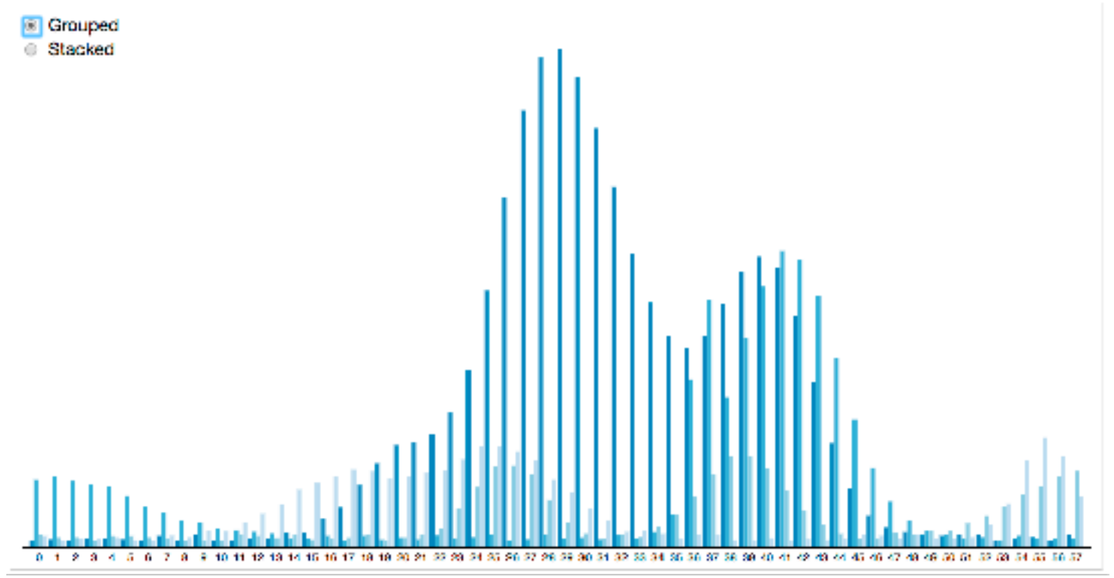
- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison

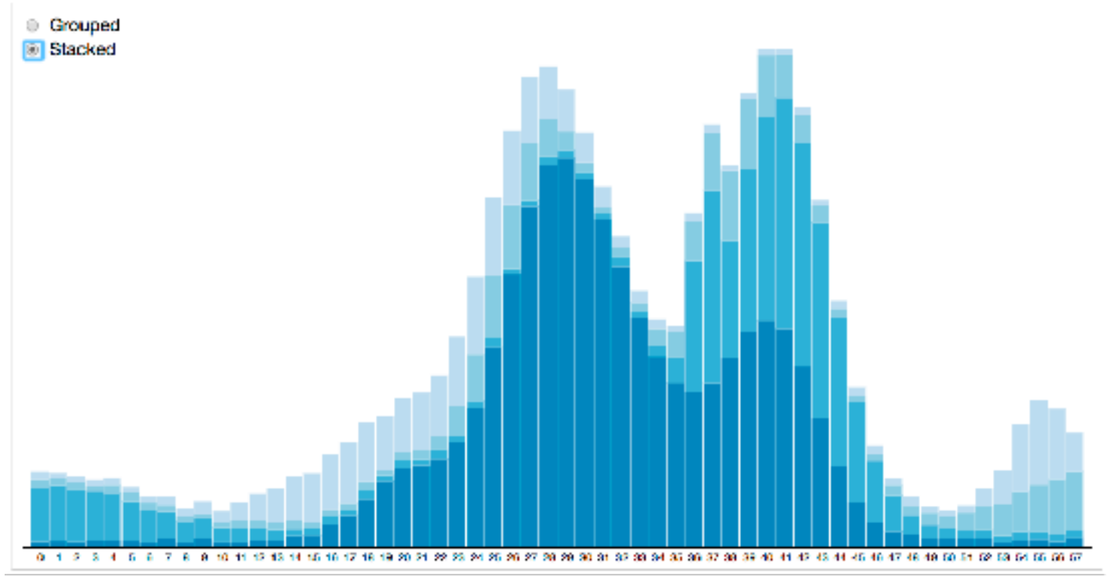


[LineUp: Visual Analysis of Multi-Attribute Rankings. Gratzl, Lex, Gehlenborg, Pfister, and Streit. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2013) 19:12 (2013), 2277–2286.]

Idiom: **Animated transitions - visual encoding change**

- smooth transition from one state to another
  - alternative to jump cuts, supports item tracking
    - best case for animation
  - staging to reduce cognitive load

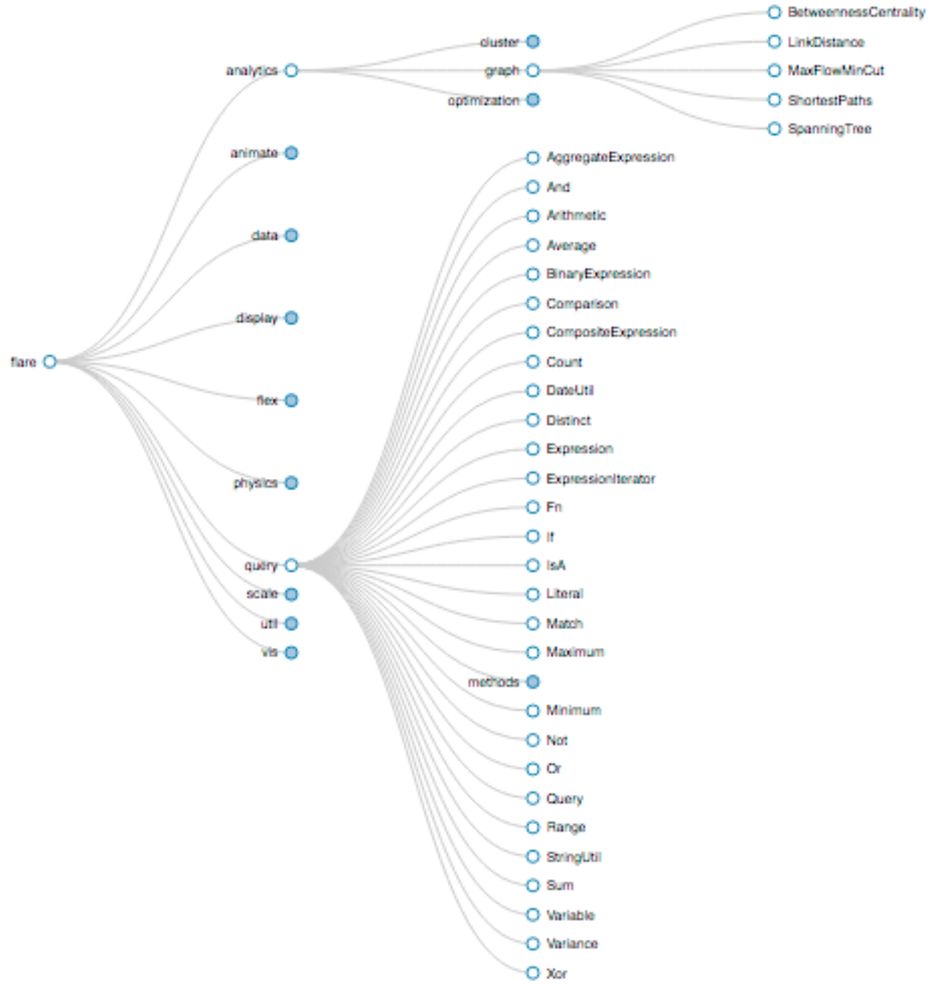




[Stacked to Grouped Bars] https://observablehq.com/@d3/stacked-to-grouped-bars

Idiom: **Animated transition - tree detail**

- animated transition
  - network drilldown/rollup

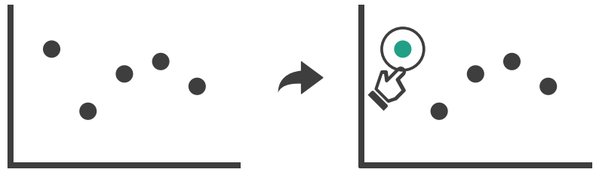


[Collapsible Tree] https://observablehq.com/@d3/collapsible-tree

Manipulate

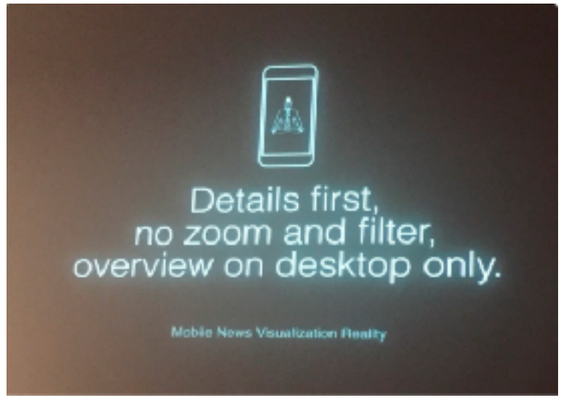
- Change over Time

- Select



Interaction technology

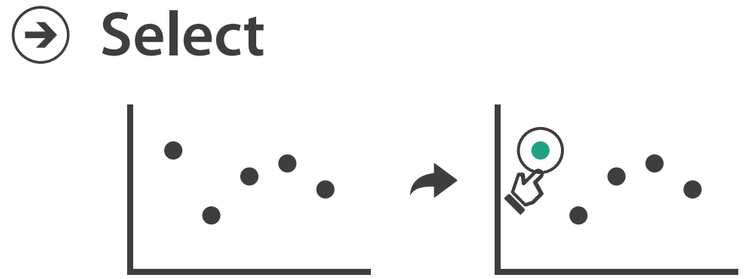
- what do you design for?
  - mouse & keyboard on desktop?
    - large screens, hover, multiple clicks
  - touch interaction on mobile?
    - small screens, no hover, just tap
  - gestures from video / sensors?
    - ergonomic reality vs movie bombast
  - eye tracking?



Data visualization and the news - Gregor Aisch (37 min)
vimeo.com/182590214

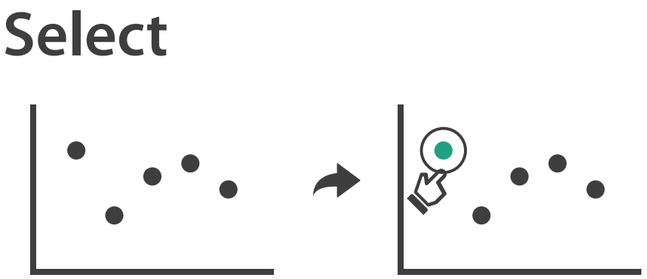
I Hate Tom Cruise - Alex Kauffmann (5 min)
www.youtube.com/watch?v=QXLfT9sFcbc

Selection



- selection: basic operation for most interaction
- design choices
  - how many selection types?
    - interaction modalities
      - click/tap (heavyweight) vs hover (lightweight but not available on most touchscreens)
      - multiple click types (shift-click, option-click, …)
      - proximity beyond click/hover (touching vs nearby vs distant)
    - application semantics
      - adding to selection set vs replacing selection
      - can selection be null?
        - ex: toggle so nothing selected if click on background
      - primary vs secondary (ex: source/target nodes in network)
      - group membership (add/delete items, name group, …)

Highlighting

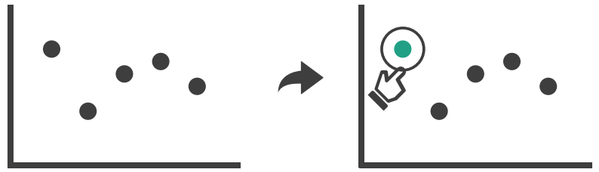


- highlight: change visual encoding for selection targets
  - visual feedback closely tied to but separable from selection (interaction)
- design choices: typical visual channels
  - change item color
    - but hides existing color coding
  - add outline mark
  - change size (ex: increase outline mark linewidth)
  - change shape (ex: from solid to dashed line for link mark)
- unusual channels: motion
  - motion: usually avoid for single view
    - with multiple views, could justify to draw attention to other views

Manipulate

Change over Time

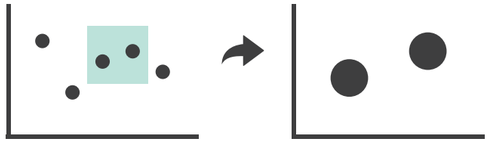
Select



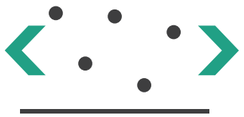
Navigate

Item Reduction

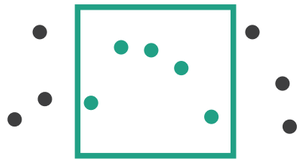
- *Zoom*
  Geometric or *Semantic*



- *Pan/Translate*



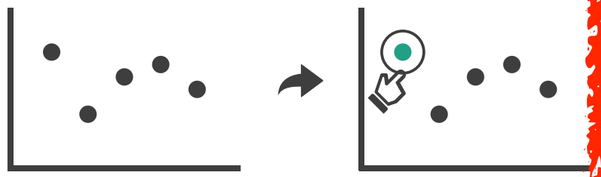
- *Constrained*



Manipulate

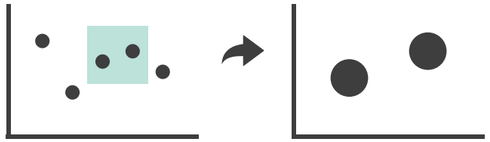
Change over Time

Select



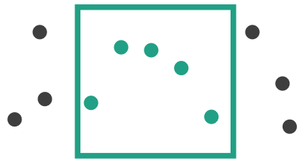
Navigate

- *Zoom*
  Geometric



- *Pan/Translate*

- *Constrained*



Navigate: Changing viewpoint/visibility

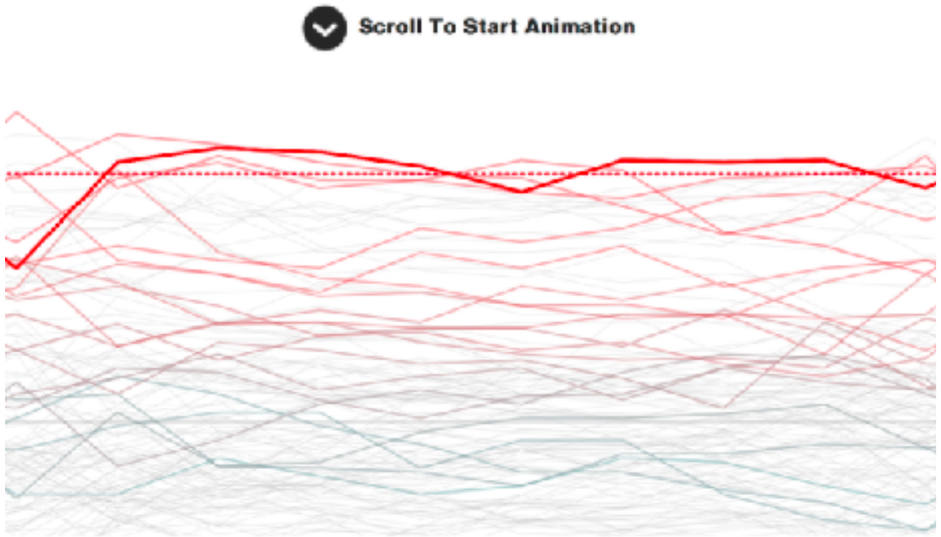
- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
    - move up/down/sideways

Navigate

→ *Pan/Translate*

Idiom: **Scrollytelling**

- how: navigate page by scrolling (panning down)
- pros:
  - familiar & intuitive, from standard web browsing
  - linear (only up & down) vs possible overload of click-based interface choices
- cons:
  - full-screen mode may lack affordances
  - scrolljacking, no direct access
  - unexpected behaviour
  - continuous control for discrete steps



[How to Scroll, Bostock](https://bost.ocks.org/mike/scroll/)

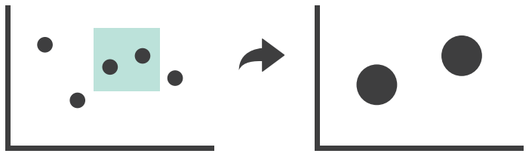
https://eagereyes.org/blog/2016/the-scrollytelling-scourge

Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
    - move up/down/sideways
  - rotate/spin
    - typically in 3D
  - zoom in/out
    - enlarge/shrink world == move camera closer/further
    - geometric zoom: standard, like moving physical object

Navigate

→ *Zoom*
Geometric



→ *Pan/Translate*

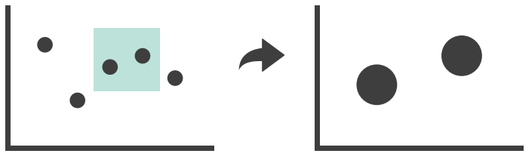
Navigate: Unconstrained vs constrained

- unconstrained navigation
  - easy to implement for designer
  - hard to control for user
    - easy to overshoot/undershoot
- constrained navigation
  - typically uses animated transitions
  - trajectory automatically computed based on selection
    - just click; selection ends up framed nicely in final viewport

Navigate

→ Item Reduction

→ *Zoom*
Geometric or *Semantic*



→ *Pan/Translate*

→ *Constrained*

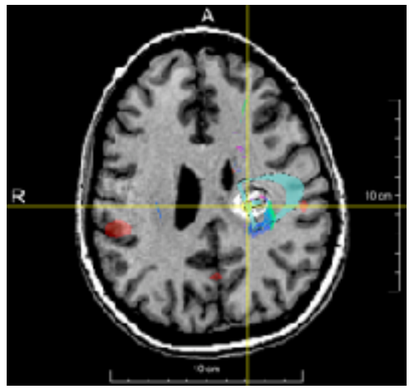
Idiom: **Animated transition + constrained navigation**

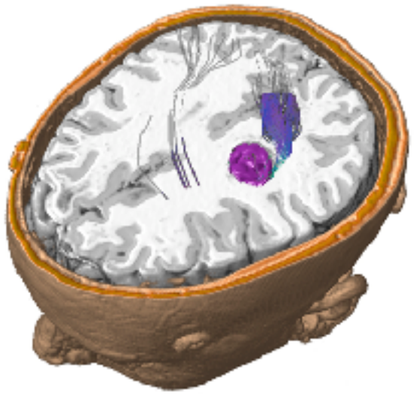
- example: geographic map
  - simple zoom, only viewport changes, shapes preserved

[Zoom to Bounding Box] https://observablehq.com/@d3/zoom-to-bounding-box

Navigate: Reducing attributes

- continuation of camera metaphor
  - slice
    - show only items matching specific value for given attribute: slicing plane
    - axis aligned, or arbitrary alignment
  - cut
    - show only items on far slide of plane from camera
  - project
    - change mathematics of image creation
      - orthographic
      - perspective
      - many others: Mercator, cabinet, ...

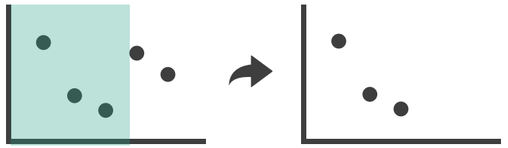




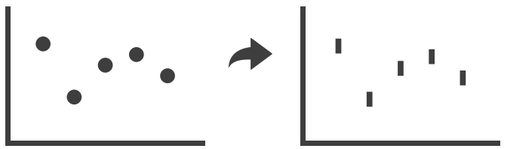
➔ Attribute Reduction

➔ *Slice*

➔ *Cut*



➔ *Project*



[Interactive Visualization of Multimodal Volume Data for Neurosurgical Tumor Treatment. Rieder, Ritter, Raspe, and Peitgen. Computer Graphics Forum (Proc. EuroVis 2008) 27:3 (2008), 1055–1062.]

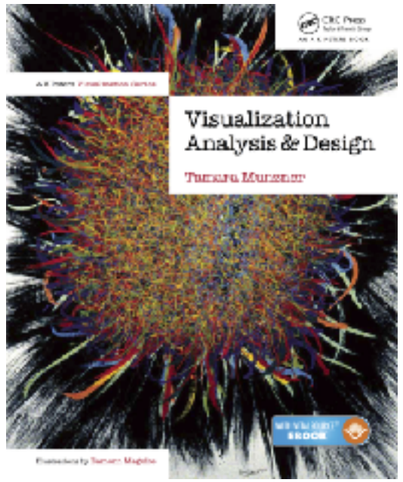
Interaction benefits

- interaction pros
  - major advantage of computer-based vs paper-based visualization
  - flexible, powerful, intuitive
    - exploratory data analysis: change as you go during analysis process
    - fluid task switching: different visual encodings support different tasks
  - animated transitions provide excellent support
    - empirical evidence that animated transitions help people stay oriented

Interaction limitations

- interaction has a time cost
  - sometimes minor, sometimes significant
  - degenerates to human-powered search in worst case
- remembering previous state imposes cognitive load
- controls may take screen real estate
  - or invisible functionality may be difficult to discover (lack of affordances)
- users may not interact as planned by designer
  - NYTimes logs show ~90% don't interact beyond scrollytelling - Aisch, 2016

Visualization Analysis & Design



Interactive Views (Ch 11/12) II

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

How to handle complexity: 1 previous strategy + 2 more

➔ *Derive*

- derive new data to show within view
- change view over time
- facet across multiple views

Manipulate

➔ Change

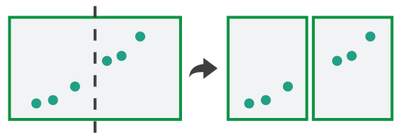
➔ Select

➔ Navigate

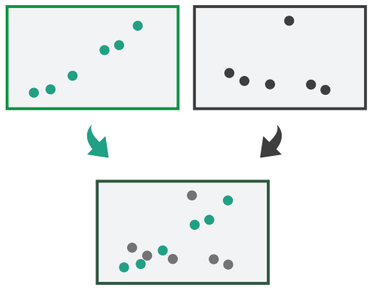
Facet

➔ Juxtapose

➔ Partition



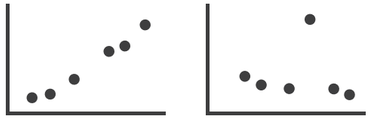
➔ Superimpose



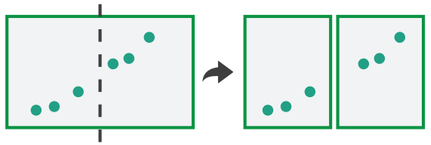
Multiple Views

Facet

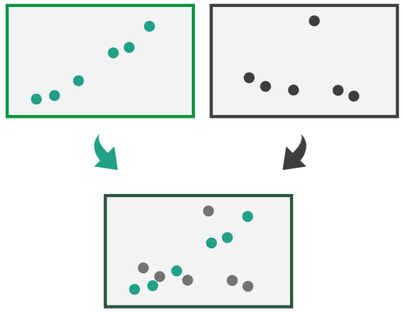
Juxtapose



Partition

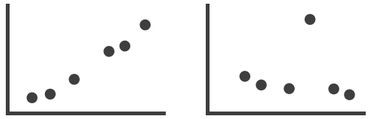


Superimpose

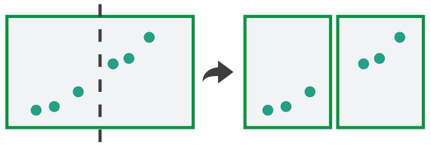


Facet

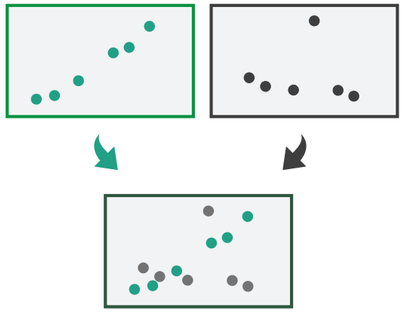
- Juxtapose



- Partition



- Superimpose



Juxtapose and coordinate views

- Share Encoding: Same/Different
  - *Linked Highlighting*

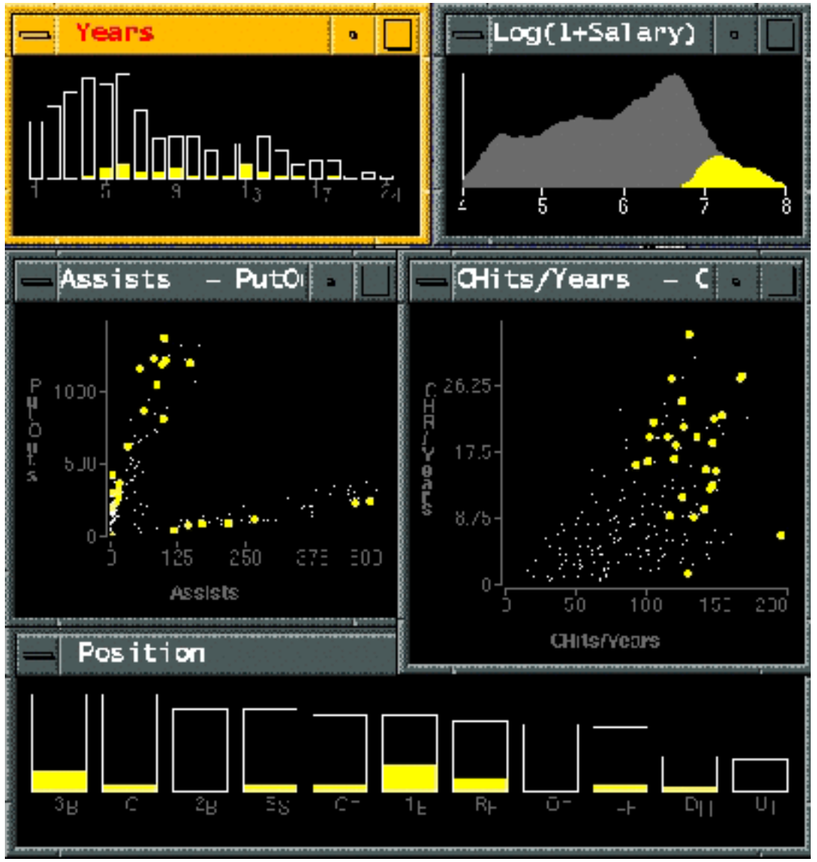
- Share Data: All/Subset/None

- Share Navigation

Idiom: **Linked highlighting**

System: **EDV**

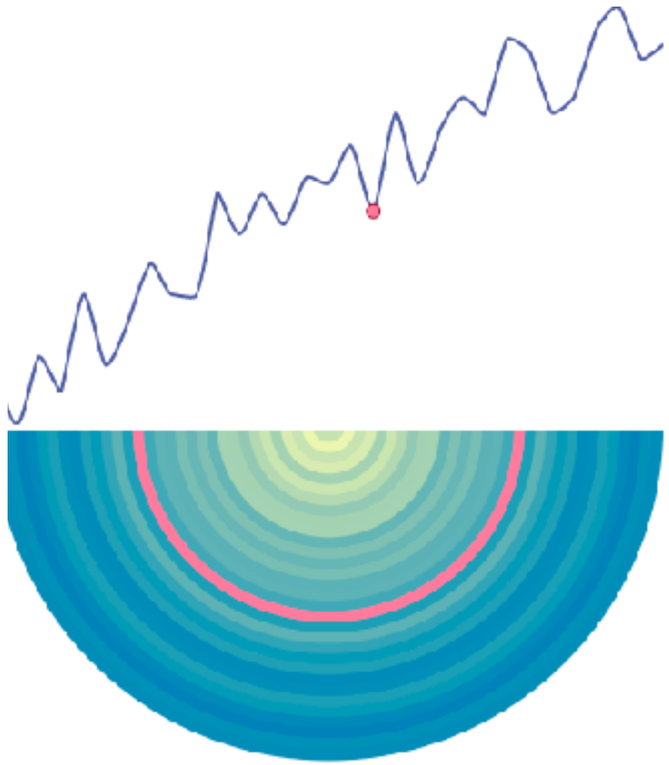
- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom
- encoding: different
  - *multiform*
- data: all shared
  - all **items** shared
  - different **attributes** across the views
- aka: brushing and linking



[Visual Exploration of Large Structured Datasets. Wills. Proc. New Techniques and Trends in Statistics (NTTS), pp. 237–246. IOS Press, 1995.]

Linked views: Directionality

- unidirectional vs bidirectional linking
  - bidirectional almost always better!



http://pbeshai.github.io/linked-highlighting-react-vega-redux/

https://medium.com/@pbesh/linked-highlighting-with-react-d3-js-and-reflux-16e9c0b2210b

Idiom: **Overview-detail views**

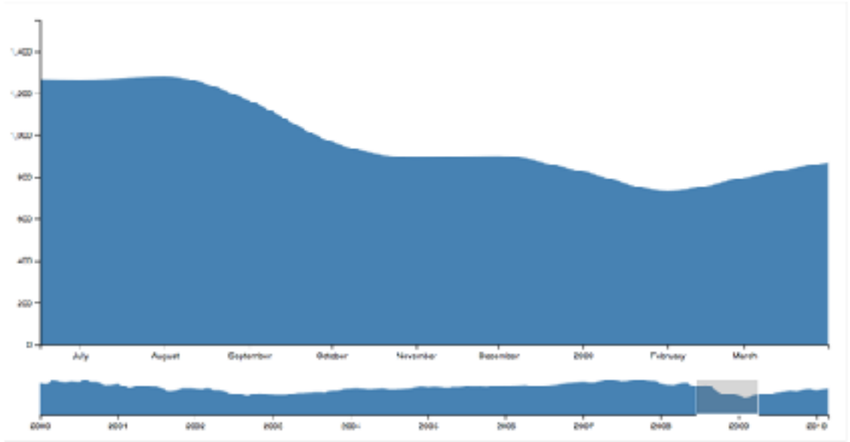
System: **Google Maps**

- encoding: same or different
  - ex: same (birds-eye map)
- data: subset shared
  - viewpoint differences: subset of data items
- navigation: shared
  - bidirectional linking
- other differences
  - (window size)

[A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. Cockburn, Karlson, and Bederson. ACM Computing Surveys 41:1 (2008), 1–31.]

Idiom: **Overview-detail navigation**

- encoding: same or different
- data: subset shared
- navigation: shared
  - unidirectional linking
  - select in small overview, change extent in large detail view



https://observablehq.com/@uwdata/interaction

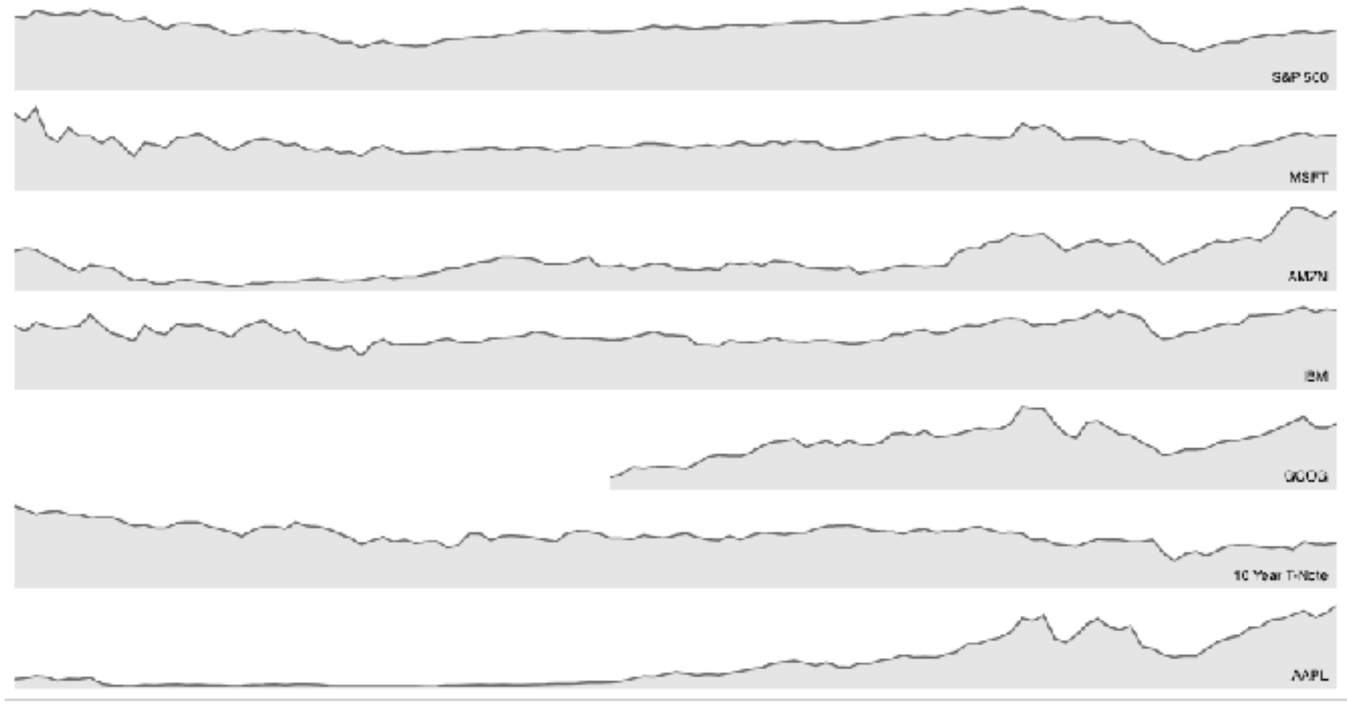
Idiom: **Tooltips**

- popup information for selection
  - hover or click
  - specific case of detail view:
    provide useful additional detail on demand
  - beware: does not support overview!
    - always consider if there's a way to visually encode directly to provide overview
    - "If you make a rollover or tooltip, assume nobody will see it. If it's important, make it explicit."
      - Gregor Aisch, NYTimes

[https://www.highcharts.com/demo/dynamic-master-detail]

Idiom: **Small multiples**

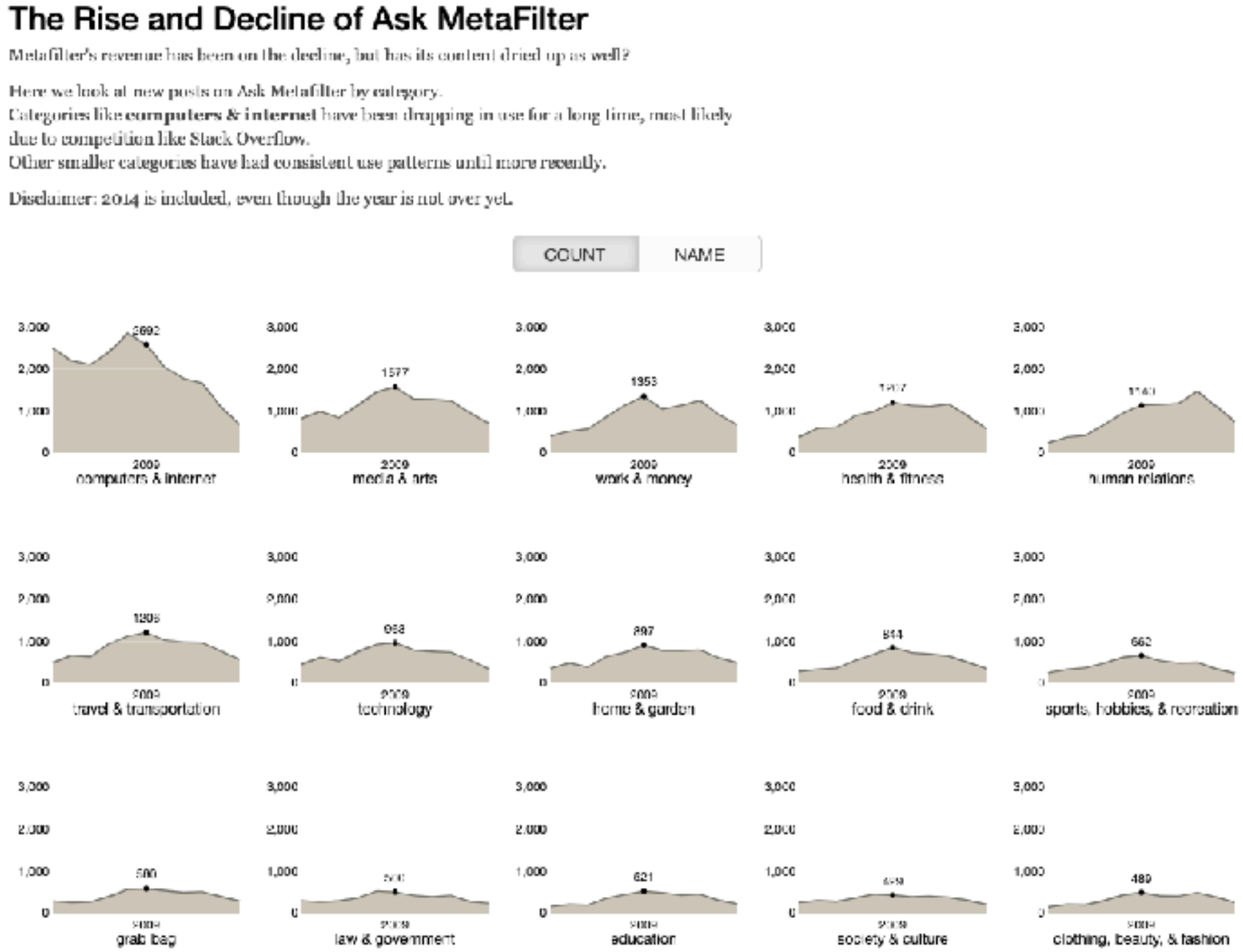
- encoding: same
  - ex: line charts
- data: none shared
  - different slices of dataset
    - items or attributes
    - ex: stock prices for different companies



[https://bl.ocks.org/mbostock/1157787]

Interactive small multiples

- linked highlighting: analogous item/attribute across views
  - same year highlighted across all charts if hover within any chart



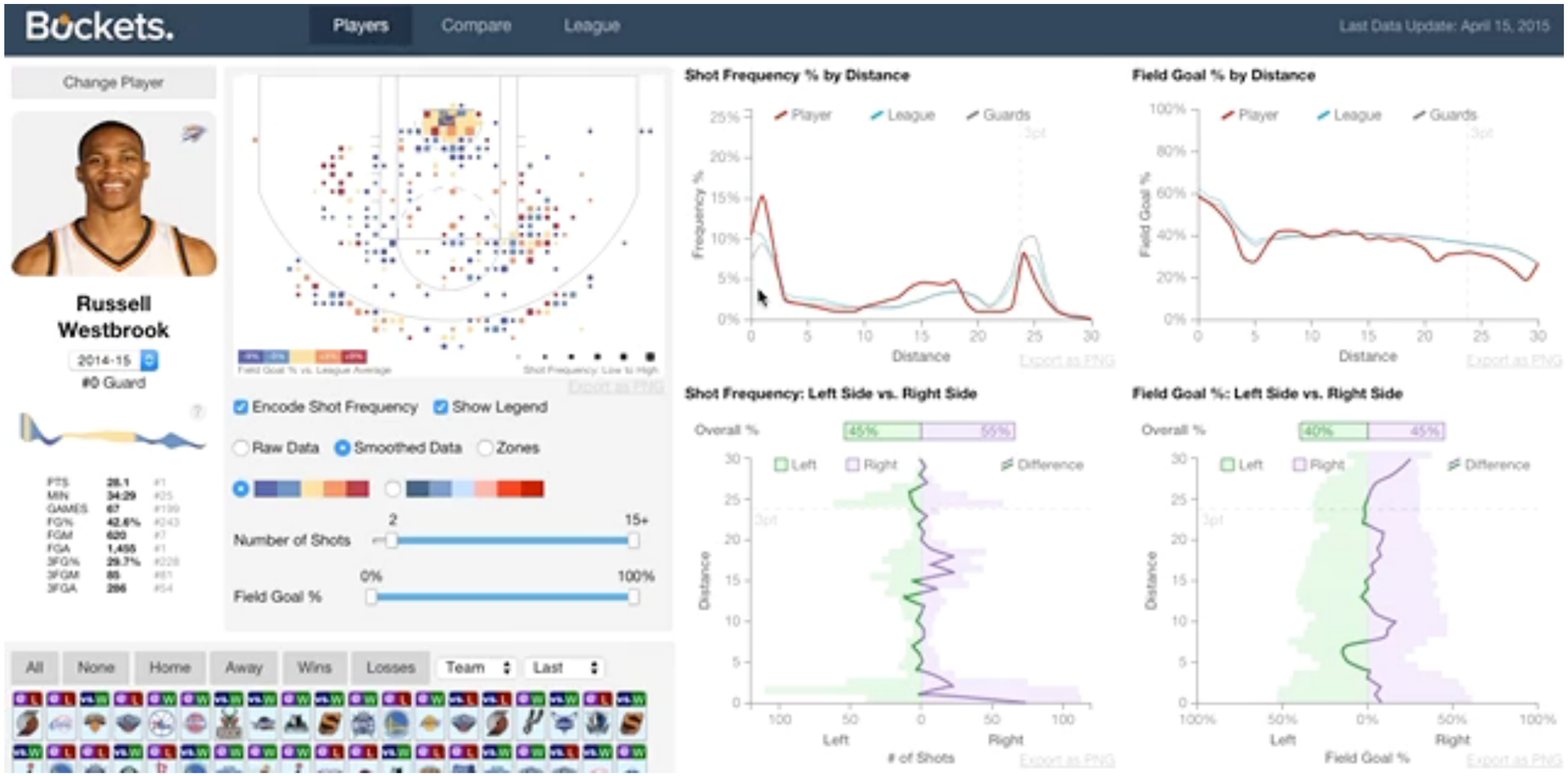
[https://bl.ocks.org/ColinEberhardt/3c780088c363d1515403f50a87a87121]

[https://blog.scottlogic.com/2017/04/05/interactive-responsive-small-multiples.html]

[http://projects.flowingdata.com/tut/linked_small_multiples_demo/]

Example: Combining many interaction idioms

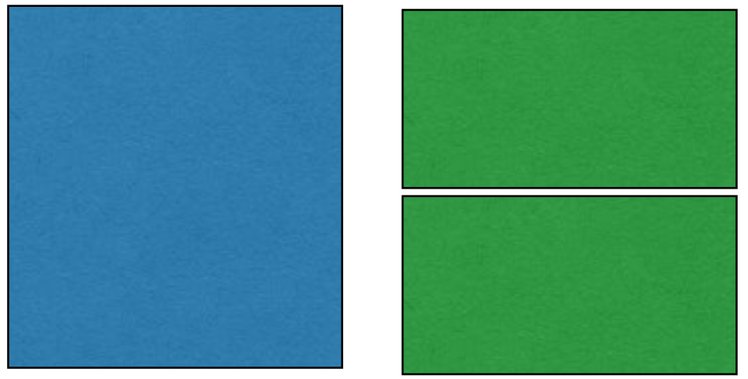
System: **Buckets**



- multiform
- multidirectional linked highlighting of small multiples
- tooltips

http://buckets.peterbeshai.com/

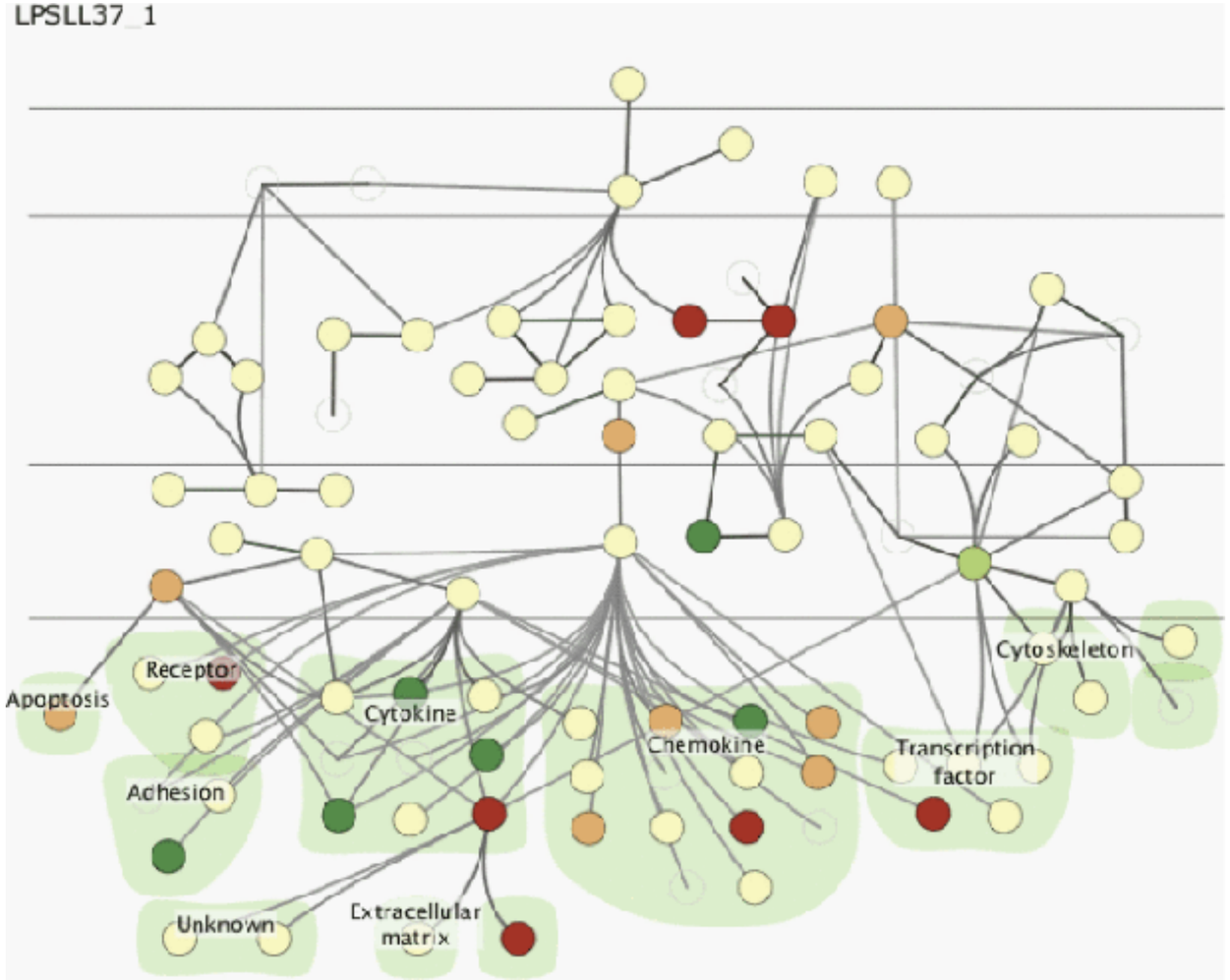
Juxtapose views: tradeoffs



- juxtapose costs
  - display area
    - 2 views side by side: each has only half the area of one view
- juxtapose benefits
  - cognitive load: eyes vs memory
    - lower cognitive load: move eyes between 2 views
    - higher cognitive load: compare single changing view to memory of previous state

Juxtapose vs animate

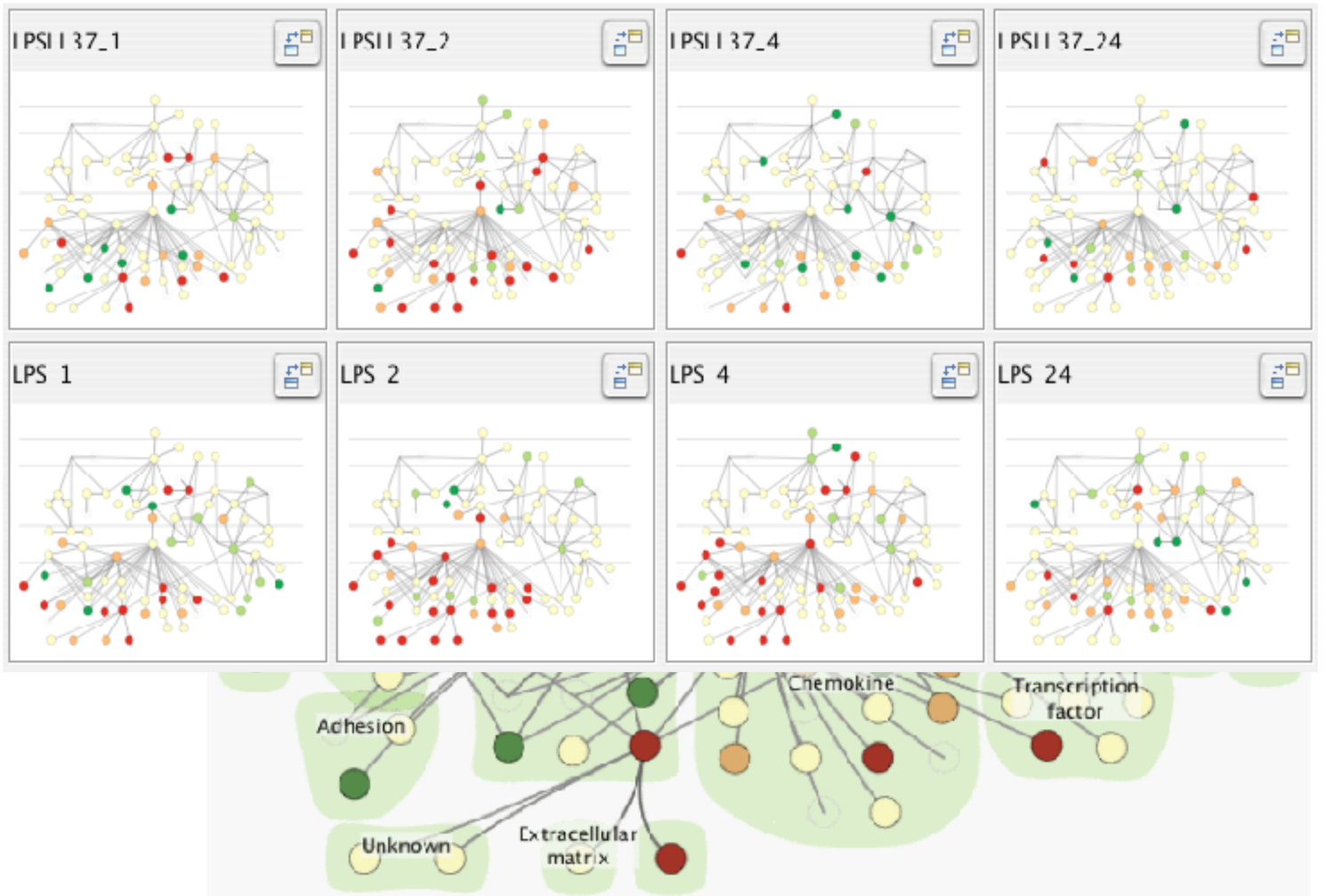
- animate: hard to follow if many scattered changes or many frames
  - vs easy special case: animated transitions



[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14:6 (2008), 1253–1260.]

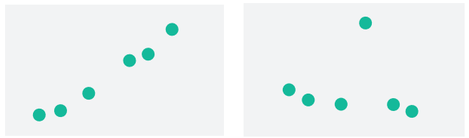
Juxtapose vs animate

- animate: hard to follow if many scattered changes or many frames
  - vs easy special case: animated transitions
- juxtapose: easier to compare across small multiples
  - different conditions (color), same gene (layout)



[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14:6 (2008), 1253–1260.]

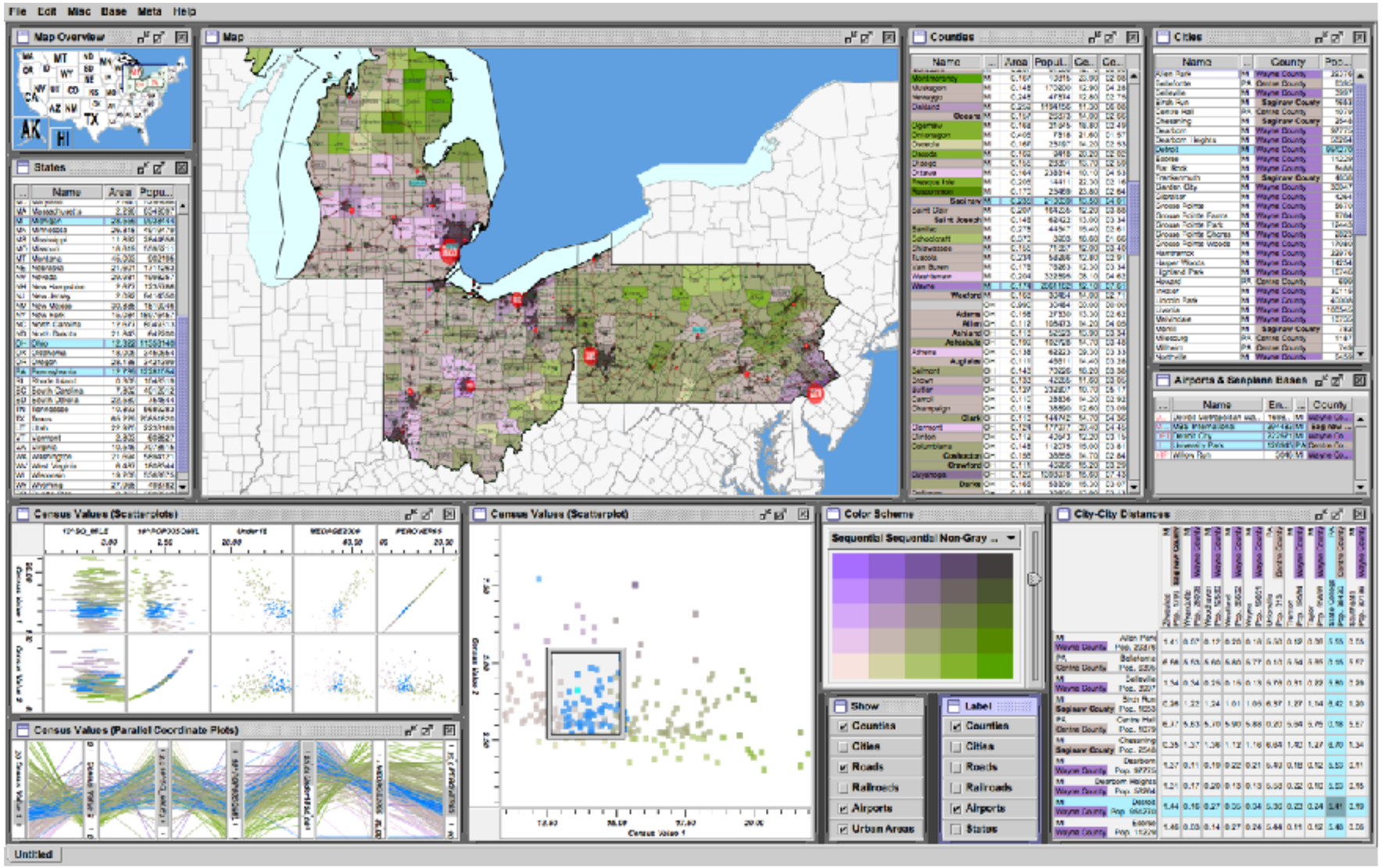
View coordination: Design choices

		Data		
		All	Subset	None
Encoding	Same	Redundant	 Overview/ Detail	 Small Multiples
	Different	 Multiform	 Multiform, Overview/ Detail	No Linkage

Idiom: **Reorderable lists**

System: **Improvise**

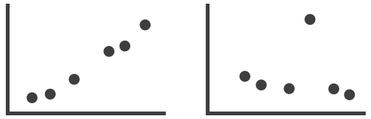
- list views
  - easy lookup
  - useful when linked to other views
- how many views is ok vs too complex?
  - open research question



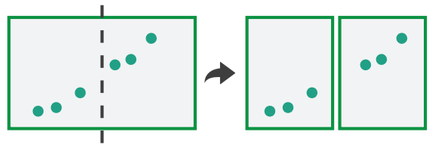
[Building Highly-Coordinated Visualizations In Improvise. Weaver. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 159–166, 2004.]

Facet

- Juxtapose



- Partition

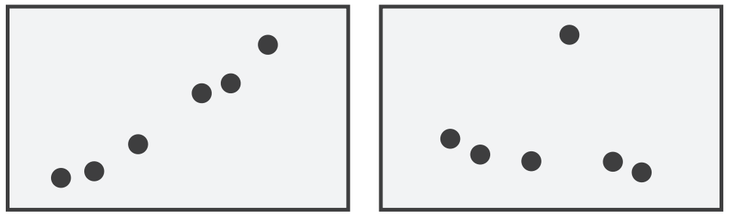


- Superimpose

Partition into views

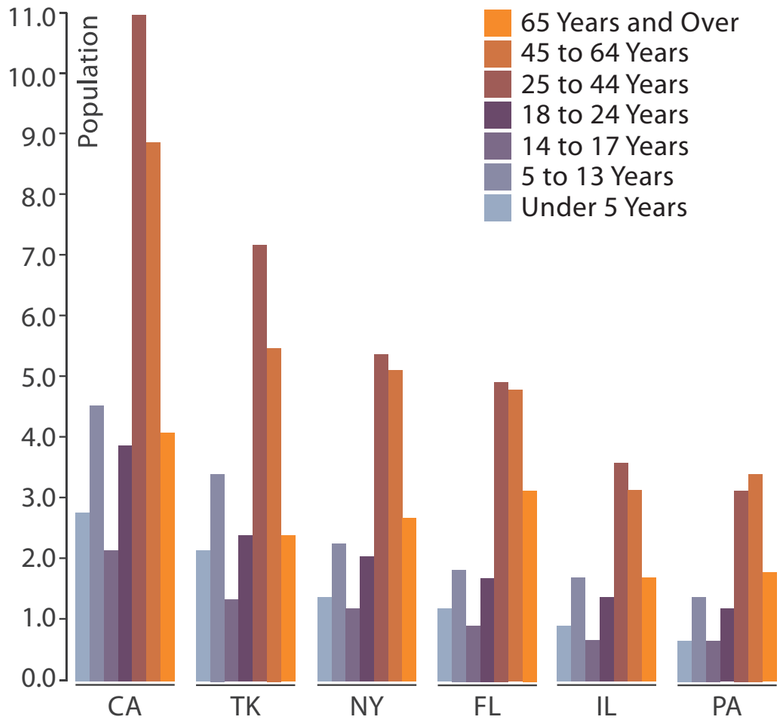
- how to divide data between views
  - split into regions by attributes
  - encodes association between items using spatial proximity
  - order of splits has major implications for what patterns are visible

➔ **Partition into Side-by-Side Views**



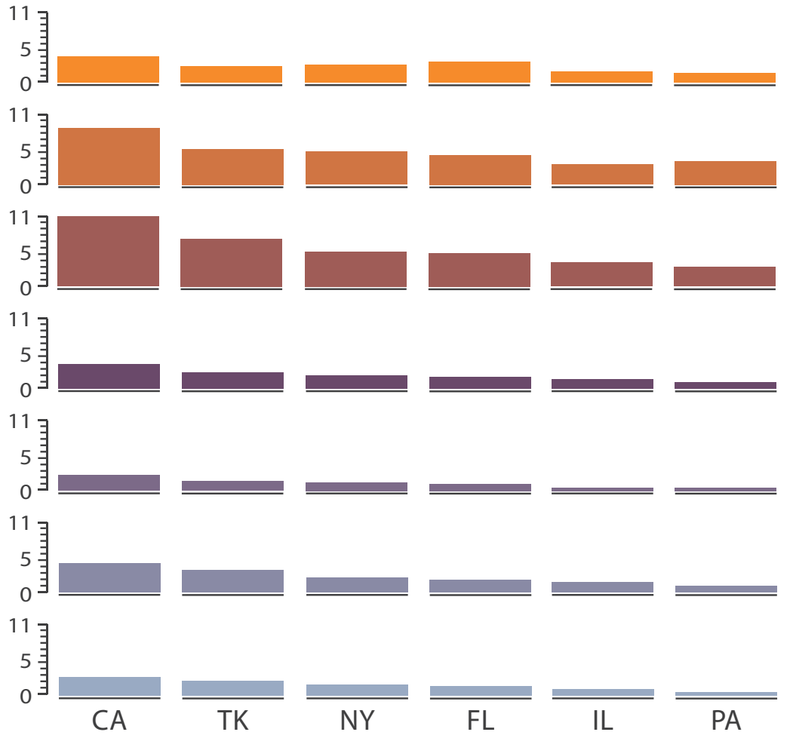
Partitioning: Grouped vs small-multiple bars

- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages



[https://observablehq.com/@d3/grouped-bar-chart]

- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states



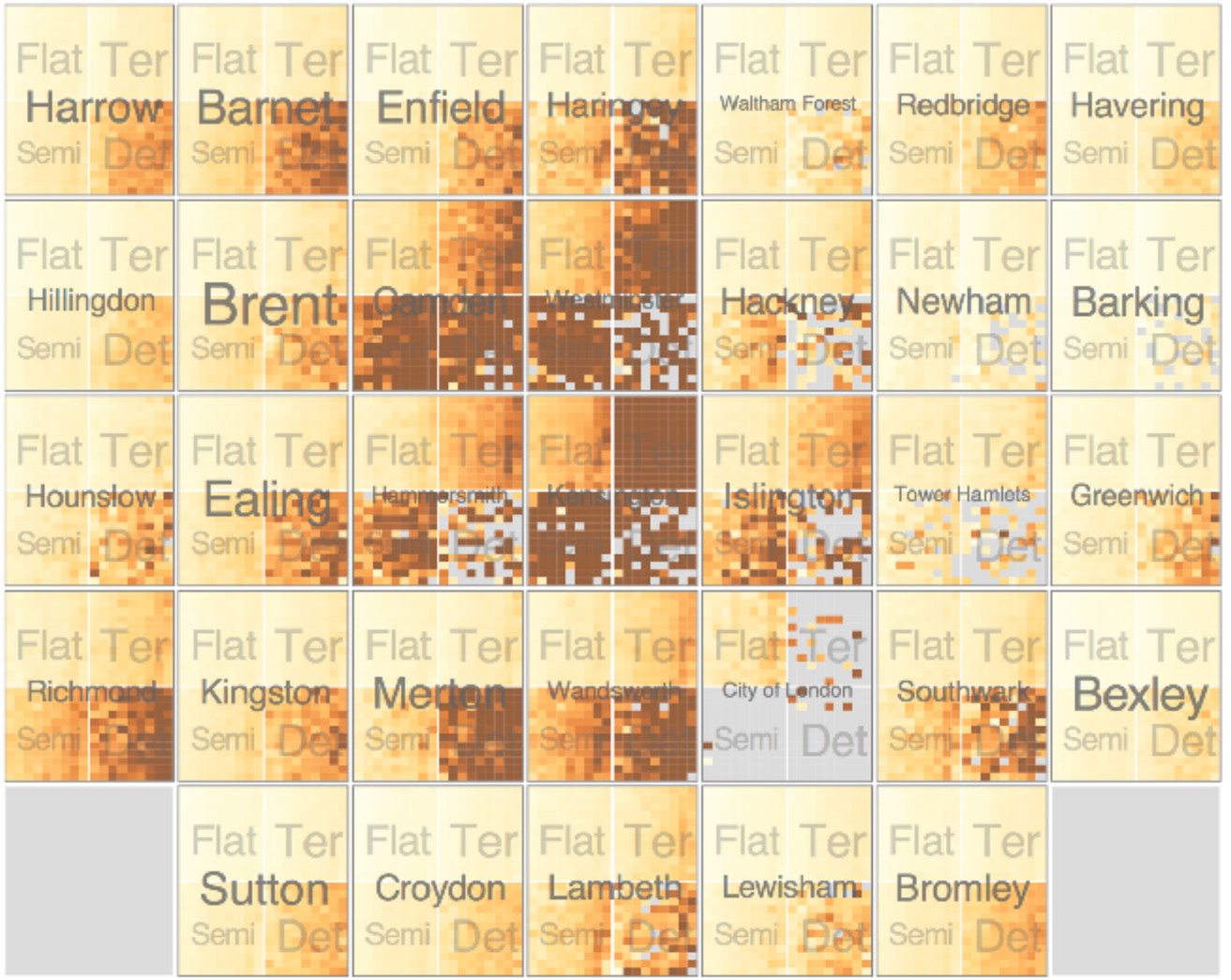
[https://bl.ocks.org/mbostock/4679202]

Partitioning: Recursive subdivision

System: **HIVE**

- split by neighborhood
- then by type
  - flat, terrace, semi-detached, detached
- then time
  - years as rows
  - months as columns
- color by price

- neighborhood patterns
  - where it's expensive
  - where you pay much more for detached type



[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

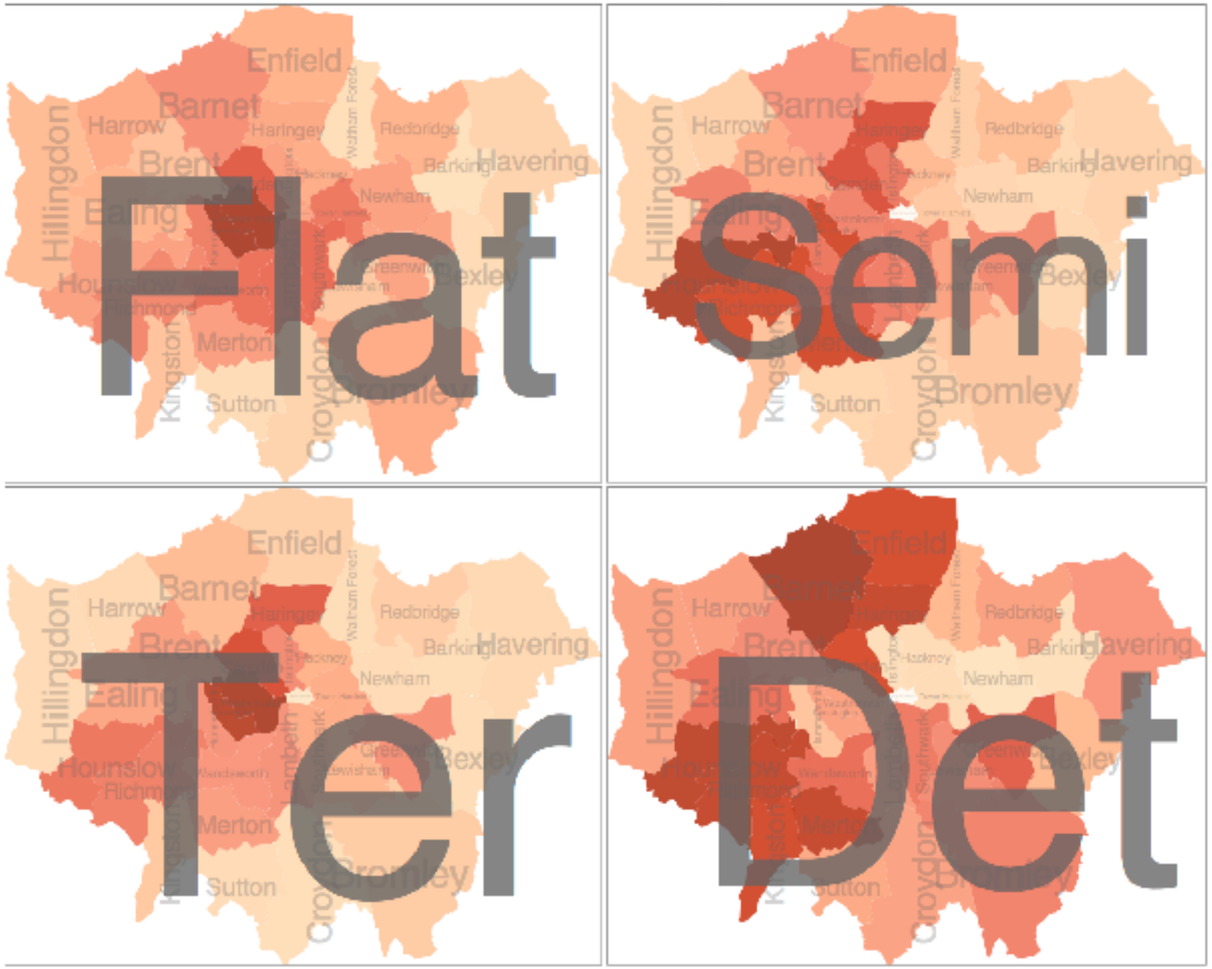
- switch order of splits
  - type then neighborhood
- switch color
  - by price variation
- type patterns
  - within specific type, which neighborhoods inconsistent

[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

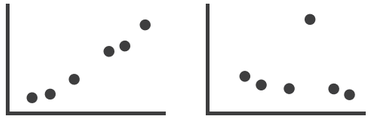
- different encoding for second-level regions
  - choropleth maps



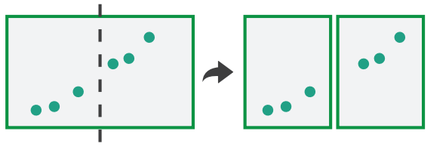
[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Facet

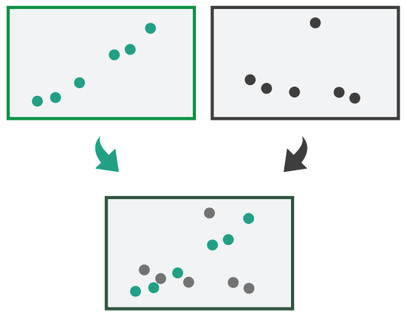
Juxtapose



Partition

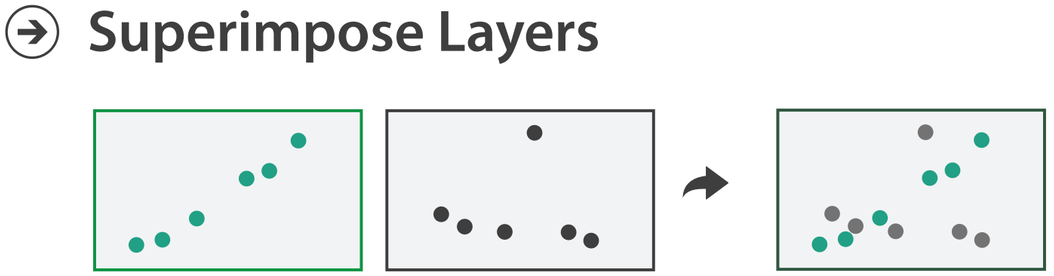


Superimpose



Superimpose layers

- layer: set of objects spread out over region
  - each set is visually distinguishable group
  - extent: whole view
- design choices
  - how many layers, how to distinguish?
    - encode with different, nonoverlapping channels
    - two layers achievable, three with careful design
  - small static set, or dynamic from many possible?



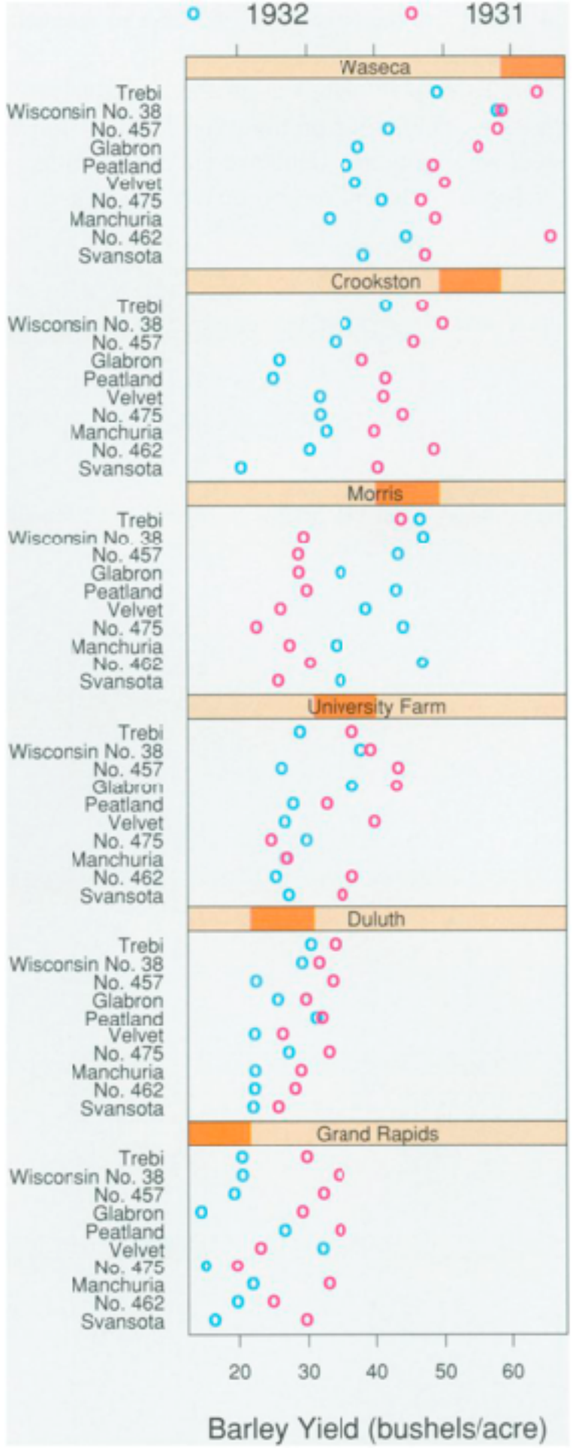
Static visual layering

- foreground layer: roads
  - hue, size distinguishing main from minor
  - high luminance contrast from background
- background layer: regions
  - desaturated colors for water, parks, land areas
- user can selectively focus attention

[Get it right in black and white. Stone. 2010. http://www.stonesc.com/wordpress/2010/03/get-it-right-in-black-and-white]

Idiom: **Trellis plots**

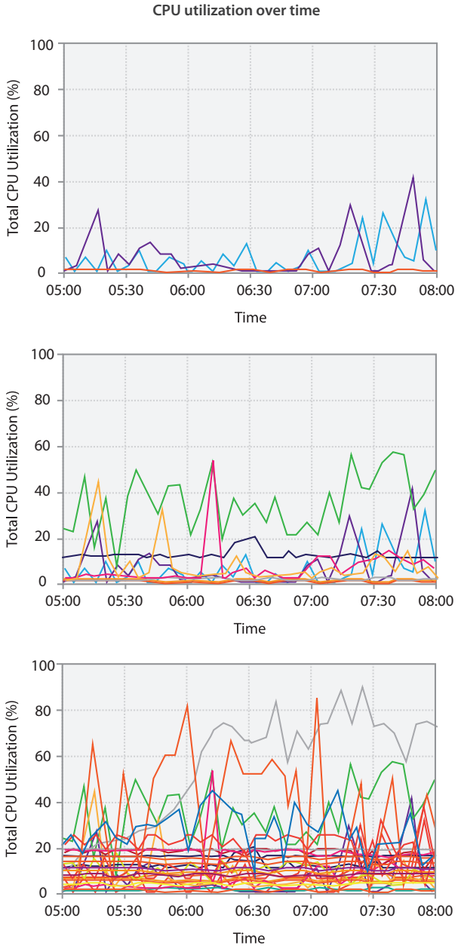
- superimpose within same frame
  - color code by year
- partitioning
  - split by site, rows are barley varieties
- main-effects ordering
  - derive value of median for group
  - order rows within view by variety median
  - order views themselves by site median

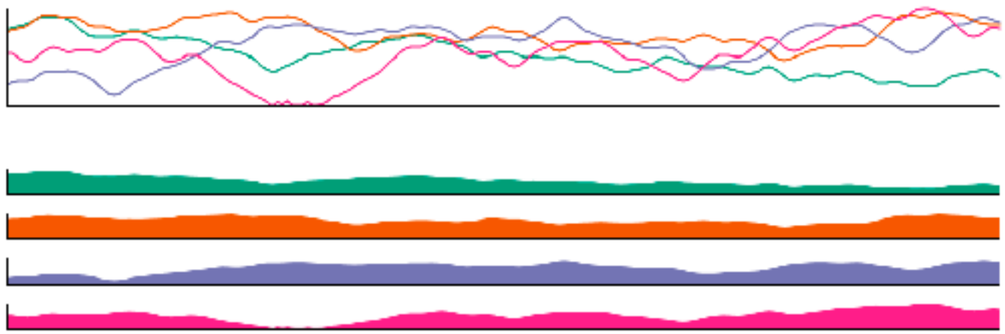


[The Visual Design and Control of Trellis Display. Becker, Cleveland, & Shyu. Journal of Computational and Graphical Statistics 5(2):123-155 1996.]

Superimposing limits (static)

- few layers, more lines
  - up to a few dozen lines
  - but not hundreds
- superimpose vs juxtapose: empirical study
  - same size: all multiples, vs single superimposed
    - superimposed: local tasks
    - juxtaposed: global tasks, esp. for many charts



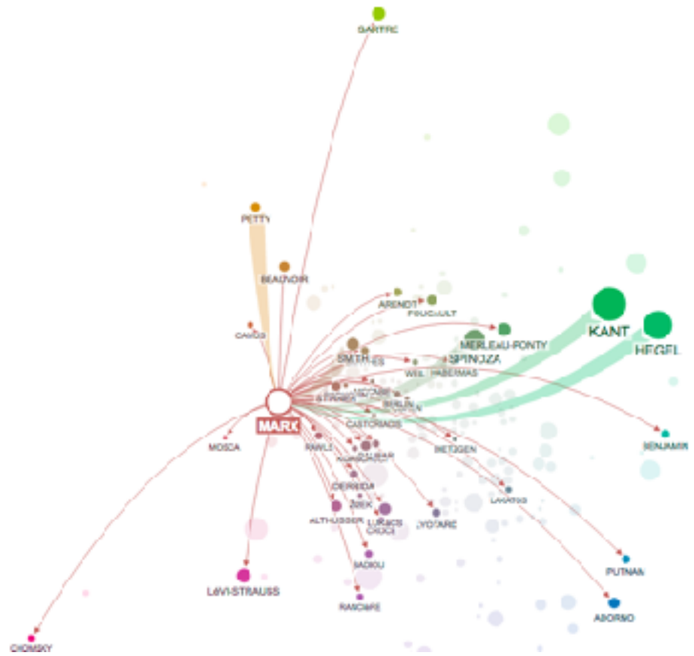


[Graphical Perception of Multiple Time Series. Javed, McDonnel, and Elmqvist. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE InfoVis 2010) 16:6 (2010), 927–934.]

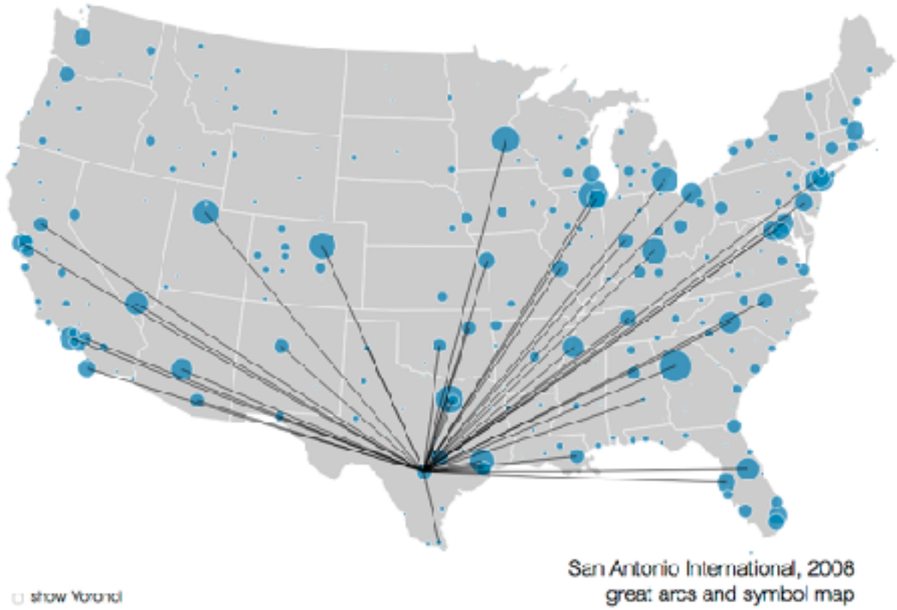
Dynamic visual layering

- interactive, based on selection
- one-hop neighbour highlighting

click (heavyweight)



hover (fast)



https://mariandoerk.de/edgemaps/demo/

http://mbostock.github.io/d3/talk/20111116/airports.html

How?

Encode

Arrange

- Express
- Separate
- Order
- Align
- Use

Map

from categorical and ordered attributes

- Color
  - *Hue*
  - *Saturation*
  - *Luminance*
- Size, Angle, Curvature, ...
- Shape
- Motion

Direction, Rate, Frequency, ...

Manipulate

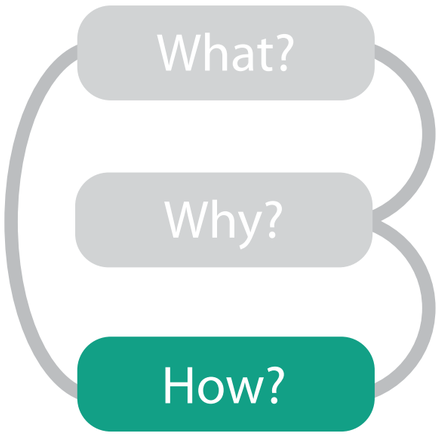
- Change
- Select
- Navigate

Facet

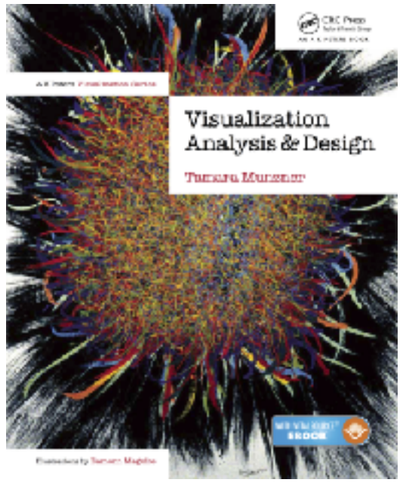
- Juxtapose
- Partition
- Superimpose

Reduce

- Filter
- Aggregate
- Embed



Visualization Analysis & Design



Reduce: Aggregation & Filtering (Ch 13)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

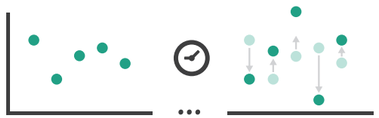
How to handle complexity: 3 previous strategies

→ *Derive*

- derive new data to show within view
- change view over time
- facet across multiple views

Manipulate

→ Change



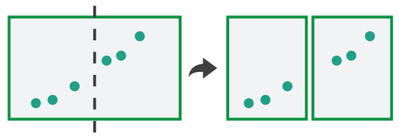
→ Select

→ Navigate

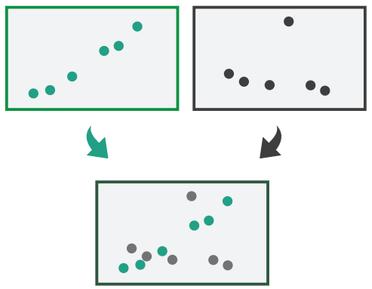
Facet

→ Juxtapose

→ Partition



→ Superimpose



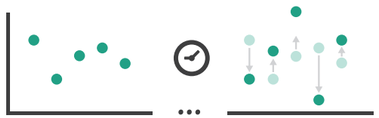
How to handle complexity: 3 previous strategies + 1 more

→ *Derive*

- derive new data to show within view
- change view over time
- facet across multiple views
- reduce items/attributes within single view

Manipulate

→ Change



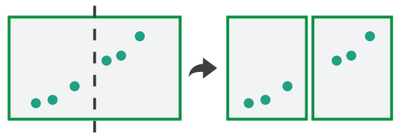
→ Select

→ Navigate

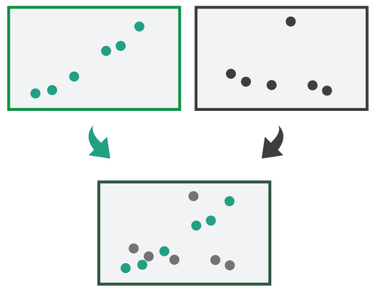
Facet

→ Juxtapose

→ Partition



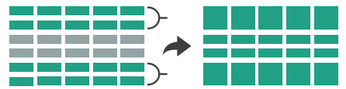
→ Superimpose



Reduce

→ Filter

→ Aggregate



→ Embed

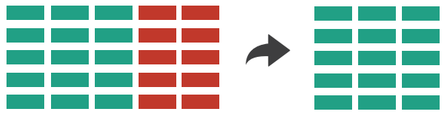
Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind

Reducing Items and Attributes

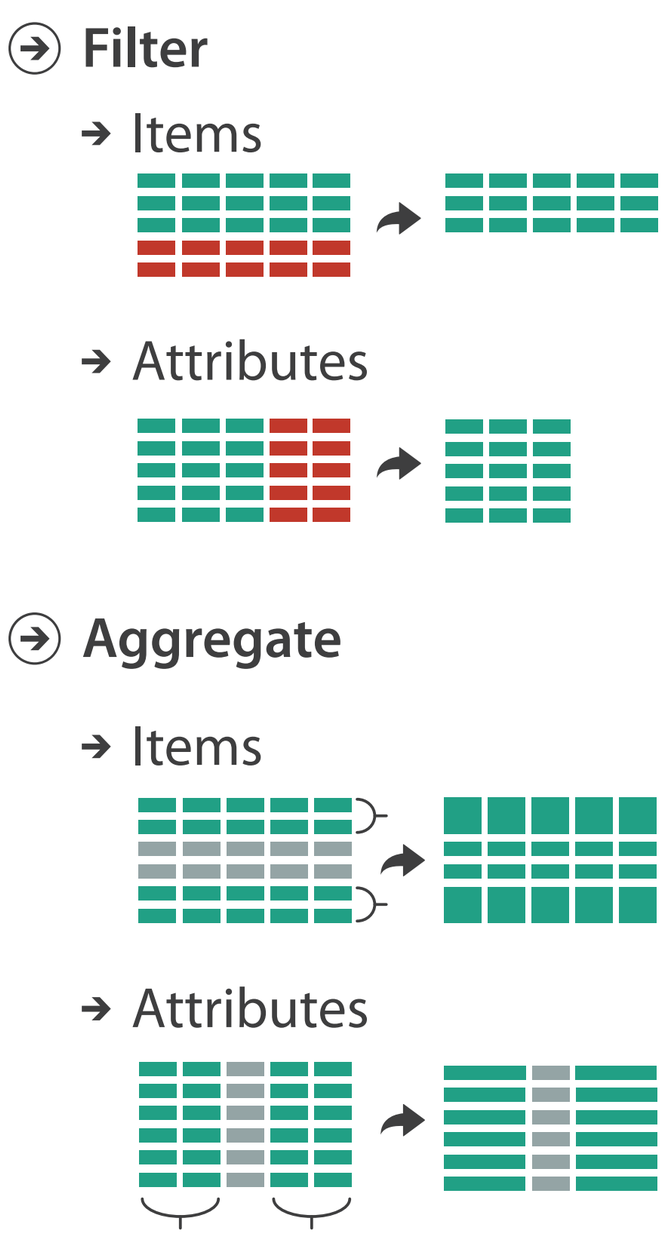
- **Filter**
  - Items

  - Attributes



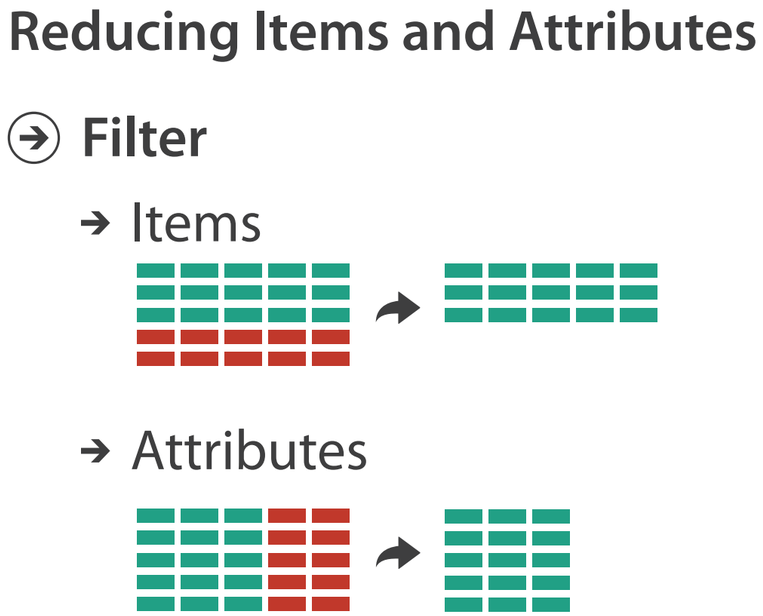
Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind
- aggregation
  - pro: inform about whole set
  - con: difficult to avoid losing signal
- not mutually exclusive
  - combine filter, aggregate
  - combine reduce, change, facet



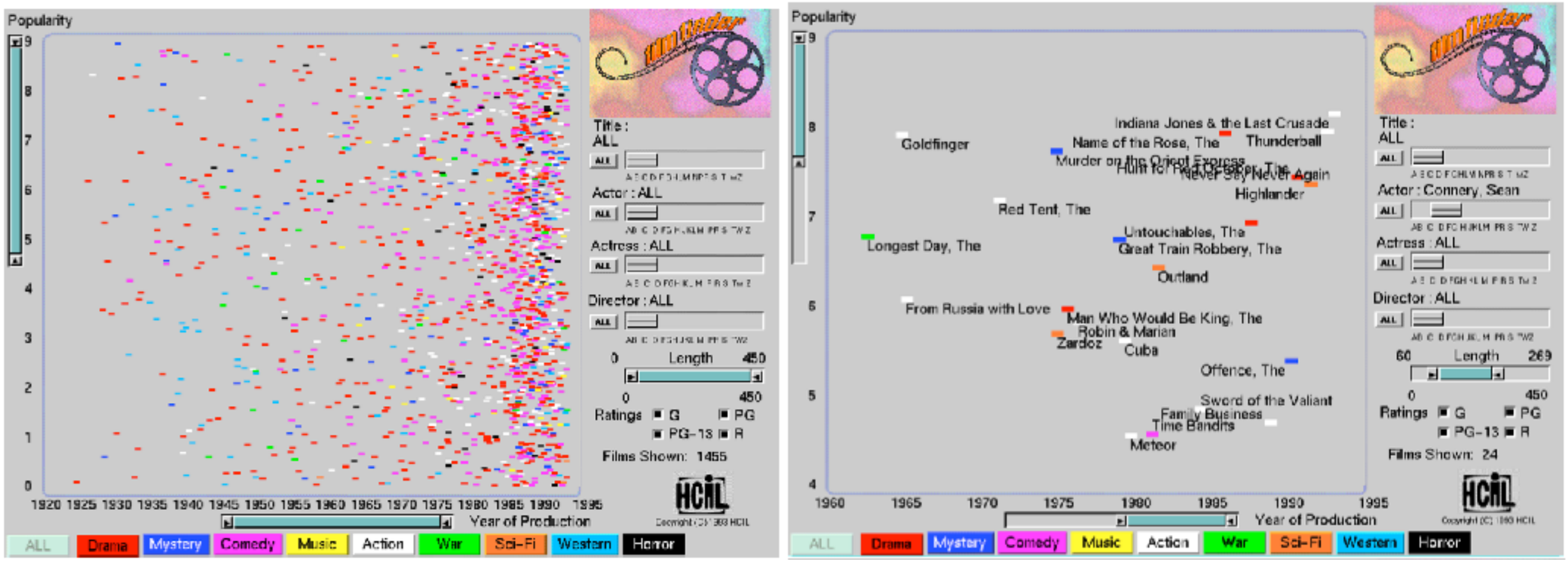
Filter

- eliminate some elements
  - either items or attributes
- according to what?
  - any possible function that partitions dataset into two sets
    - attribute values bigger/smaller than x
    - noise/signal
- filters vs queries
  - query: start with nothing, add in elements
  - filters: start with everything, remove elements
  - best approach depends on dataset size



Idiom: **FilmFinder**

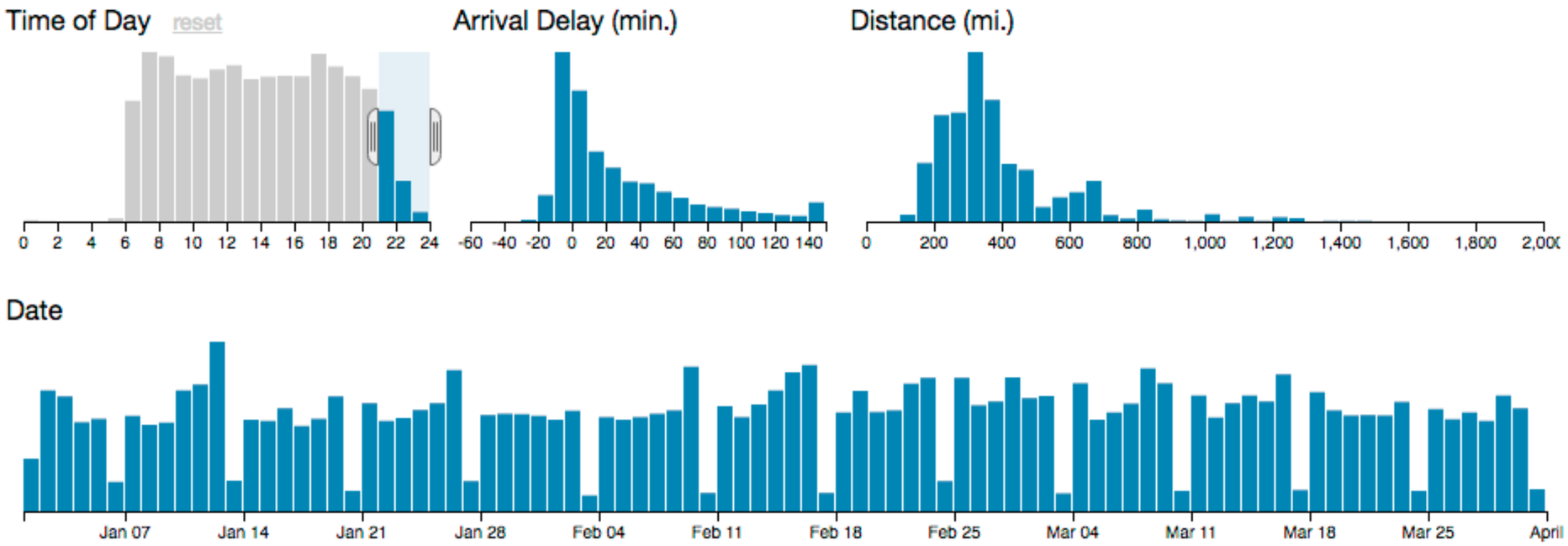
- dynamic queries/filters for items
  - tightly coupled interaction and visual encoding idioms, so user can immediately see results of action



[Ahlberg & Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. CHI 1994.]

Idiom: **cross filtering** System: **Crossfilter**

- item filtering
- coordinated views/controls combined
  - all scented histogram bisliders update when any ranges change

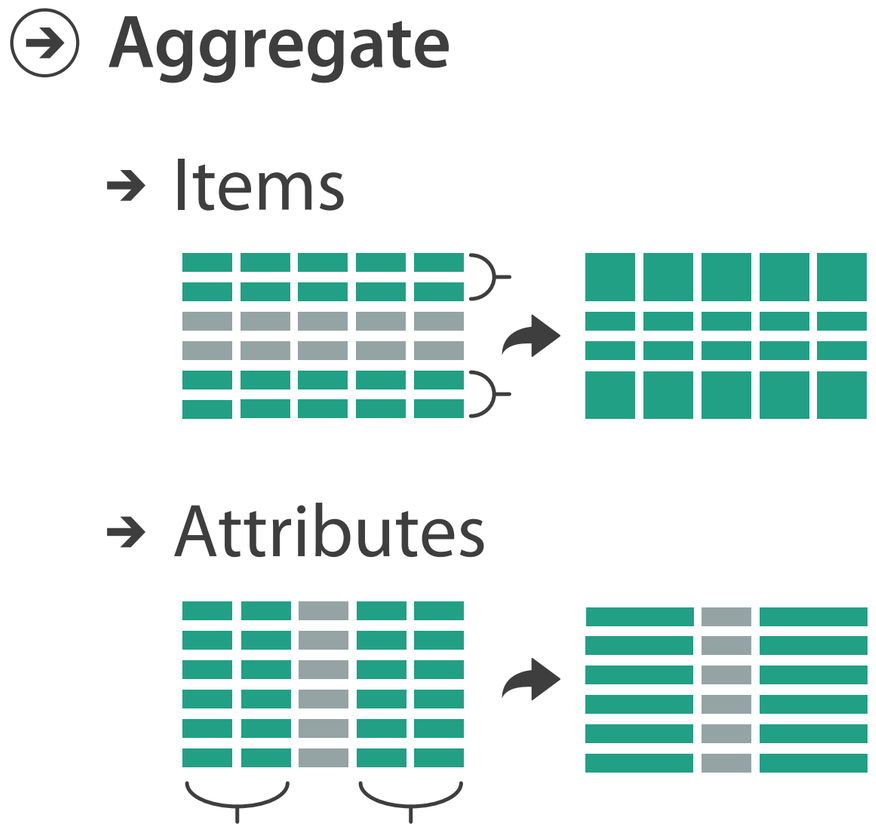


http://square.github.io/crossfilter/

https://observablehq.com/@uwdata/interaction

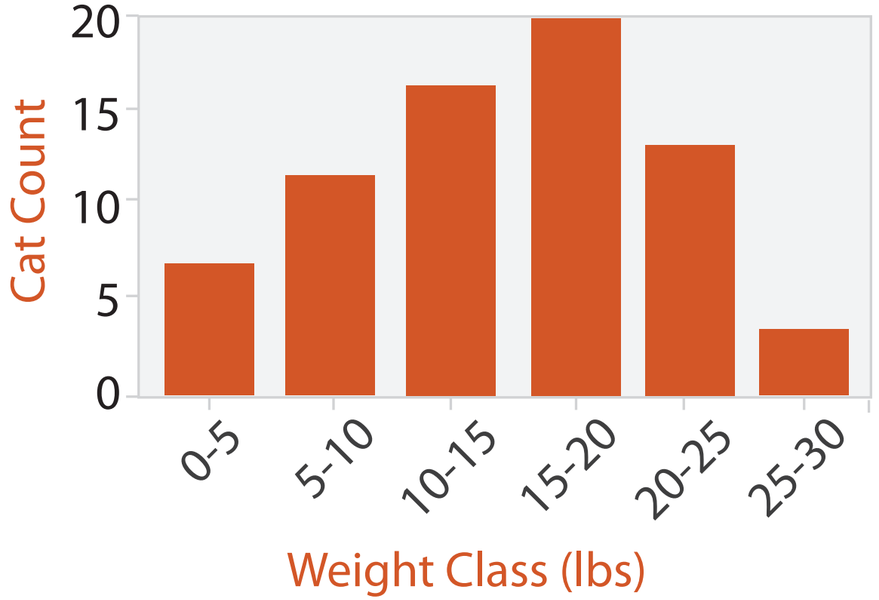
Aggregate

- a group of elements is represented by a smaller number of derived elements



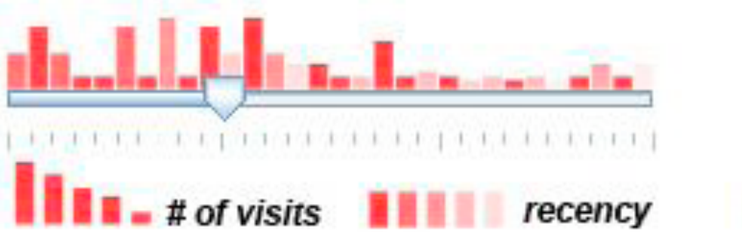
Idiom: **histogram**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly



Idiom: **scented widgets**

- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider

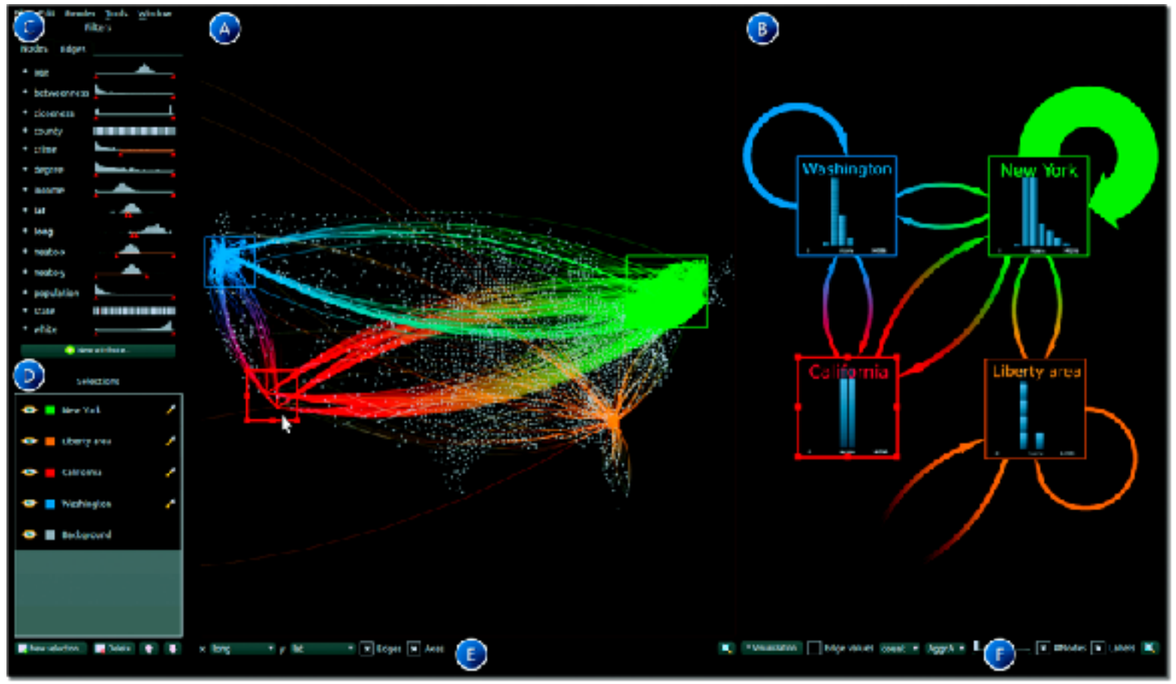


[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]

Idiom: **scented widgets**

- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider

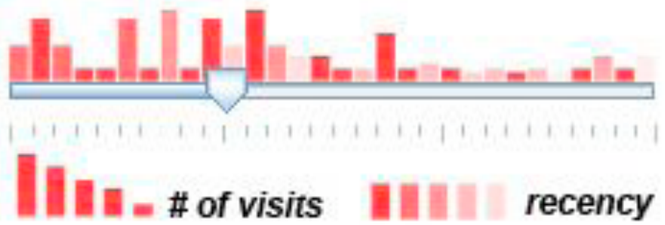
[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]



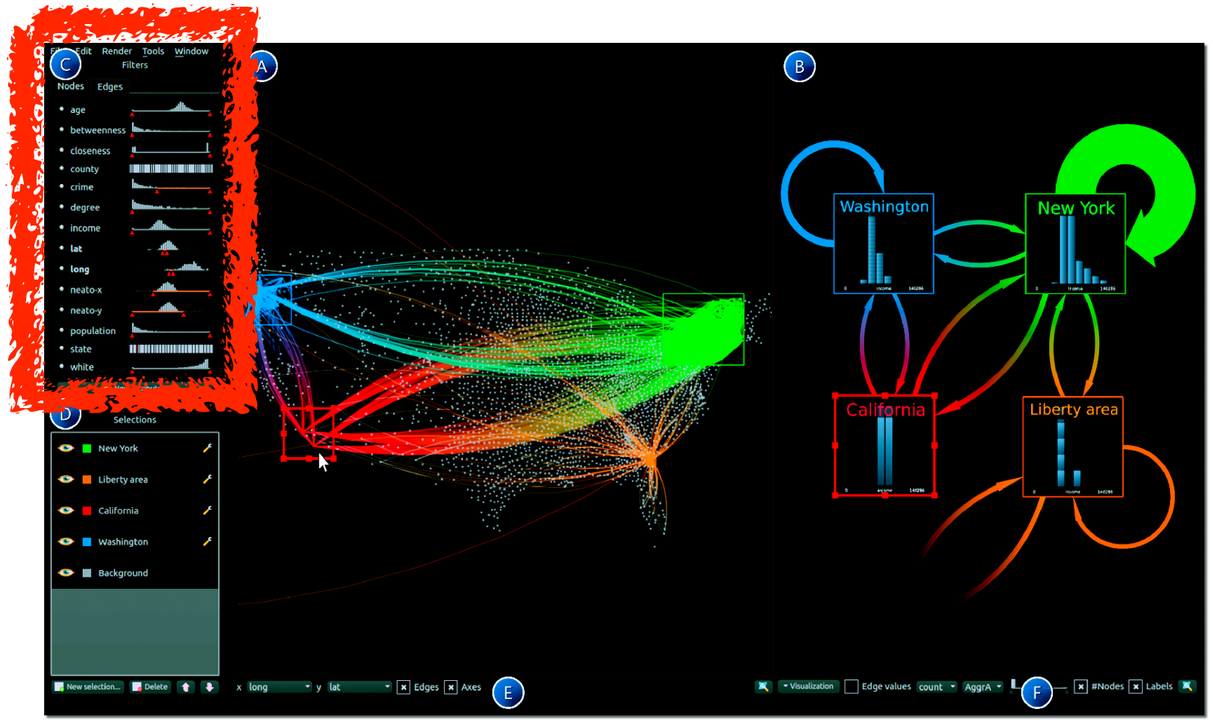
[Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. van den Elzen, van Wijk, IEEE TVCG 20(12): 2014 (Proc. InfoVis 2014).]

Idiom: **scented widgets**

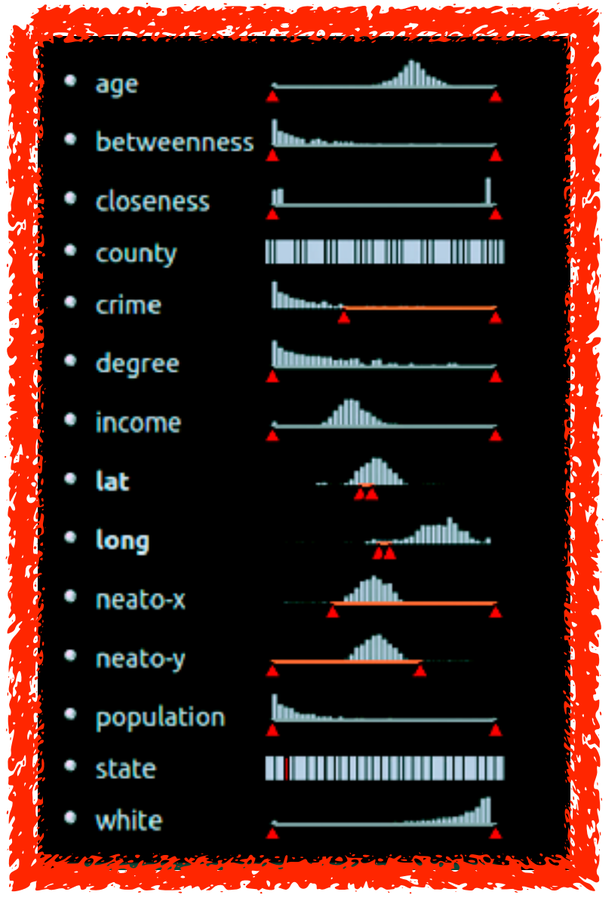
- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider



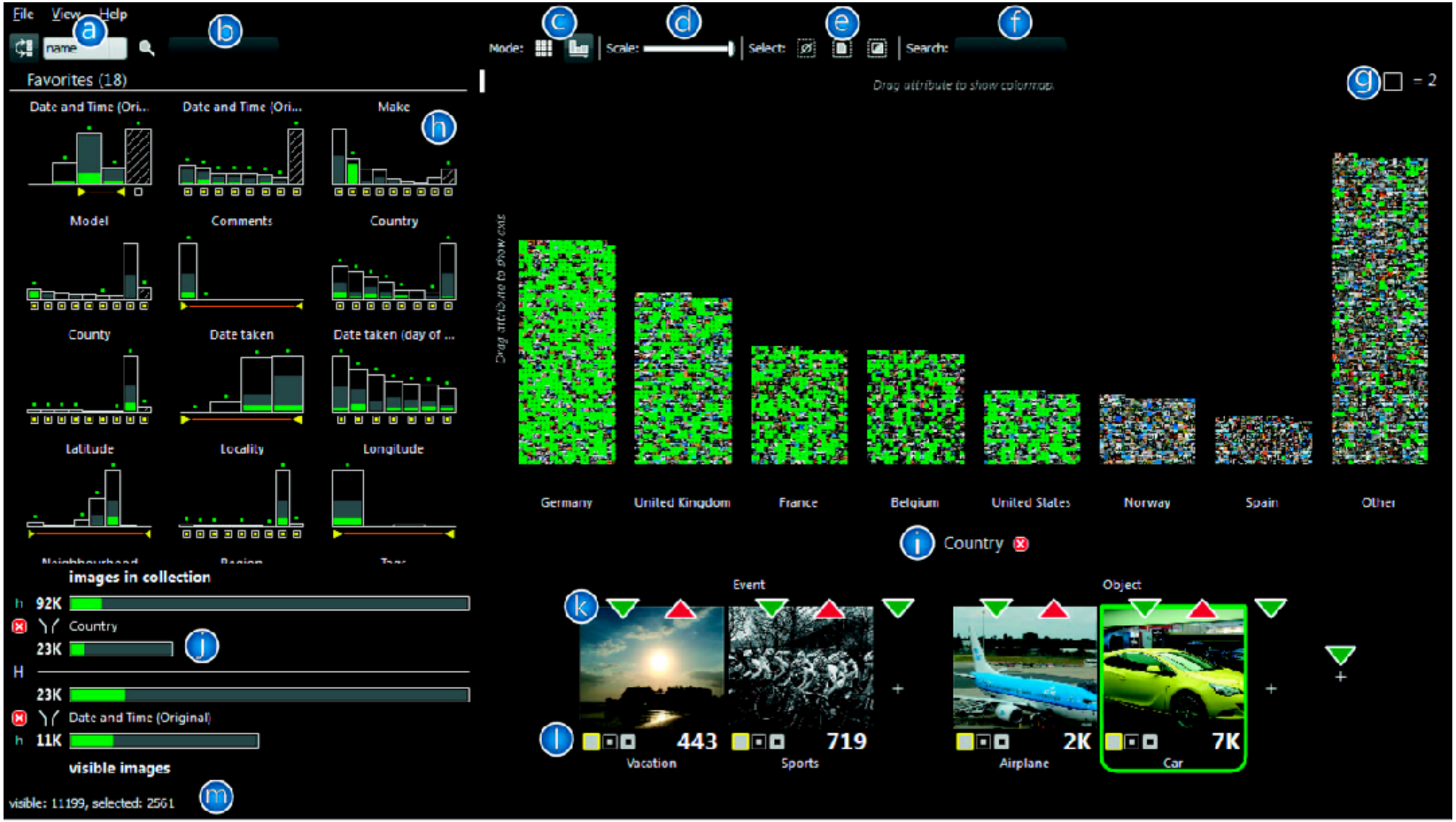
[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]



[Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. van den Elzen, van Wijk, IEEE TVCG 20(12): 2014 (Proc. InfoVis 2014).]



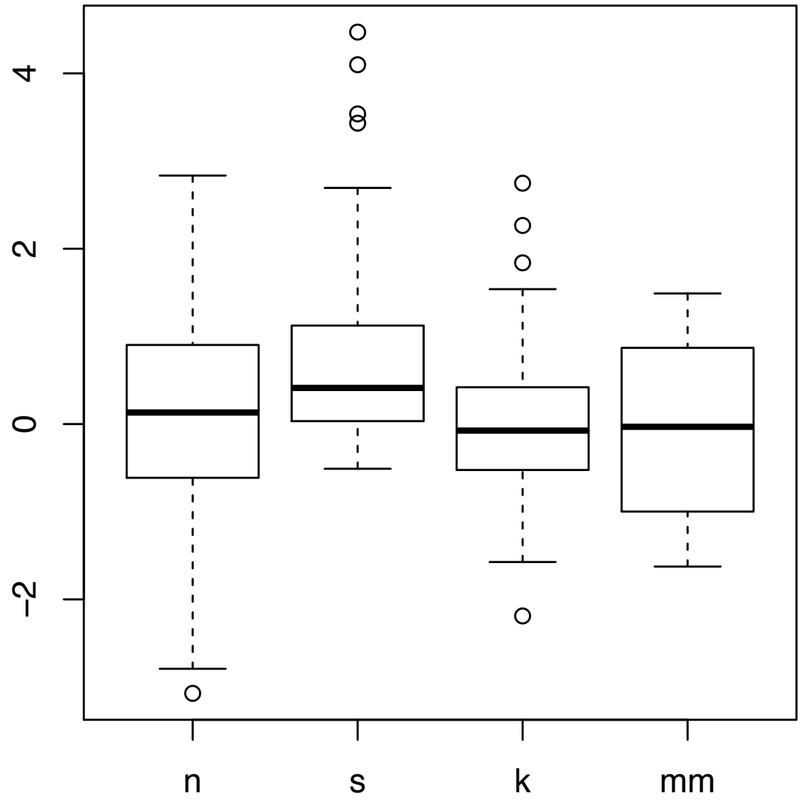
Scented histogram bisliders: detailed



[ICLIC: Interactive categorization of large image collections. van der Corput and van Wijk. Proc. PacificVis 2016.]

Idiom: **boxplot**

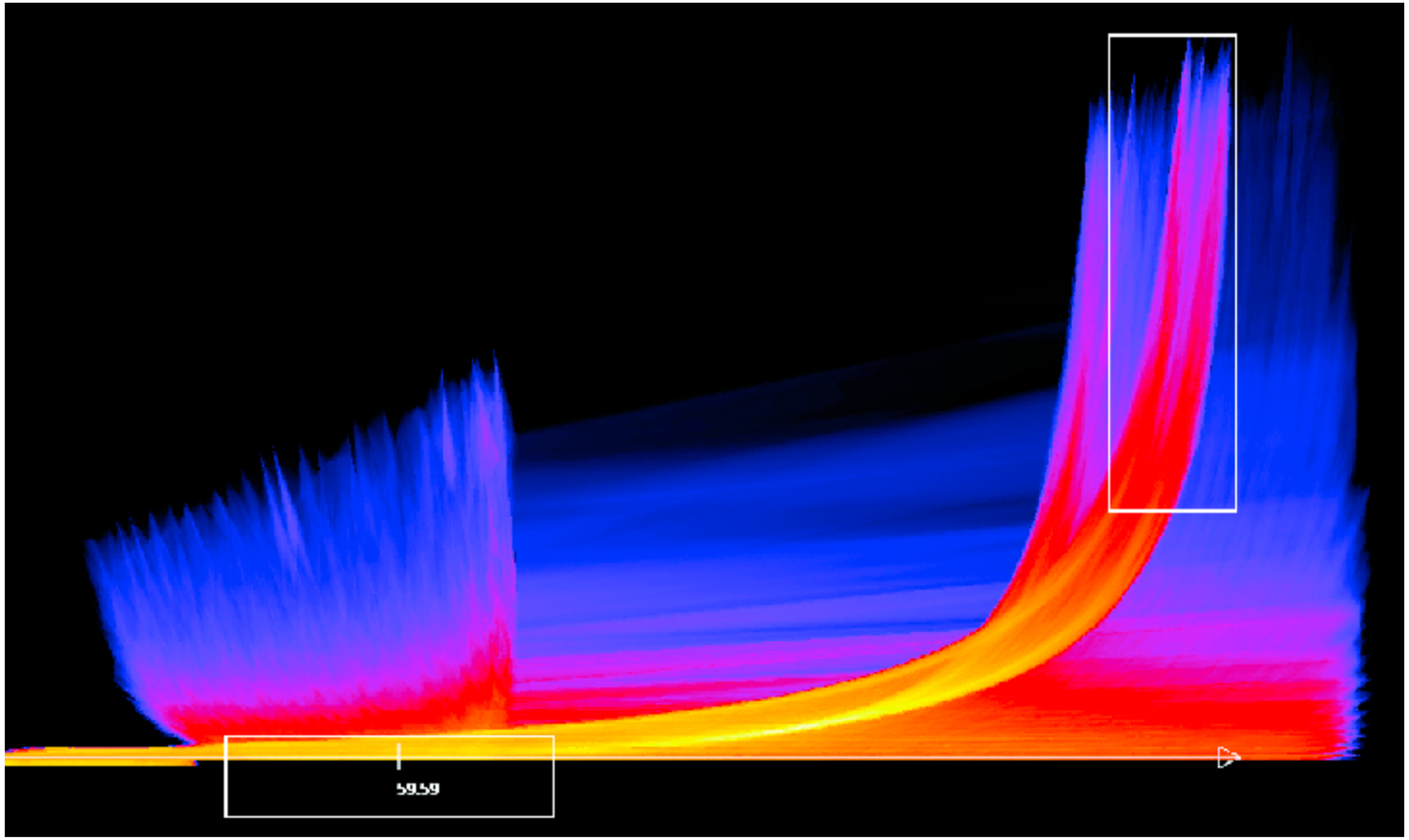
- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown
- scalability
  - unlimited number of items!



[40 years of boxplots. Wickham and Stryjewski. 2012]

Idiom: **Continuous scatterplot**

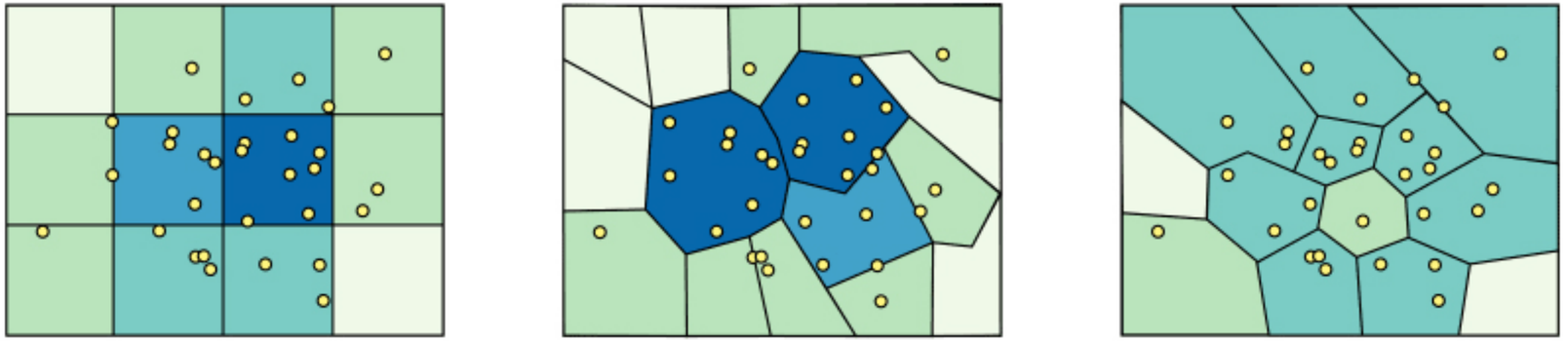
- static item aggregation
- data: table
- derived data: table
  - key attribs x,y for pixels
  - quant attrib: overplot density
- dense space-filling 2D matrix
- color:
  sequential categorical hue +
  ordered luminance colormap
- scalability
  - no limits on overplotting:
    millions of items



[Continuous Scatterplots. Bachthaler and Weiskopf. IEEE TVCG (Proc. Vis 08) 14:6 (2008), 1428–1435. 2008.]

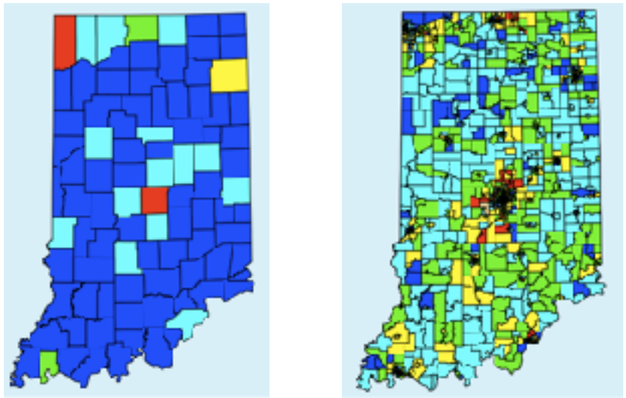
Spatial aggregation

- MAUP: Modifiable Areal Unit Problem
  - changing boundaries of cartographic regions can yield dramatically different results
  - zone effects



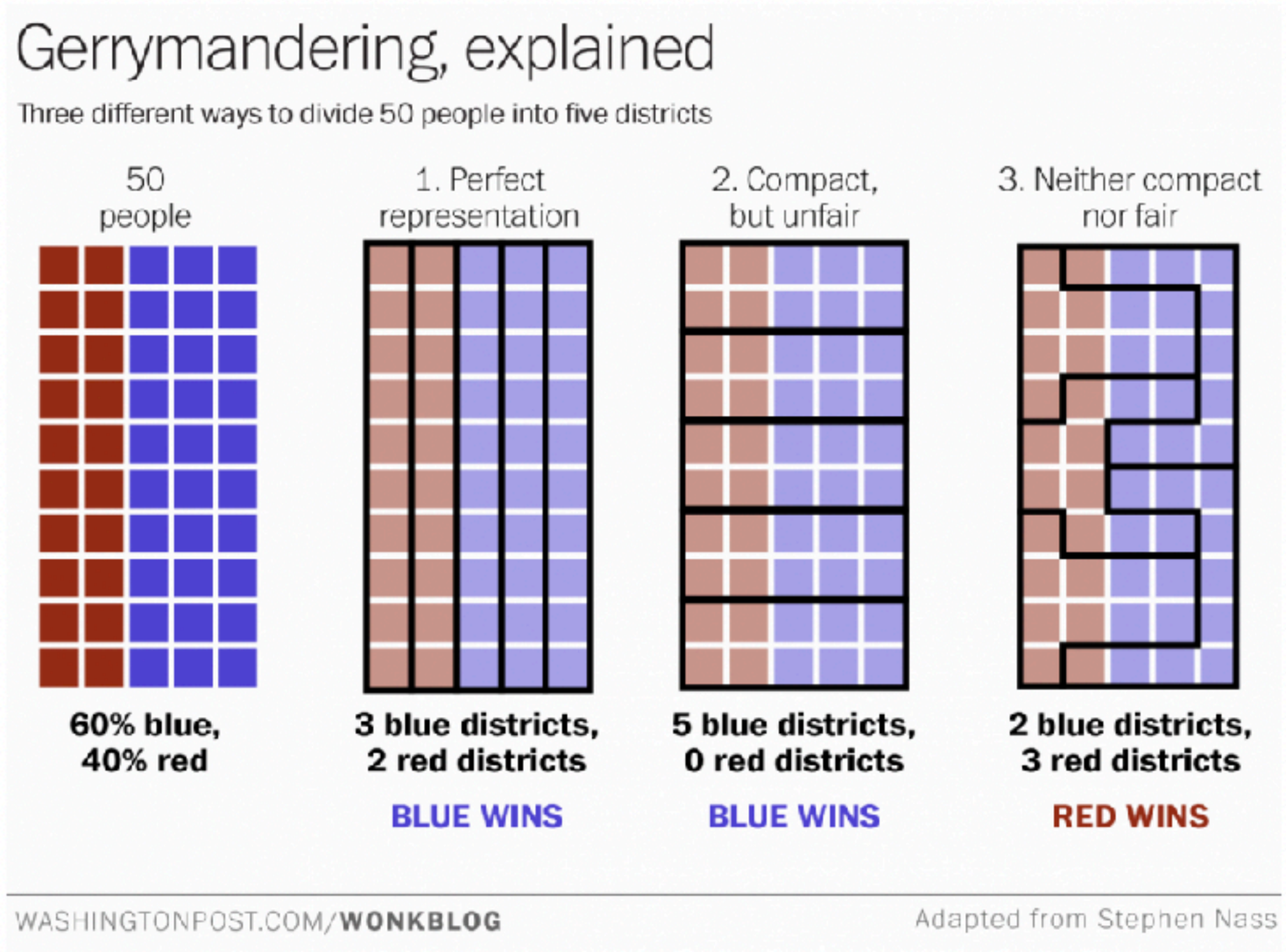
[http://www.e-education.psu/edu/geog486/l4_p7.html, Fig 4.cg.6]

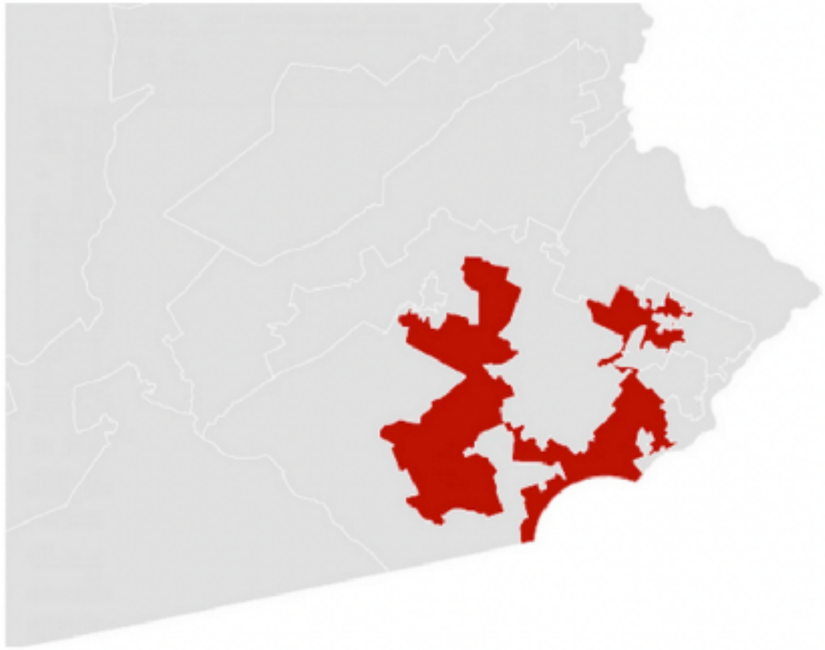
  - scale effects



https://blog.cartographica.com/blog/2011/5/19/the-modifiable-areal-unit-problem-in-gis.html

Gerrymandering: MAUP for political gain





A real district in Pennsylvania:
Democrats won 51% of the vote but only 5 out of 18 house seats

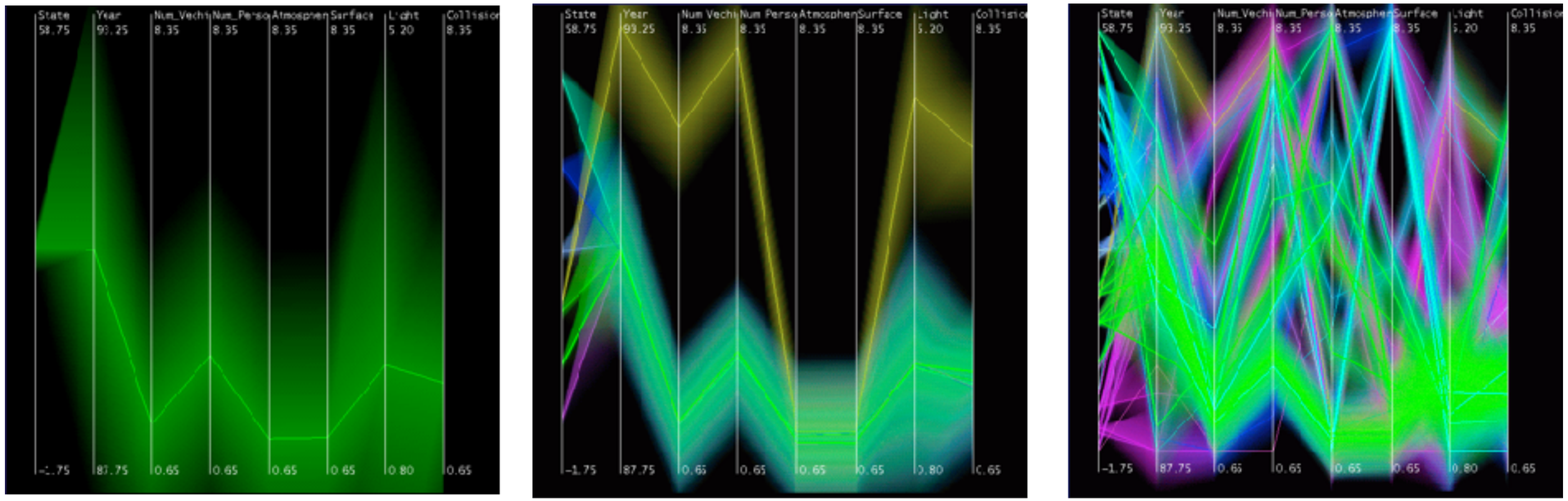
https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/

Dynamic aggregation: Clustering

- clustering: classification of items into similar bins
  - based on similiarity measure
  - hierarchical algorithms produce "similarity tree": cluster hierarchy
    - agglomerative clustering: start w/ each node as own cluster, then iteratively merge
- cluster hierarchy: derived data used w/ many dynamic aggregation idioms
  - cluster more homogeneous than whole dataset
    - statistical measures & distribution more meaningful

Idiom: **Hierarchical parallel coordinates**

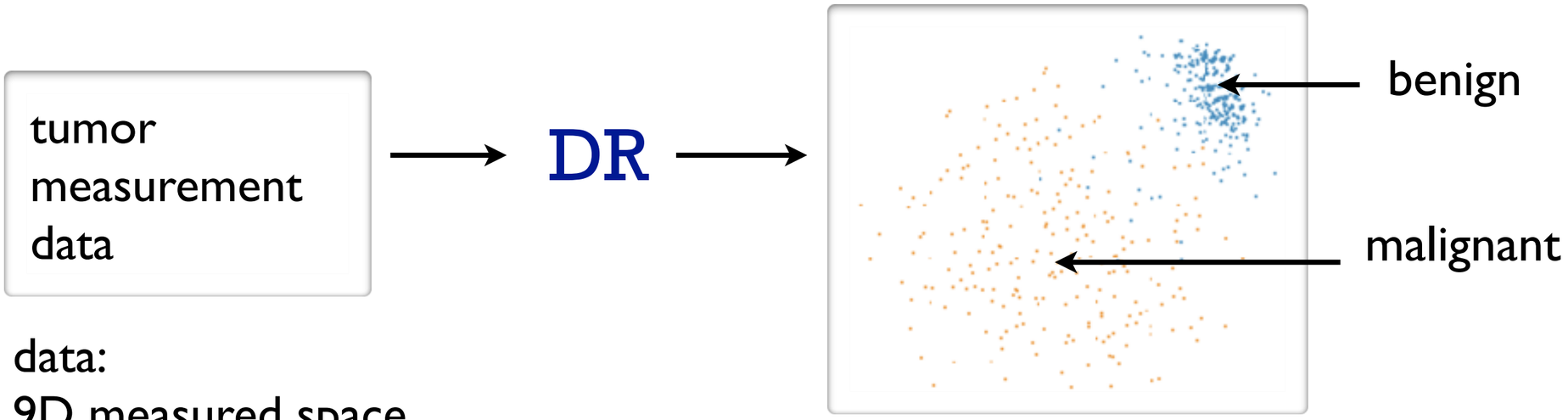
- dynamic item aggregation
- derived data: **cluster hierarchy**
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy



[Hierarchical Parallel Coordinates for Exploration of Large Datasets. Fua, Ward, and Rundensteiner. Proc. IEEE Visualization Conference (Vis '99), pp. 43– 50, 1999.]

Attribute aggregation: Dimensionality reduction

- attribute aggregation
  - derive low-dimensional target space from high-dimensional measured space
    - capture most of variance with minimal error
  - use when you can't directly measure what you care about
    - true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    - latent factors, hidden variables



data:
9D measured space

derived data:
2D target space

Idiom: **Dimensionality reduction for documents**

Task 1

In
HD data

Out
2D data

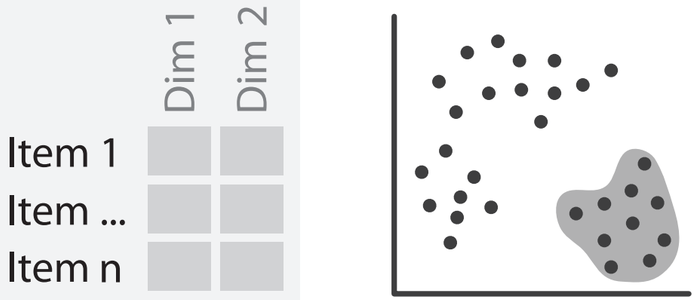
What?

- In High-dimensional data
- Out 2D data

Why?

- Produce
- Derive

Task 2



In
2D data

Out
Scatterplot
Clusters & points

What?

- In 2D data
- Out Scatterplot
- Out Clusters & points

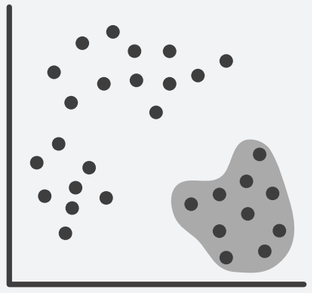
Why?

- Discover
- Explore
- Identify

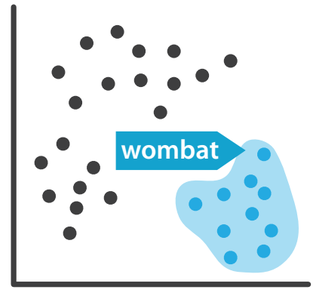
How?

- Encode
- Navigate
- Select

Task 3



In
Scatterplot
Clusters & points



Out
Labels for
clusters

What?

- In Scatterplot
- In Clusters & points
- Out Labels for clusters

Why?

- Produce
- Annotate

How?

Encode

Arrange

- Express
- Separate
- Order
- Align
- Use

Map

from categorical and ordered attributes

- Color
  - *Hue*
  - *Saturation*
  - *Luminance*
- Size, Angle, Curvature, ...
- Shape
- Motion

Direction, Rate, Frequency, ...

Manipulate

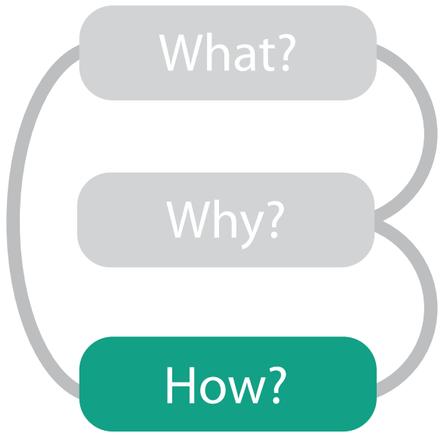
- Change
- Select
- Navigate

Facet

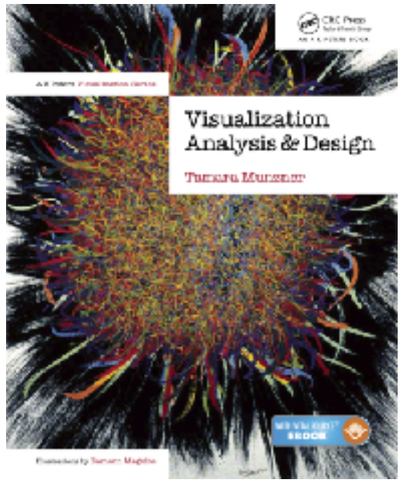
- Juxtapose
- Partition
- Superimpose

Reduce

- Filter
- Aggregate
- Embed



Visualization Analysis & Design



Embed: Focus+Context (Ch 14)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

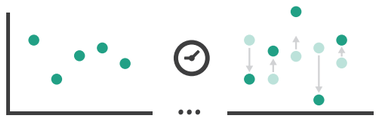
How to handle complexity: 4 strategies

→ *Derive*

- derive new data to show within view
- change view over time
- facet across multiple views
- reduce items/attributes within single view

Manipulate

→ Change



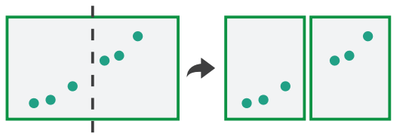
→ Select

→ Navigate

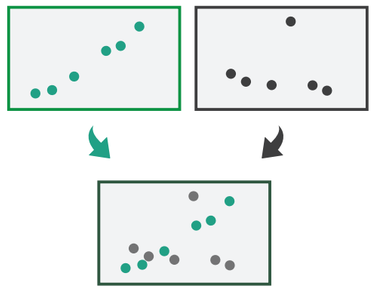
Facet

→ Juxtapose

→ Partition



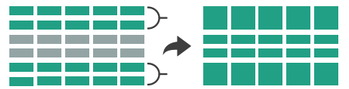
→ Superimpose



Reduce

→ Filter

→ Aggregate



→ Embed

Embed: Focus+Context

- combine focus + context info within single view
  - vs standard navigation within view
  - vs multiple views

Embed: Focus+Context

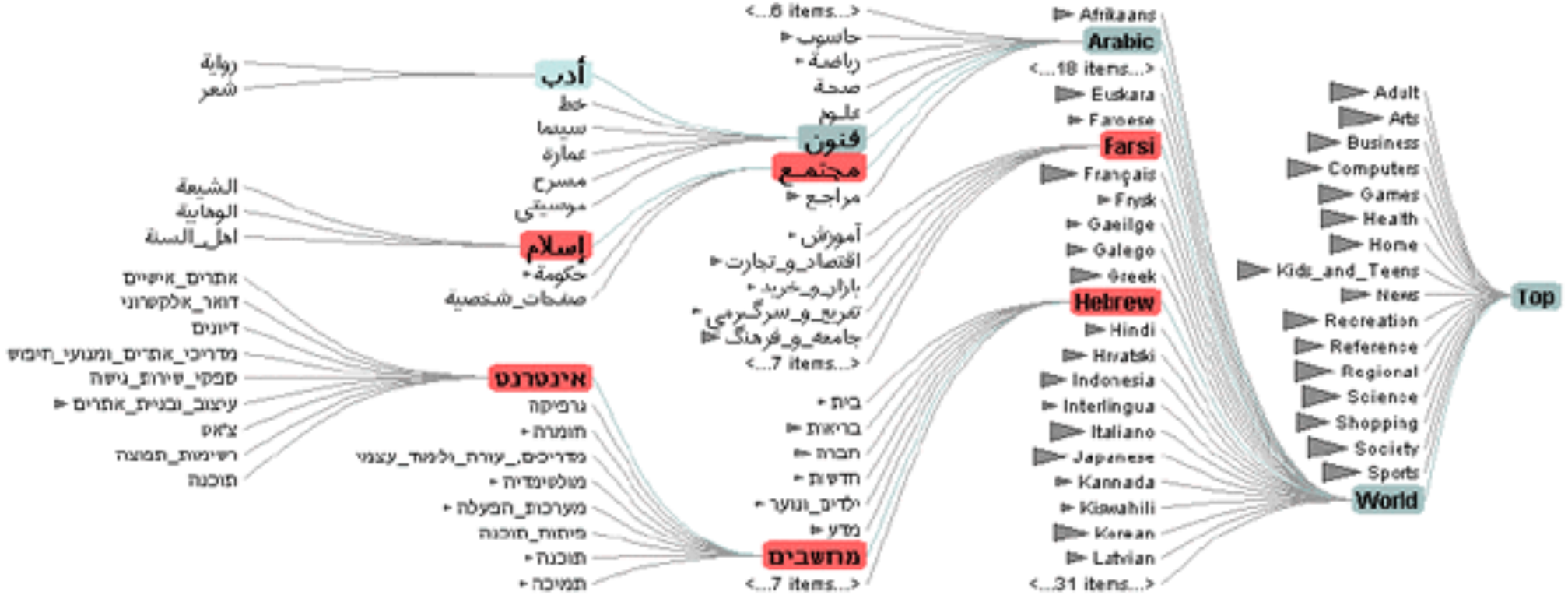
- combine focus + context info within single view
  - vs standard navigation within view
  - vs multiple views
- elide data
  - selectively filter and aggregate

➔ **Embed**

➔ Elide Data

Idiom: **DOITrees Revisited**

- focus+context choice: elide
  - some items dynamically filtered out
  - some items dynamically aggregated together
  - some items shown in detail



[DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. Heer and Card. Proc. Advanced Visual Interfaces (AVI), pp. 421–424, 2004.]

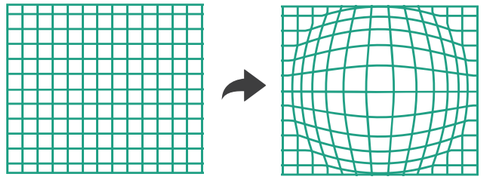
Embed: Focus+Context

- combine focus + context info within single view
  - vs standard navigation within view
  - vs multiple views
- elide data
  - selectively filter and aggregate
- distort geometry
  - carefully chosen to integrate F+C

➔ **Embed**

➔ Elide Data

➔ Distort Geometry



Idiom: **Fisheye Lens**

- F+C choice: distort geometry
  - shape: radial
  - focus: single extent
  - extent: local
  - metaphor: draggable lens

[D3 Fisheye Lens] https://bost.ocks.org/mike/fisheye/

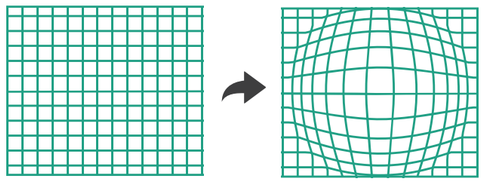
Embed: Focus+Context

- combine focus + context info within single view
  - vs standard navigation within view
  - vs multiple views
- elide data
  - selectively filter and aggregate
- distort geometry: design choices
  - region shape: radial, rectilinear, complex
  - how many regions: one, many
  - region extent: local, global
  - interaction metaphor

➔ **Embed**

➔ Elide Data

➔ Distort Geometry

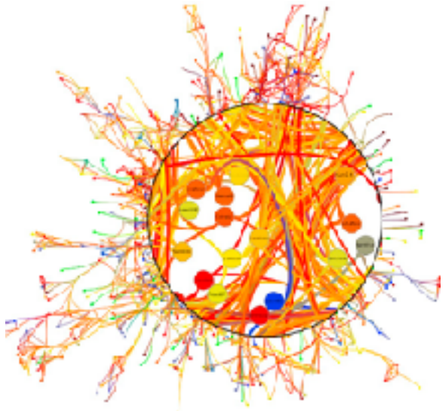


Distortion costs and benefits

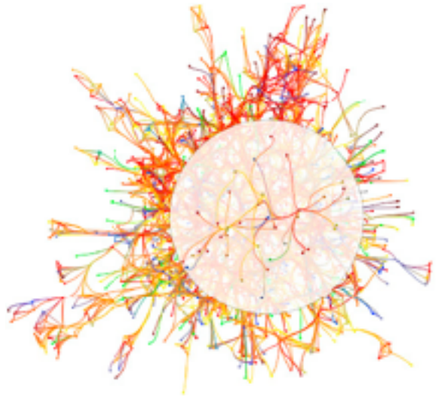
- benefits
  - combine focus and context information in single view
- costs
  - length comparisons impaired
    - topology comparisons unaffected: connection, containment
  - effects of distortion unclear if original structure unfamiliar
  - object constancy/tracking may be impaired

fisheye lens

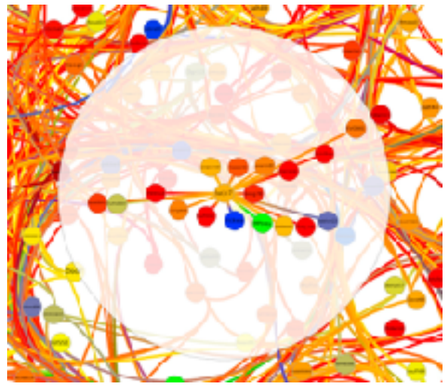
magnifying lens



neighborhood layering



Bring and Go



[Living Flows: Enhanced Exploration of Edge-Bundled Graphs Based on GPU-Intensive Edge Rendering. Lambert, Auber, and Melançon. Proc. Intl. Conf. Information Visualisation (IV), pp. 523–530, 2010.]