

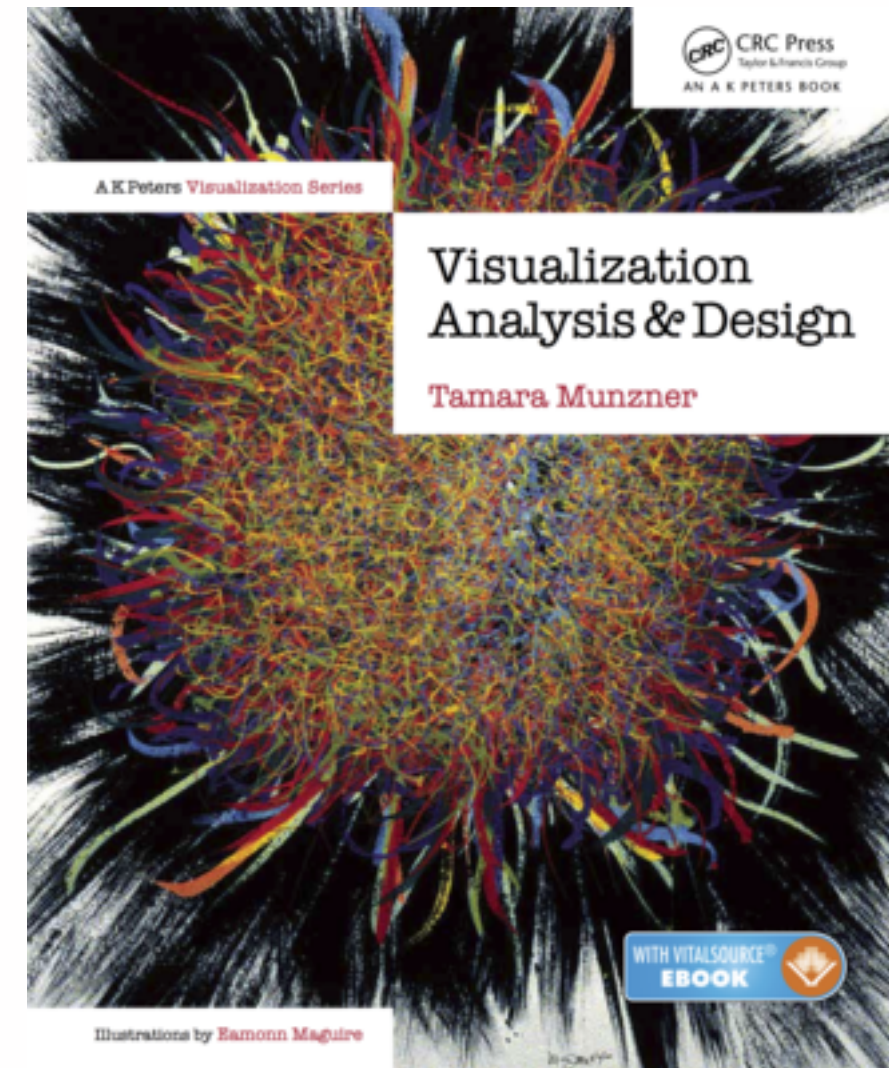
Visualization Analysis & Design

Tamara Munzner

Department of Computer Science
University of British Columbia

*Seminar “Bridging Information Visualization with Machine Learning”
March 1 2015, Schloss Dagstuhl, Wadern DE*

<http://www.cs.ubc.ca/~tmm/talks.html#vad15dagstuhl>



Defining visualization (vis)

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Why?...

Why have a human in the loop?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

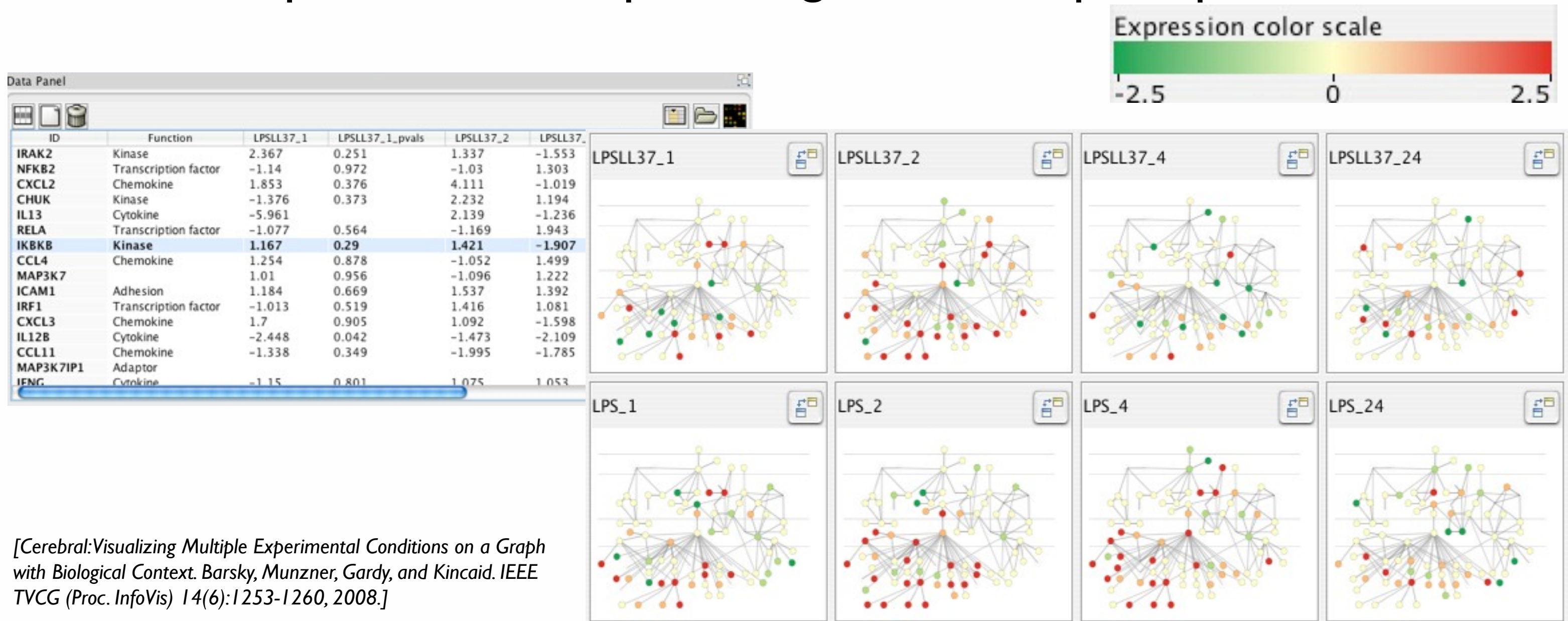
Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

- don't need vis when fully automatic solution exists and is trusted
- many analysis problems ill-specified
 - don't know exactly what questions to ask in advance
- possibilities
 - long-term use for end users (e.g. exploratory analysis of scientific data)
 - presentation of known results
 - stepping stone to better understanding of requirements before developing models
 - help developers of automatic solution refine/debug, determine parameters
 - help end users of automatic solutions verify, build trust

Why use an external representation?

Computer-based visualization systems provide **visual representations** of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception



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

Why represent all the data?

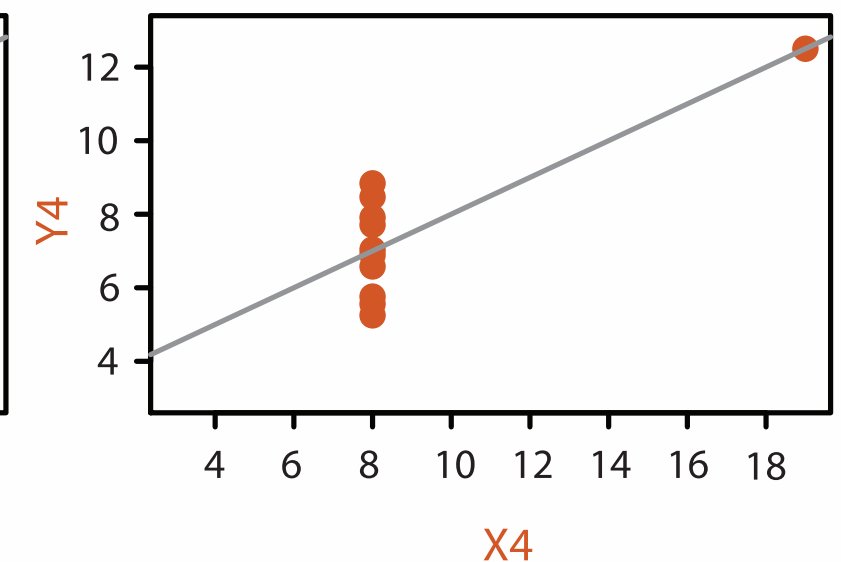
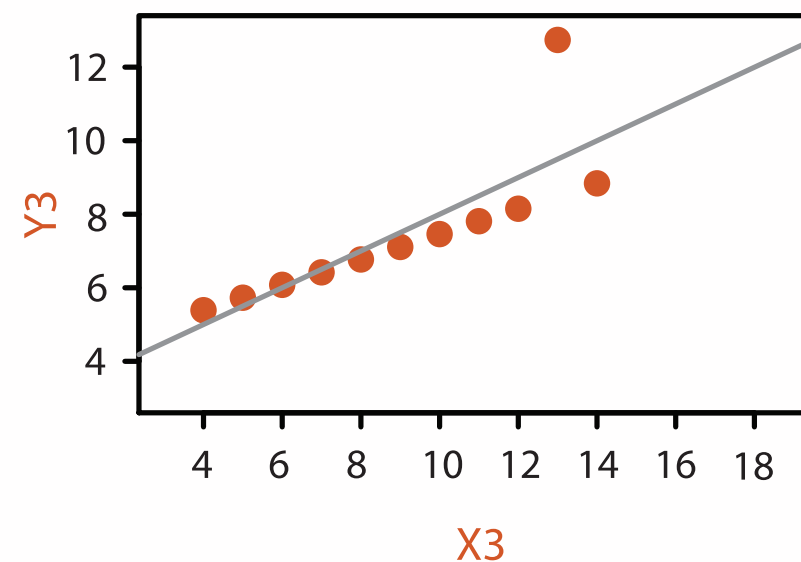
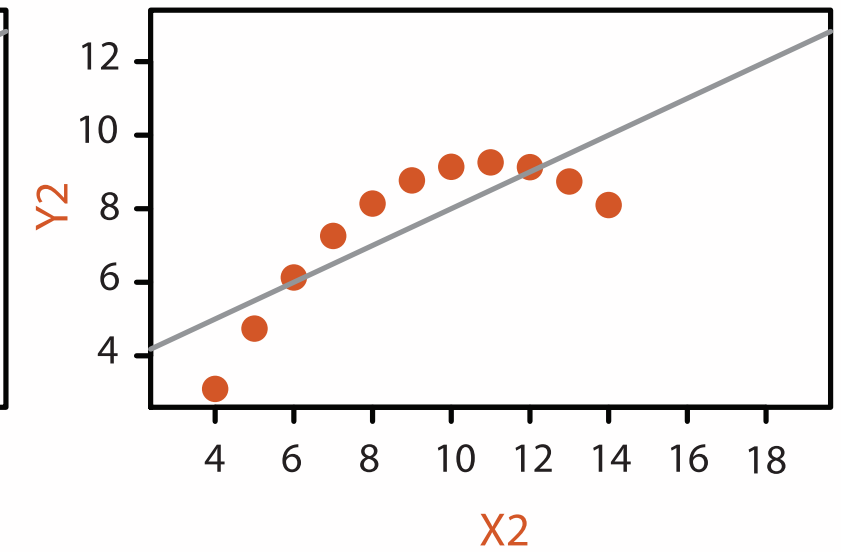
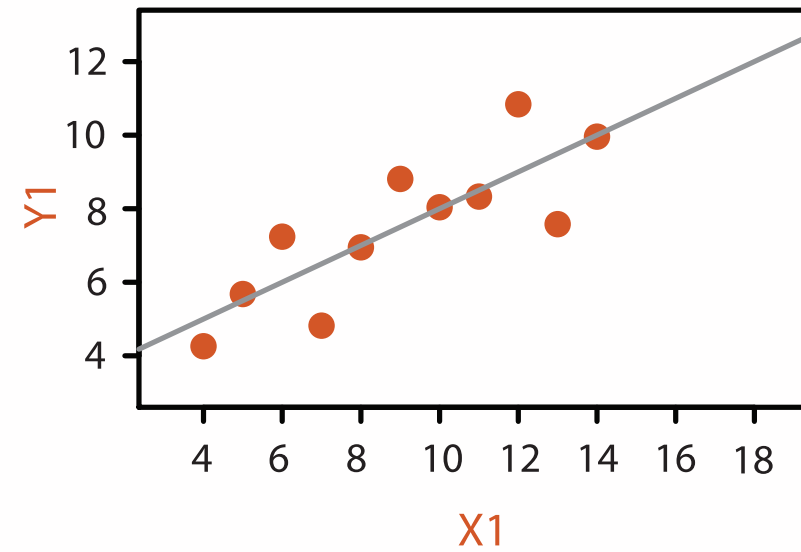
Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
 - confirm expected and find unexpected patterns
 - assess validity of statistical model

Anscombe's Quartet

Identical statistics

x mean	9
x variance	10
y mean	8
y variance	4
x/y correlation	1



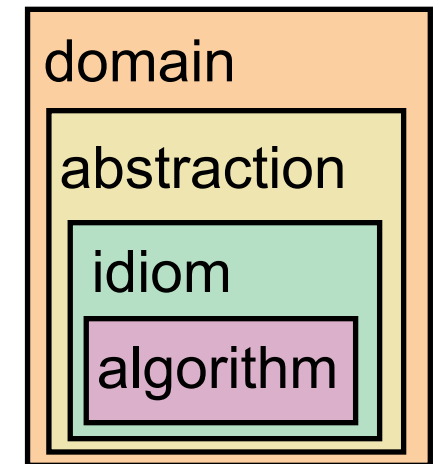
Why are there resource limitations?

Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.

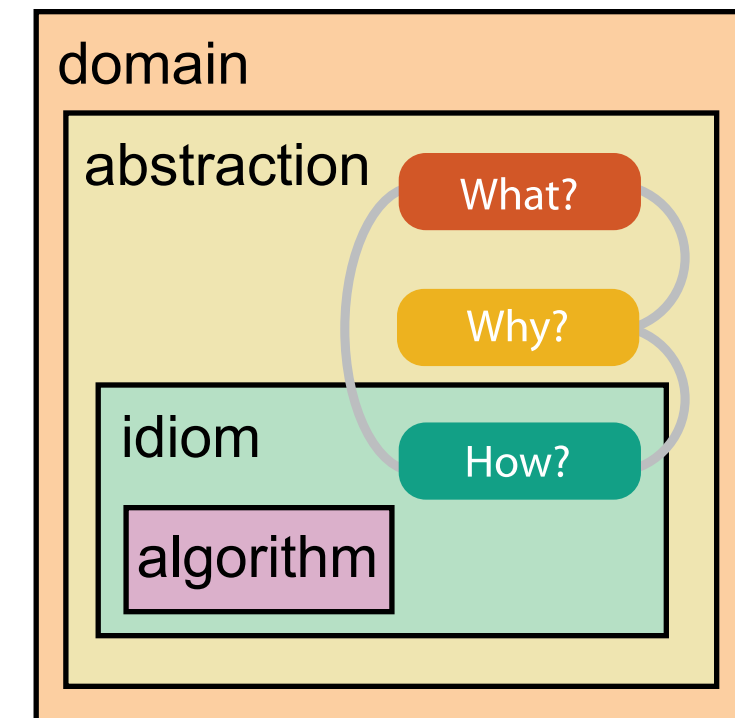
- computational limits
 - processing time
 - system memory
- human limits
 - human attention and memory
- display limits
 - pixels are precious resource, the most constrained resource
 - **information density**: ratio of space used to encode info vs unused whitespace
 - tradeoff between clutter and wasting space, find sweet spot between dense and sparse

Analysis framework: Four levels, three questions

- *domain* situation
 - who are the target users?
- *abstraction*
 - translate from specifics of domain to vocabulary of vis
 - **what** is shown? **data abstraction**
 - **why** is the user looking at it? **task abstraction**
- *idiom*
 - **how** is it shown?
 - **visual encoding idiom**: how to draw
 - **interaction idiom**: how to manipulate
- *algorithm*
 - efficient computation



[A Nested Model of Visualization Design and Validation.
Munzner. *IEEE TVCG* 15(6):921-928, 2009 (Proc. InfoVis 2009).]



[A Multi-Level Typology of Abstract Visualization Tasks
Brehmer and Munzner. *IEEE TVCG* 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]


Validation methods from different fields for each level

anthropology/
ethnography

 **Domain situation**
Observe target users using existing tools

 **Data/task abstraction**

 **Visual encoding/interaction idiom**
Justify design with respect to alternatives

 **Algorithm**
Measure system time/memory
Analyze computational complexity

Analyze results qualitatively
Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

design

computer
science

cognitive
psychology

anthropology/
ethnography

- mismatch: cannot show idiom good with system timings
- mismatch: cannot show abstraction good with lab study

Why analyze?

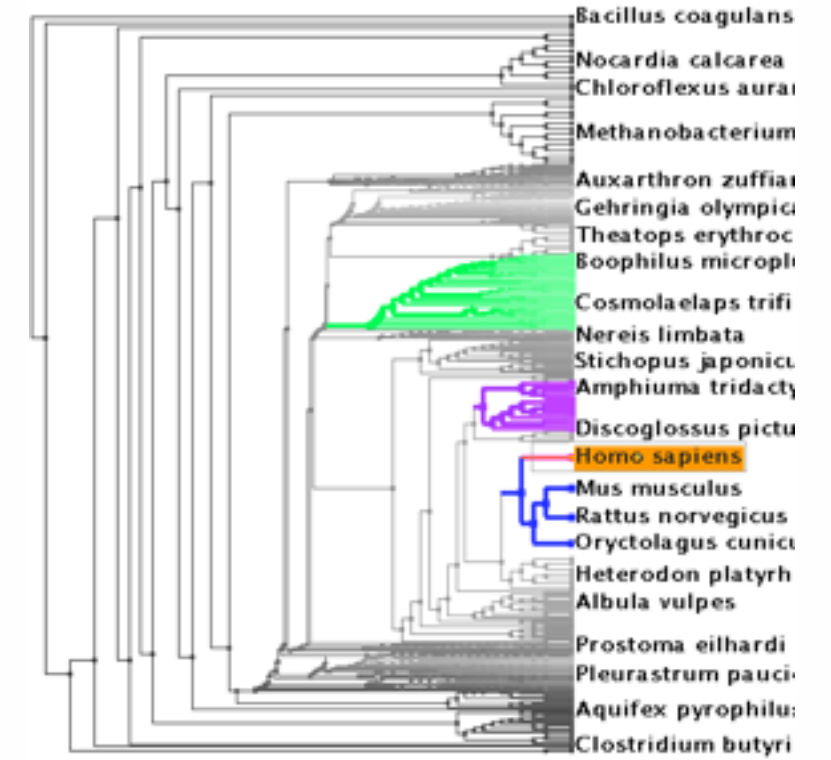
- imposes a structure on huge design space
 - scaffold to help you think systematically about choices
 - analyzing existing as stepping stone to designing new

SpaceTree



[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

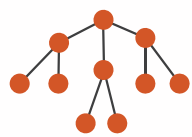
TreeJuxtaposer



[Tree]juxtaposer: Scalable Tree Comparison Using Focus +Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]

What?

→ Tree



Why?

→ Actions

→ Present → Locate → Identify



→ Targets

→ Path between two nodes



How?

→ SpaceTree

→ Encode → Navigate → Select → Filter → Aggregate



→ TreeJuxtaposer

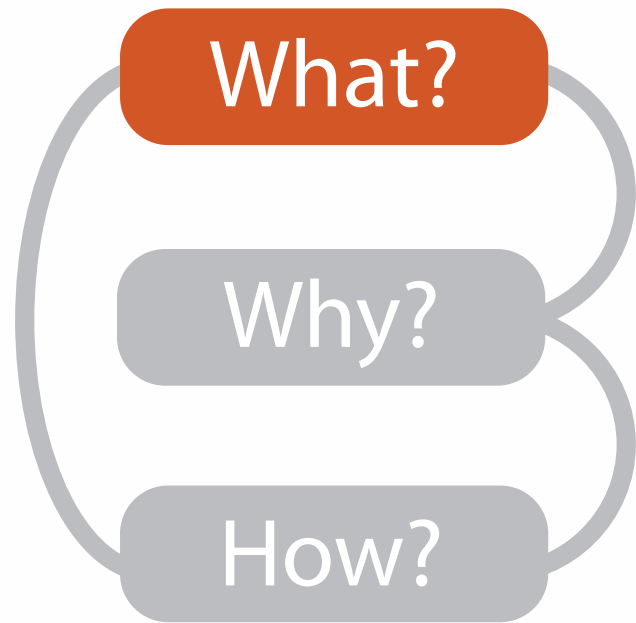
→ Encode → Navigate → Select → Arrange



What?

Why?

How?



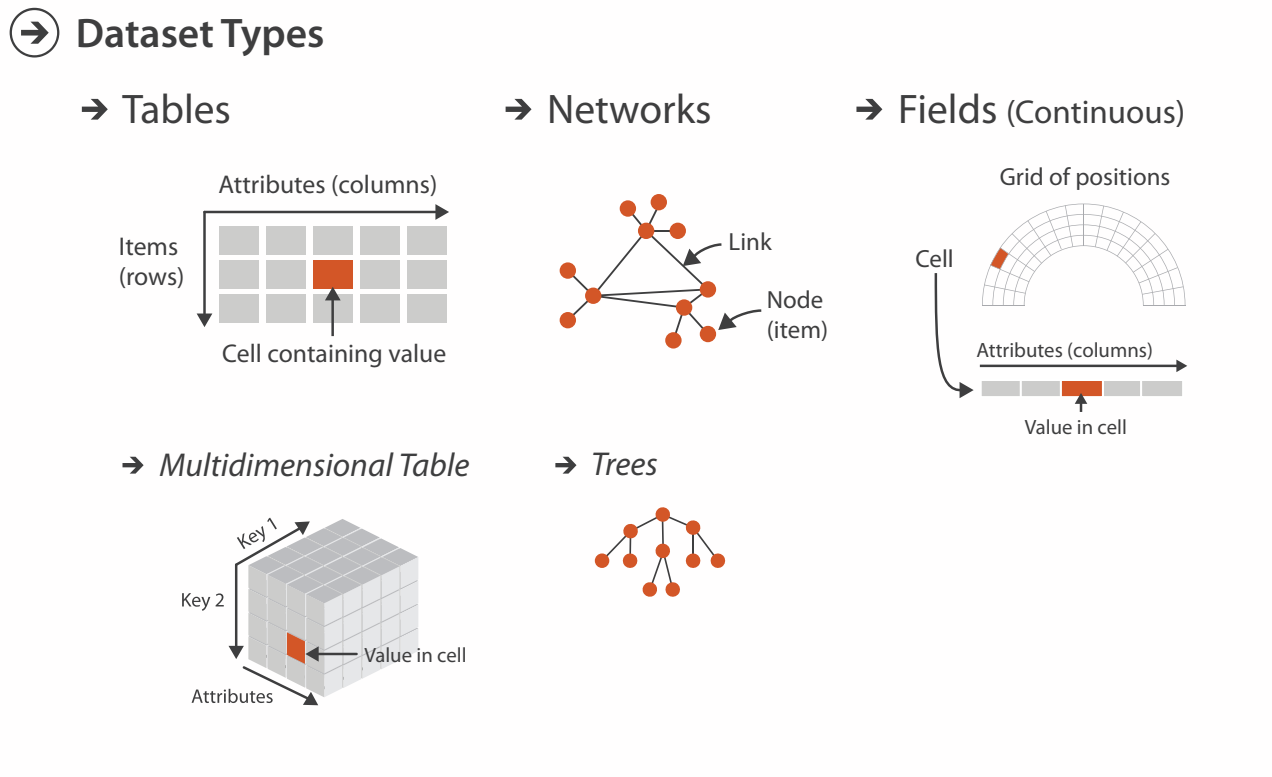
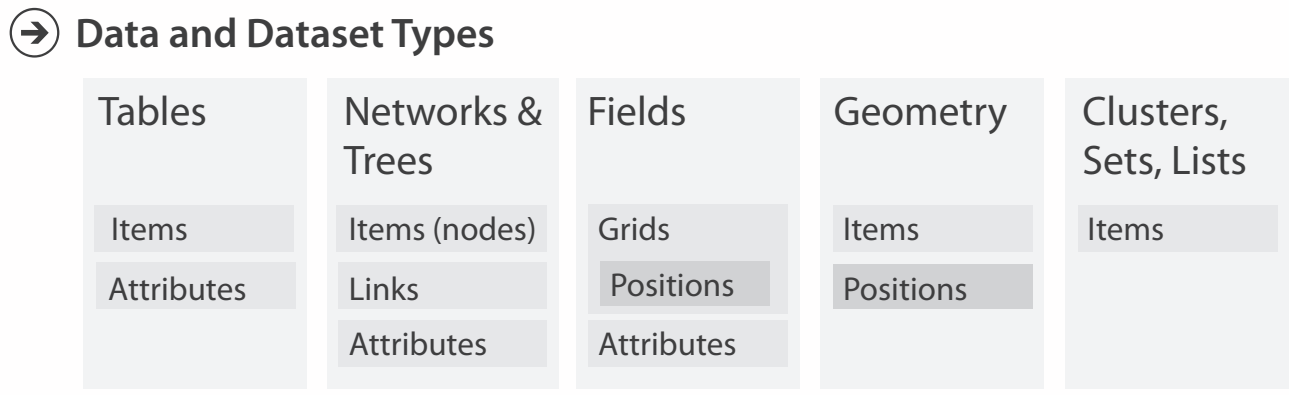
What?

Datasets

Attributes

- ➔ **Data Types**
 - ➔ Items
 - ➔ Attributes
 - ➔ Links
 - ➔ Positions
 - ➔ Grids

- ➔ **Attribute Types**
 - ➔ Categorical
 - + ● ■ ▲
 - ➔ Ordered
 - ➔ Ordinal
 - 👕 👕 👕
 - ➔ Quantitative
 - ┆┆ ┆┆┆ ┆┆┆┆┆



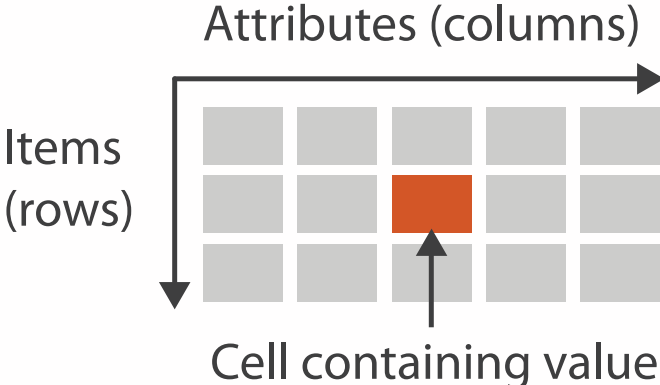
- ➔ **Ordering Direction**
 - ➔ Sequential
 -
 - ➔ Diverging
 - ←→
 - ➔ Cyclic
 - ↻



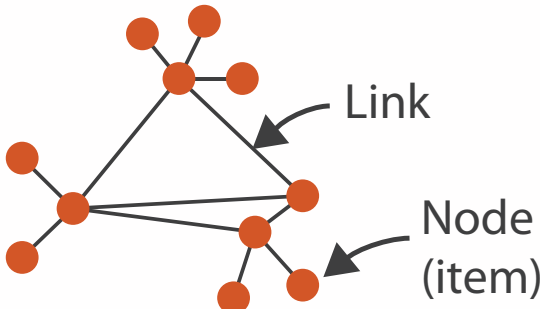
Dataset and data types

→ Dataset Types

→ Tables



→ Networks



Spatial

→ Fields (Continuous)

A semi-circular grid of positions. One cell is highlighted in red and labeled 'Cell'. Below the grid, a horizontal row of five cells is shown, with the third cell highlighted in red and labeled 'Value in cell'. An arrow labeled 'Attributes (columns)' points to the right above this row.

→ Geometry (Spatial)

A map of the British Isles with five red dots indicating specific locations. One dot is labeled 'Position' with an arrow pointing to it.

→ Attribute Types

→ Categorical

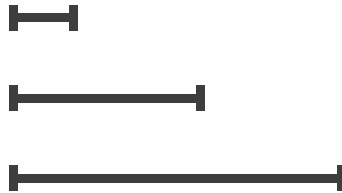


→ Ordered

→ Ordinal



→ Quantitative



Why?

👉 Actions

🎯 Targets

➔ **Analyze**

- ➔ Consume
 - ➔ Discover
 - ➔ Present
 - ➔ Enjoy
- ➔ Produce
 - ➔ Annotate
 - ➔ Record
 - ➔ Derive

➔ **All Data**

- ➔ Trends
- ➔ Outliers
- ➔ Features

➔ **Attributes**

- ➔ One
 - ➔ Distribution
 - ➔ Extremes
- ➔ Many
 - ➔ Dependency
 - ➔ Correlation
 - ➔ Similarity

➔ **Search**

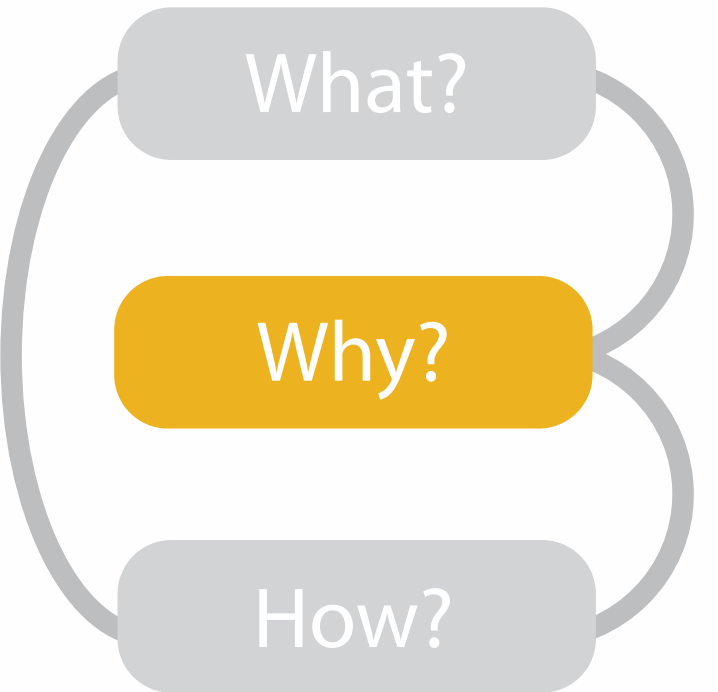
	Target known	Target unknown
Location known	<i>Lookup</i>	<i>Browse</i>
Location unknown	<i>Locate</i>	<i>Explore</i>

➔ **Network Data**

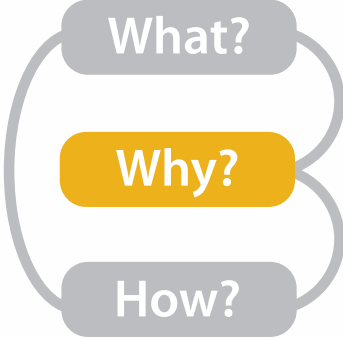
- ➔ Topology
- ➔ Paths

➔ **Spatial Data**

- ➔ Shape



- {action, target} pairs
 - discover distribution
 - compare trends
 - locate outliers
 - browse topology



Actions I: Analyze

- consume
 - discover vs present
 - classic split
 - aka explore vs explain
 - enjoy
- produce
 - newcomer
 - aka casual, social
- produce
 - annotate, record
 - derive
 - crucial design choice

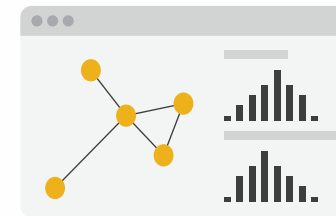
→ Analyze

→ Consume

→ Discover



→ Present

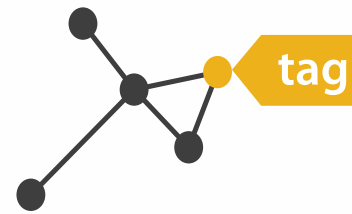


→ Enjoy



→ Produce

→ Annotate



→ Record







→ Derive



Actions II: Search

- what does user know?
 - target, location





➔ Search

	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>

Actions III: Query

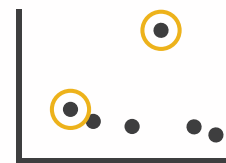
- what does user know?
 - target, location
- how much of the data matters?
 - one, some, all

➔ Search

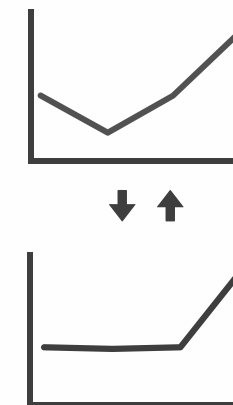
	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>

➔ Query

➔ Identify



➔ Compare



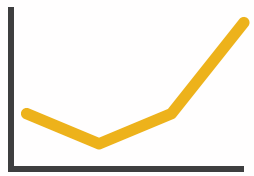
➔ Summarize



Targets

→ All Data

→ Trends



→ Outliers



→ Features



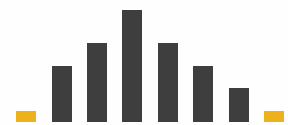
→ Attributes

→ One

→ *Distribution*

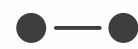


→ *Extremes*

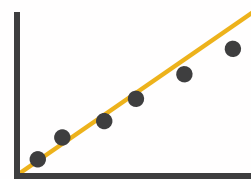


→ Many

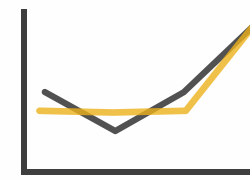
→ *Dependency*



→ *Correlation*

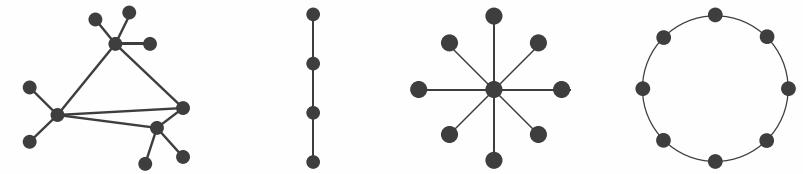


→ *Similarity*

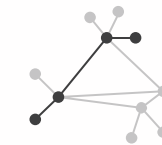


→ Network Data

→ Topology

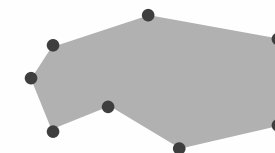


→ *Paths*



→ Spatial Data

→ Shape



How?

Encode

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



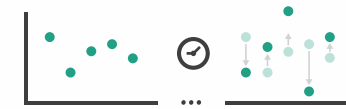
→ Motion

Direction, Rate, Frequency, ...

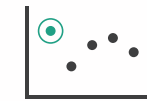


Manipulate

→ Change



→ Select



→ Navigate

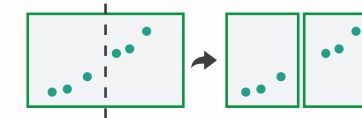


Facet

→ Juxtapose



→ Partition



→ Superimpose



Reduce

→ Filter



→ Aggregate



→ Embed



What?

Why?

How?

How?

Encode

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



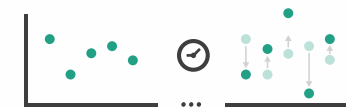
→ Motion

Direction, Rate, Frequency, ...

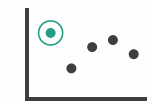


Manipulate

→ Change



→ Select



→ Navigate

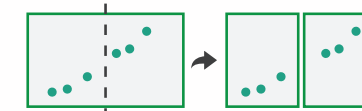


Facet

→ Juxtapose



→ Partition



→ Superimpose



Reduce

→ Filter



→ Aggregate



→ Embed



What?

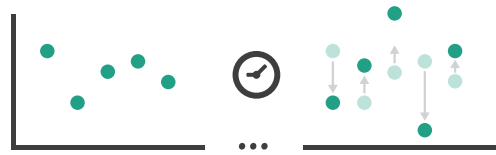
Why?

How?

How to handle complexity: 3 more strategies + 1 previous

Manipulate

➔ Change



➔ Select

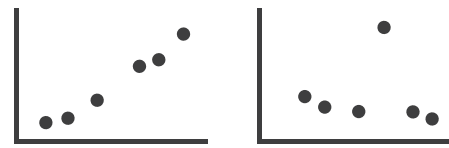


➔ Navigate

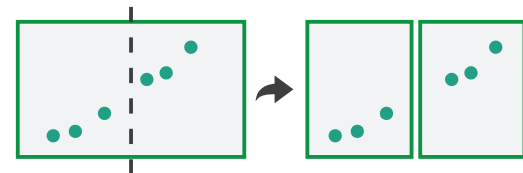


Facet

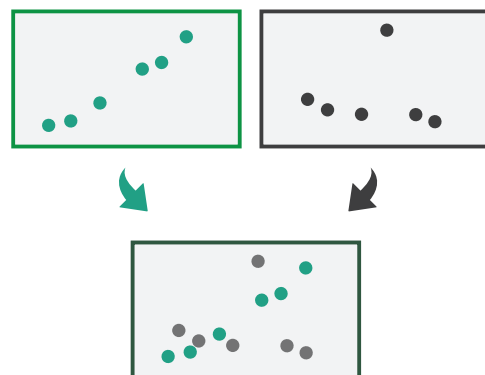
➔ Juxtapose



➔ Partition



➔ Superimpose



Reduce

➔ Filter



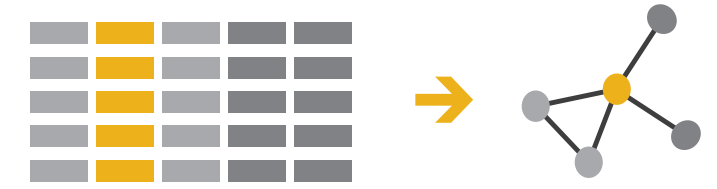
➔ Aggregate



➔ Embed



➔ *Derive*




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

How to handle complexity: 3 more strategies

+ 1 previous

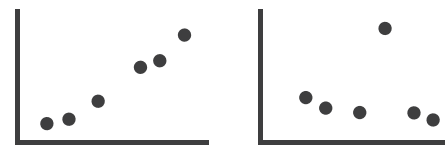
Manipulate

→ Change



Facet

→ Juxtapose

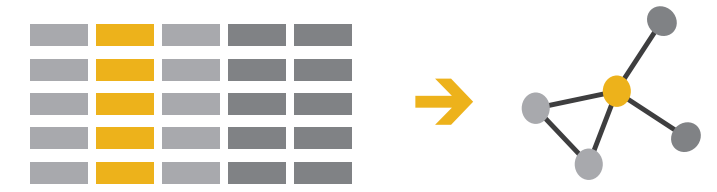


Reduce

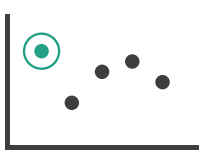
→ Filter



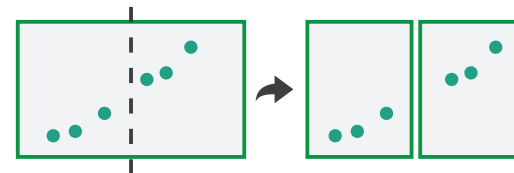
→ *Derive*



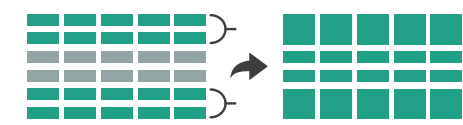
→ Select



→ Partition



→ Aggregate

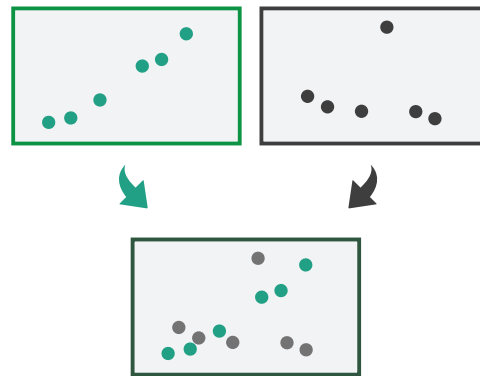


- change over time
- most obvious & flexible of the 4 strategies


→ Navigate



→ Superimpose

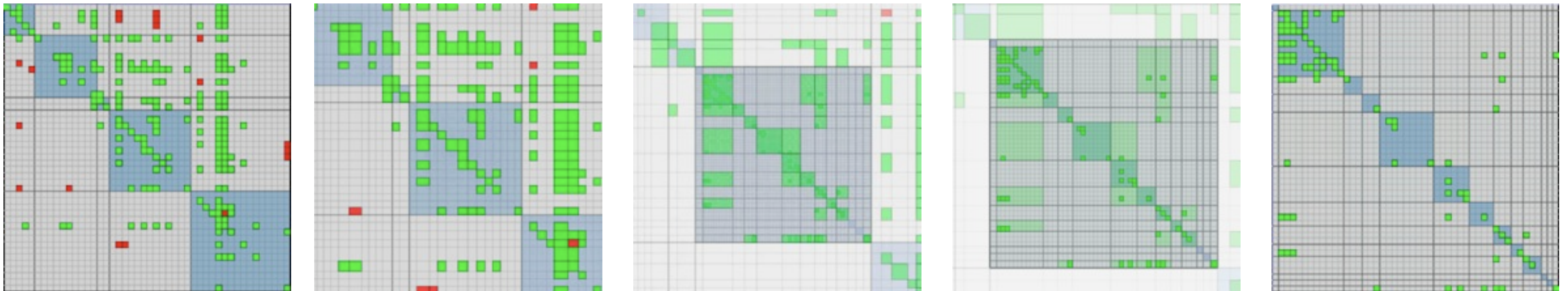


→ Embed



Idiom: **Animated transitions**

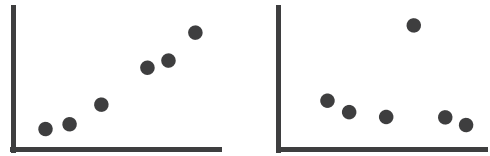
- smooth transition from one state to another
 - alternative to jump cuts
 - support for item tracking when amount of change is limited
- example: multilevel matrix views
 - scope of what is shown narrows down
 - middle block stretches to fill space, additional structure appears within
 - other blocks squish down to increasingly aggregated representations



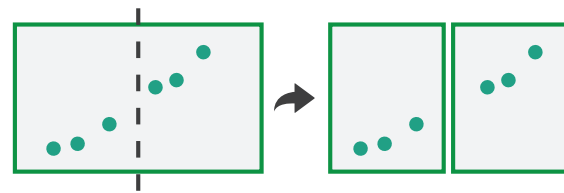
[Using Multilevel Call Matrices in Large Software Projects. van Ham. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 227–232, 2003.]

Facet

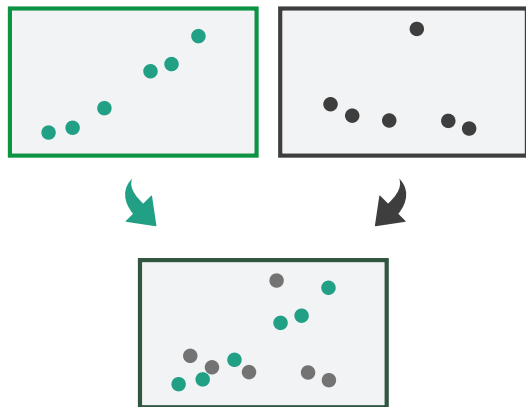
→ Juxtapose



→ Partition



→ Superimpose



→ Coordinate Multiple Side By Side Views

→ Share Encoding: Same/Different

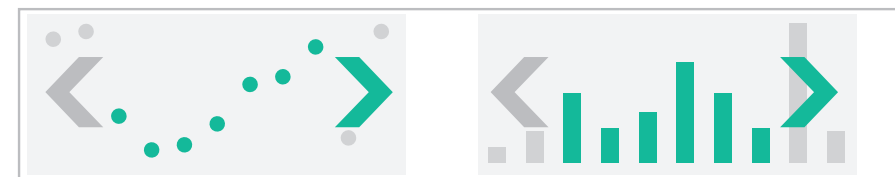
→ *Linked Highlighting*



→ Share Data: All/Subset/None



→ Share Navigation

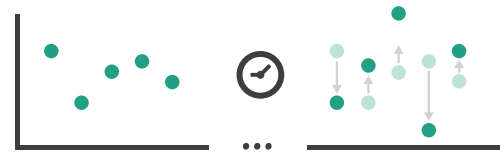


How to handle complexity: 3 more strategies

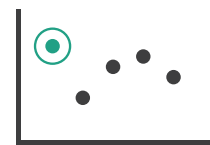
+ 1 previous

Manipulate

➔ Change



➔ Select

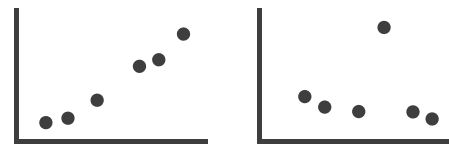


➔ Navigate

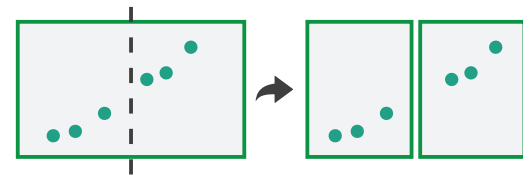


Facet

➔ Juxtapose



➔ Partition



➔ Superimpose

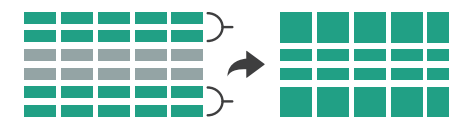


Reduce

➔ Filter



➔ Aggregate



➔ Embed



➔ *Derive*

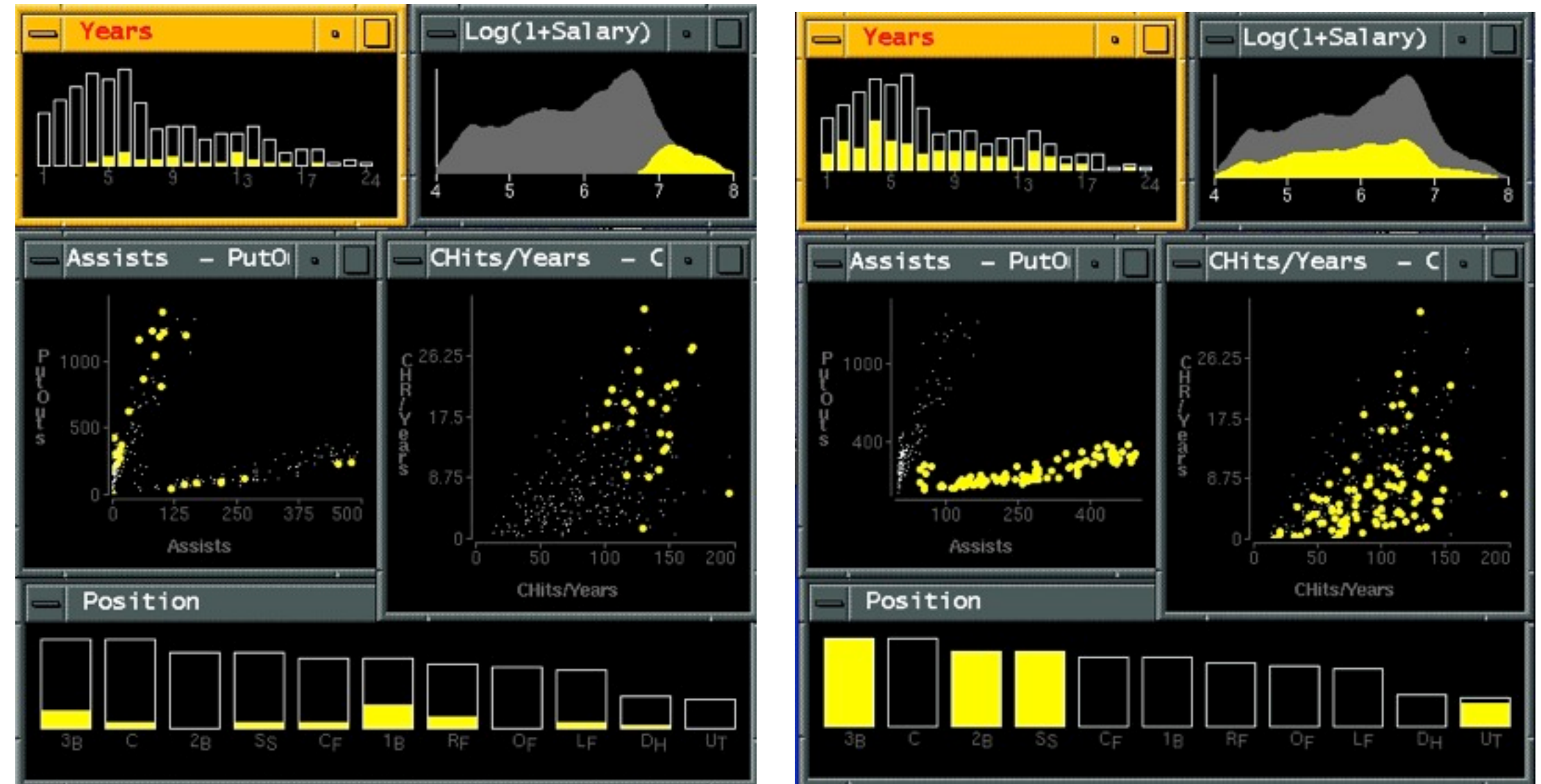


- facet data across multiple views

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



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

Idiom: **bird's-eye maps**

System: **Google Maps**

- encoding: same
- data: subset shared
- navigation: shared
 - bidirectional linking

- differences
 - viewpoint
 - (size)

- **overview-detail**



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

Idiom: **Small multiples**

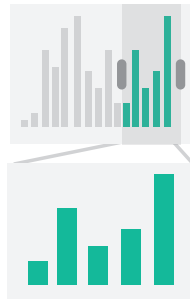
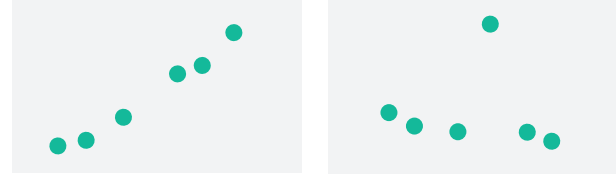


System: **Cerebral**

- encoding: same
- data: none shared
 - different attributes for node colors
 - (same network layout)
- navigation: shared



[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.]

Coordinate views: Design choice interaction

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

- why juxtapose views?

- benefits: eyes vs memory

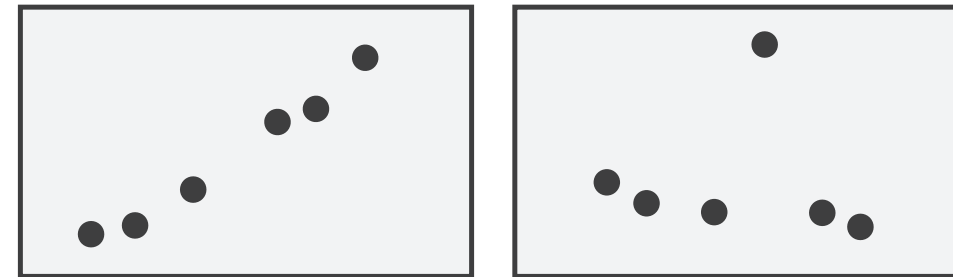
- lower cognitive load to move eyes between 2 views than remembering previous state with single changing view

- costs: display area, 2 views side by side each have only half the area of one view

Partition into views

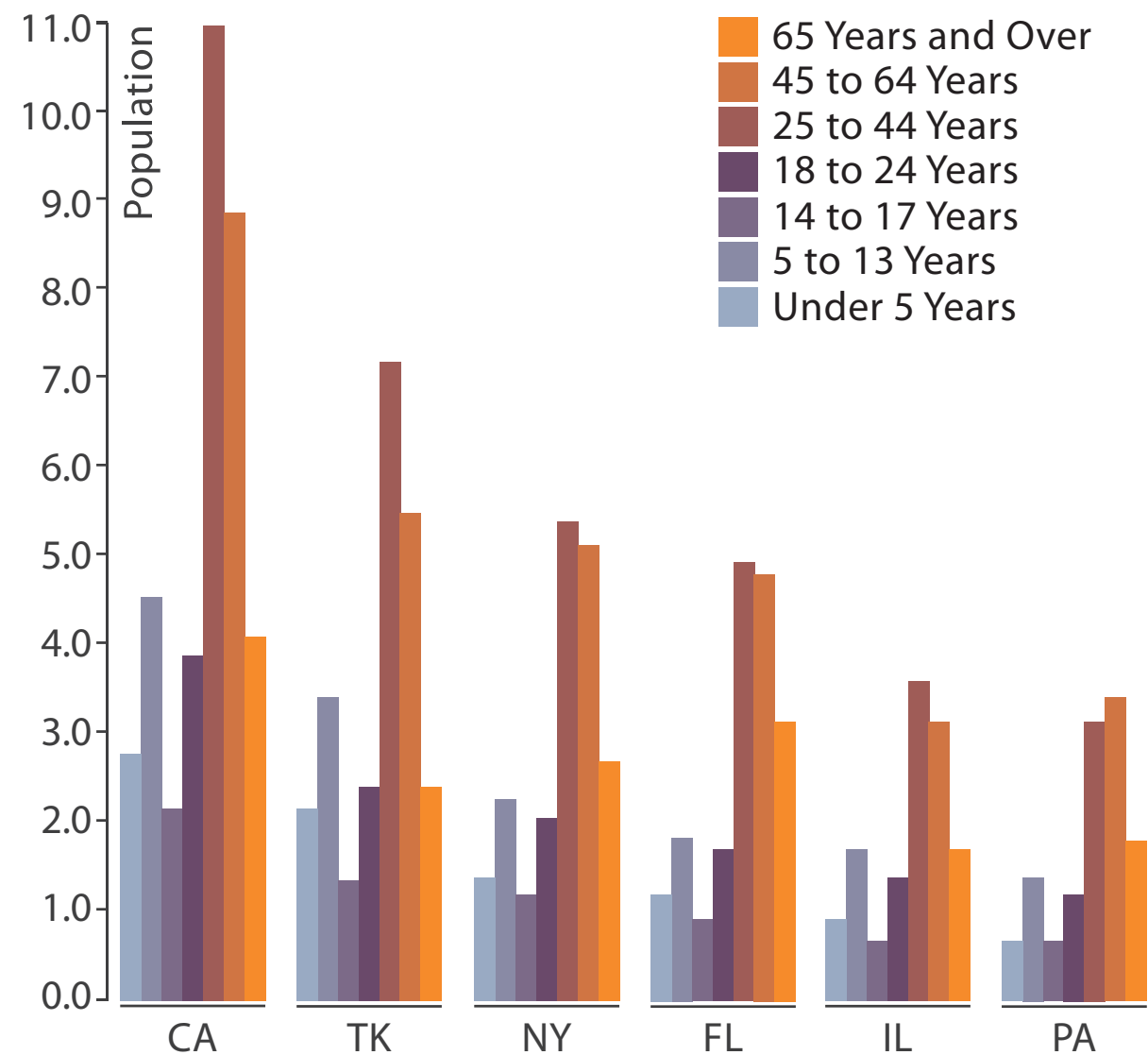
- how to divide data between views
 - encodes association between items using spatial proximity
 - major implications for what patterns are visible
 - split according to attributes
- design choices
 - how many splits
 - all the way down: one mark per region?
 - stop earlier, for more complex structure within region?
 - order in which attribs used to split
 - how many views

➔ Partition into Side-by-Side Views

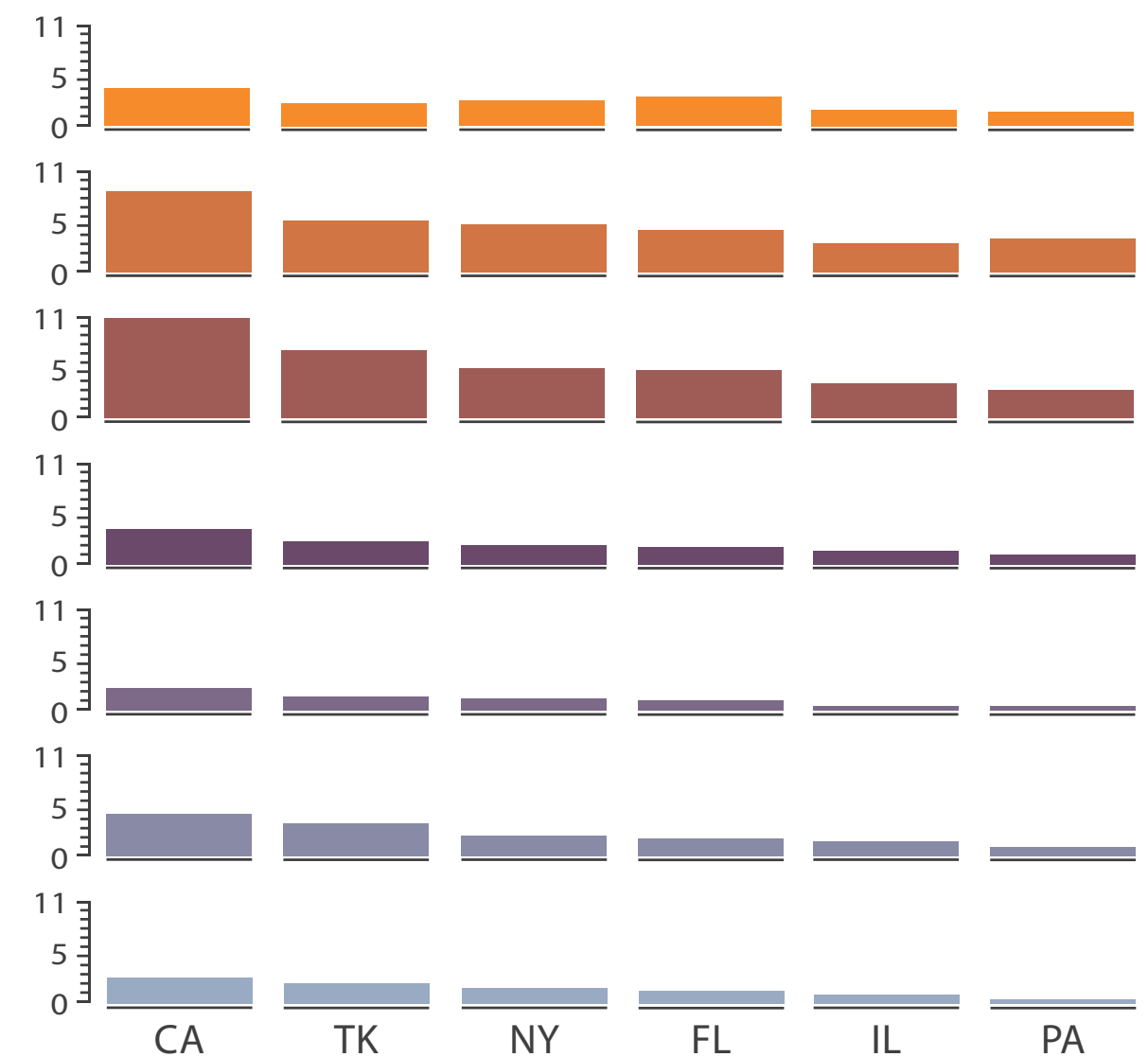


Partitioning: List alignment

- 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



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



Partitioning: Recursive subdivision

System: **HIVE**

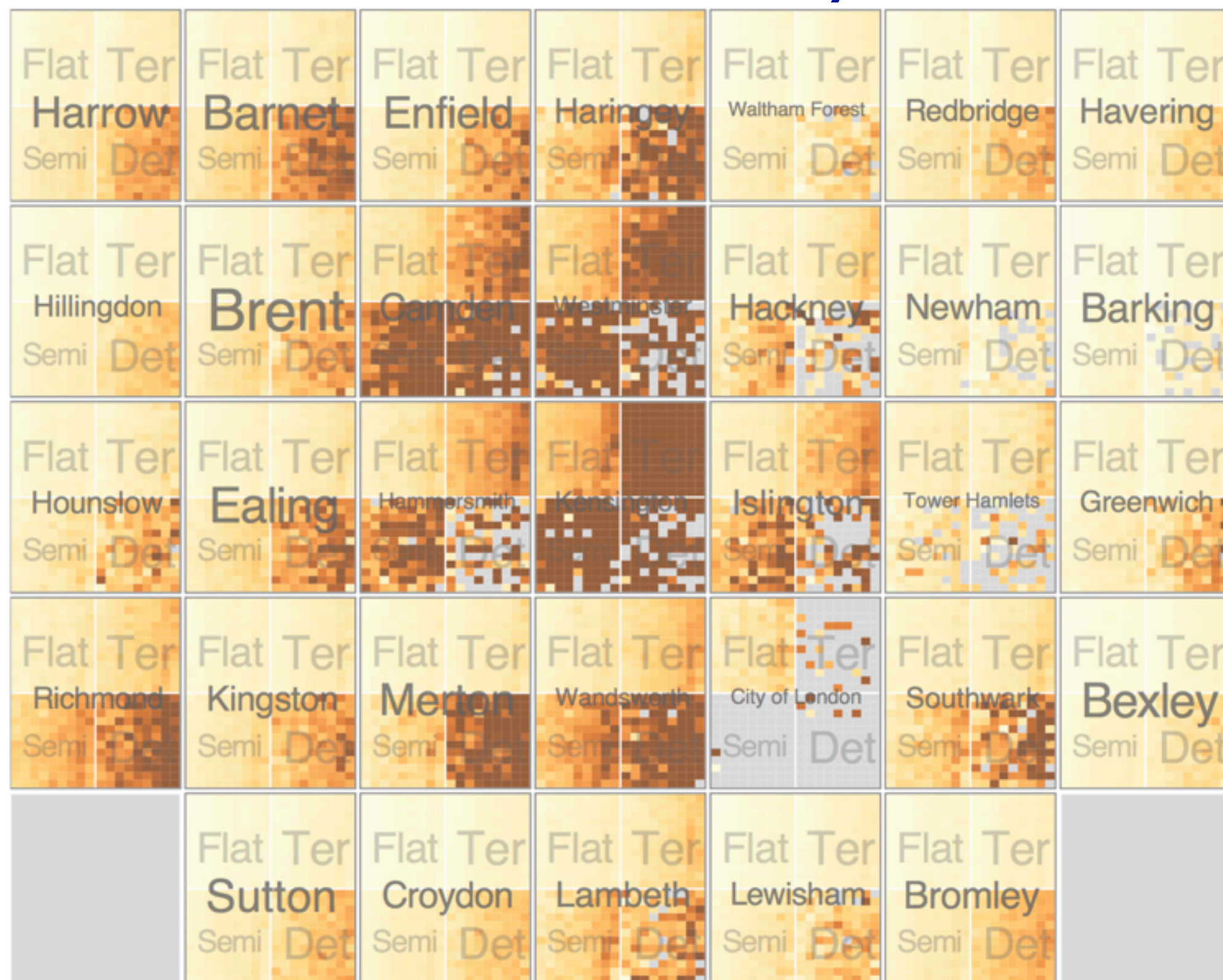
- split by type
- then by neighborhood
- then time
 - years as rows
 - months as columns



Partitioning: Recursive subdivision

System: **HIVE**

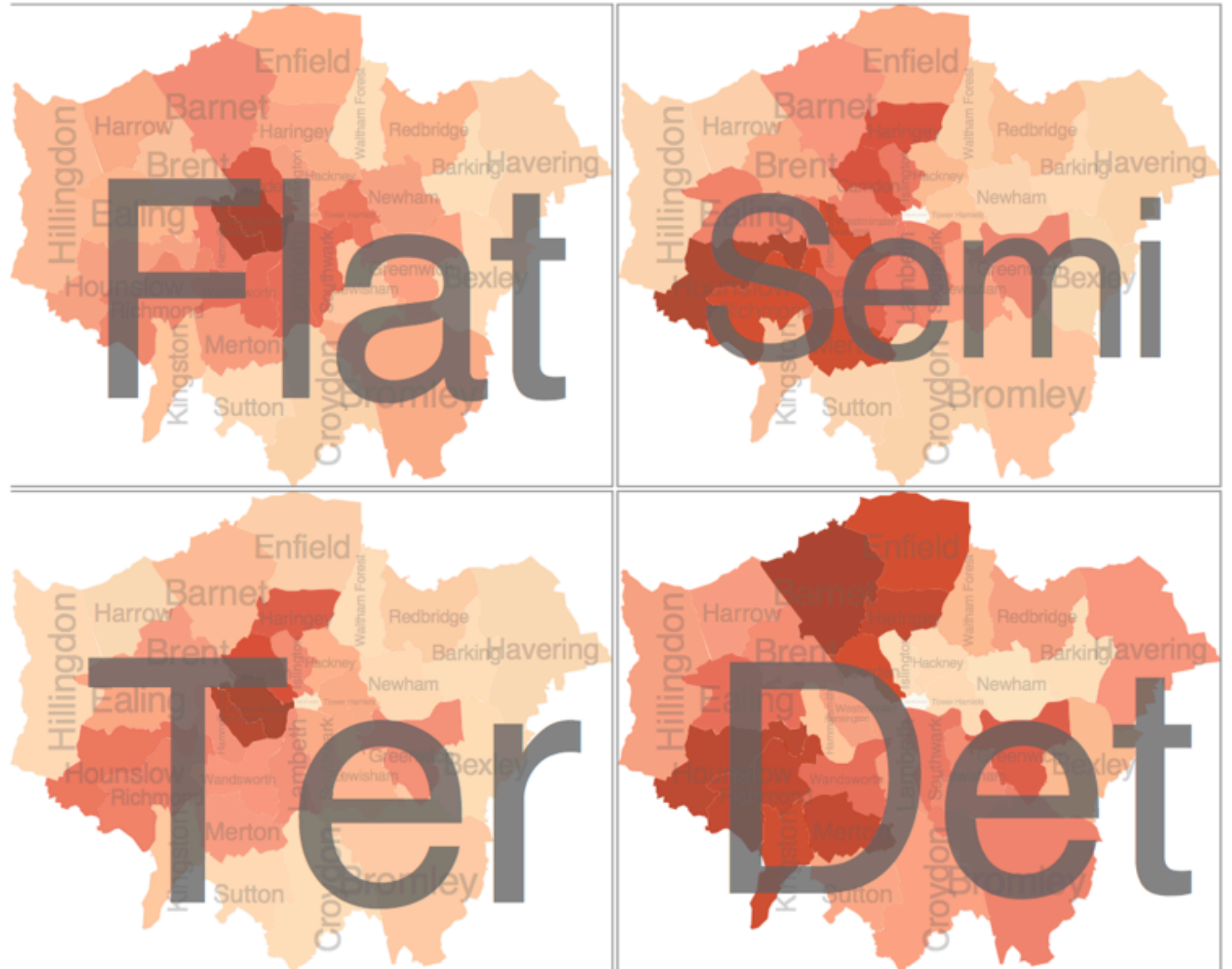
- switch order of splits
 - neighborhood then type
- very different patterns



Partitioning: Recursive subdivision

System: **HIVE**

- different encoding for second-level regions
 - choropleth maps

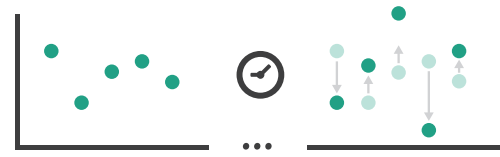


How to handle complexity: 3 more strategies

+ 1 previous

Manipulate

➔ Change



➔ Select

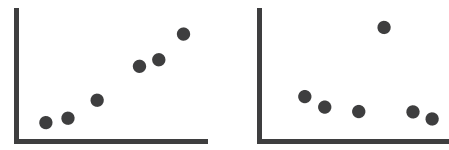


➔ Navigate

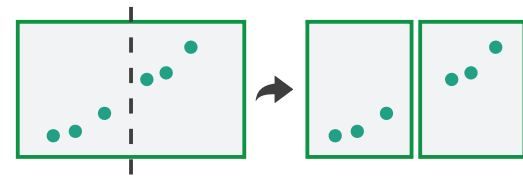


Facet

➔ Juxtapose



➔ Partition



➔ Superimpose



Reduce

➔ Filter



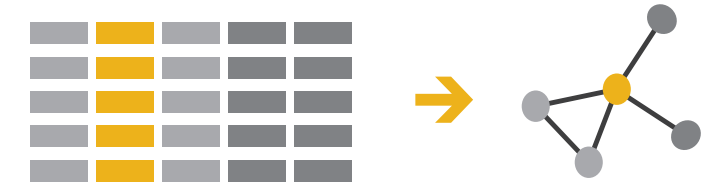
➔ Aggregate



➔ Embed



➔ *Derive*



- reduce what is shown within single view

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, facet, change, derive

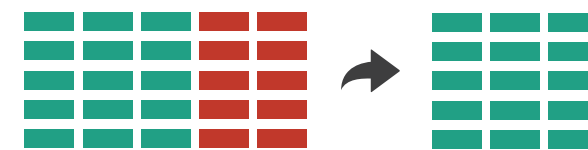
Reducing Items and Attributes

→ Filter

→ Items

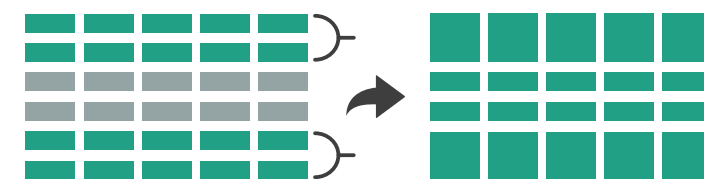


→ Attributes

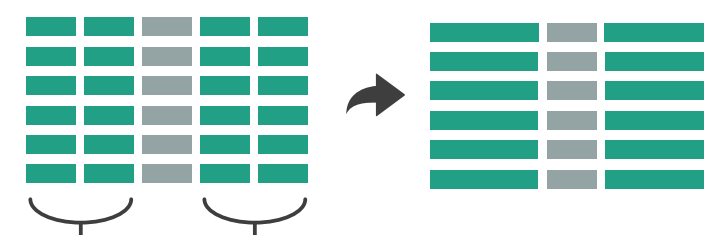


→ Aggregate

→ Items



→ Attributes

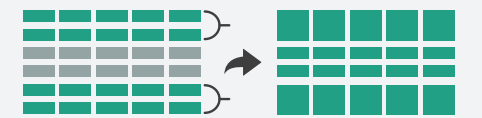


Reduce

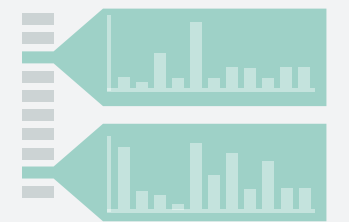
→ Filter



→ Aggregate

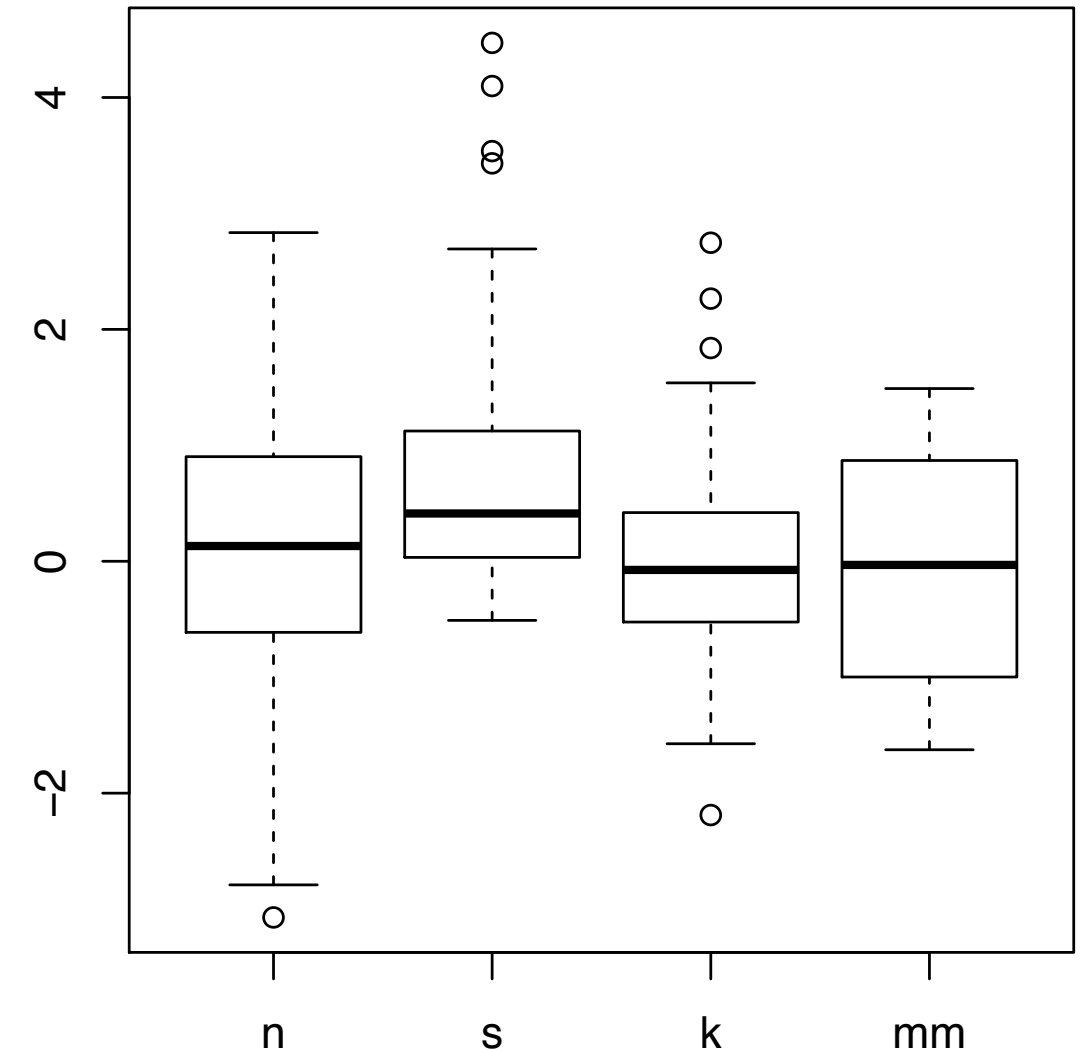


→ Embed



Idiom: **boxplot**

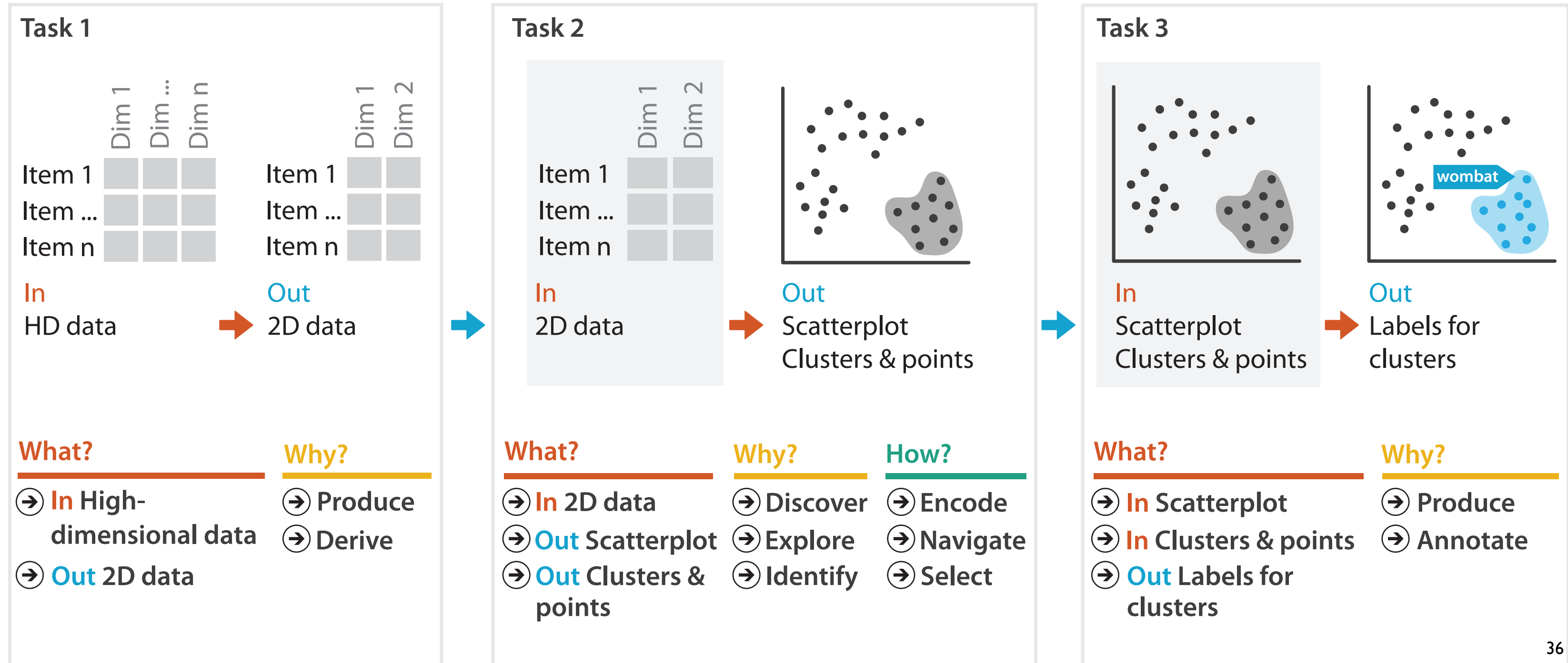
- static item aggregation
- task: find distribution
- data: table
- derived data
 - 5 quant attribs
 - median: central line
 - lower and upper quartile: boxes
 - lower upper fences: whiskers
 - values beyond which items are outliers
 - outliers beyond fence cutoffs explicitly shown



[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]

Idiom: Dimensionality reduction for documents

- attribute aggregation
 - derive low-dimensional target space from high-dimensional measured space



How?

Encode

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



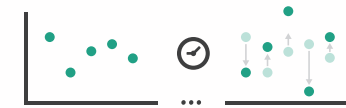
→ Motion

Direction, Rate, Frequency, ...

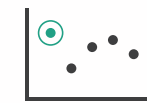


Manipulate

→ Change



→ Select



→ Navigate

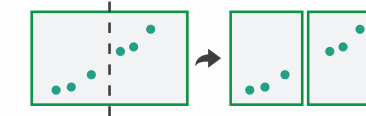


Facet

→ Juxtapose



→ Partition



→ Superimpose



Reduce

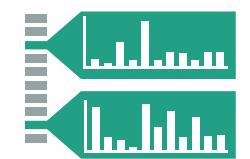
→ Filter



→ Aggregate



→ Embed



What?

Why?

How?

How to encode: Arrange space, map channels

Encode

➔ Arrange

➔ Express



➔ Order



➔ Use



➔ Separate



➔ Align



➔ Map

from **categorical** and **ordered** attributes

➔ Color

➔ Hue



➔ Saturation



➔ Luminance



➔ Size, Angle, Curvature, ...



➔ Shape



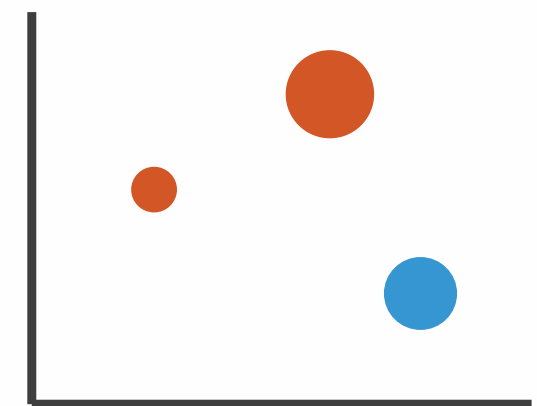
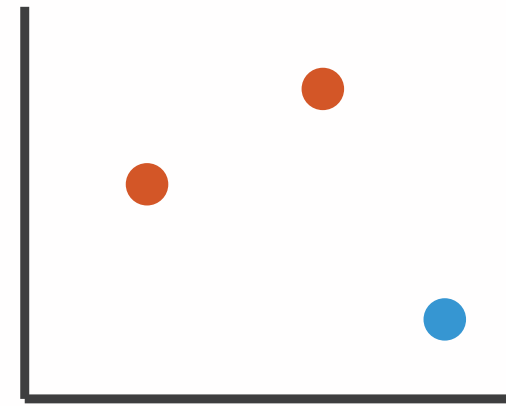
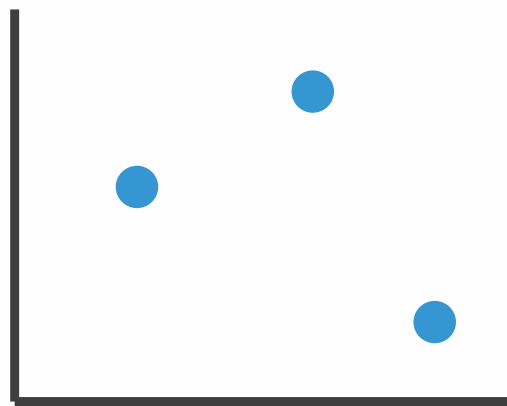
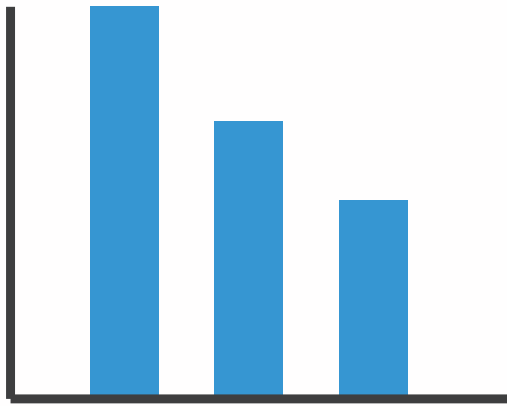
➔ Motion

Direction, Rate, Frequency, ...



Encoding visually

- analyze idiom structure



Definitions: Marks and channels

- marks

 - geometric primitives

→ Points



→ Lines



→ Areas



- channels

 - control appearance of marks

→ Position

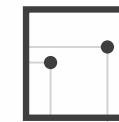
→ Horizontal



→ Vertical



→ Both



→ Color



→ Shape



→ Tilt



→ Size

→ Length



→ Area

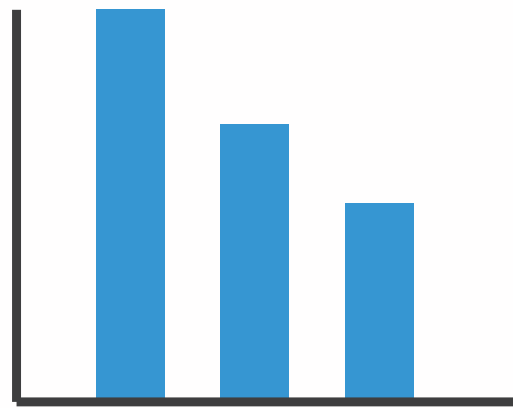


→ Volume



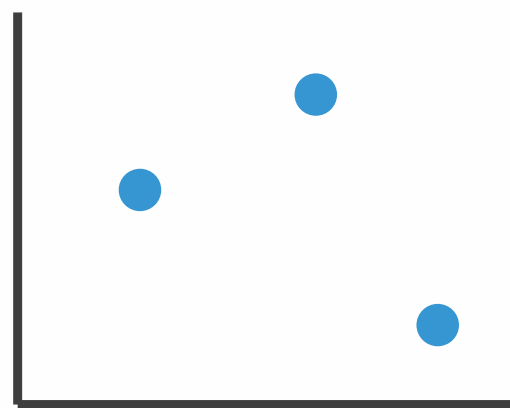
Encoding visually with marks and channels

- analyze idiom structure
 - as combination of marks and channels



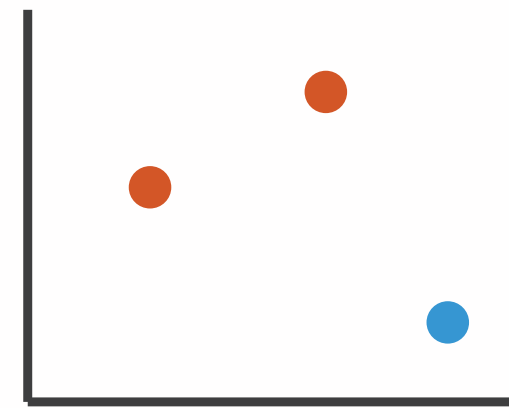
1:
vertical position

mark: line



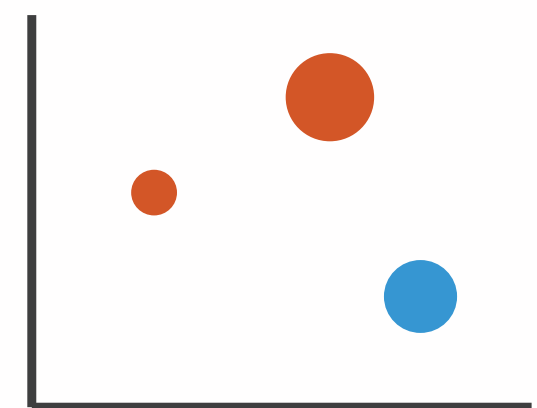
2:
vertical position
horizontal position

mark: point



3:
vertical position
horizontal position
color hue

mark: point



4:
vertical position
horizontal position
color hue
size (area)

mark: point

Channels

Position on common scale



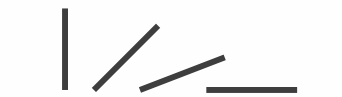
Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



Depth (3D position)



Color luminance



Color saturation



Curvature



Volume (3D size)



Same

Spatial region



Color hue



Motion



Shape



Channels: Rankings

➔ Magnitude Channels: Ordered Attributes



➔ Identity Channels: Categorical Attributes



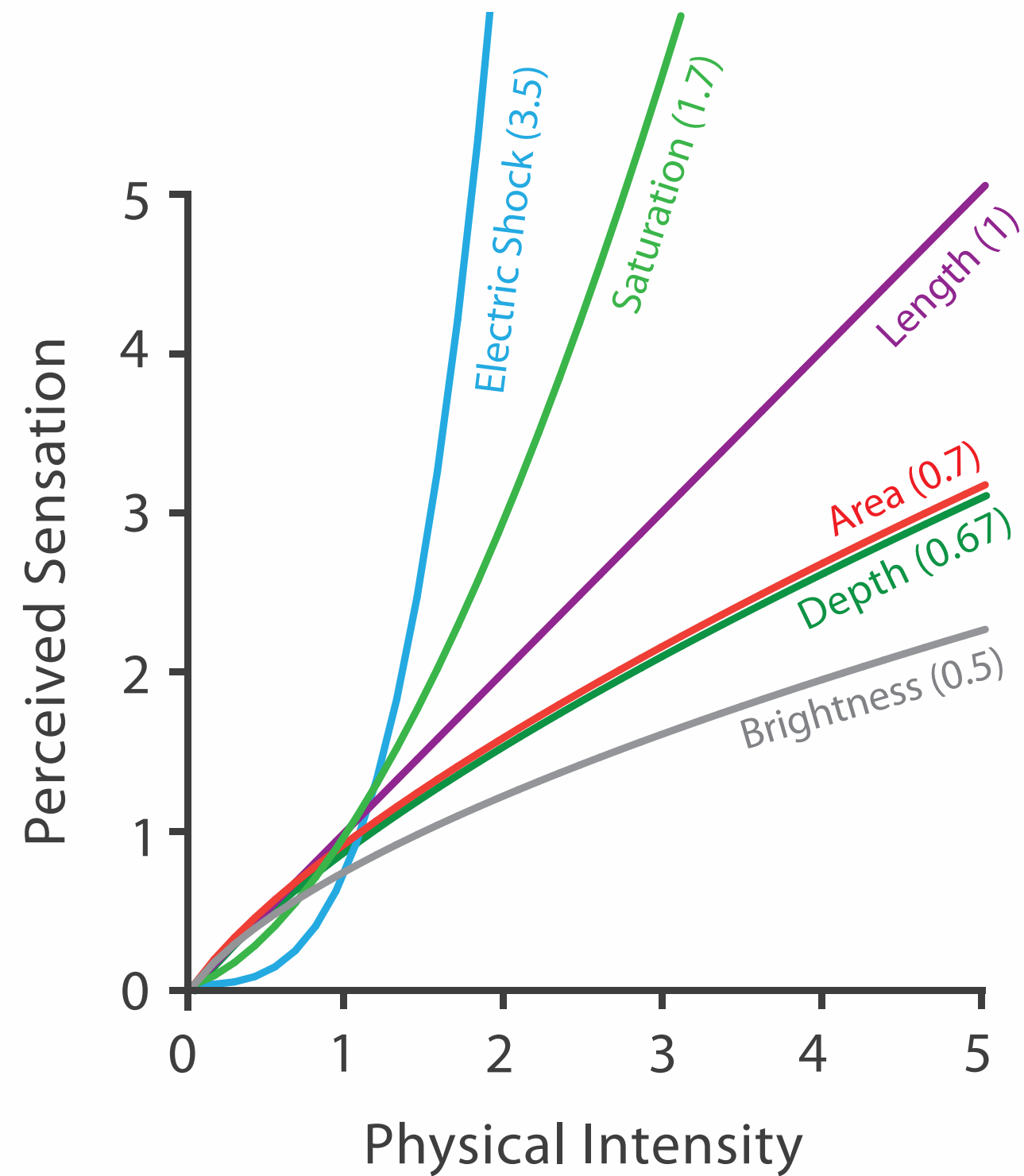
Best
Effectiveness
Least

Same
Same

- effectiveness principle
 - encode most important attributes with highest ranked channels
- expressiveness principle
 - match channel and data characteristics

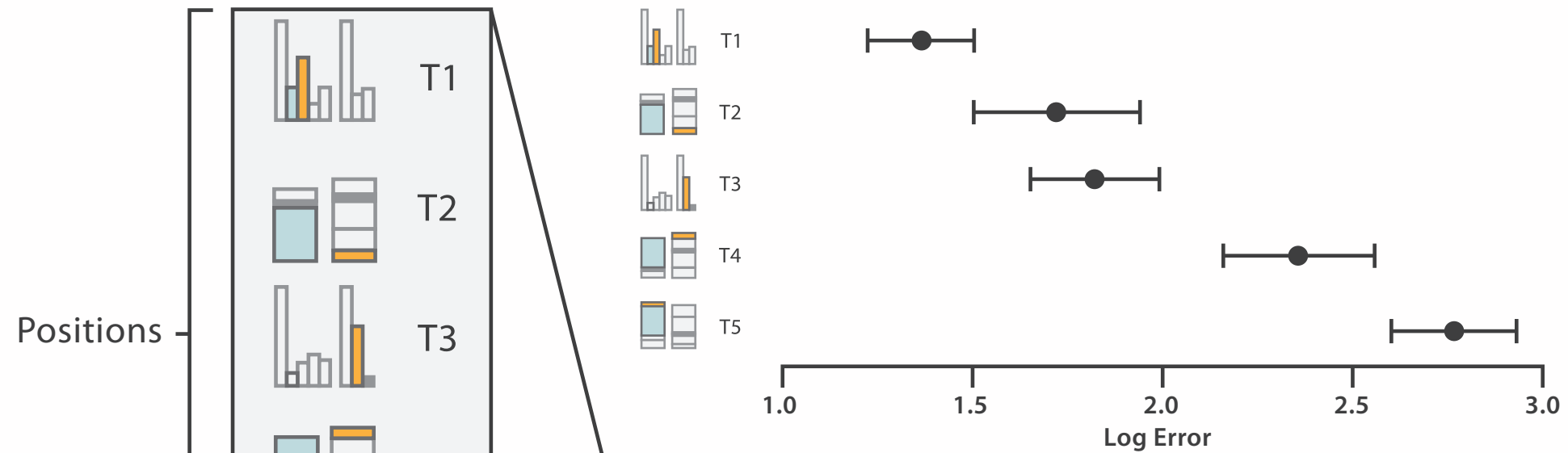
Accuracy: Fundamental Theory

Steven's Psychophysical Power Law: $S = I^N$

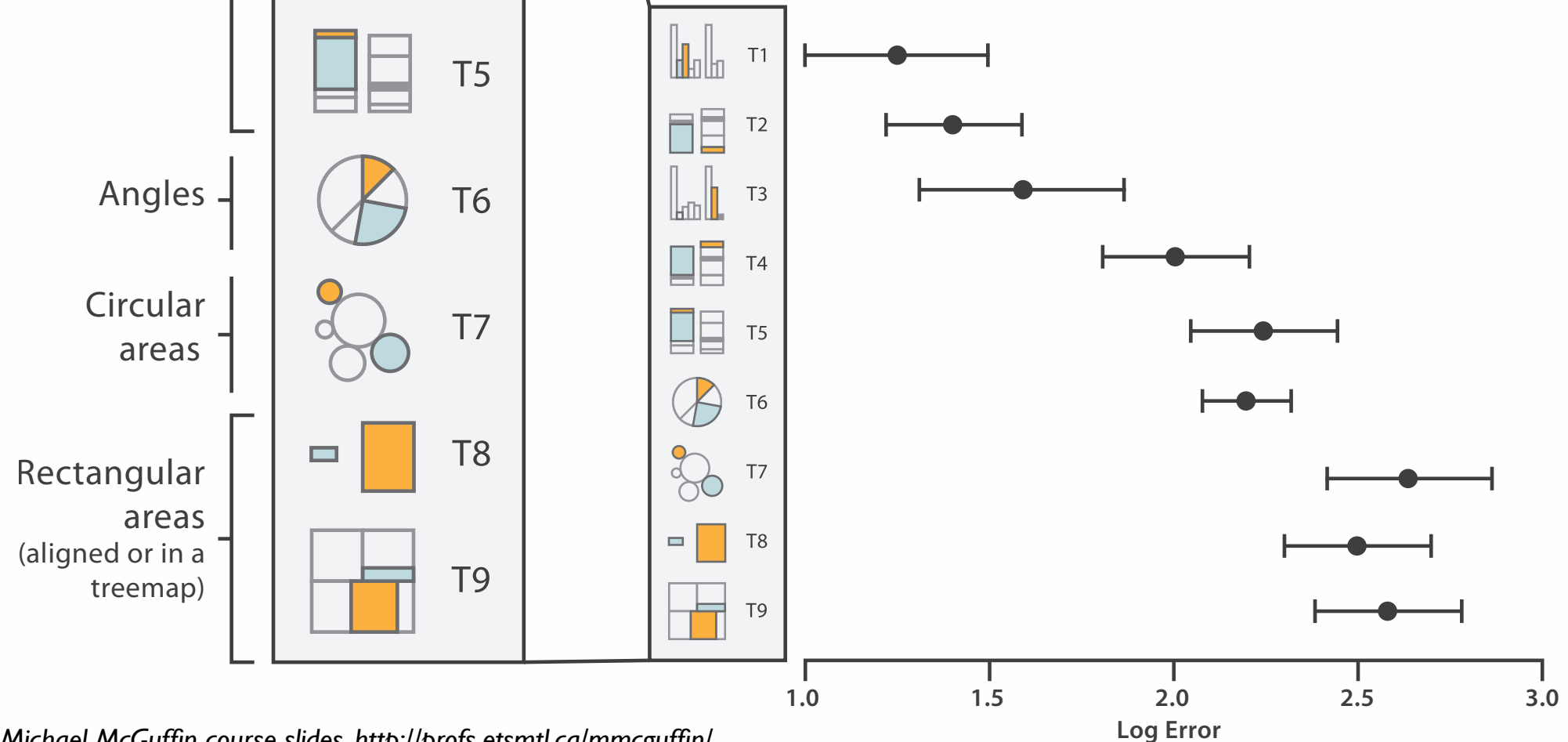


Accuracy: Vis experiments

Cleveland & McGill's Results



Crowdsourced Results



[Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. Heer and Bostock. Proc ACM Conf. Human Factors in Computing Systems (CHI) 2010, p. 203–212.]

How to encode: Arrange position and region

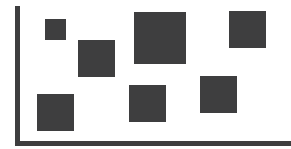
Encode

→ Arrange

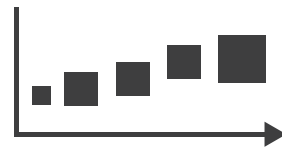
→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...

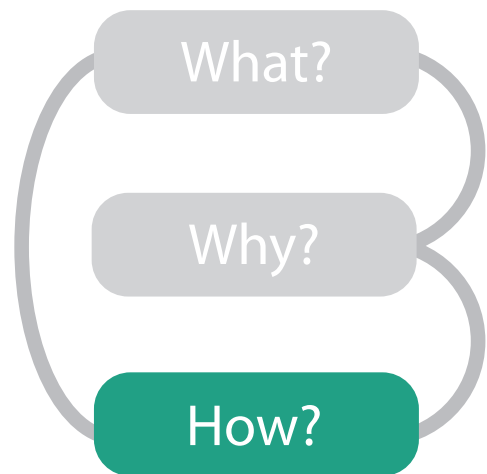


→ Shape



→ Motion

Direction, Rate, Frequency, ...



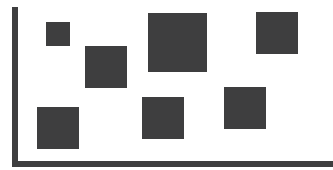
Arrange tables

② Express Values

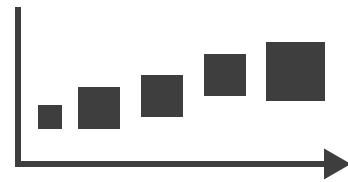


② Separate, Order, Align Regions

→ Separate



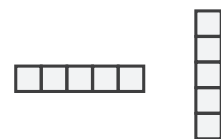
→ Order



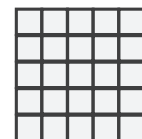
→ Align



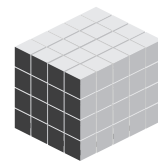
→ 1 Key
List



→ 2 Keys
Matrix



→ 3 Keys
Volume

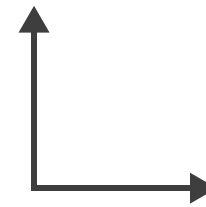


→ Many Keys
Recursive Subdivision

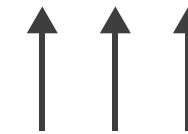


② Axis Orientation

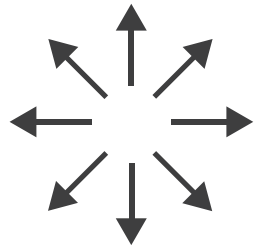
→ Rectilinear



→ Parallel

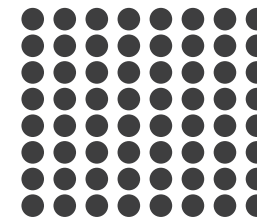


→ Radial



② Layout Density

→ Dense



→ Space-Filling



Idioms: dot chart, line chart

- one key, one value

- data

- 2 quant attribs

- mark: points

- dot plot: + line connection marks between them

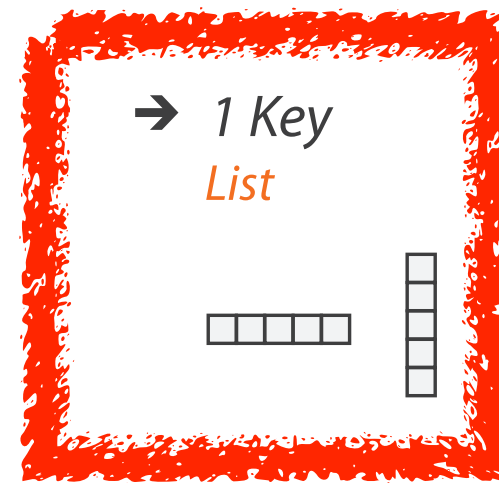
- channels

- aligned lengths to express quant value
- separated and ordered by key attrib into horizontal regions

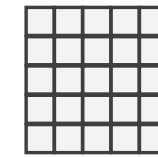
- task

- find trend

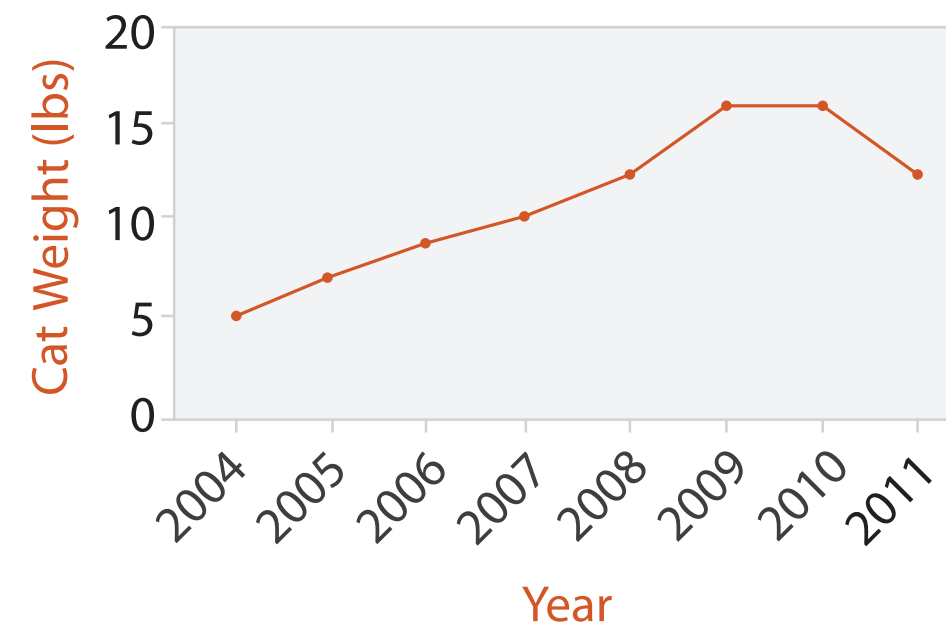
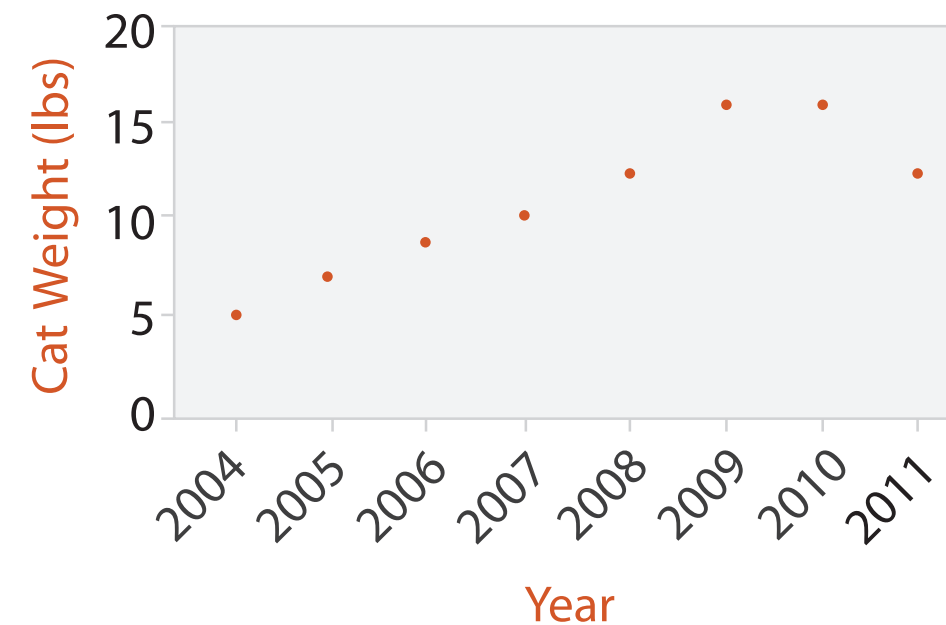
- connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next



→ 2 Keys
Matrix

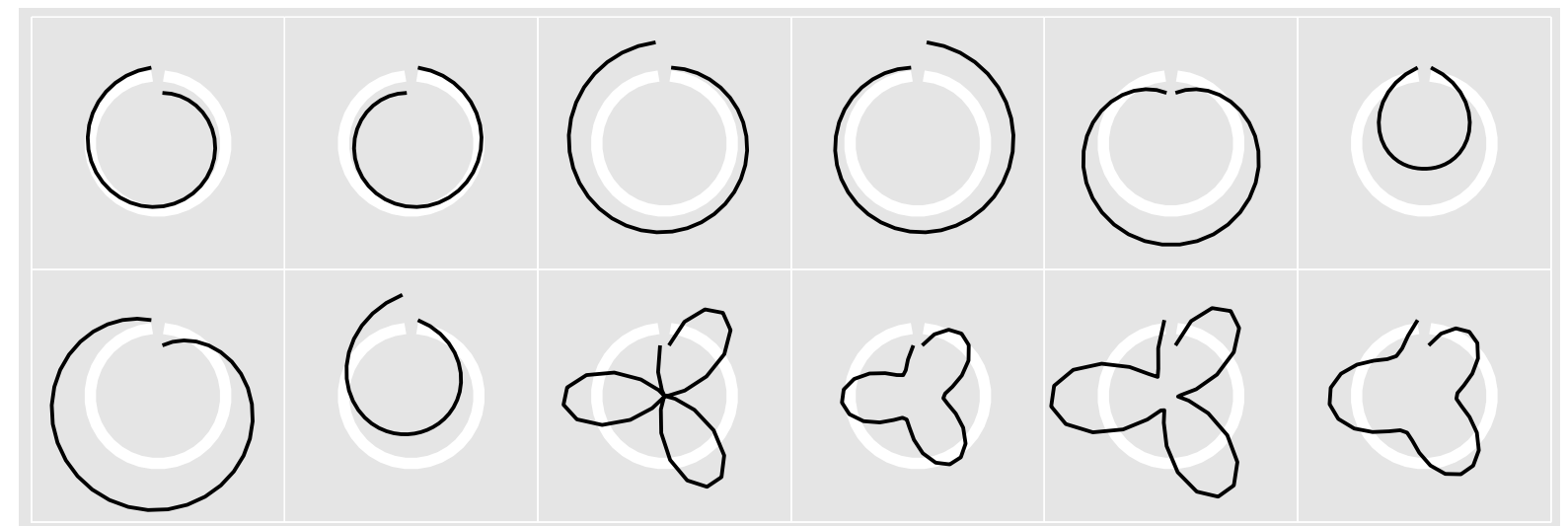
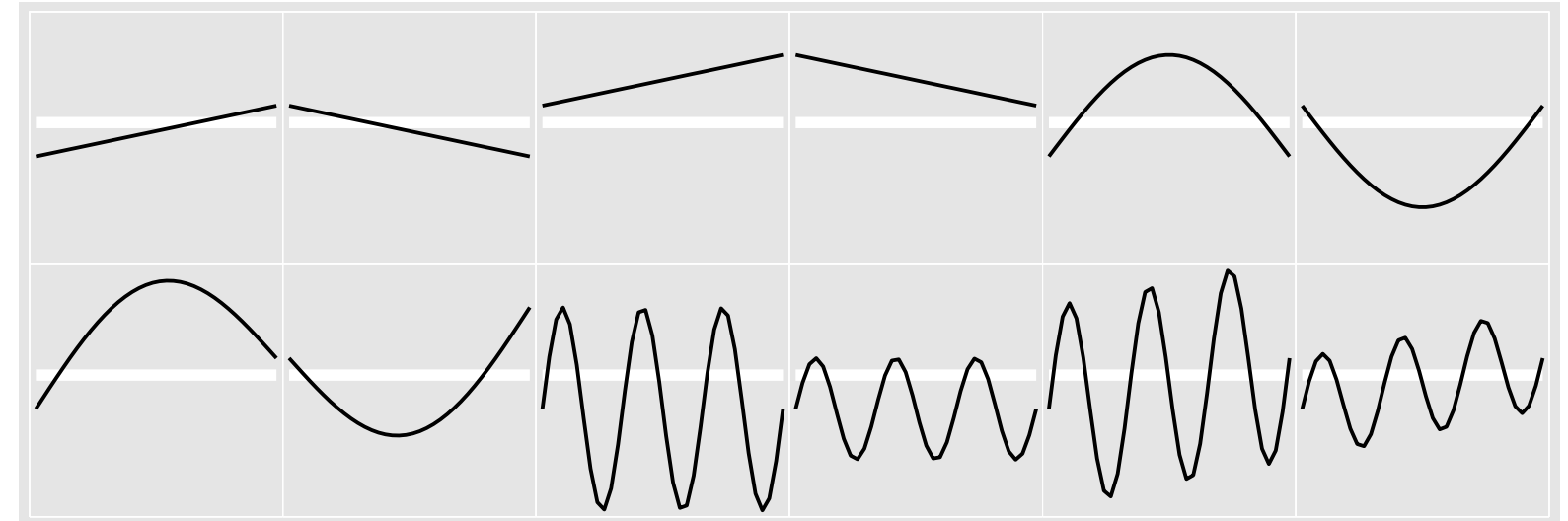


→ Many Keys
Recursive Subdivision

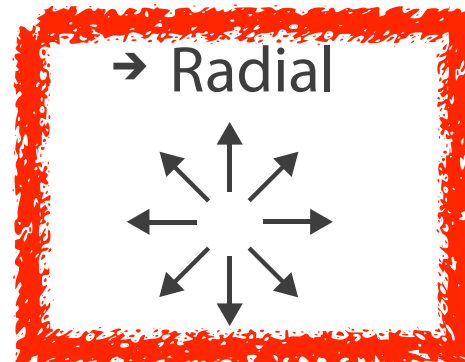
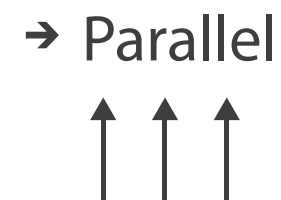
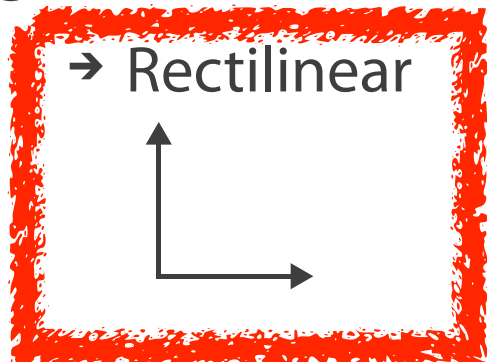


Idiom: glyphmaps

- rectilinear good for linear vs nonlinear trends
- radial good for cyclic patterns



➔ Axis Orientation



[Glyph-maps for Visually Exploring Temporal Patterns in Climate Data and Models. Wickham, Hofmann, Wickham, and Cook. *Environmetrics* 23:5 (2012), 382–393.]

Idiom: heatmap

- two keys, one value

- data

- 2 categ attribs (gene, experimental condition)
- 1 quant attrib (expression levels)

- marks: area

- separate and align in 2D matrix
 - indexed by 2 categorical attributes

- channels

- color by quant attrib
 - (ordered diverging colormap)

- task

- find clusters, outliers

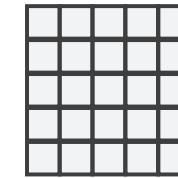
- scalability

- 1M items, 100s of categ levels, ~10 quant attrib levels

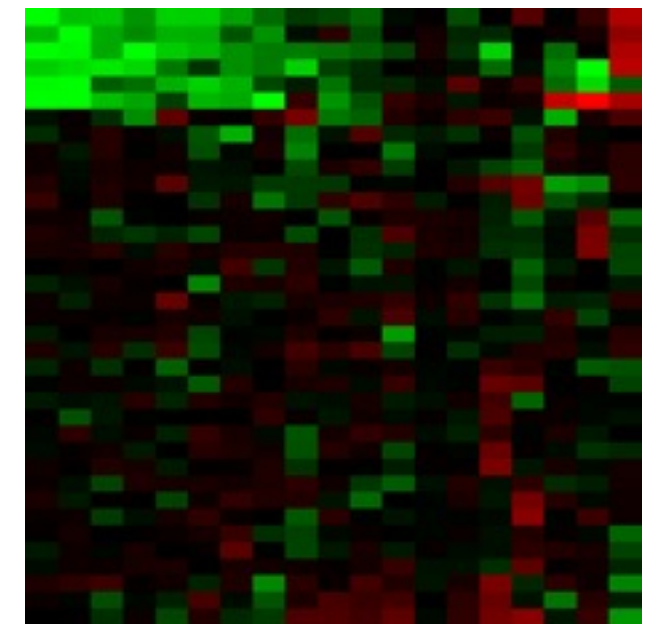
→ 1 Key
List



→ 2 Keys
Matrix

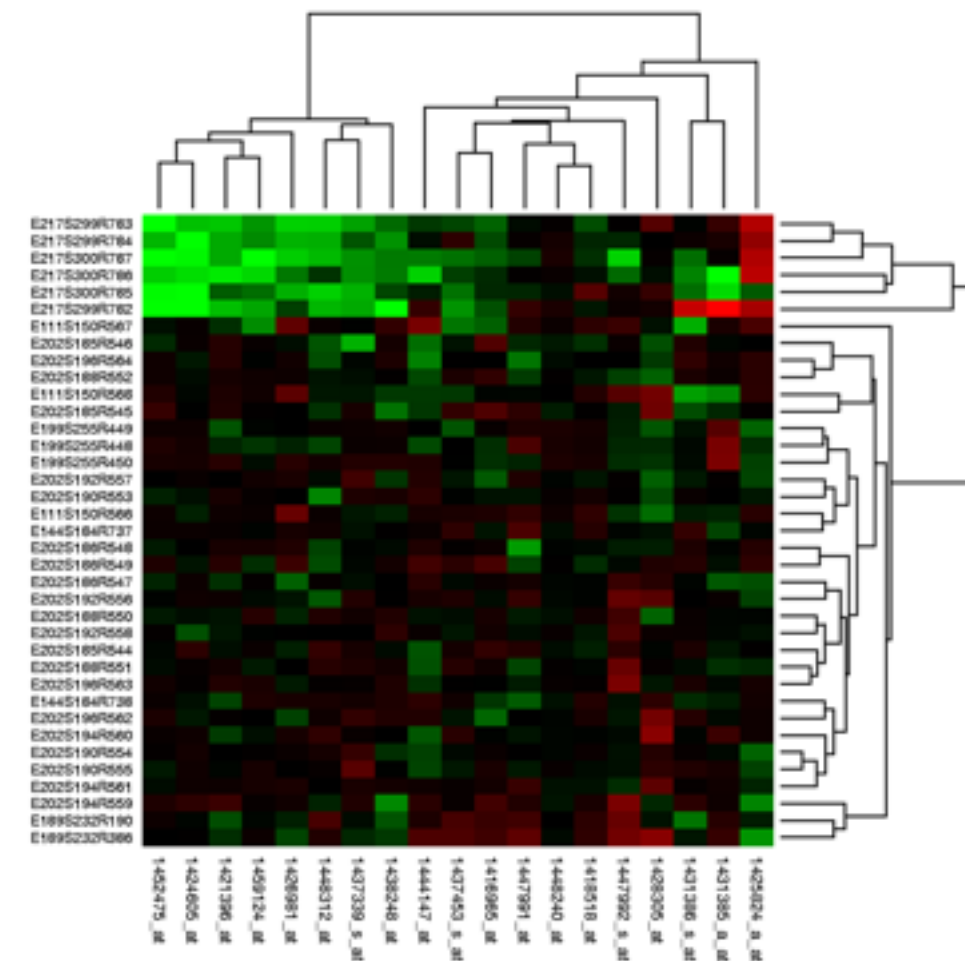


→ Many Keys
Recursive Subdivision



Idiom: cluster heatmap

- in addition
 - derived data
 - 2 cluster hierarchies
 - dendrogram
 - parent-child relationships in tree with connection line marks
 - leaves aligned so interior branch heights easy to compare
 - heatmap
 - marks (re-)ordered by cluster hierarchy traversal



Arrange spatial data

→ Use Given

→ Geometry

→ *Geographic*

→ *Other Derived*

→ Spatial Fields

→ *Scalar Fields (one value per cell)*

→ *Isocontours*

→ *Direct Volume Rendering*

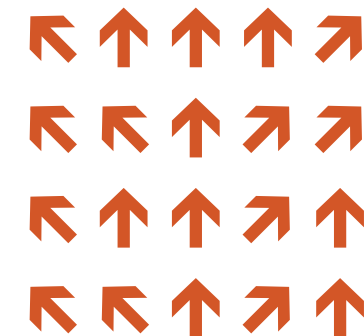
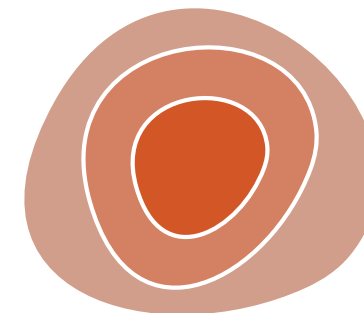
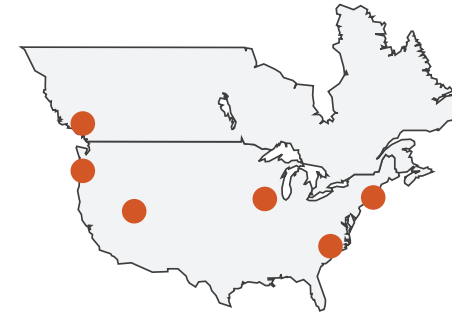
→ *Vector and Tensor Fields (many values per cell)*

→ *Flow Glyphs (local)*

→ *Geometric (sparse seeds)*

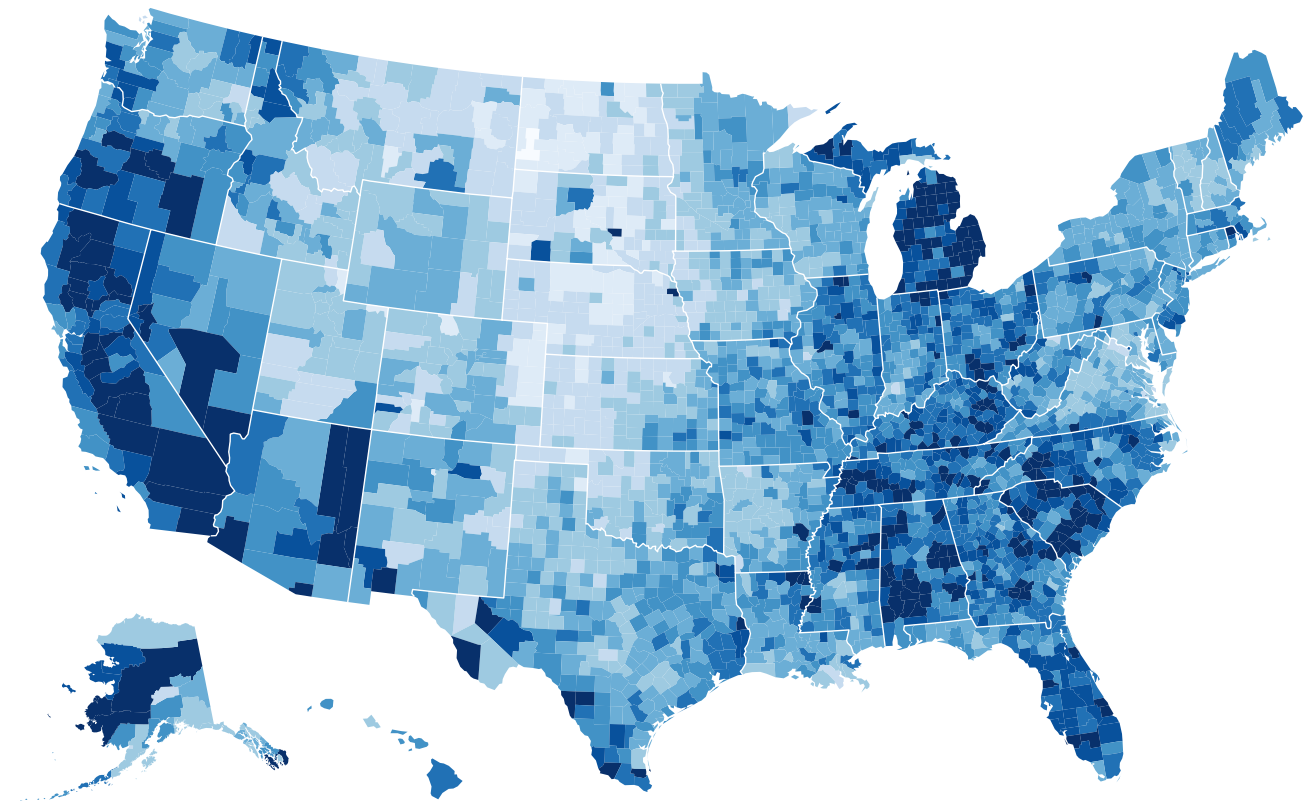
→ *Textures (dense seeds)*

→ *Features (globally derived)*



Idiom: **choropleth map**

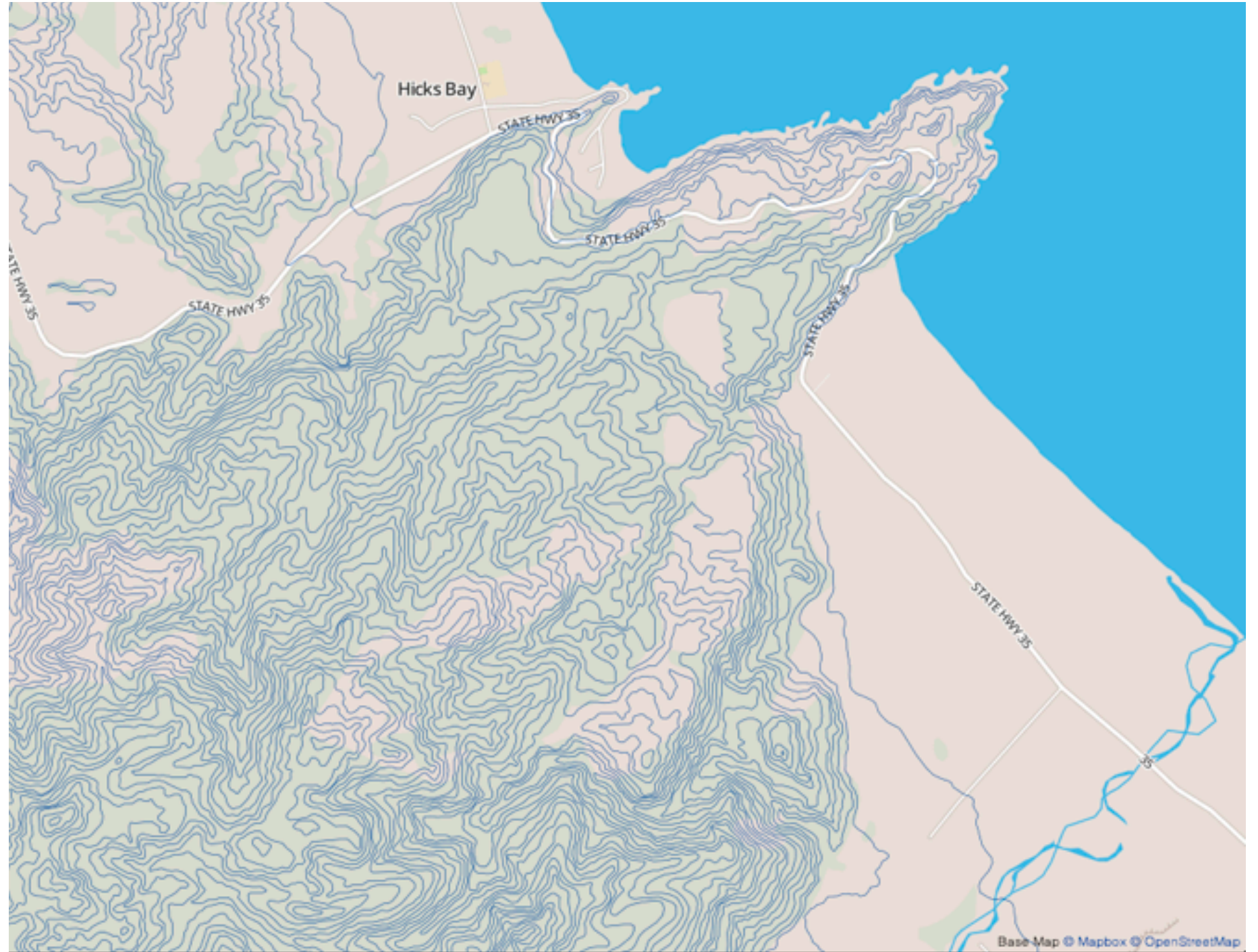
- **use** given spatial data
 - when central task is understanding spatial relationships
- data
 - geographic geometry
 - table with 1 quant attribute per region
- encoding
 - use given geometry for area mark boundaries
 - sequential segmented colormap



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

Idiom: **topographic map**

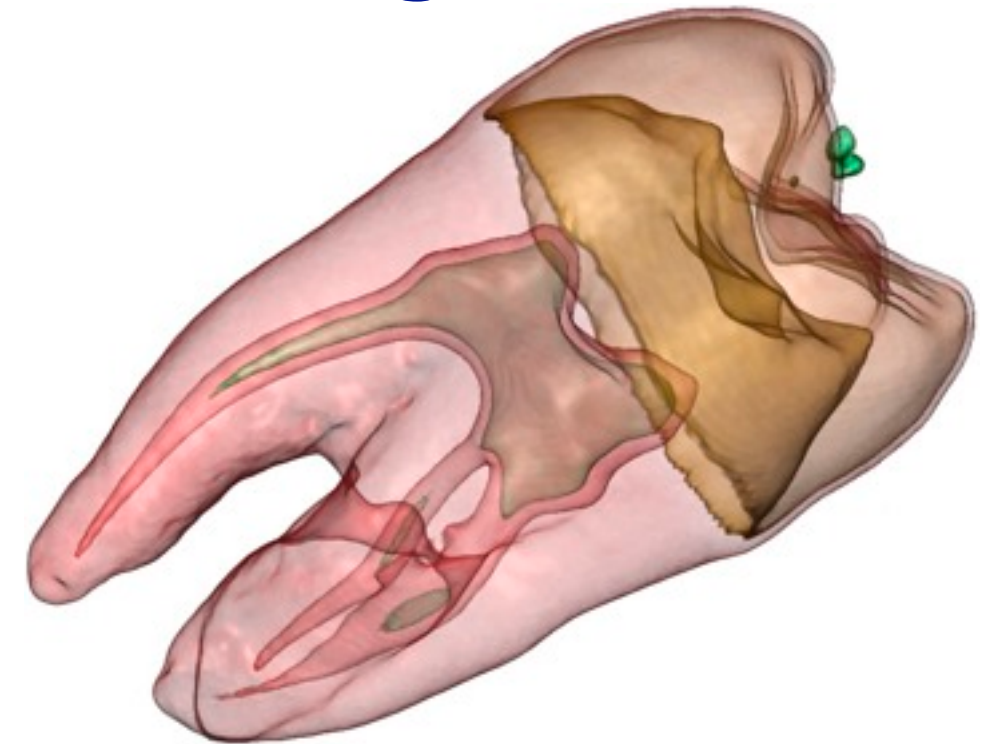
- data
 - geographic geometry
 - scalar spatial field
 - 1 quant attribute per grid cell
- derived data
 - isoline geometry
 - isocontours computed for specific levels of scalar values



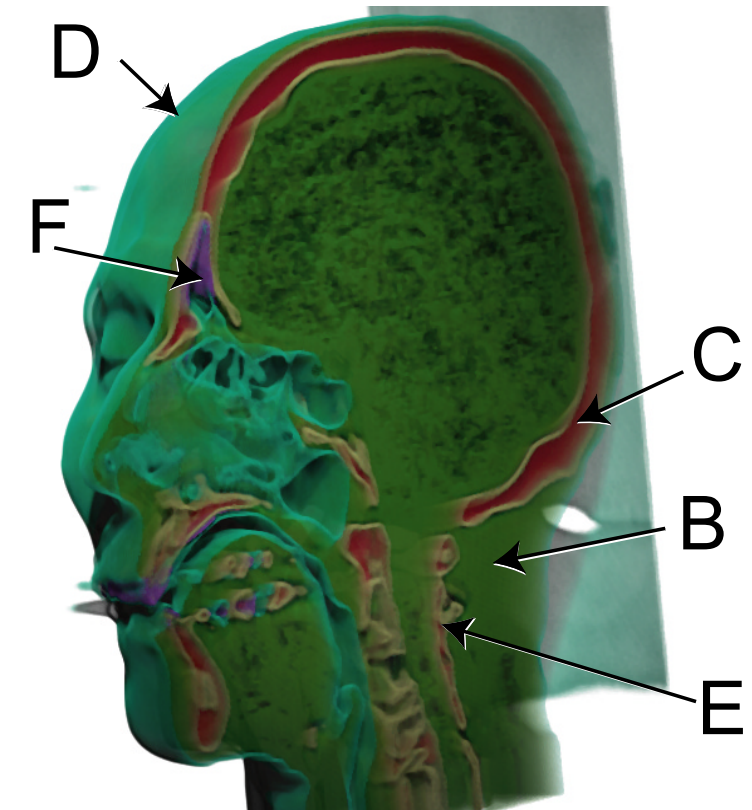
Land Information New Zealand Data Service

Idioms: **isosurfaces**, **direct volume rendering**

- data
 - scalar spatial field
 - 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



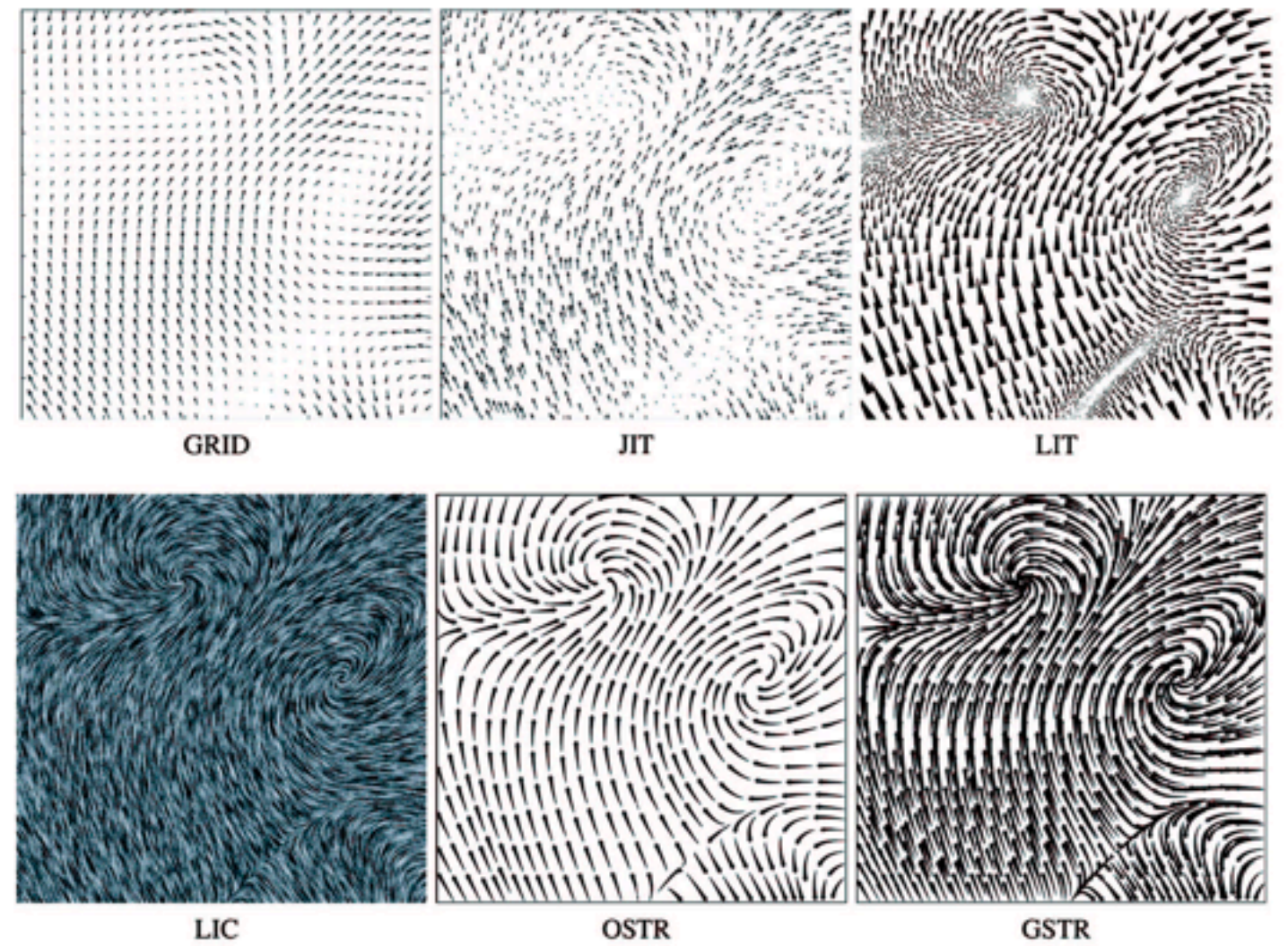
[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]



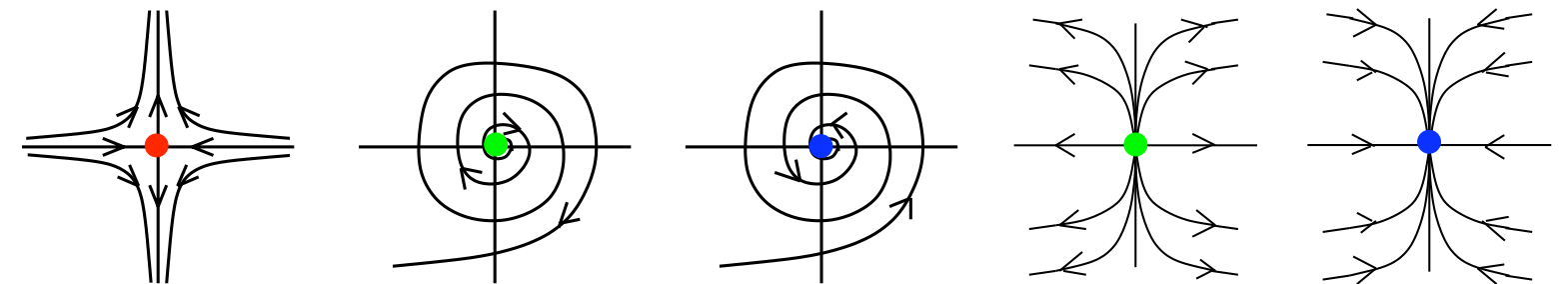
[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]

Idioms: **vector glyphs**

- 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)



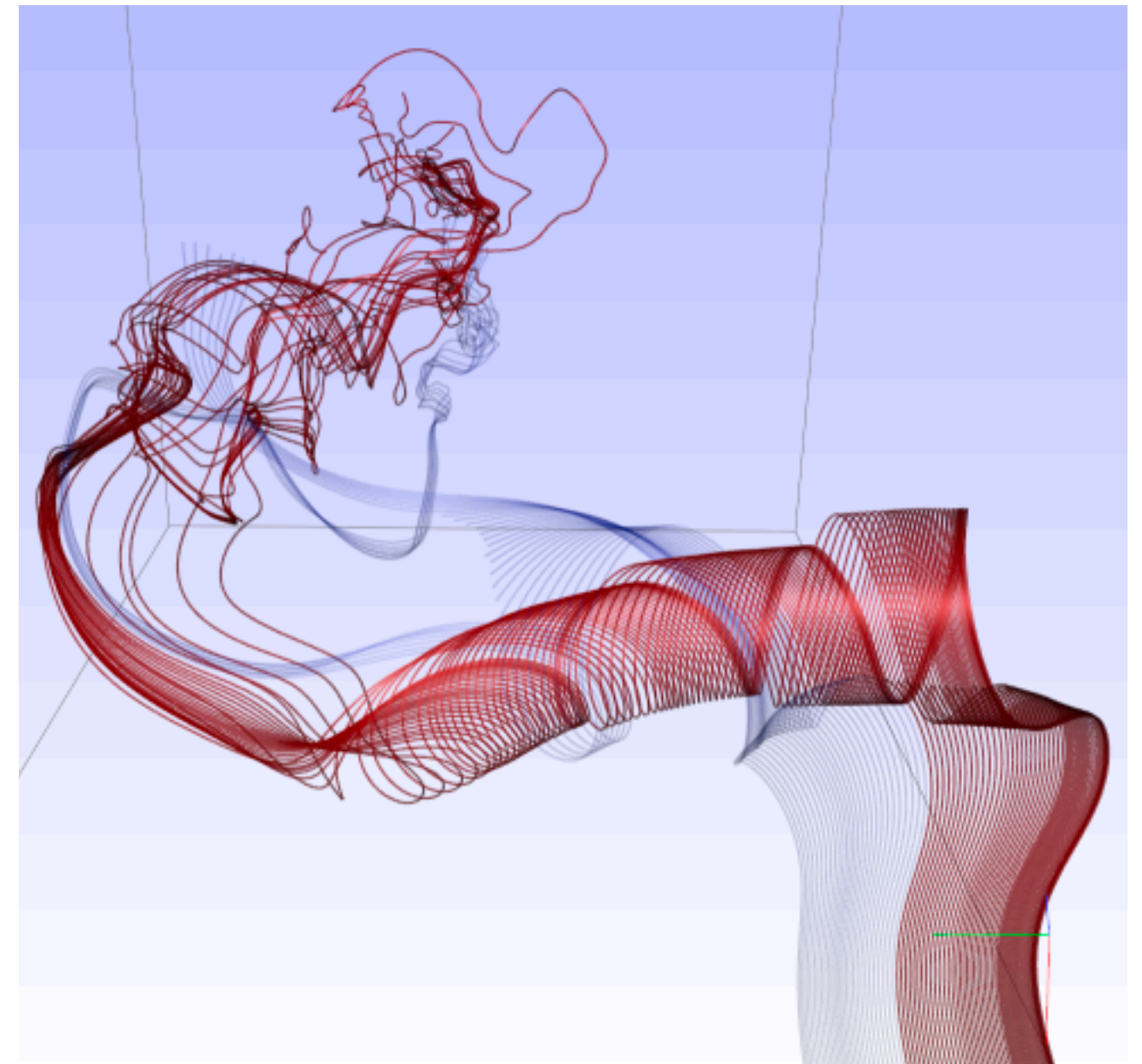
[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. *IEEE Trans. Visualization and Computer Graphics (TVCG)* 11:1 (2005), 59–70.]



[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



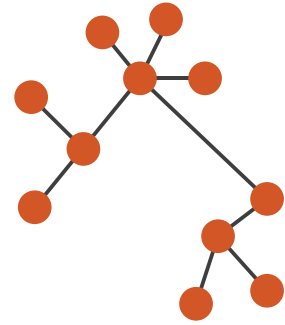
[Similarity Measures for Enhancing Interactive Streamline Seeding. McLoughlin, Jones, Laramee, Malki, Masters, and Hansen. IEEE Trans. Visualization and Computer Graphics 19:8 (2013), 1342–1353.]

Arrange networks and trees

→ Node–Link Diagrams Connection Marks

✓ NETWORKS

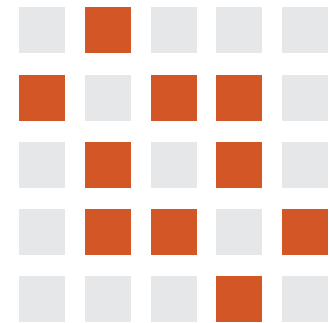
✓ TREES



→ Adjacency Matrix Derived Table

✓ NETWORKS

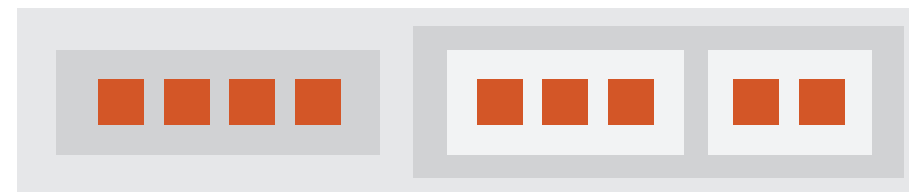
✓ TREES



→ Enclosure Containment Marks

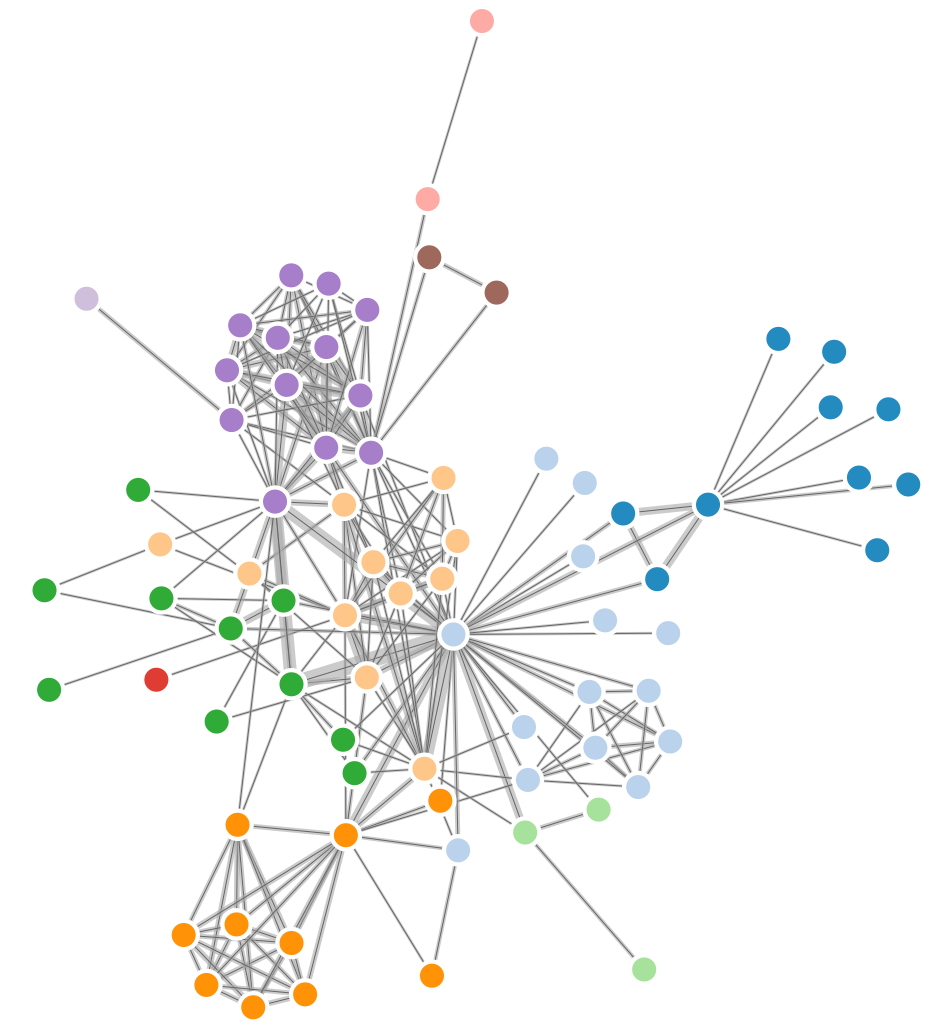
✗ NETWORKS

✓ TREES



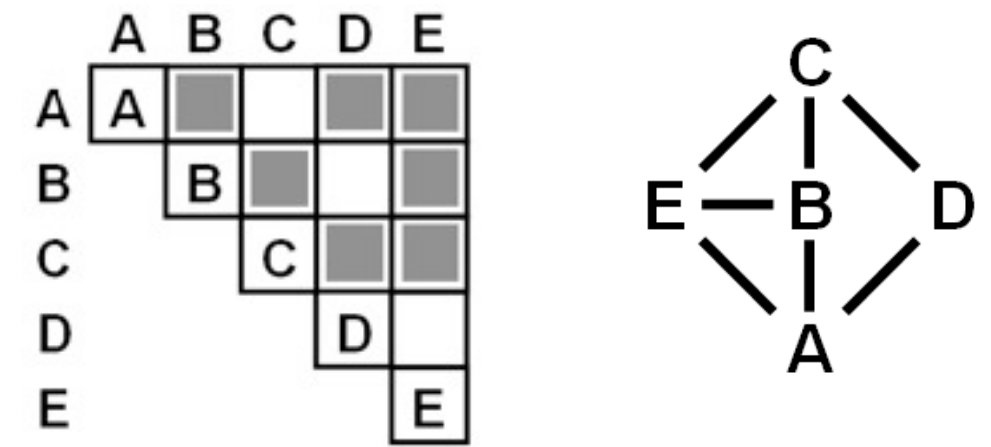
Idiom: **force-directed placement**

- 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 $E < 4N$



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.]

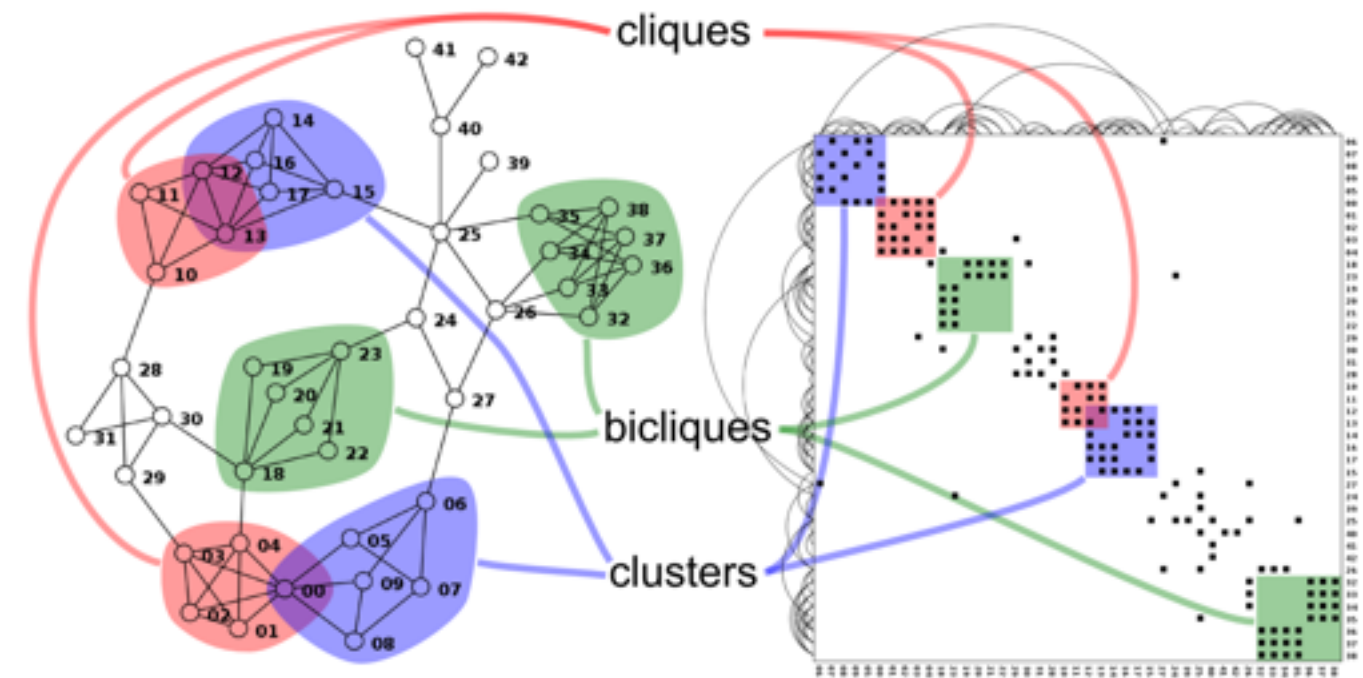


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

Connection vs. adjacency comparison

- adjacency matrix strengths
 - predictability, scalability, supports reordering
 - some topology tasks trainable
- node-link diagram strengths
 - topology understanding, path tracing
 - intuitive, no training needed
- empirical study
 - node-link best for small networks
 - matrix best for large networks
 - if tasks don't involve topological structure!

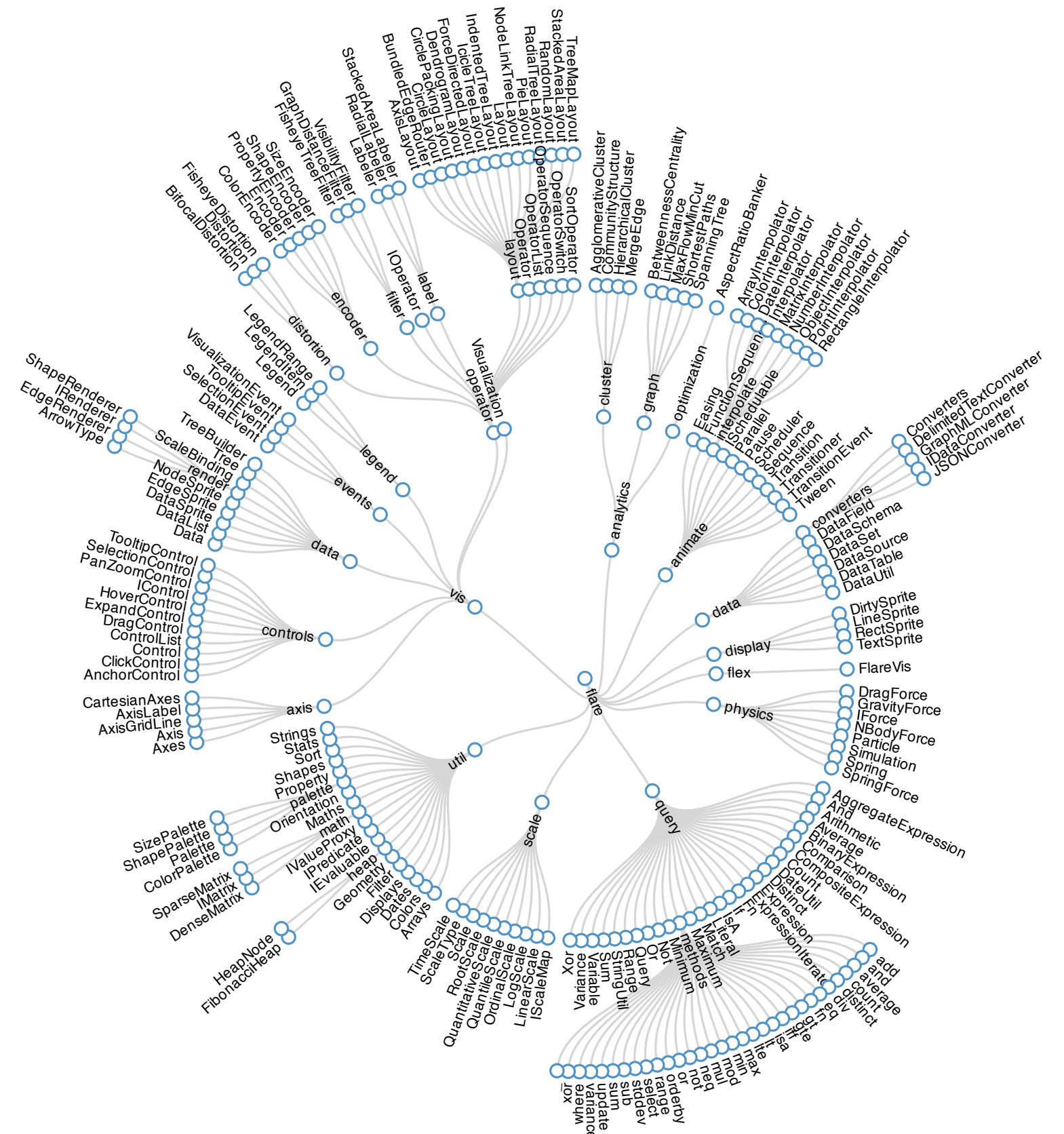
[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]



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

Idiom: 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



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
- **scalability**
 - 1M leaf nodes

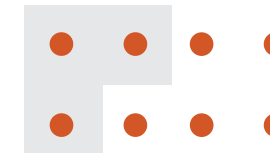


http://tulip.labri.fr/Documentation/3_7/userHandbook/html/ch06.html

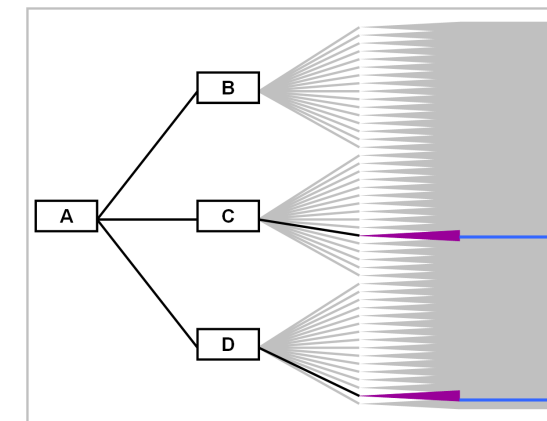
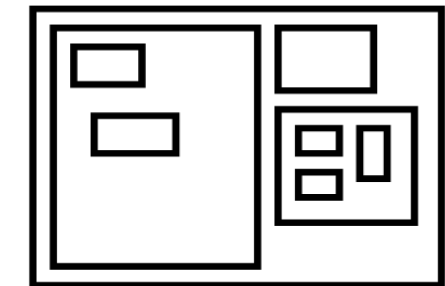
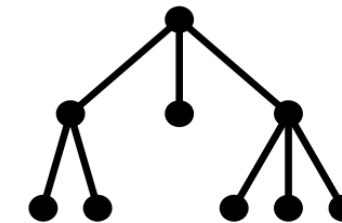
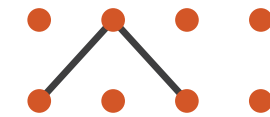
Connection vs. containment comparison

- 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

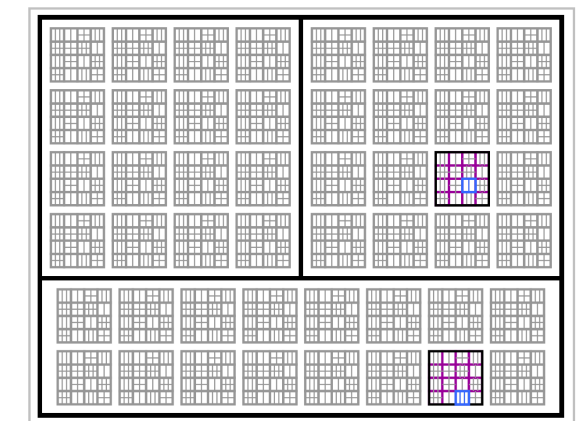
➔ Containment



➔ Connection



Node-Link Diagram



Treemap

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

How to encode: Mapping color

Encode

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...

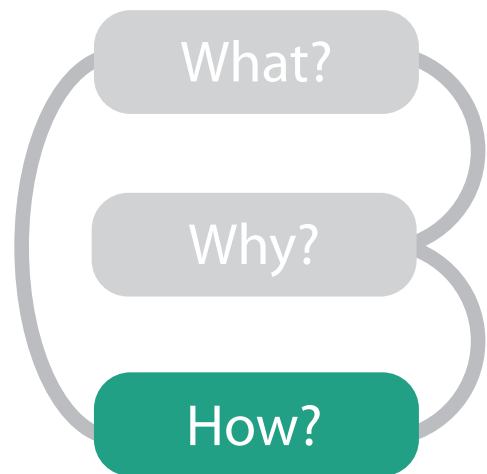


→ Shape



→ Motion

Direction, Rate, Frequency, ...



Color: Luminance, saturation, hue

- 3 channels

- identity for categorical

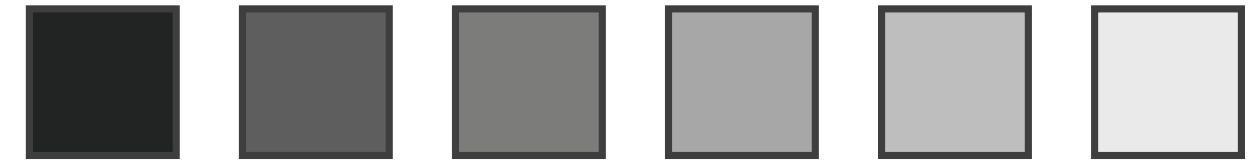
- hue

- magnitude for ordered

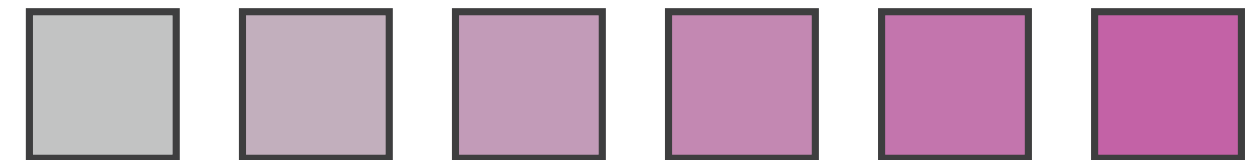
- luminance
- saturation

- better match for visual encoding than RGB color space from graphics

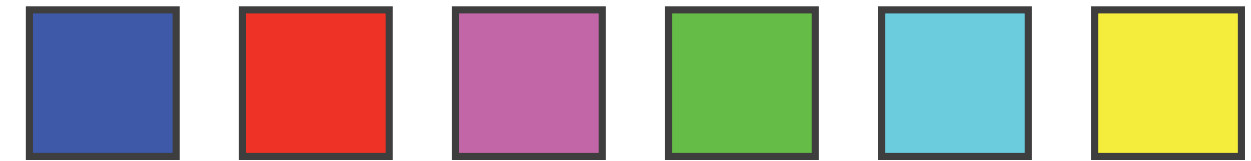
Luminance



Saturation

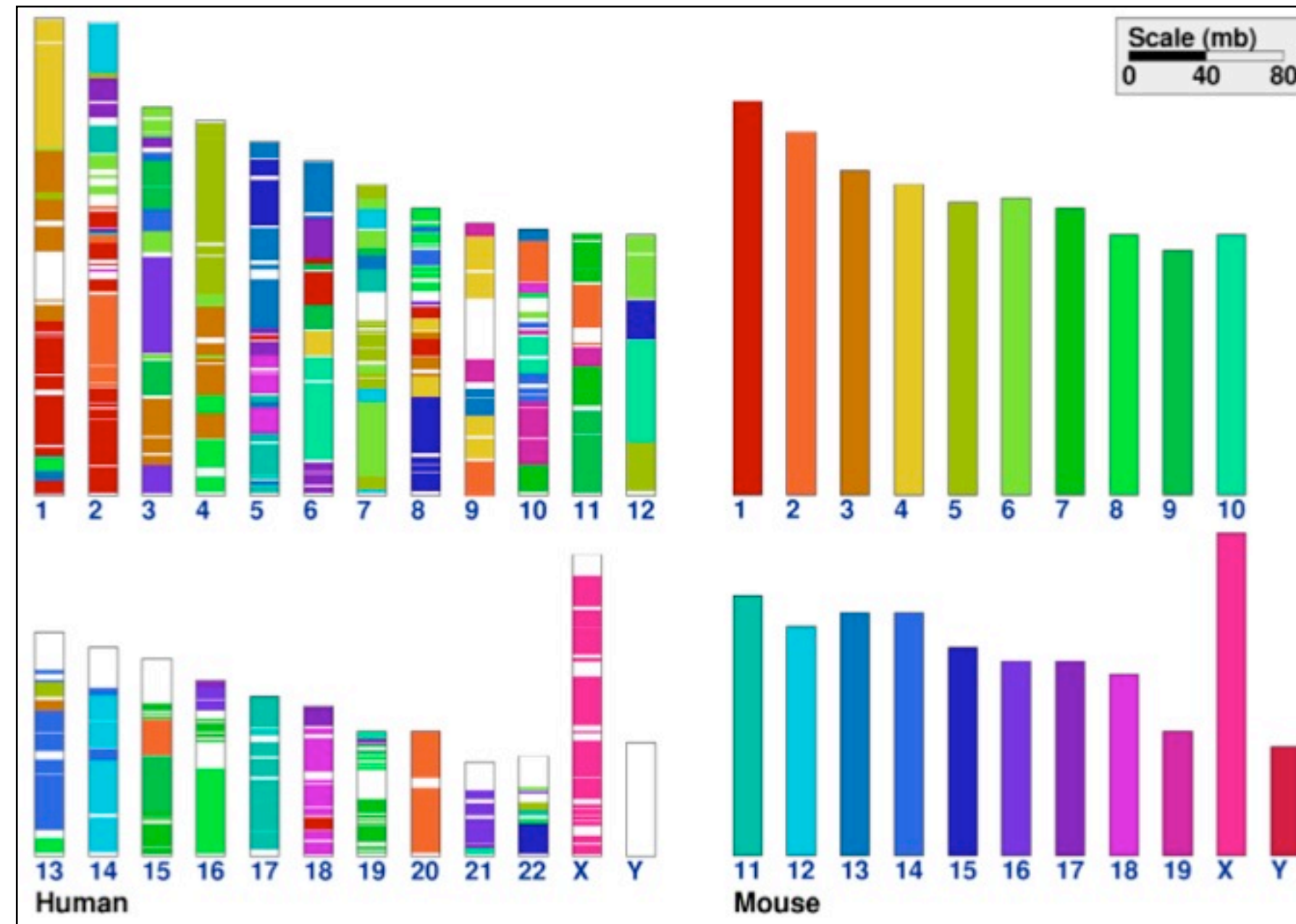


Hue



Categorical color: Discriminability constraints

- noncontiguous small regions of color: only 6-12 bins



[Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. BMC Bioinformatics, 8:82, 2007.]

What?

Datasets

Attributes

domain

abstraction

What?

Why?

idiom

How?

algorithm

Why?

Actions

Targets

→ Data Types

→ Items

→ Data and D

Tables

Items

Attributes

→ Analyze

→ Consume

→ Discover



→ Present



→ Enjoy



→ All Data

→ Trends



→ Outliers

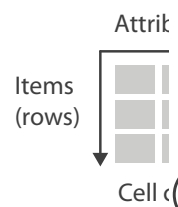


→ Features



→ Dataset Typ

→ Tables



→ Produce

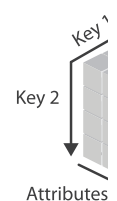
→ Annotate



→ Search

	Tar
Location known	••
Location unknown	<••

→ Multidir



→ Geometr



→ Query

→ Identify



How?

Encode

Manipulate

Facet

Reduce

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



→ Motion

Direction, Rate, Frequency, ...



→ Change



→ Select



→ Navigate



→ Juxtapose



→ Partition



→ Superimpose



→ Filter



→ Aggregate



→ Embed



What?

Why?

More Information

- this talk

<http://www.cs.ubc.ca/~tmm/talks.html#vad15dagstuhl>

- book page (including tutorial lecture slides)

<http://www.cs.ubc.ca/~tmm/vadbook>

– 20% promo code for book+ebook combo:
HVN17

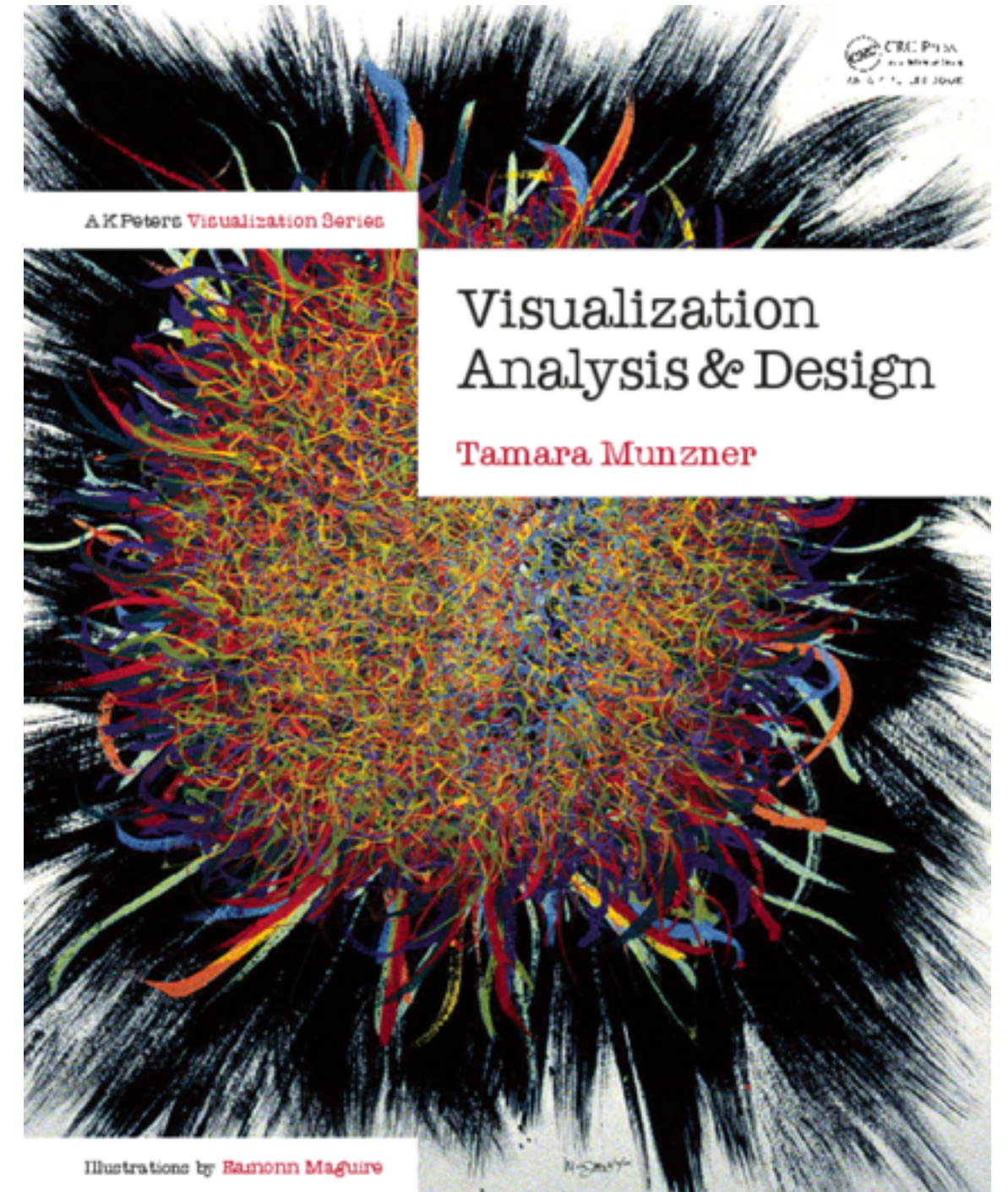
– <http://www.crcpress.com/product/isbn/9781466508910>

– illustrations: Eamonn Maguire

- papers, videos, software, talks, full courses

<http://www.cs.ubc.ca/group/infovis>

<http://www.cs.ubc.ca/~tmm>



Visualization Analysis and Design.
Munzner. A K Peters Visualization Series, CRC Press, Visualization Series, 2014.