Information Visualisation (4) Tables, Spatial data, Networks and Trees

Prof. Bruno Dumas

UNIVERSITÉ DE NAMUR

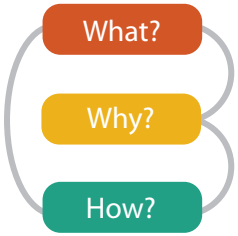
Course Outline

- Introduction to Information Visualisation
- Data, Task and Validation
- Marks and Channels; Color Mapping
- Tables, Spatial data, Networks and Trees
- Manipulating View, Facetting, Focus + Context
- Reduce Items and Attributes + Some Cases Analysis

Program for Today

- Catalogue of vis idioms
- Arranging tables
  - Express quantitative values
  - Express categories: separate, order, align
- Arranging spatial data
  - Geographic data
  - Scalar fields
- Arranging networks and trees
  - Node-link layouts
  - Matrix layouts
  - Containment: hierarchy marks

Catalogue of Vis Idioms



- For each idiom, summarising table
  - What: kind of data? Quantitative, ordered categorical? Key, value?
    - Derived data?
  - How: main characteristics of the vis idiom
  - Why: which tasks are best supported by each idiom
  - Scale: how many items can safely be displayed

Idiom	
What: Data	
How: Encode	
Why: Task	
Scale	

Arranging Tables: Cheat Sheet

Arrange Tables

- **Express Values**

- **Separate, Order, Align Regions**
  - Separate

  - Order

  - Align

    - *1 Key* — *List*

    - *2 Keys* — *Matrix*

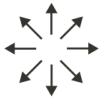
    - *3 Keys* — *Volume*

    - *Many Keys* — *Recursive Subdivision*

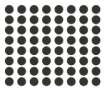
- **Axis Orientation**
  - Rectilinear

  - Parallel

  - Radial



- **Layout Density**
  - Dense

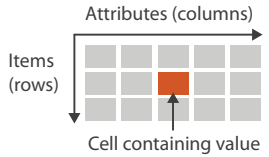


  - Space-Filling

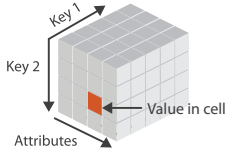
Arranging Tables

- Key: independent attribute used as a unique index to look up items in a table
  - Categorical or ordinal
- Value: dependent attribute: value of a cell
  - Categorical, ordinal, quantitative
- Design choices typically related to the semantics of the table's attributes
  - How many keys, how many values?
  - What do you want to put the focus on?

➔ Tables

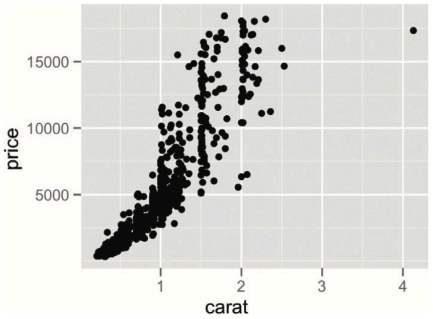


➔ *Multidimensional Table*

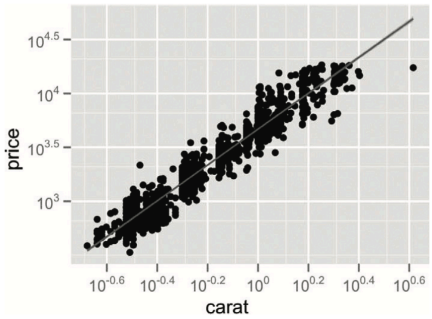


Express Quantitative Values: Scatterplots

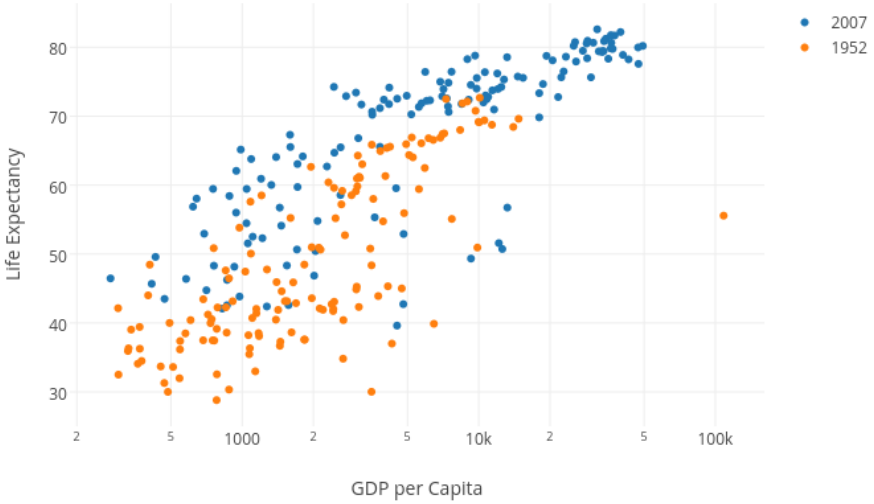
- Attribute mapped to a spatial position along an axis
- Mark can encode more attributes with color, size, shape…
- Scatterplots:
  - Effective for overviews and spotting distributions
  - Finding outliers and extreme values
  - Correlations
  - Axes can be manipulated (logs...)



(a)

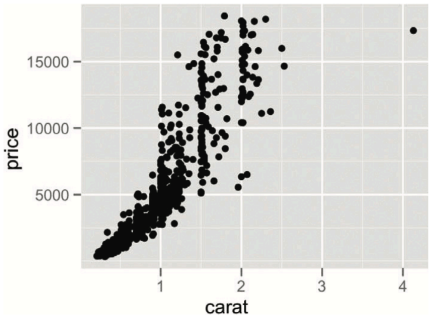


(b)

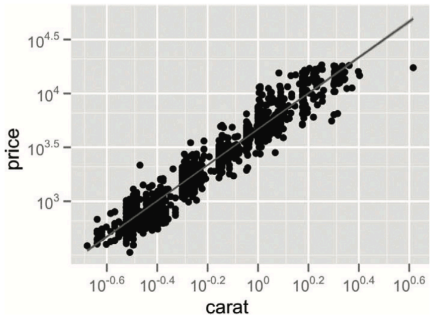


Scatterplots

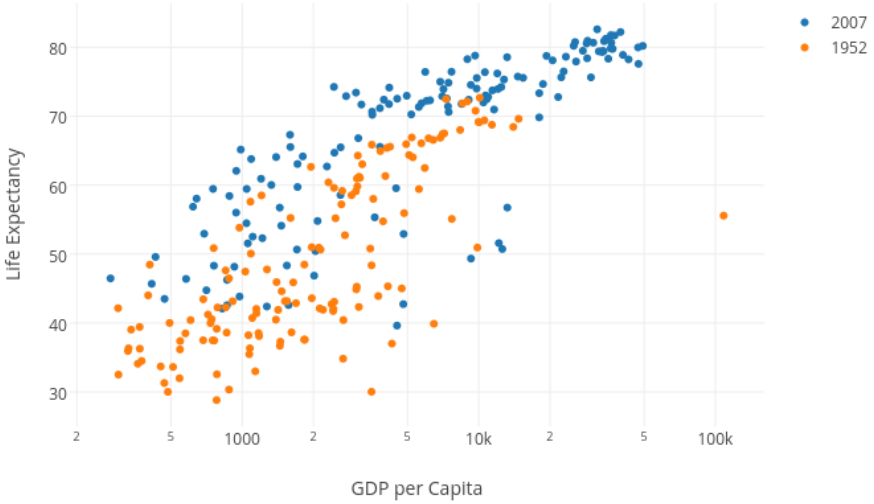
Idiom	**Scatterplots**
What: Data	Table: two quantitative value attributes
How: Encode	Express values with horizontal and vertical spatial position and point marks
Why: Task	Find trends, outliers, distribution, correlation; locate clusters
Scale	Items: hundreds



(a)



(b)

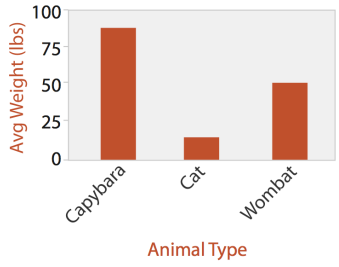


Separate, Order, Align

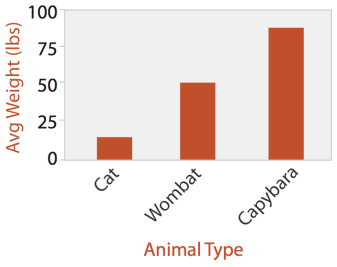
- How to encode categorical attributes?
  - Spatial position implies ordering... (violation of expressiveness principle)
- Defining regions
  - Regions: contiguous bounded areas that are distinct from each other
  - Use of spatial proximity -> expressiveness principle
- How to distribute these regions on a plane?
  1. Separating into regions
  2. Aligning the regions
  3. Ordering the regions

Separate, Order, Align: Bar Charts, Stacked Bar Charts, Streamgraphs, Dot/Line Charts

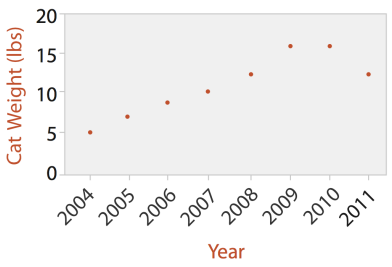
- One key = one region per item
- Typical arrangement: list arrangement
  - One dimension for the keys, one for the values
- Arrangement shared by bar charts, stacked bar charts, streamgraphs and dot/line charts



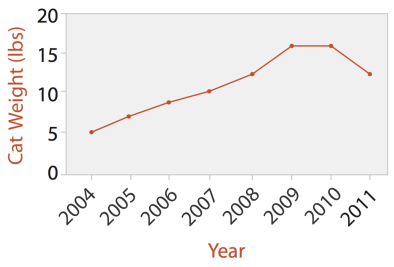
(a)



(b)



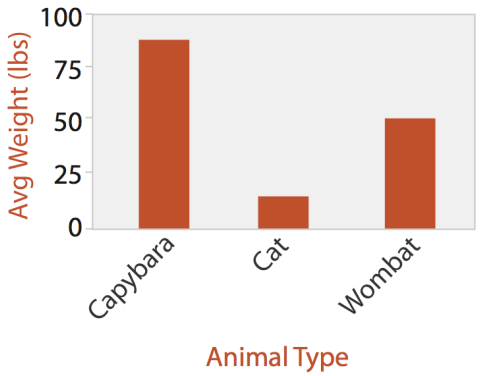
(a)



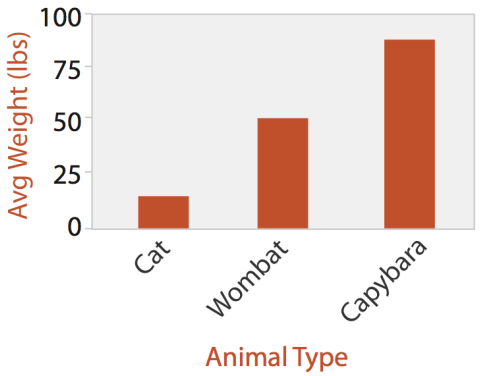
(b)

Bar Charts

Idiom	**Bar Charts**
What: Data	Table: one quantitative value attribute, one categorical key attribute
How: Encode	Line marks, express value attribute with aligned vertical position, separate key attribute with horizontal position
Why: Task	Lookup and compare values
Scale	Key attribute: dozens to hundreds of levels



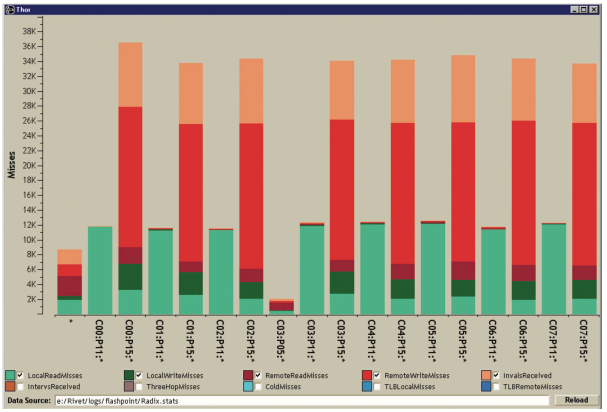
(a)



(b)

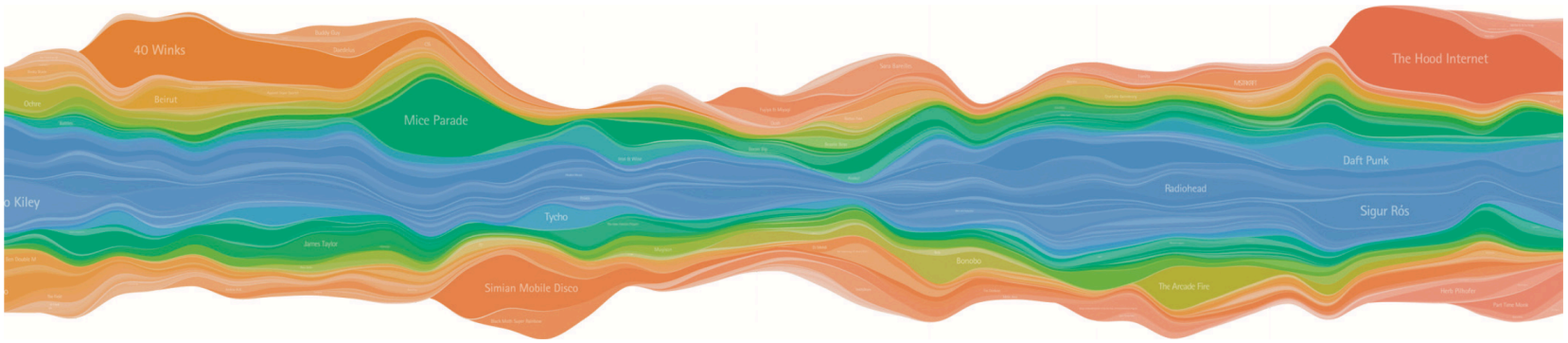
Stacked Bar Charts

Idiom	**Stacked Bar Charts**
What: Data	Multidimensional table: one quantitative value attribute, two categorical key attributes
How: Encode	Bar glyph with length-coded subcomponents of value attribute for each category of secondary key attribute. Separate bars by category of primary key attribute.
Why: Task	Part-to-whole relationship, lookup values, find trends
Scale	Key attribute (main axis): dozens to hundreds of levels. Key attribute (stacked glyph axis): several to one dozen



Streamgraphs

Idiom	**Streamgraphs**
What: Data	Multidimensional table: one quantitative value attribute (counts), one ordered key attribute (time), one categorical key attribute (artist)
What: Derived	One quantitative attribute (for layer ordering)
How: Encode	Use derived geometry showing layers across time, layer height encodes count
Scale	Key attributes (time, main axis): hundreds of time points. Key attributes (artists, short axis): dozens to hundreds

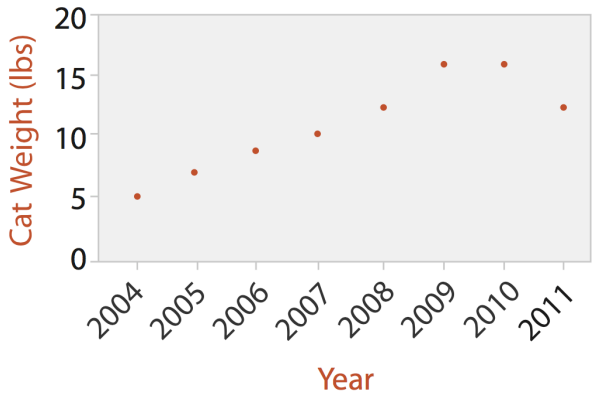


Dot and Line Charts

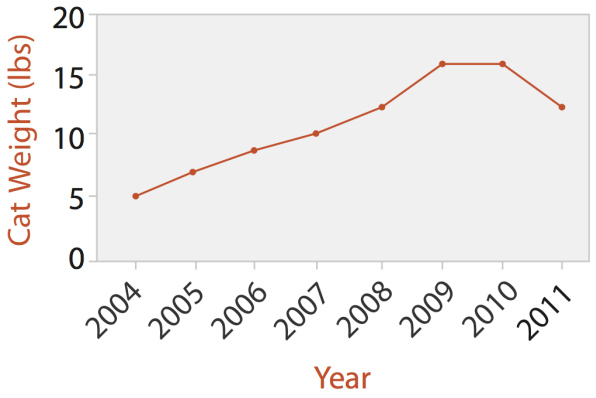
Idiom	**Dot Charts**
What: Data	Table: one quantitative value attribute, one ordered key attribute
How: Encode	Express value attribute with aligned vertical position and point marks. Separate/order into horizontal regions by key attribute
Idiom	**Line Charts**
What: Data	Table: one quantitative value attribute, one ordered key attribute
How: Encode	Dot chart with connection marks between dots
Why: Task	Show trend
Scale	Key attribute: hundreds of levels

Line Charts: Interesting Aspects

- « *A line chart is basically a dot chart with the dots connected* »
- Connection strongly implies dependency
  - Line charts are very good for illustrating trends
  - ... And should be used as such (cf. next slide for a bad example)

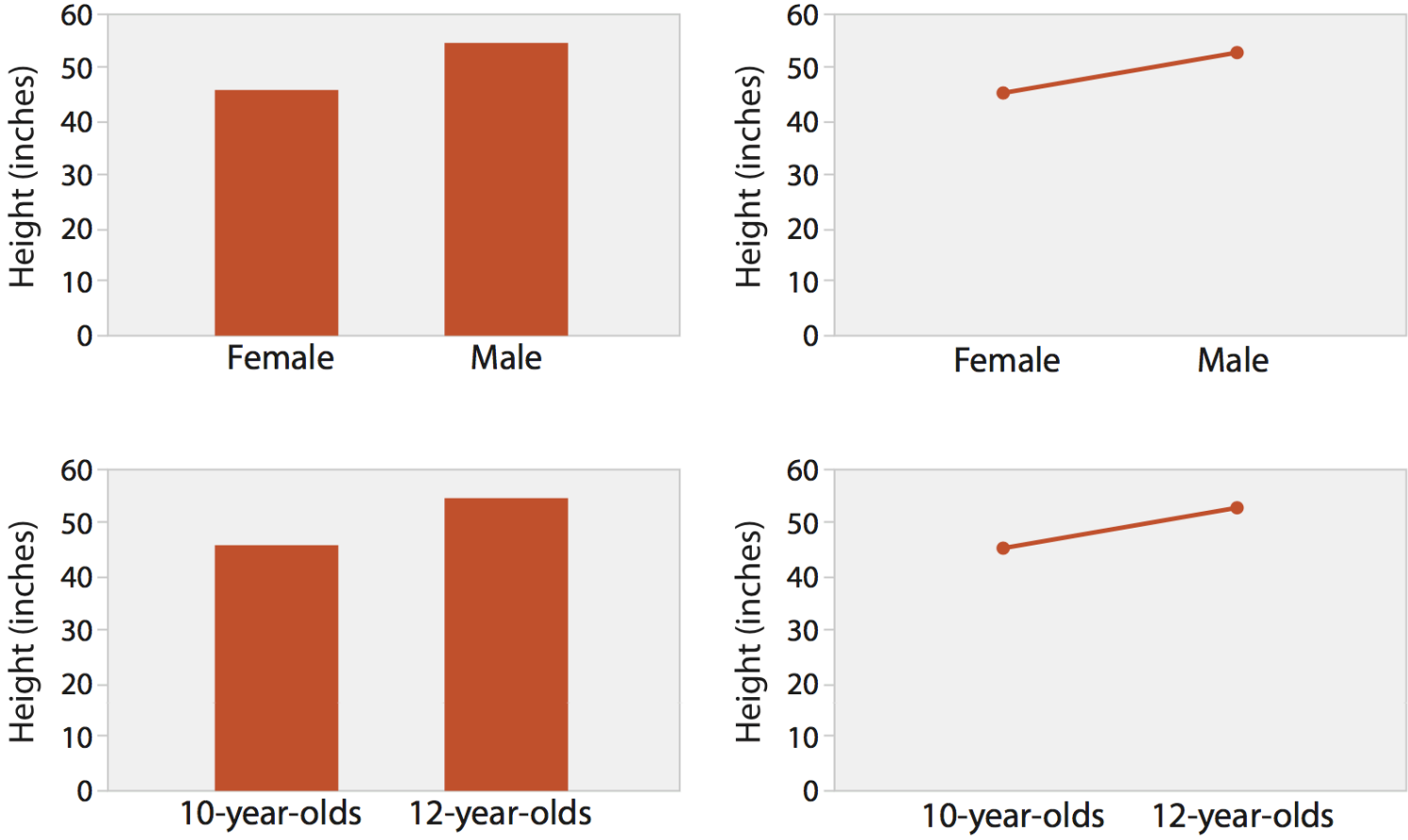


(a)



(b)

Bar Charts vs. Line Charts



Line Charts: the 45° Idiom

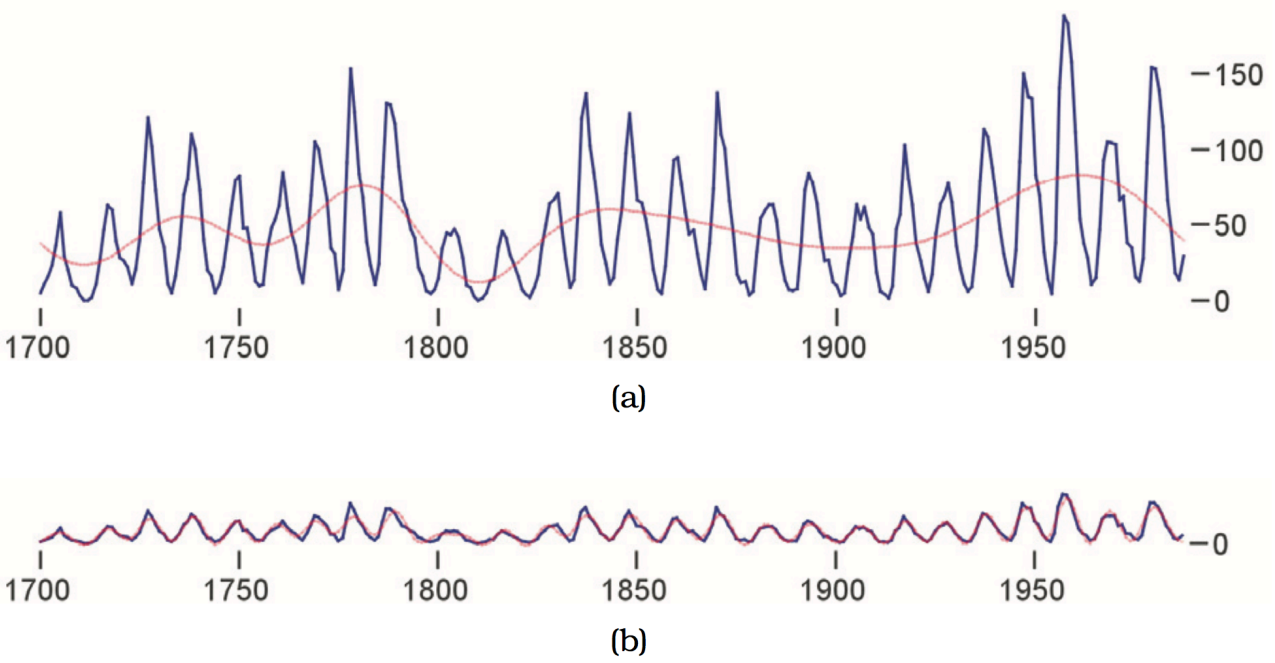
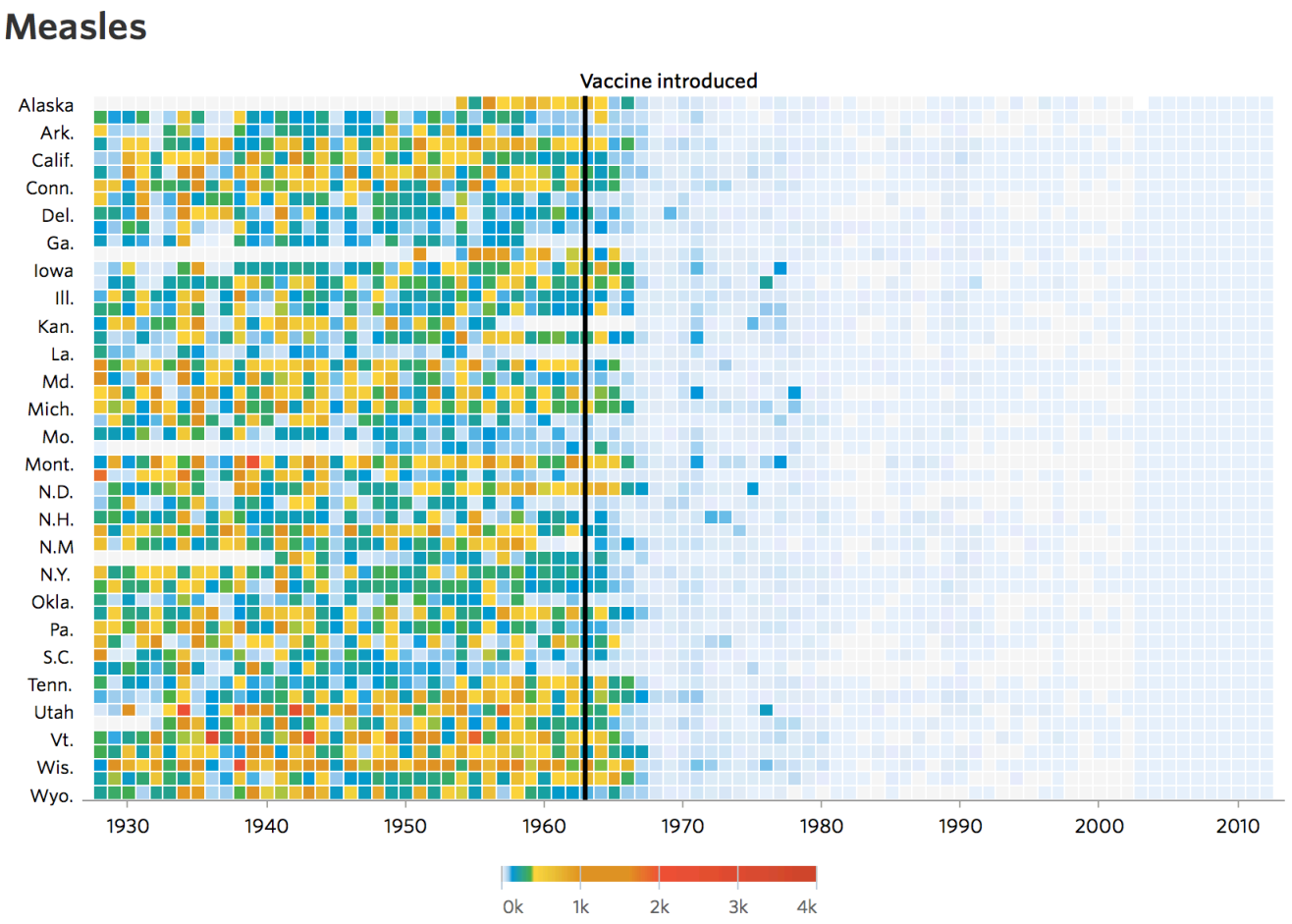


Figure 7.10. Sunspot cycles. The multiscale banking to 45° idiom exploits our orientation resolution accuracy at the diagonal. (a) An aspect ratio close to 4 emphasizes low-frequency structure. (b) An aspect ratio close to 22 shows higher-frequency structure: cycle onset is mostly steeper than the decay. From [Heer and Agrawala 06, Figure 5].

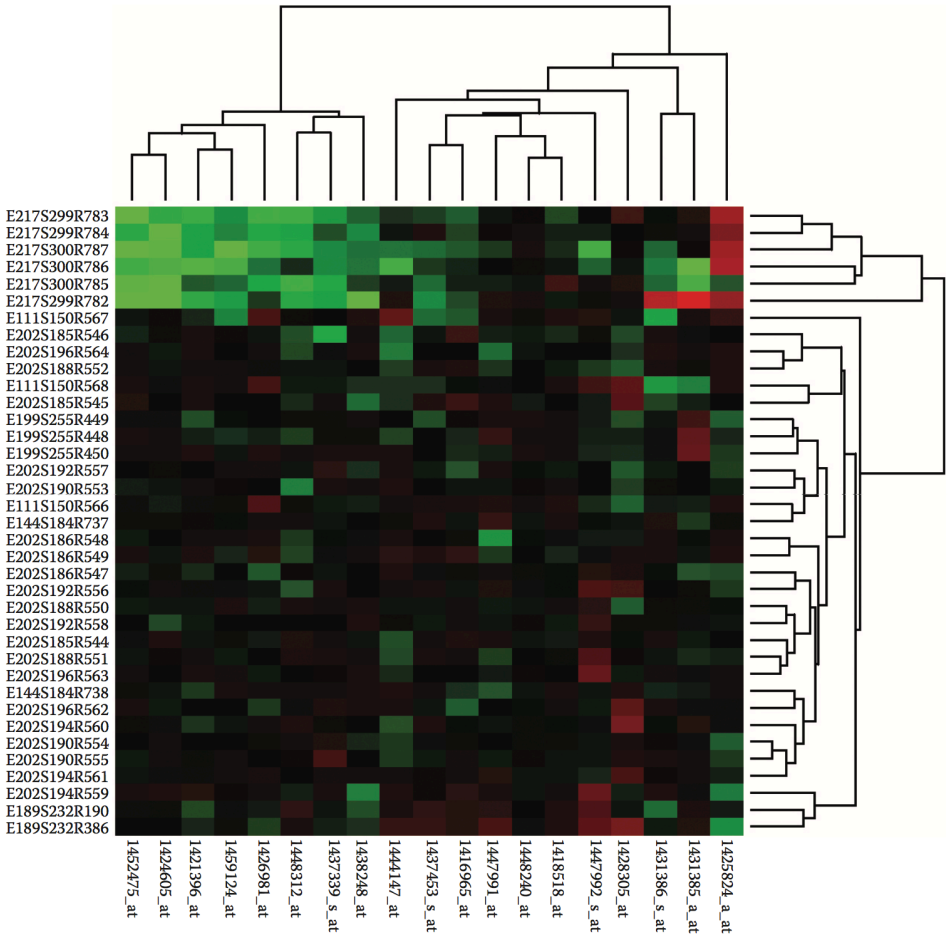
Separate, Order, Align: Heatmaps, Scatterplot Matrices

- Two keys -> matrix alignment
  - One key on the X axis, the other on the Y axis, values in the grid
- Two idioms:
- Heatmaps
  - Heatmaps are good for encoding dense information
    - Each pixel can be one value !
  - Variant: Cluster heatmap: heatmap with dendrograms (trees) encoding hierarchical information for the categorical keys
- Scatterplot matrices
  - Matrix where each cell contains an entire scatterplot chart
  - Shows all possible pairwise combinations of attributes

Heatmap Example



Cluster Heatmaps Example

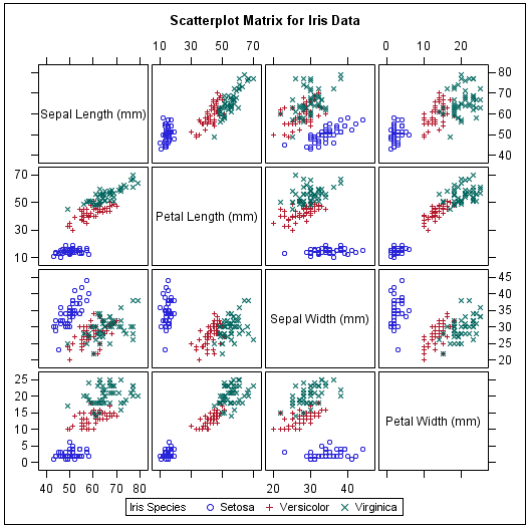


Heatmaps and Cluster Heatmaps

Idiom	**Heatmaps**
What: Data	Table: two categorical/ordered key attributes, one quantitative attribute
How: Encode	2D matrix alignment of area marks, with colourmap
Why: Task	Find clusters, outliers; summarize
Scale	Items: one million. Categorical/ordered attribute levels: hundreds. Quantitative attribute levels: 3-11
Idiom	**Cluster Heatmaps**
What: Data	Two cluster hierarchies for table rows and columns
How: Encode	Heatmap: 2D matrix alignment, ordered by both cluster hierarchies. Dendrogram: connection line for parent-child relationship in tree

Scatterplot Matrix (SPLOM)

Idiom	**Scatterplot Matrix (SPLOM)**
What: Data	Table
What: Derived	Ordered key attribute: list of original attributes
How: Encode	Scatterplots in 2D matrix alignment
Why: Task	Find correlation, trends, outliers
Scale	Attributes: one dozen. Items: dozens to hundreds



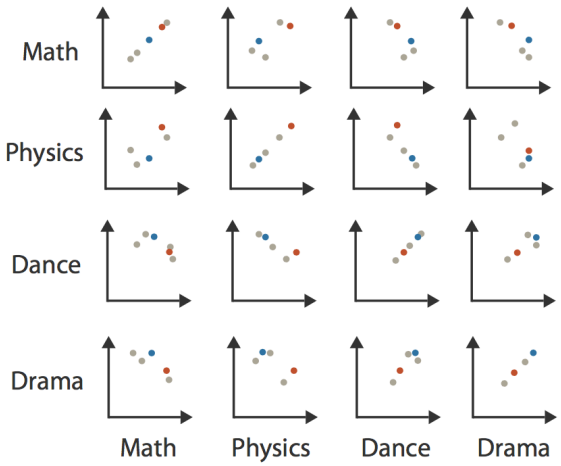
Spatial Axis Orientation: Parallel Coordinates

- Useful approach for visualizing many quantitative attributes at once using spatial position
- Each attribute = one axis, linked one to the other
- Very good to check for correlations
- Scatterplots don't scale well, though

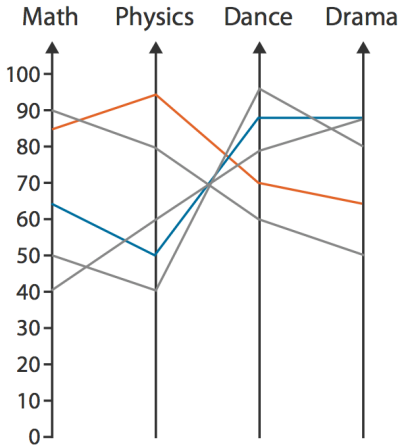
Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

Scatterplot Matrix

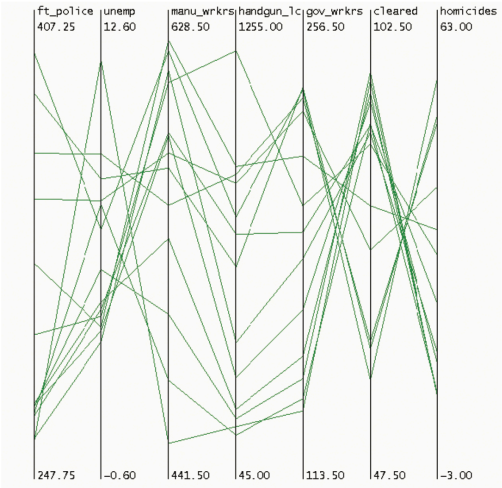


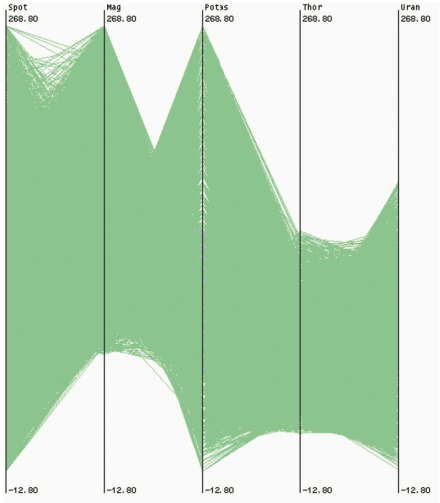
Parallel Coordinates



Parallel Coordinates

Idiom	**Parallel coordinates**
What: Data	Table: many value attributes
How: Encode	Parallel layout: horizontal spatial position used to separate axes, vertical position used to express value along each aligned axis with connection line marks as segments between them
Why: Task	Find trends, outliers, extremes, correlations
Scale	Attributes: dozens along secondary axis; Items: hundreds



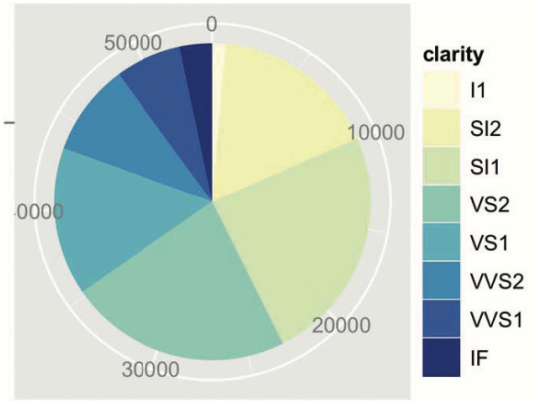


Spatial Axis Orientation: Radial Layouts

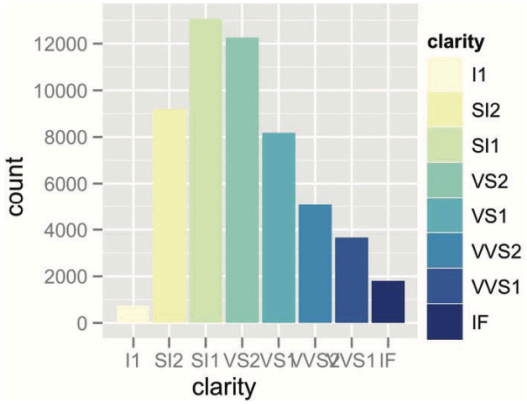
- Polar coordinates vs. Rectilinear coordinates
- Infamous example of radial layout: pie charts
- Perceptually, multiple issues with radial layouts:
  - Angles less well perceived than length
  - Inherently cyclic (i.e. if your data is not cyclic/periodic, don't use a radial layout -> expressiveness principle)
  - Parts at the top of a pie chart « feel » more important
- Pie charts are good at showing the relative contribution of parts to a whole
  - But they're not the only vis idioms that can do so...

Pie Charts: Comparison With Other Vis Idioms

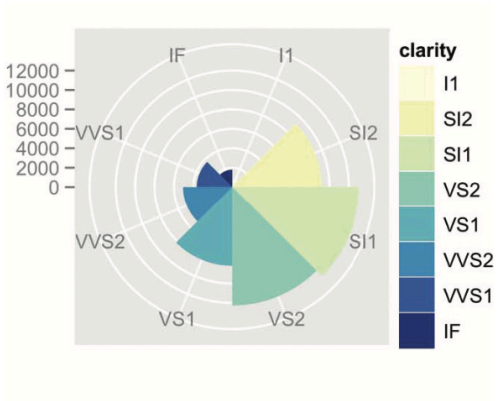
Pie chart

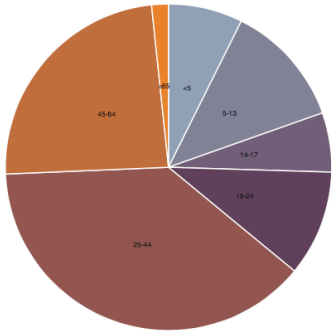


Bar chart

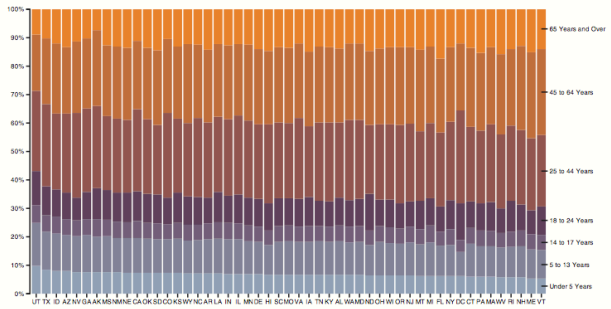


Polar area chart

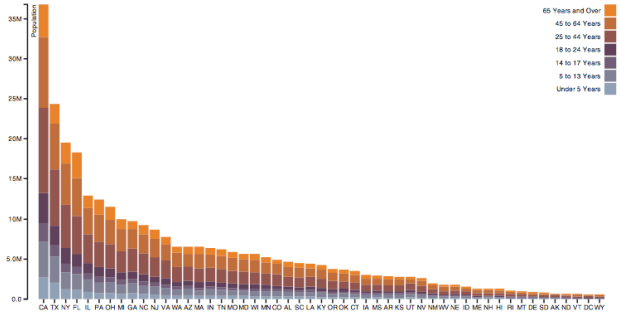




Pie chart



Normalized stacked bar chart



Stacked bar chart

Pie Chart, Polar Area Chart vs. Normalized Stacked Bar Chat

Idiom	**Pie Chart; Polar Area Chart**
What: Data	Table: one quantitative attribute, one categorical attribute
How: Encode	Area marks (wedges) with angle channel; (varying length) radial layout.
Why: Task	Part–whole relationship
Scale	One dozen categories
Idiom	**Normalized Stacked Bar Chart**
What: Data	Multidimensional table: one quantitative value attribute, two categorical key attributes
How: Encode	Line marks with length channel; rectilinear layout
Why: Task	Part–whole relationship
Scale	One dozen categories for stacked attribute. Several dozen categories for axis attribute

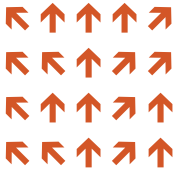
Arranging Spatial Data

Arrange Spatial Data

- **Use Given**
  - Geometry
    - *Geographic*
    - *Other Derived*

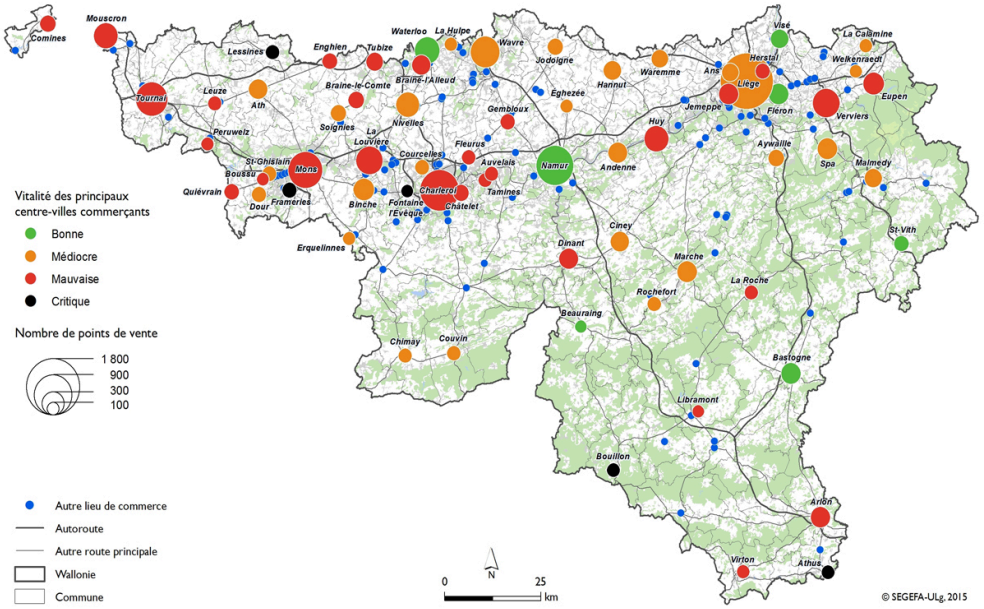
  - Spatial Fields
    - *Scalar Fields (one value per cell)*
      - *Isocontours*
      - *Direct Volume Rendering*

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



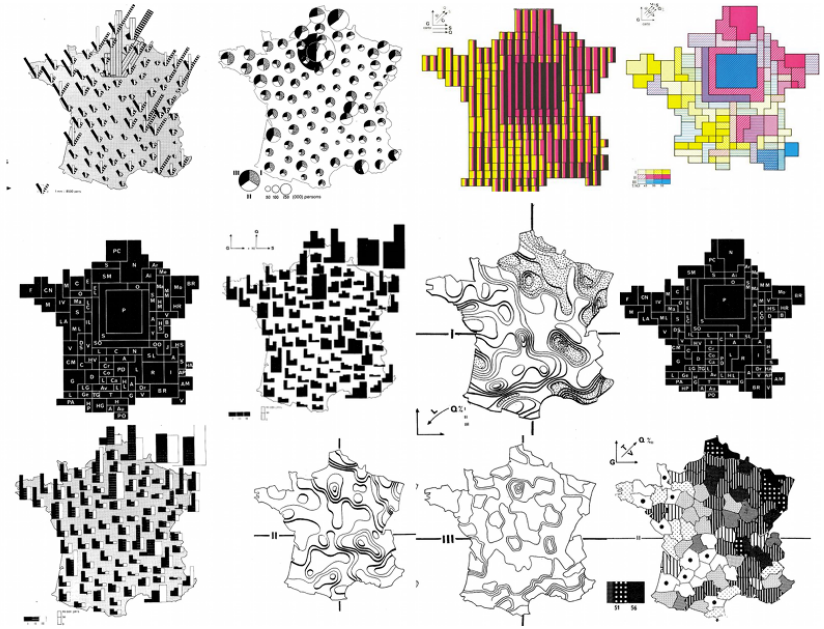
Arranging Spatial Data Common Case: Use Given Geometry/Spatial Field

- Typically, the given spatial position is the attribute of primary importance
  - Central tasks revolve around understanding spatial relationships
  - The provided spatial attributes should be used as the basis
  - Effectiveness principle; common sense, too...



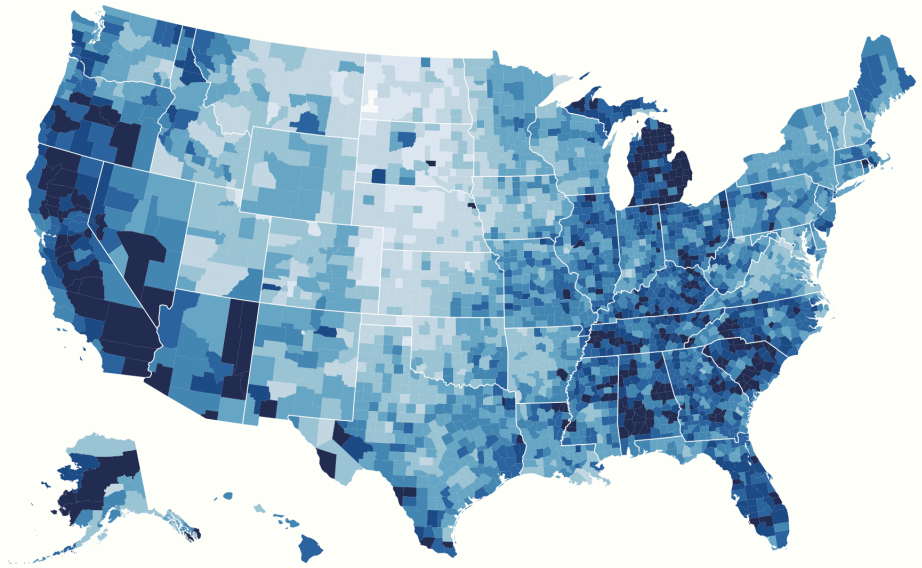
Geometry: Geographic Data

- Huge body of work already done by cartographers
- Integrating non-spatial data with spatial data: thematic cartography
  - E.g. population of cities with the localisation of said cities...



Choropleth Map

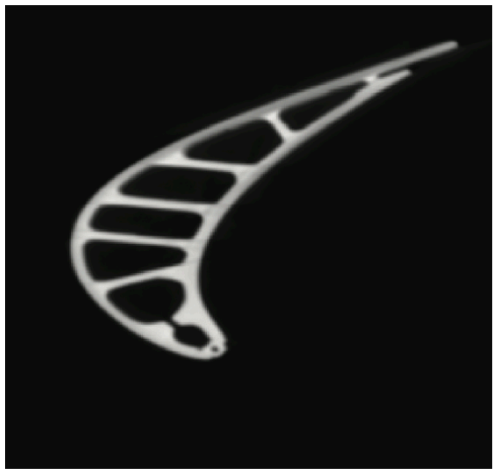
Idiom	**Choropleth Map**
What: Data	Geographic geometry data. Table with one quantitative attribute per region
How: Encode	Space: use given geometry for area mark boundaries. Color: sequential segmented colormap



Geometry: Scalar Fields – One Value

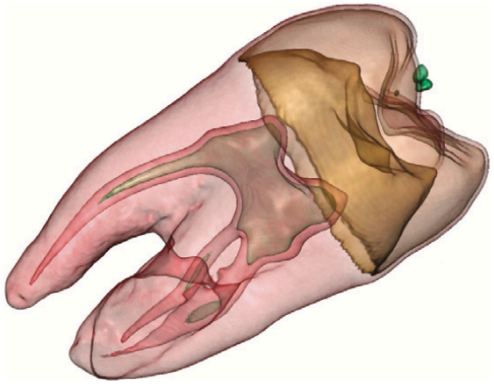
- Scalar spatial field: a single value associated with each spatially defined cell
- Three major families of idioms:
  - Slicing: typically only 2 dimensions shown at once
  - Isocontours: only derived data of lower-dimensional surface geometry
  - Direct volume rendering: as name implies !
- Closely related to scientific visualisation

Geometry: Scalar Field: Examples



(a)

Slicing



(b)

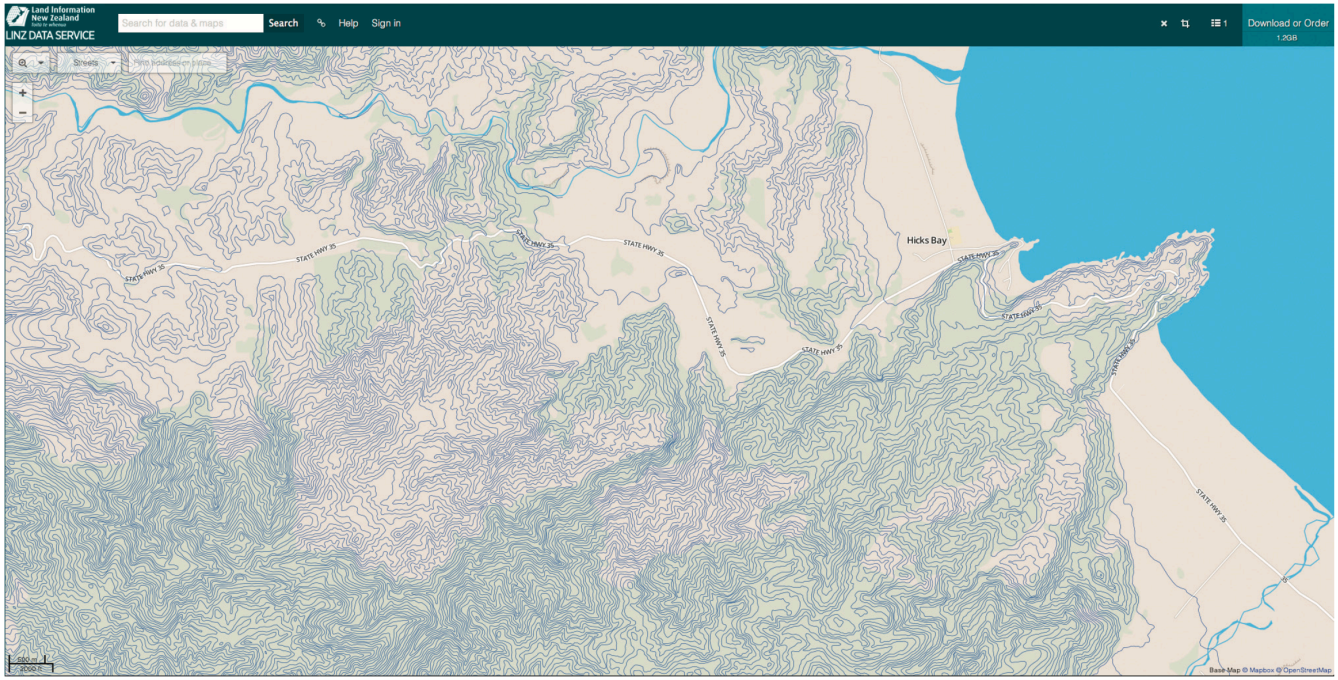
Isocontours

(c)

Direct volume rendering

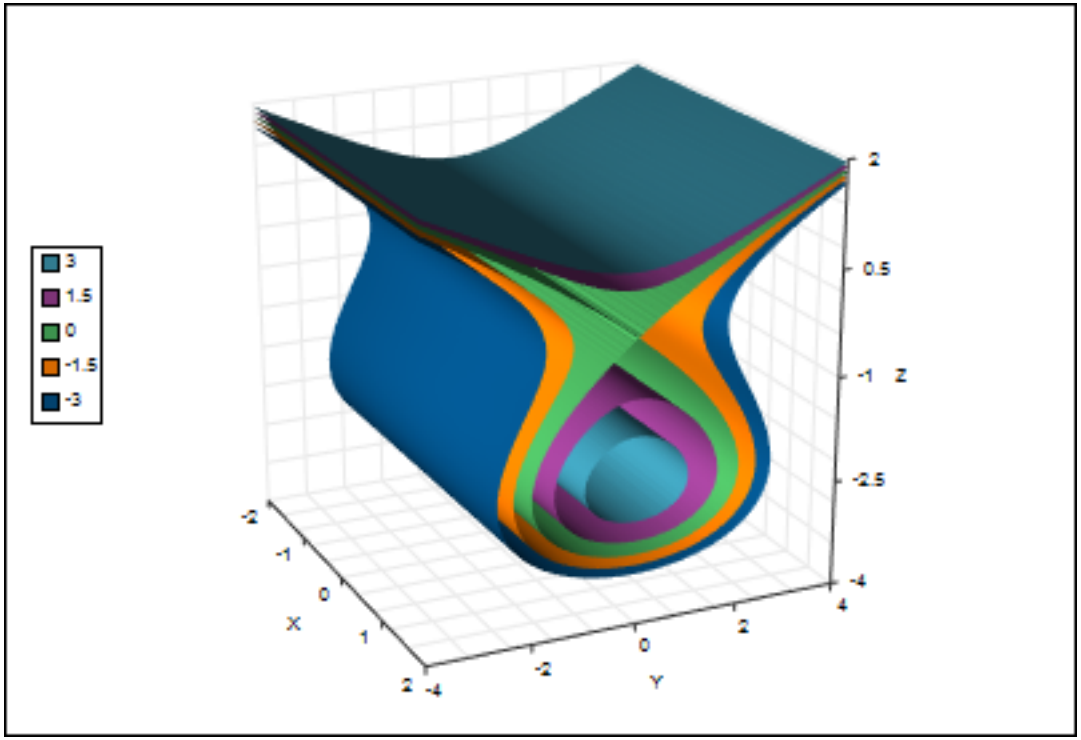
Scalar Fields – One Value: Isocontours

- Derivation of isolines
  - Isolines: lines that represent the contours of a particular level of the scalar value
  - Classical example: topographic terrain maps



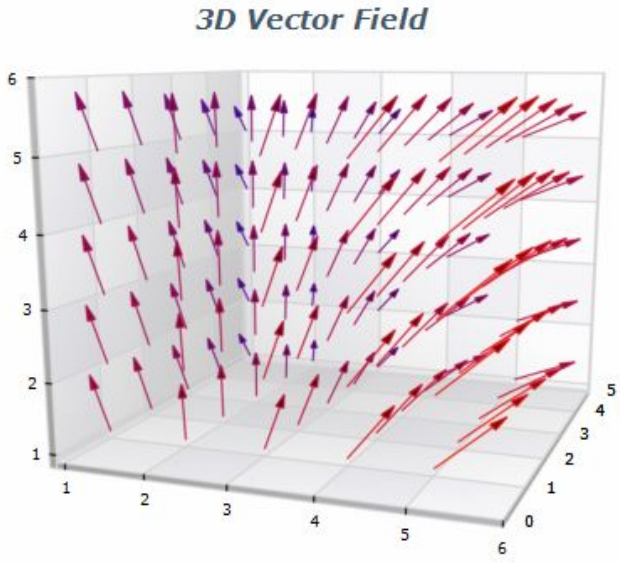
Scalar Fields – One Value: Isosurfaces

- Transformation of a 3D scalar spatial field into one or more derived 2D surfaces that represent a level of one value
  - Typically, one colour mapped to a given value

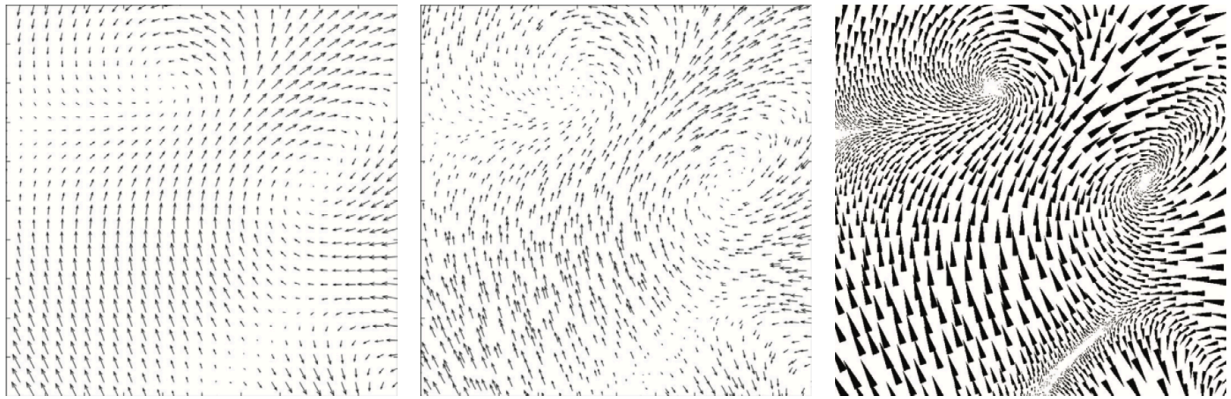


Geometry: Vector Fields – Multiple Values

- Typically used for visualising flow simulations or measurements
  - E.g. velocity field
- Three main alternatives:
  - Purely 2D spatial fields
  - Purely 3D spatial field
  - Flow on a 2D surface embedded in a 3D space
- Unsteady (time-varying) vs. steady (fixed) flow datasets



Geometry: Vector Fields Examples



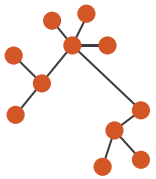
Laidlaw, D. H., Kirby, R. M., Jackson, C. D., Davidson, J. S., Miller, T. S., Da Silva, M., ... & Tarr, M. J. (2005). Comparing 2D vector field visualization methods: A user study. IEEE Transactions on Visualization and Computer Graphics, 11(1), 59-70.

Arranging Networks and Trees

Arrange Networks and Trees

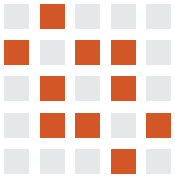
Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



Adjacency Matrix
Derived Table

✔ NETWORKS ✔ TREES



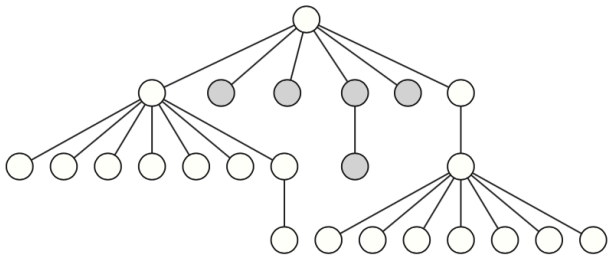
Enclosure
Containment Marks

✘ NETWORKS ✔ TREES

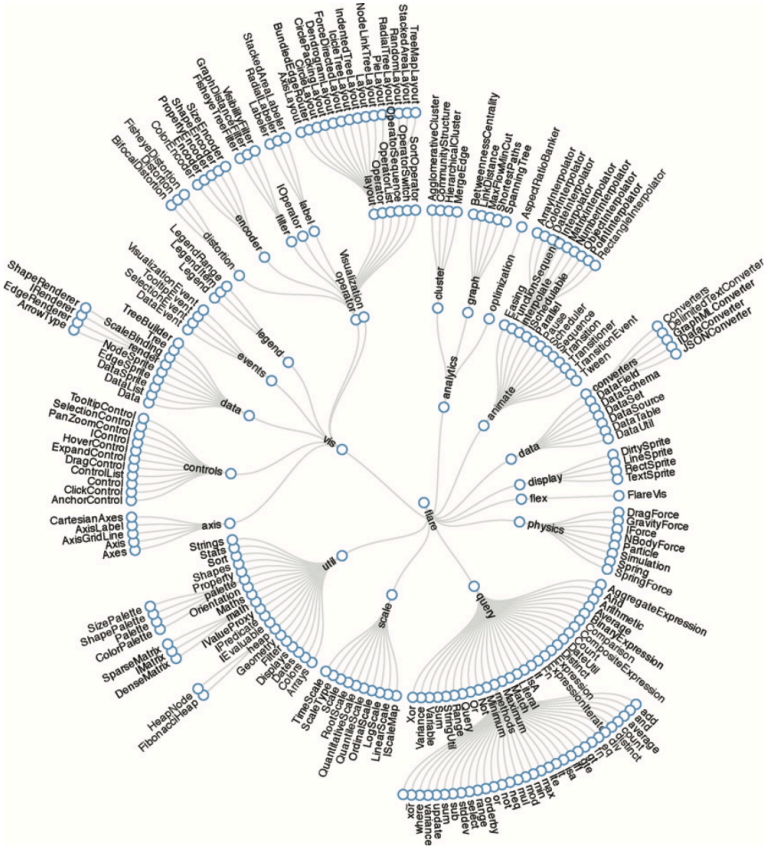
Arranging Networks and Trees: Connecting with Link Marks

- most common visual encoding idiom for tree and network data: node–link diagrams
  - Connection marks indicate relationships between items
- Used for trees and networks alike (examples follow)
- Good for understanding a network's topology
  - finding all possible paths from one node to another,
  - finding the shortest path between two nodes
  - finding all the adjacent nodes one hop away from a target node
  - finding nodes that act as a bridge between two components of the network that would otherwise be disconnected.

Connecting with Link Marks: Layout



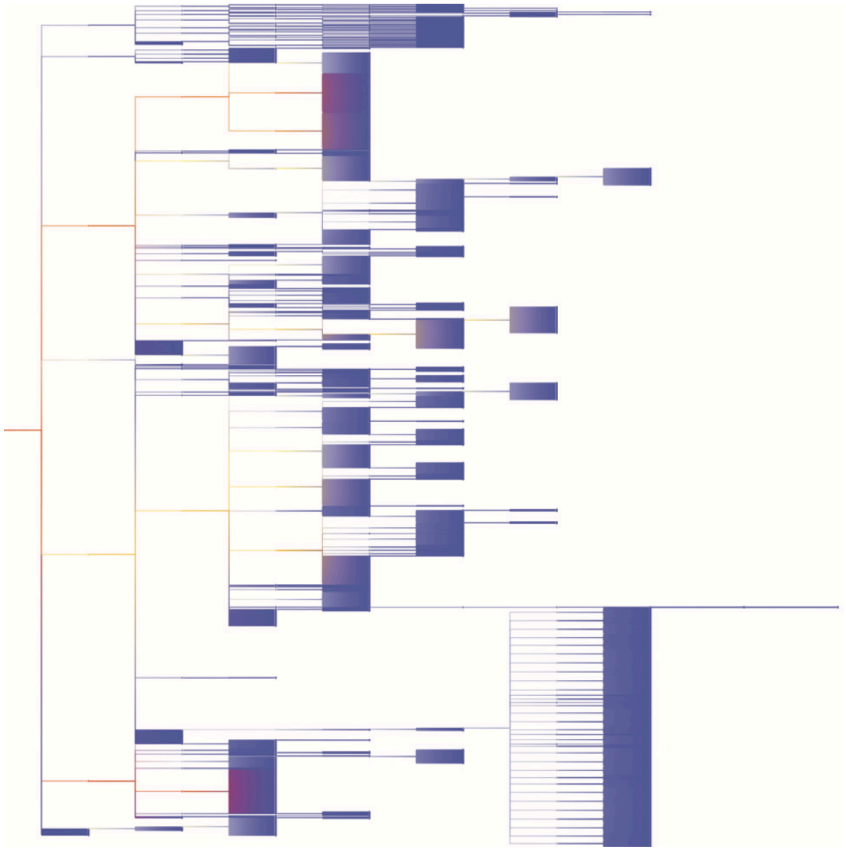
(a)



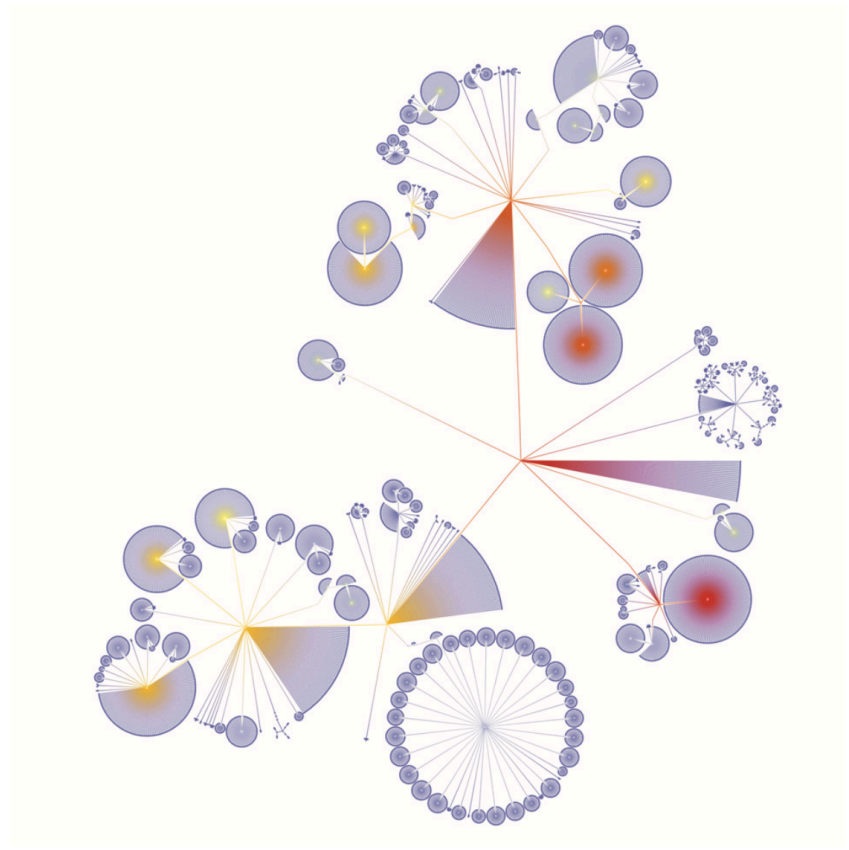
(b)

Connecting with Link Marks: Density

(Rectangular Horizontal Node-Link Layout vs. BubbleTree Node-Link Layout)



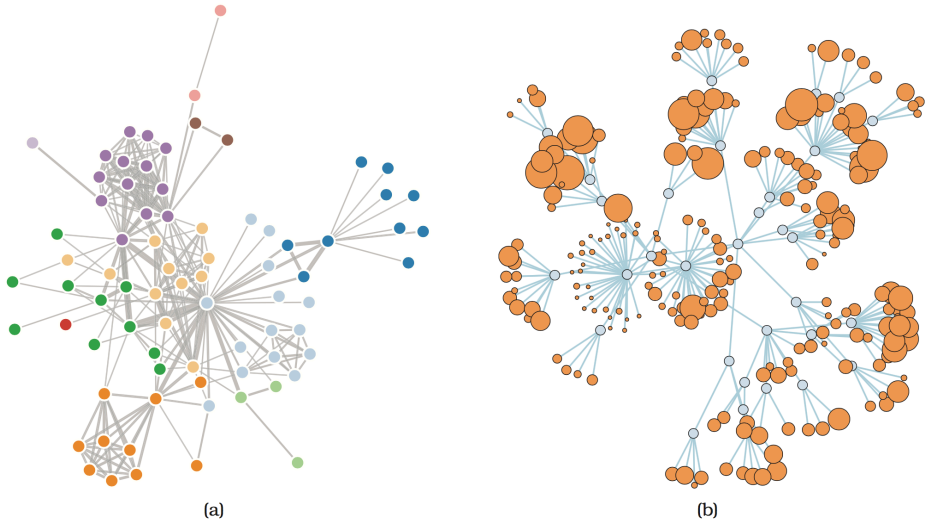
(a)



(b)

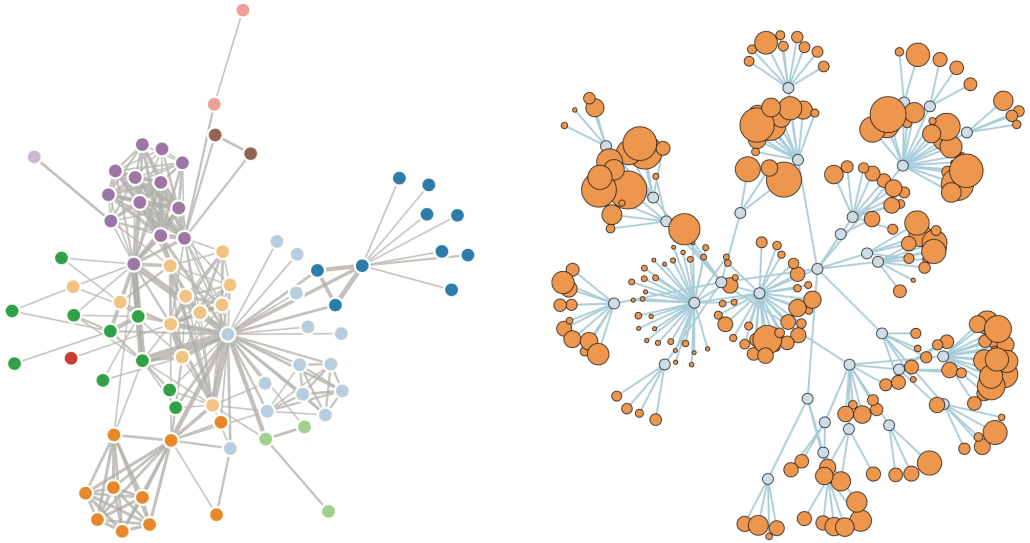
Connecting with Link Marks: Force-Directed Placement

- Widely used vis idiom (present in D3.js)
- Typically, network elements are positioned according to simulated physical forces pushing nodes away from each other
  - Importance of visual clusters for graph analysis
- Weaknesses: layout is typically nondeterministic; scalability
  - Also, if the graph is dynamic, movement of nodes is distracting



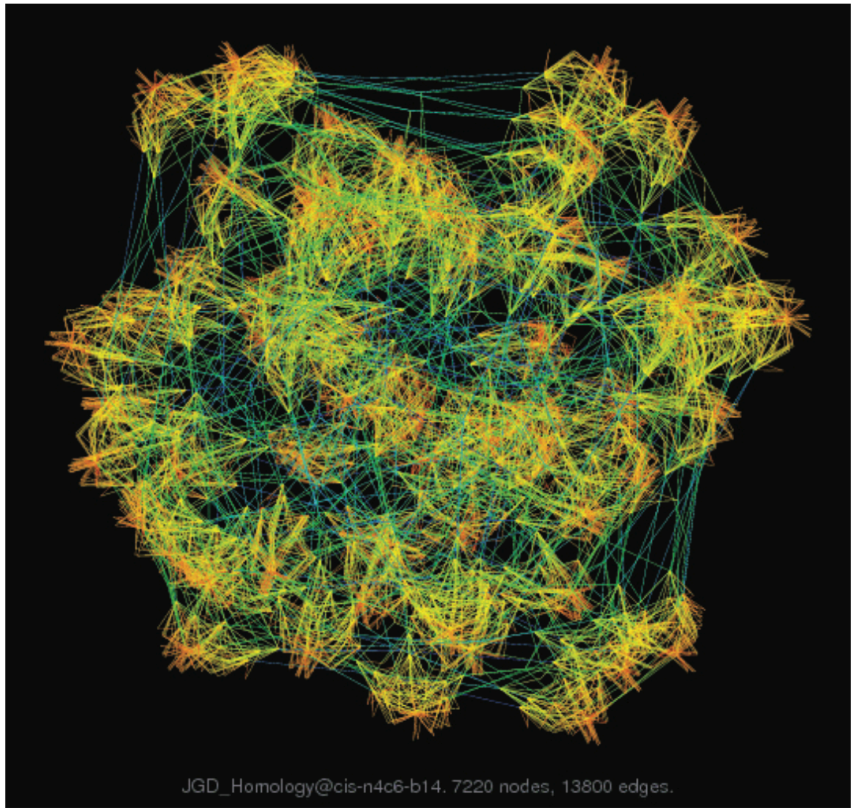
Force-Directed Placement

Idiom	**Force-Directed Placement**
What: Data	Network
How: Encode	Point marks for nodes, connection marks for links
Why: Task	Explore topology, locate paths
Scale	Nodes: dozens/hundreds. Links: hundreds. Node/link density: L < 4N

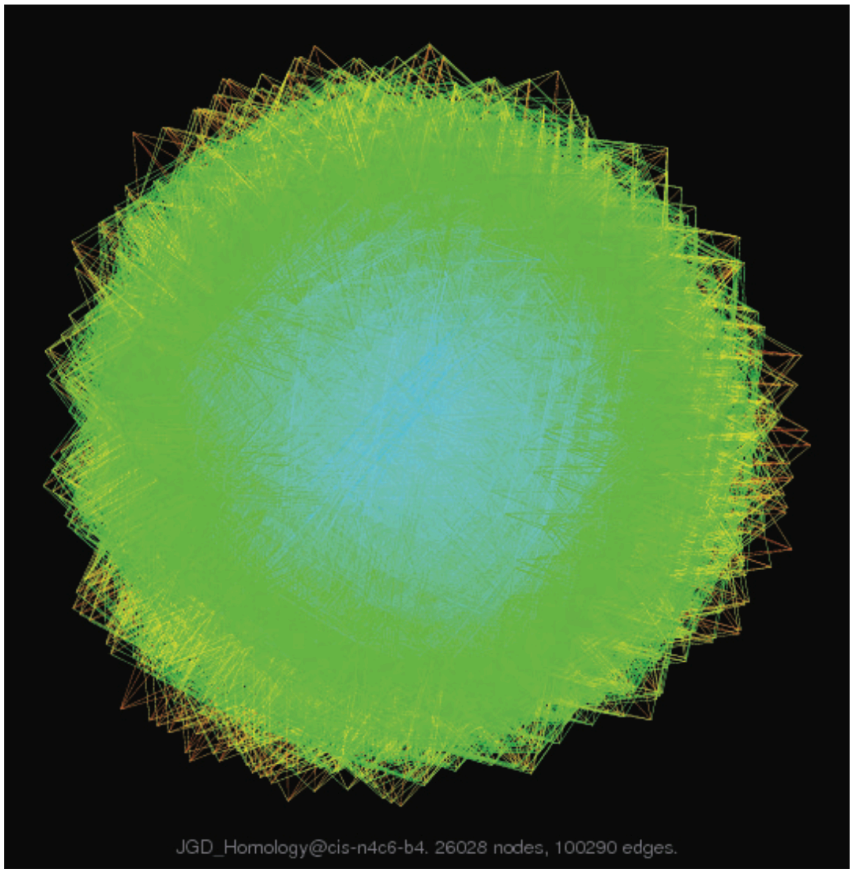


Connecting with Link Marks: Scalability with Multi-Level Networks

7220 nodes and 13,800 edges

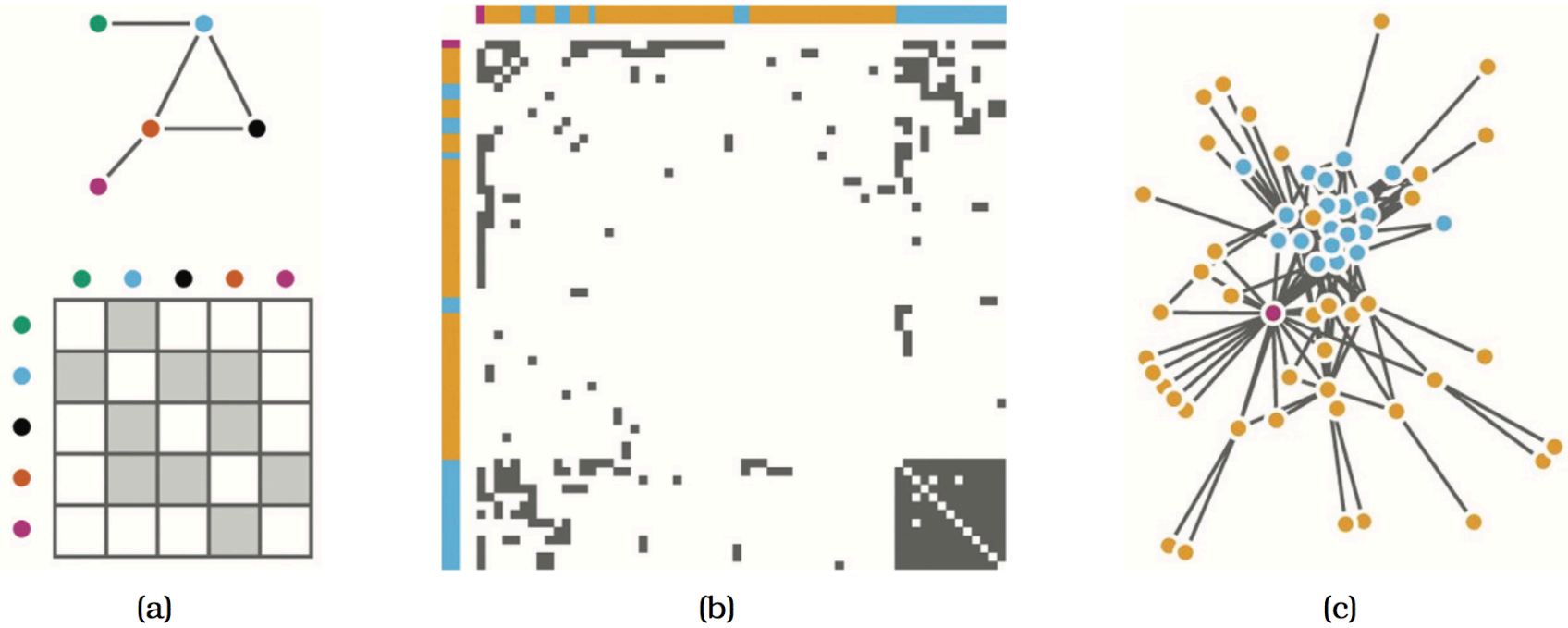


26,028 nodes and 100,290 edge



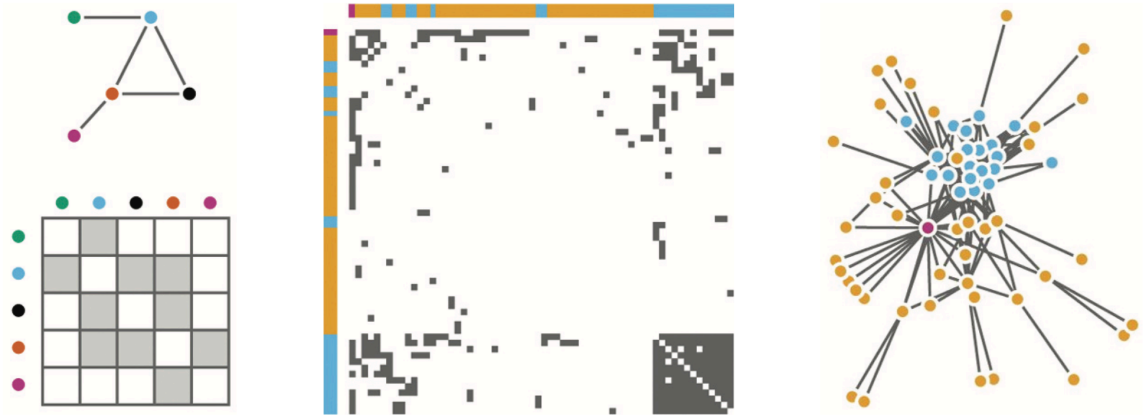
Arranging Networks and Trees: Connecting with Matrix Views

- Alternative to connecting with node-links
- Adjacency matrix: nodes displayed in X and Y axes, connections between nodes are marked as marks in matrix



Adjacency Matrix View

Idiom	**Adjacency Matrix View**
What: Data	Network
What: Derived	Table: network nodes as keys, link status between two nodes as values
How: Encode	Area marks in 2D matrix alignment
Scale	Nodes: 1000. Links: one milllion

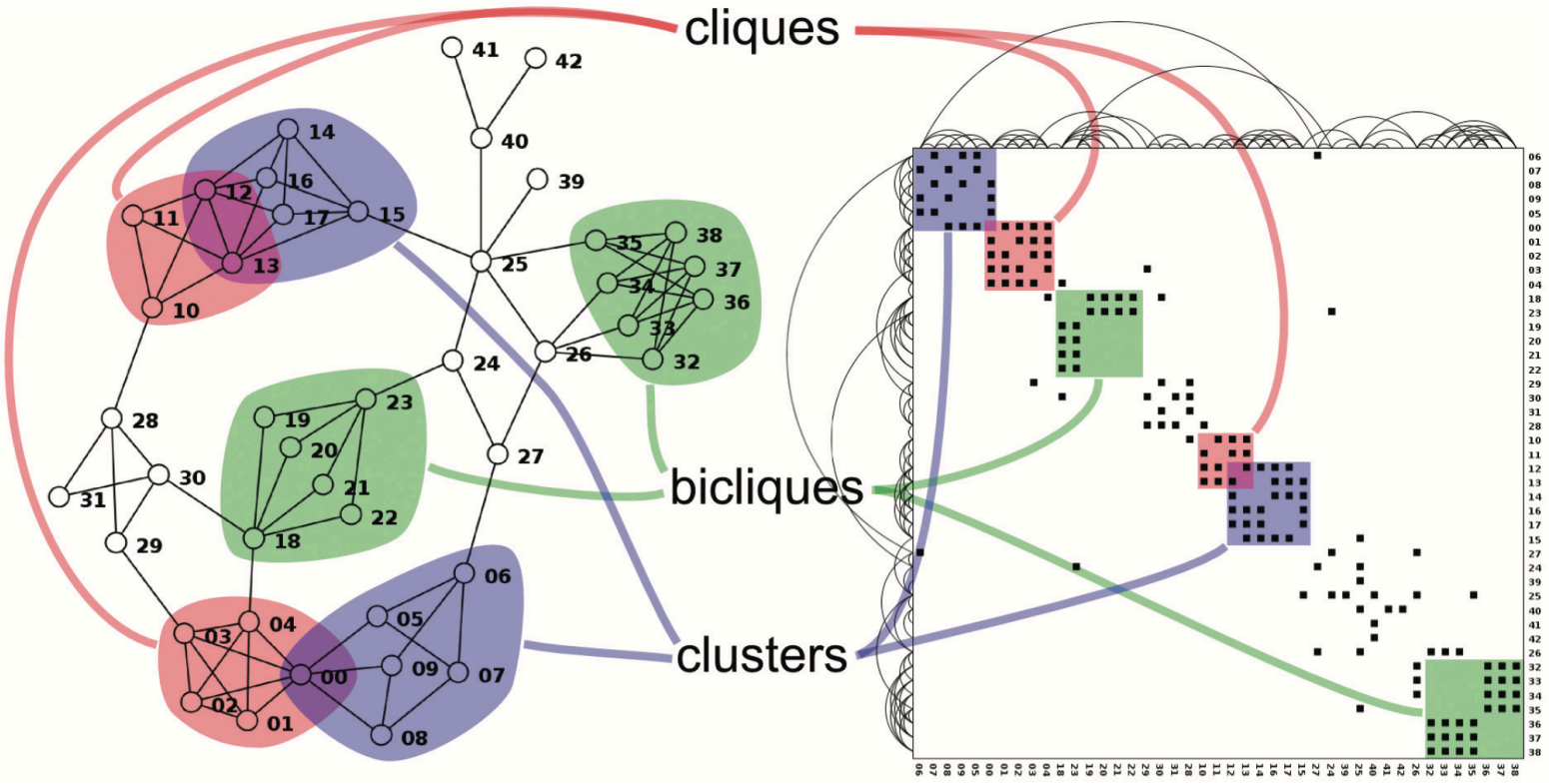


(a) (b) (c)

Node-Links vs. Matrix Views

- Node-link layouts:
  - Very intuitive for sufficiently small networks
  - Perfect for tasks involving analysis of topology
  - Also good for general overview tasks or finding substructures
  - Beyond a certain node to link density, get unreadable because of occlusion issues
- Matrix views:
  - Good perceptual scalability for both large and dense graphs
  - Zero occlusion issue
  - Predictable, stable; support reordering tasks
  - Very good for tasks such as estimating number of nodes, or node lookup
  - Less intuitive than node-link layouts

Node-Links vs. Matrix Views: Cliques



Containment: Hierarchy Marks

- Alternative to node-link tree drawing
- Hierarchical relationships shown with containment
- Good for understanding attribute values of leaves of the tree
- Best used with shallow hierarchies, and in combination with detail-on-demand interaction techniques

Treemaps

Idiom	**Treemaps**
What: Data	Tree
How: Encode	Area marks and containment, with rectilinear layout
Why: Tasks	Query attributes at leaf nodes
Scale	Leaf nodes: one million. Links: one million