02: Data Abstraction

Enrico Puppo
Department of Computer Science, Bioengineering, Robotics and Systems Engineering
University of Genova

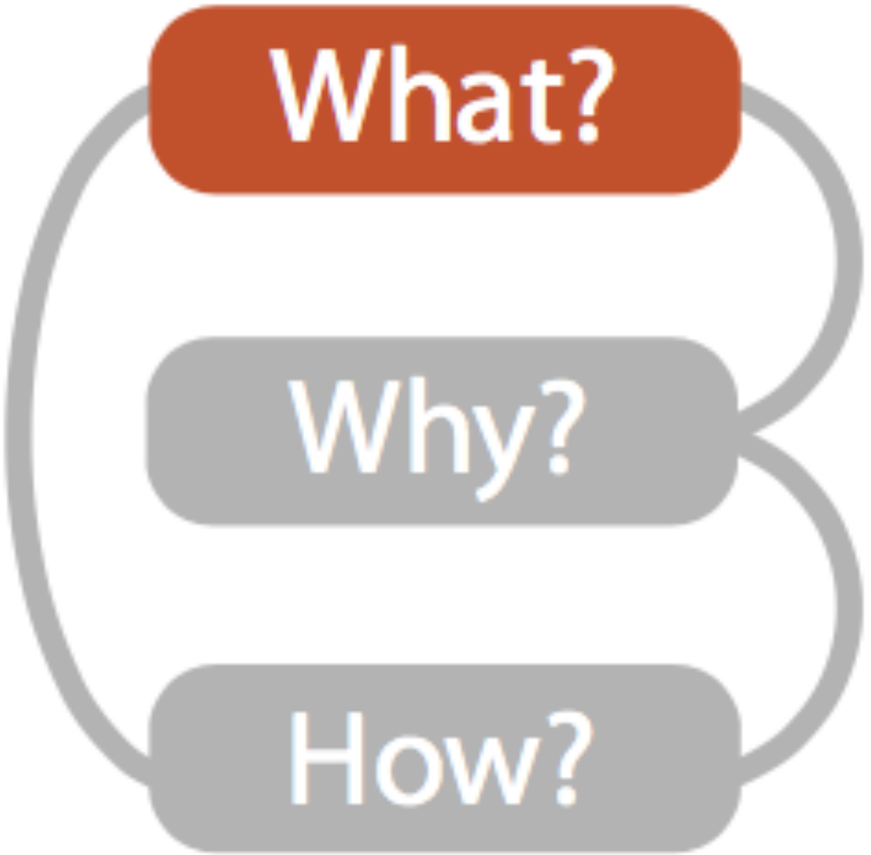
90529 Data Visualization
28 September - 1 October 2020

https://2020.aulaweb.unige.it/course/view.php?id=4293

Credits:
material in these slides is partially taken from
- T. Munzner, University of British Columbia
- A. Lex, University of Utah

other credits in the slides

Design cycle



Data abstraction

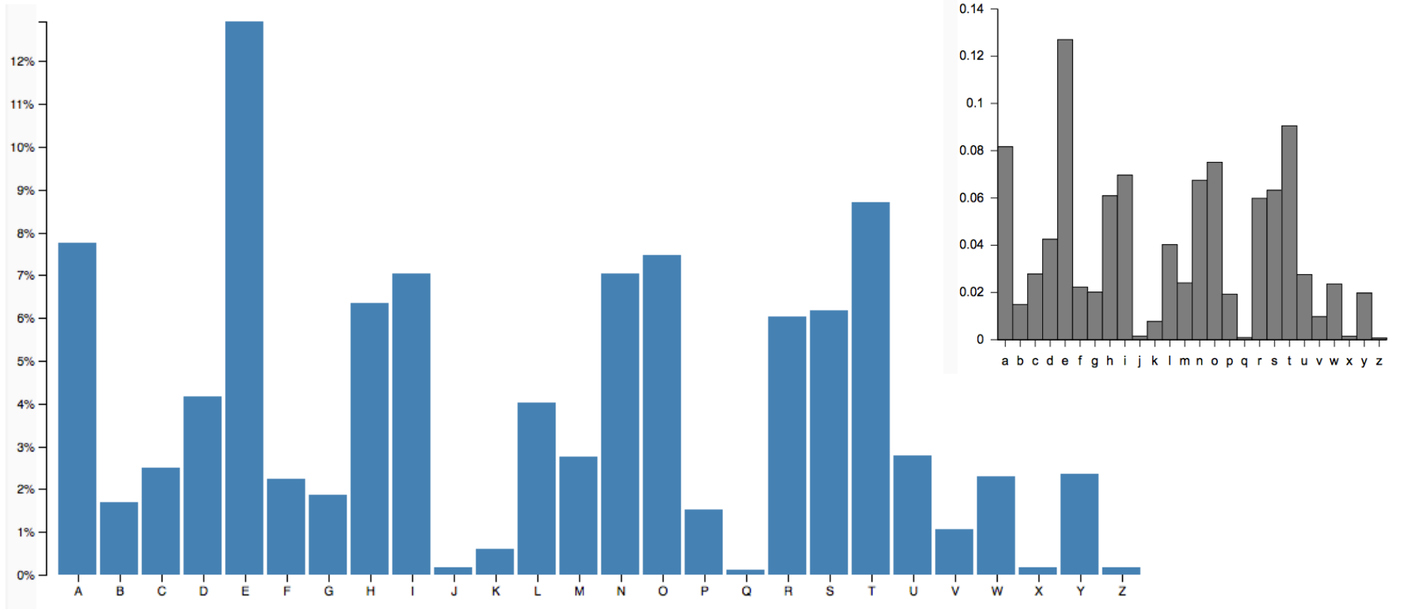
- Vis is concerned with the visualization of *data*
  - data live in *abstract domains* (infovis) or *spatial domains* (scivis)
  - structure, domain and semantics of data are *specific* and *application dependent*
- We need more abstract, general, *application-independent schemes*
- Basic concepts:
  - dataset
  - data type & structure
  - item
  - attribute
  - domain
  - key & value
  - semantics

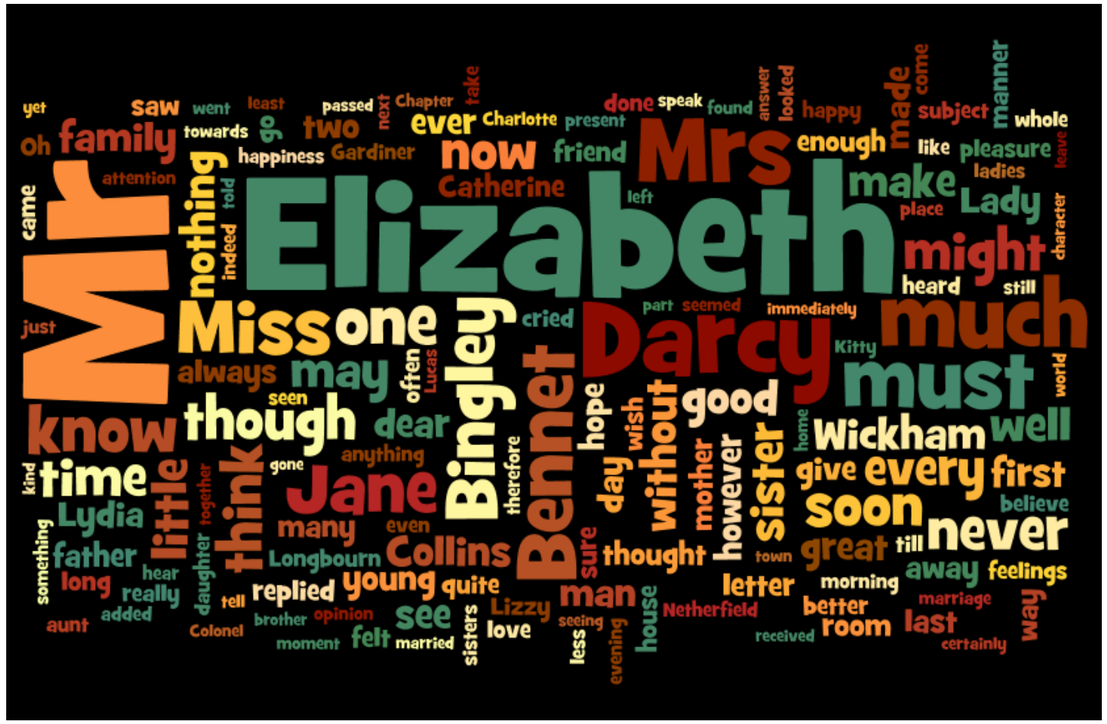
Data abstraction

- Example: book
  - stream of chars
  - stream of words
  - list of sentences
  - hierarchical structure into chapters and paragraphs
  - figures? captions? tables? charts? indexes?
  - Characters in a novel:
    - when do they appear first?
    - how much of the book deals about each?
    - how do they relate?
  - Cross-references in a science book:
    - how do different chapters / paragraphs / sentences / terms relate?

Data abstraction

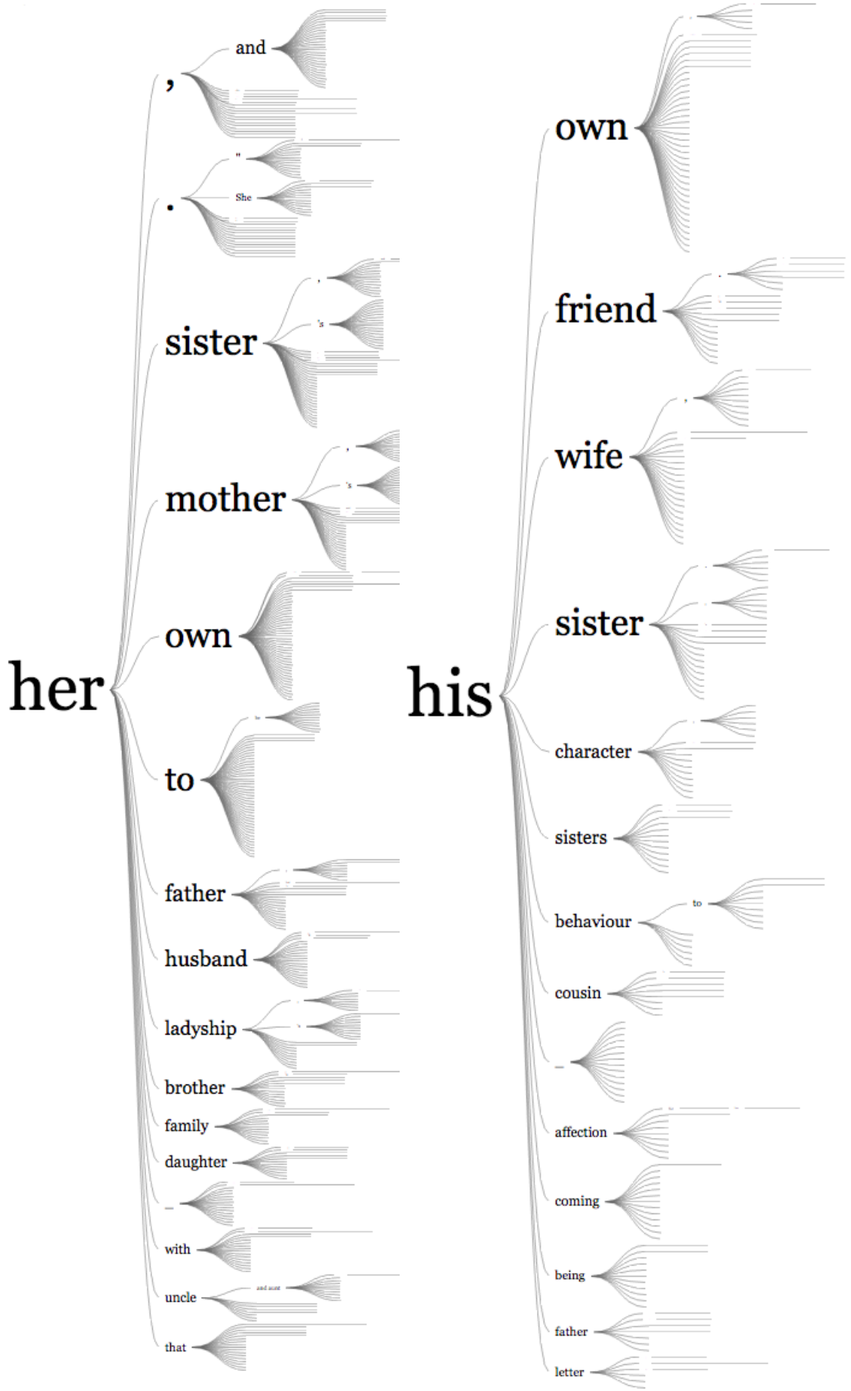
- Example: *Pride and prejudice*, J. Austen, 1813 - novel
  - as a stream of chars: count frequencies of letters
  - as a stream of words: highlight most frequent words (*word cloud*)





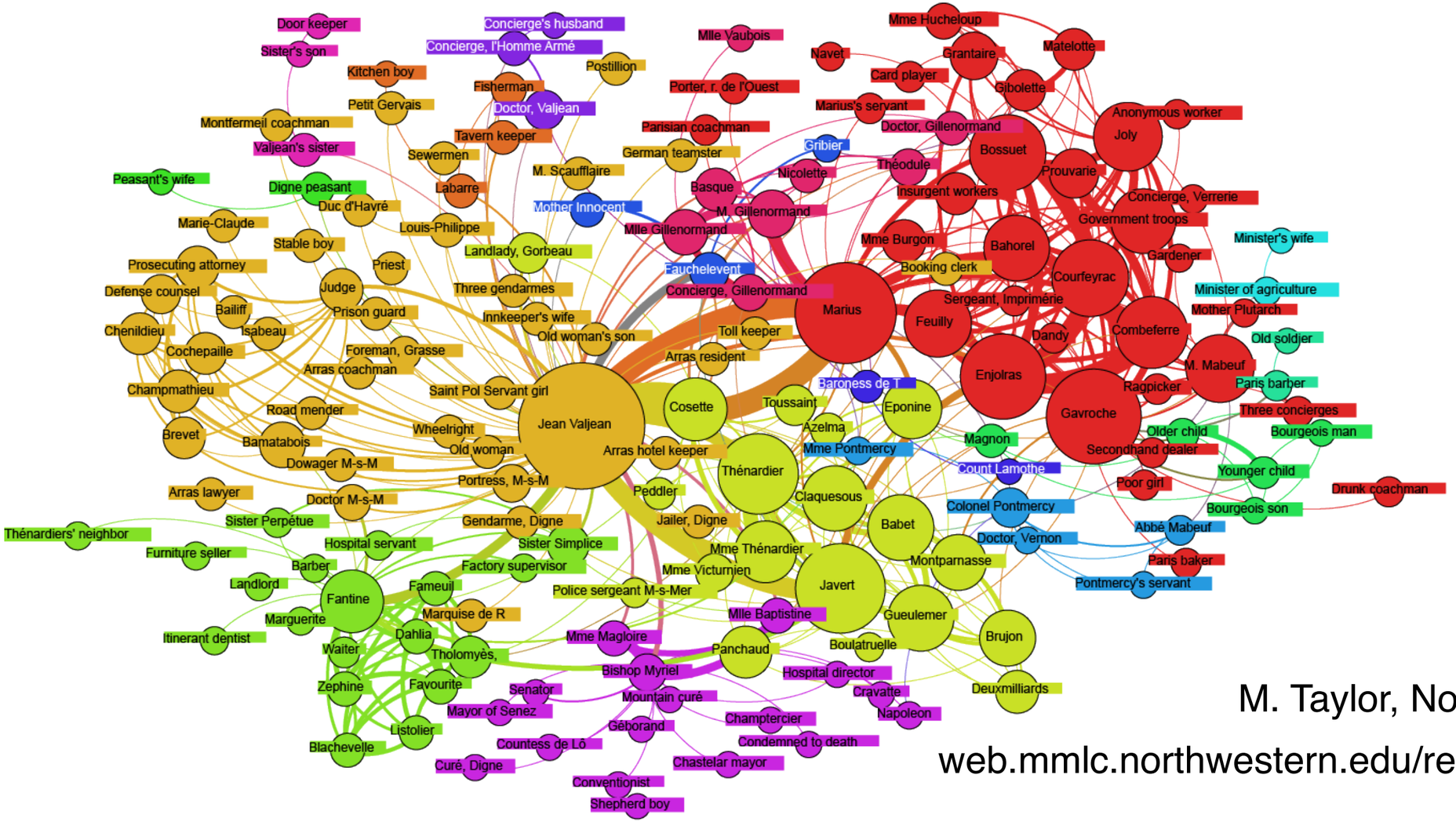
Data abstraction

- Example: *Pride and prejudice*, J. Austen, 1813 - novel
  - as a list of sentences: trace word dependences
  - as a hierarchy of book / chapters / paragraphs /sentences: tree
    - make your own :-)



Data abstraction

- Example: *Les misérables*, V. Hugo, 1862 - novel
  - relations between characters: graph

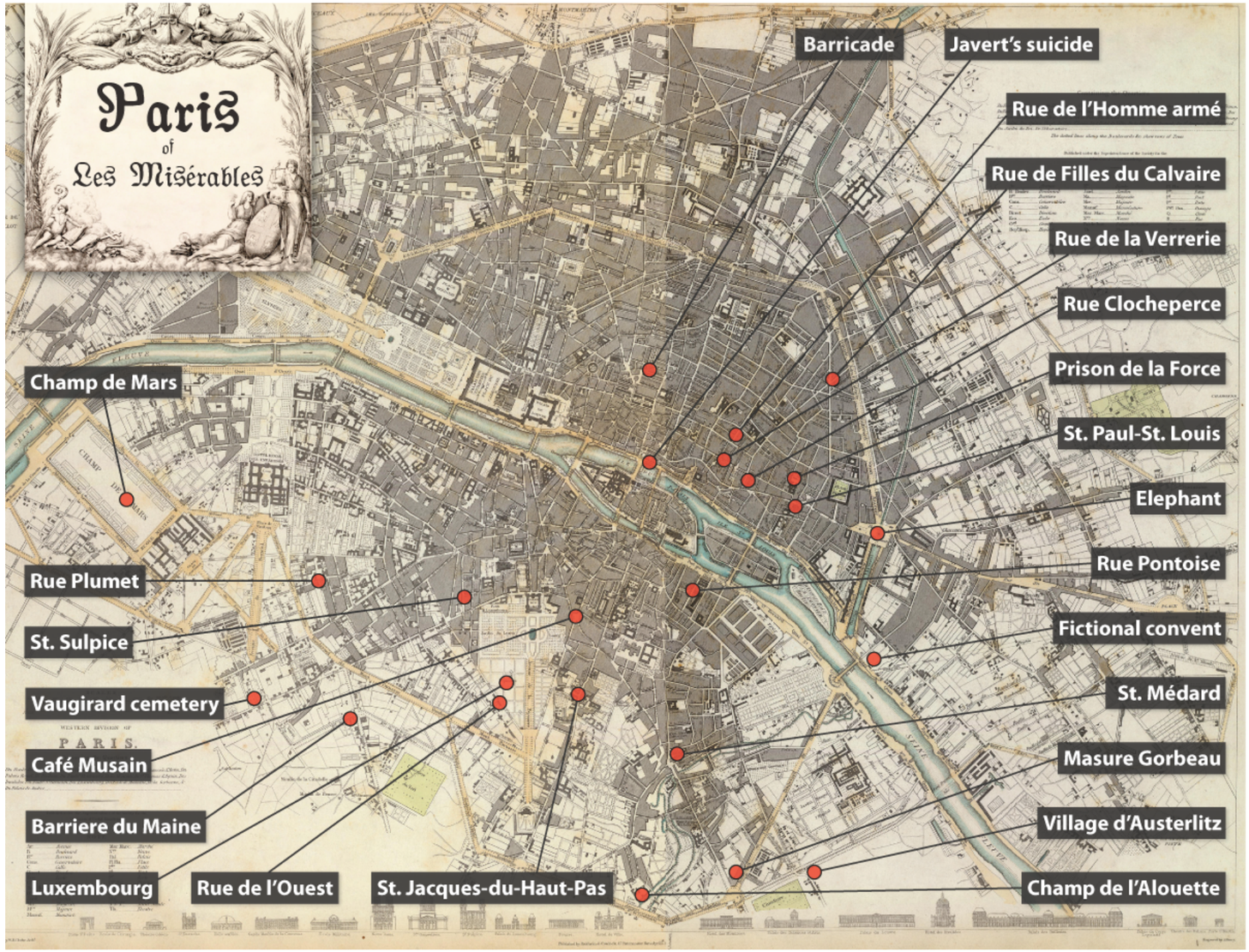


M. Taylor, Northwestern University

web.mmlc.northwestern.edu/re-visualizing-the-novel/

Data abstraction

- Example: *Les misérables*, V. Hugo, 1862 - novel
  - locations of the play: map



M. Taylor, Northwestern University

Domains

Domain, or sort, or type, like in typed programming languages:

- Algebraic approach: a domain is a *set* of *elements* together with a set of *operations* that can be performed on them
- Coarse classification [S.S. Stevens 1946]:
  - Nominal (categories, labels)
    - Operations: =, ≠
  - Ordinal (ordered set)
    - Operations: =, ≠, >, <
  - Interval (location of zero is arbitrary)
    - Operations: =, ≠, >, <, +, − (distance)
  - Ratio (zero fixed)
    - Operations: =, ≠, >, <, +, −, ×, ÷ (proportions)
  - Special case: Boolean
    - Operations: =, ≠, ¬, ∧, ∨

Domains

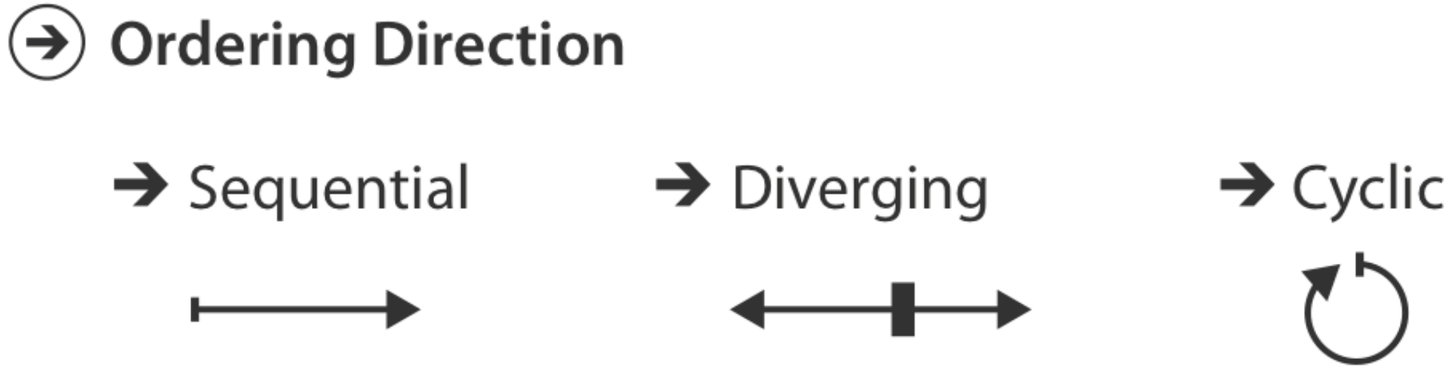
Domains in Vis adopt a simplified classification:

- Categorical (nominal)
  - set identity for individuals or classes
  - function: compare equality
  - *Fruit, Gender, Movie genres, File types*
- Ordinal (ordinal)
  - relative ranking of individuals
  - *Shirt size, Rankings, Qualitative scales*
- Quantitative (interval or ratio)
  - integer, real
  - precise arithmetic
  - *Length, Weight, Count, Date*

Domains

Ordering direction of ordinal domains:

- Sequential: homogeneous sequence from min to max
  - number of people in a country
- Diverging: two or more sequences that meet at a neutral point (zero)
  - elevation above & below sea level
- Cyclic: quotient algebra, modulo-X
  - time (minutes in an hour, hours in a day, days in a week / month)
  - angles



Domains

Common domains (example: *Tableau* classification)

- Character, string (categorical, but can be treated as ordinal)
- Boolean
- Number: decimal or whole (quantitative, ratio)
- Date (quantitative, interval)
- Date & Time (quantitative, interval)

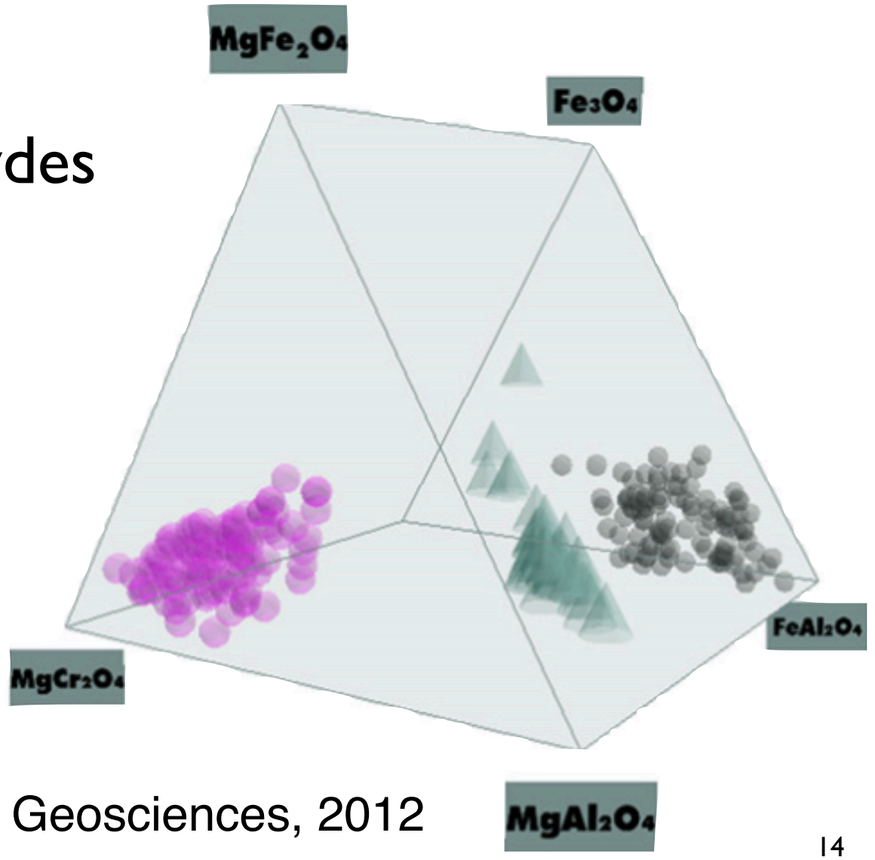
  - Date and time have a hierarchical structure:
    - second, minute, hour, day, month, year, decade, century, millennium, age
    - weeks group day but they are not aligned with months and larger groups

Items

- An *item* is an individual entity within a dataset
- structured collection of pieces of information:
  - an *identity* that uniquely identifies each item from the other items in the dataset
  - a set of *attributes* that provide relevant information about it
- both identity and attributes are specified by values from given domains
  - identity is determined by one or more *keys*: an item is identified by a combination of keys that is unique within the dataset
  - attributes store *values*
- distinction between attributes and keys not always determined a priori
- sometimes visualization is also aimed at finding a meaningful set of keys among attributes
- *implicit key*: order number in the collection of items forming a dataset

Positions

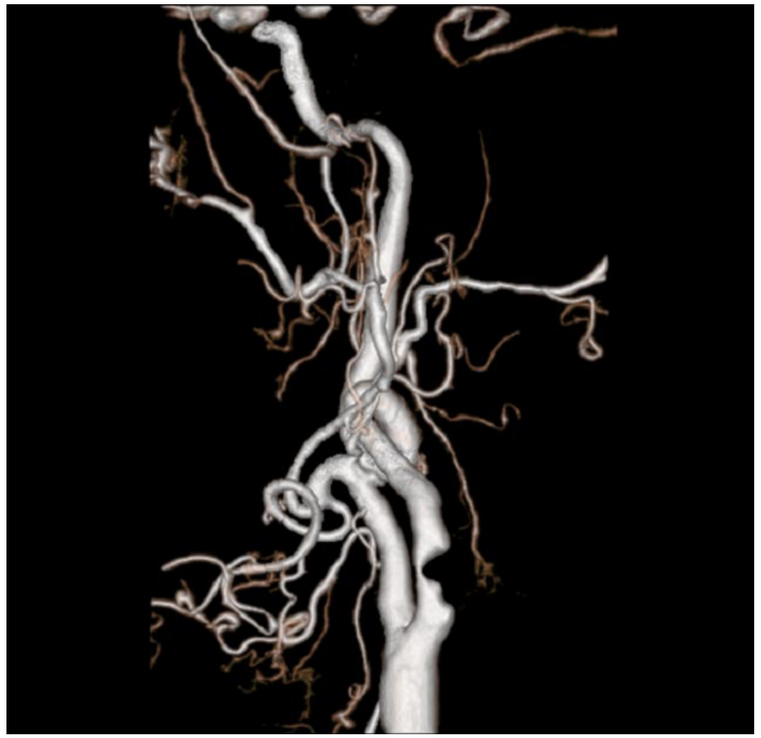
- Position: specific key / attribute referring to a place in a space endowed with a set of coordinates
  - e.g.: (latitude,longitude) in geographic space, (x,y,z) in 3D space, (azimuth,elevation) in horizontal coordinate system
- Embedding space is not necessarily physical
  - e.g.: rocks in a space that relates to the amount of oxydes they contain

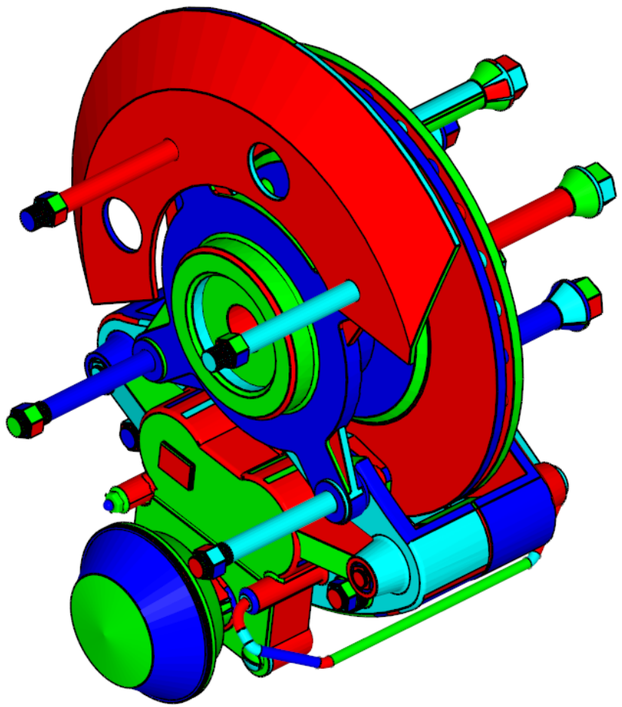


M-L- Ganuza et al., Computer and Geosciences, 2012

Geometry

- Geometry: specific, complex attribute assigning a *shape* to an item, in the context of some physical space
  - e.g.: boundaries of a country, layout of vessels from angiography, CAD data of a mechanical object



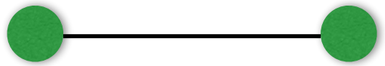


Classes

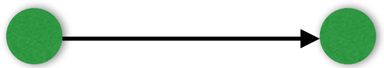
- Items can be arranged in classes
- A *class* is a group of items having a common value or range of values for some specific attribute(s)
  - ex.: all European countries, all members of the Pink team, all people with income in range [25KEuro,50KEuro), all bank customers with positive balance, …
- Classes may come with the dataset, but they are more often derived from analysis
  - class selectors may be derived from existing data or may require additional attributes

Links

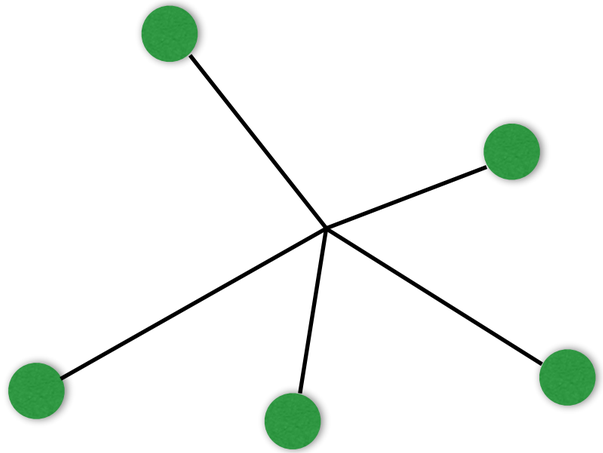
- Items can be connected by *links*
- Undirected link: connects two items with no order (a,b)=(b,a)
- Directed link: connects a source to a destination (a,b)≠(b,a)
- Hyperlink: connects more than two items (a,b,c,d,e)



undirected



directed



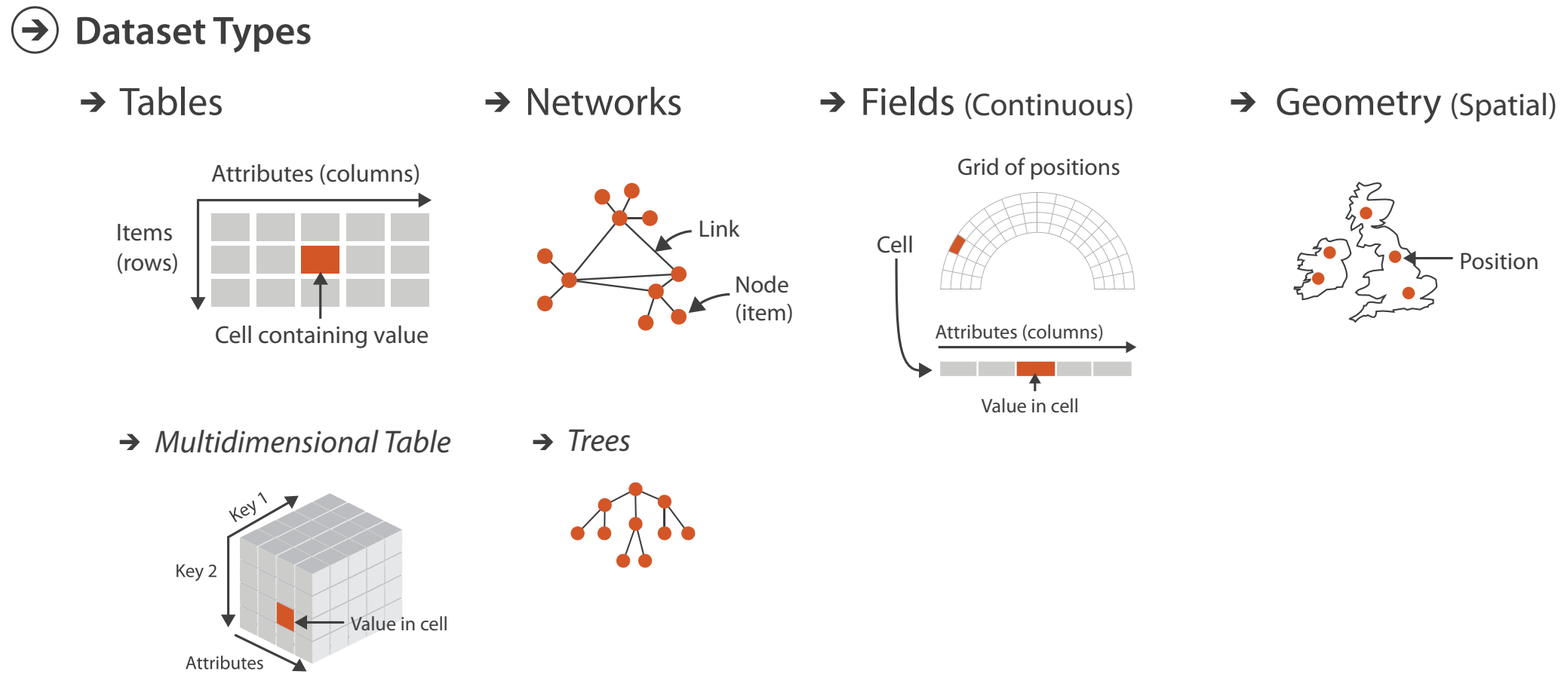
hyperlink

item

link

Dataset types

- Discrete datasets:
  - Tables
  - Networks & trees
  - Geometries
- Continuous datasets:
  - Fields (scalar, vector, tensor)
  - Discretization:
    - sampling
    - grid
    - reconstruction (interpolation / approximation)

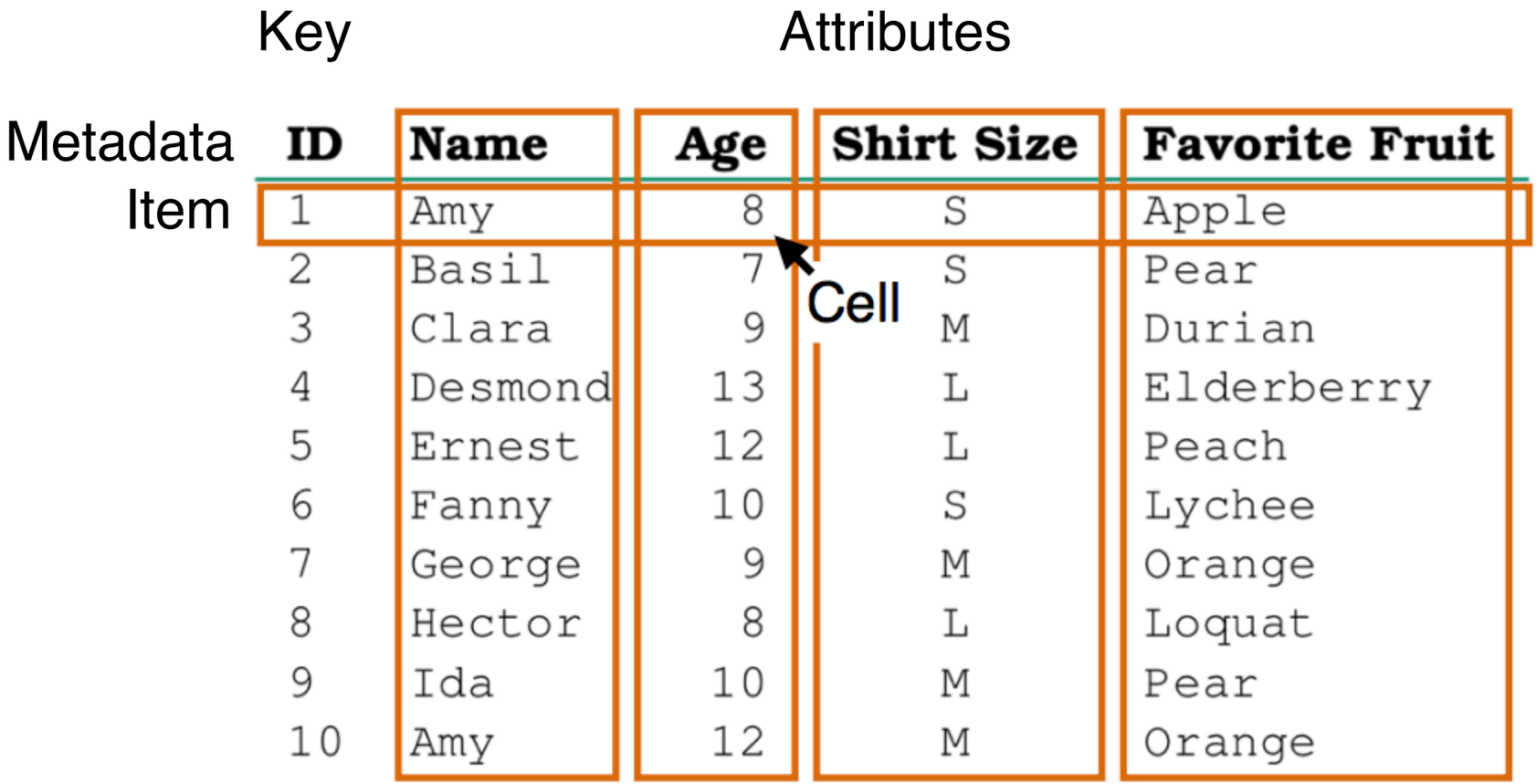


Tables

- *Table*: collection of items - items x attributes
- *Flat table*: structure of a two-dimensional array rows x columns
  - Row ↔ Item
  - Column ↔ Attribute
  - Key is implicit (but can be also one of the attributes)
  - No duplicates (each item appears once)
- *Multidimensional table*: structure of a n-dimensional array
  - Multiple keys (one for each dimension)
  - Last dimension can be used for differentiating attributes

Tables

- Flat table
- Metadata explain the *type* and *semantics* of attributes



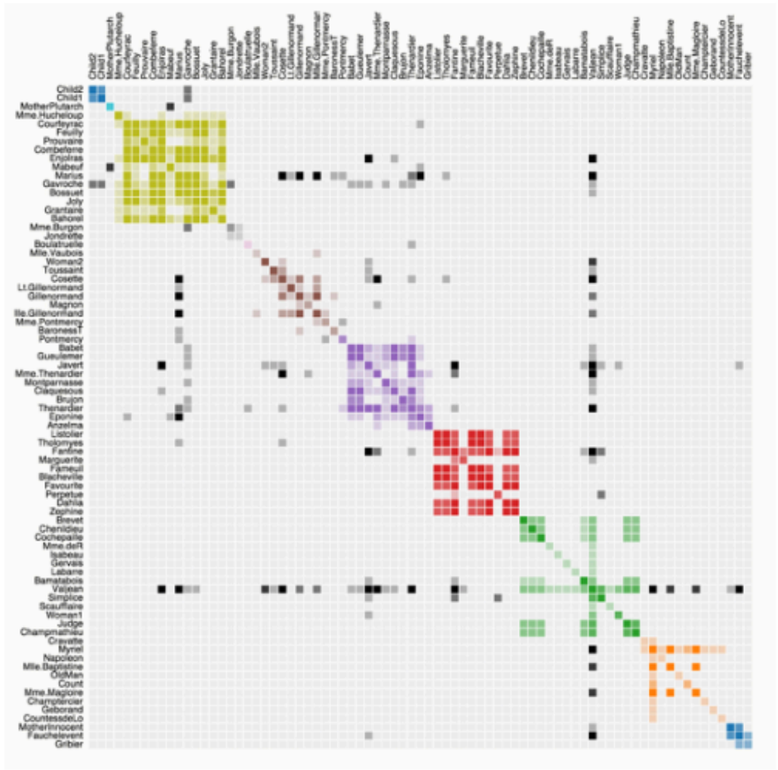
ID	Name	Age	Shirt Size	Favorite Fruit
1	Amy	8	S	Apple
2	Basil	7	S	Pear
3	Clara	9	M	Durian
4	Desmond	13	L	Elderberry
5	Ernest	12	L	Peach
6	Fanny	10	S	Lychee
7	George	9	M	Orange
8	Hector	8	L	Loquat
9	Ida	10	M	Pear
10	Amy	12	M	Orange

Networks & trees

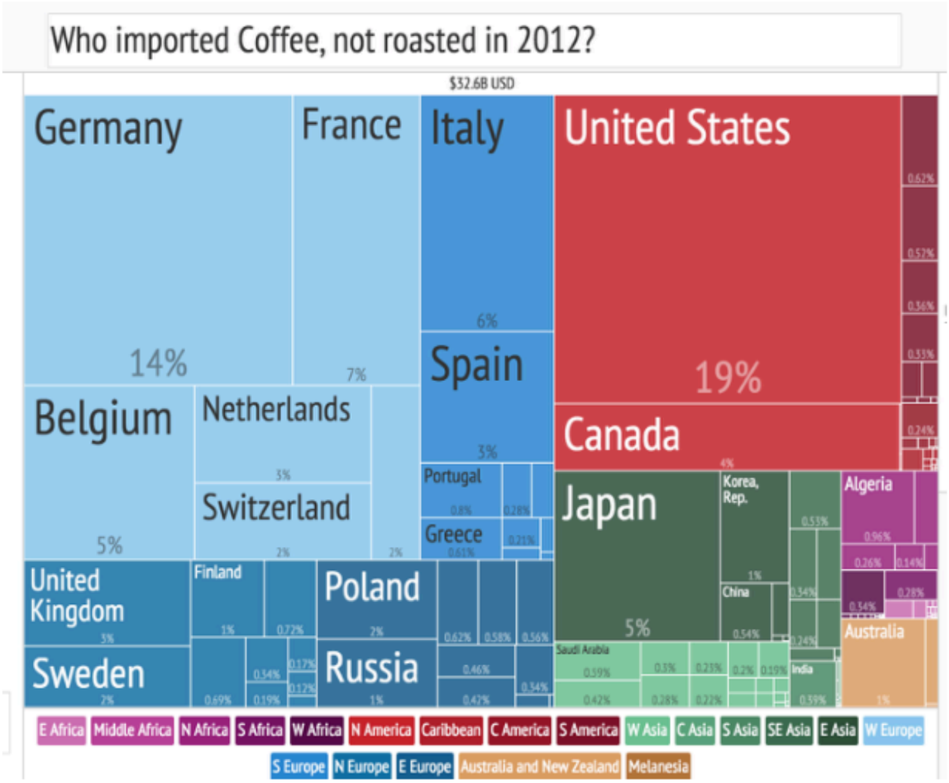
- Network: arrangement of items and links (graph: nodes and arcs)
- Tree: network without cycles
- Attributes can be attached to both items and links
  - arbitrarily many attributes on items
  - usually just "weights" on links (e.g.: cost, importance, number)
- Many different ways to visualize networks
  - depend on information that must be highlighted
  - not all method scale equally well with #nodes and #arcs
  - specific methods for trees

Networks & trees

Node-Link Diagram



Matrix



Treemap (Implicit Tree Visualization)

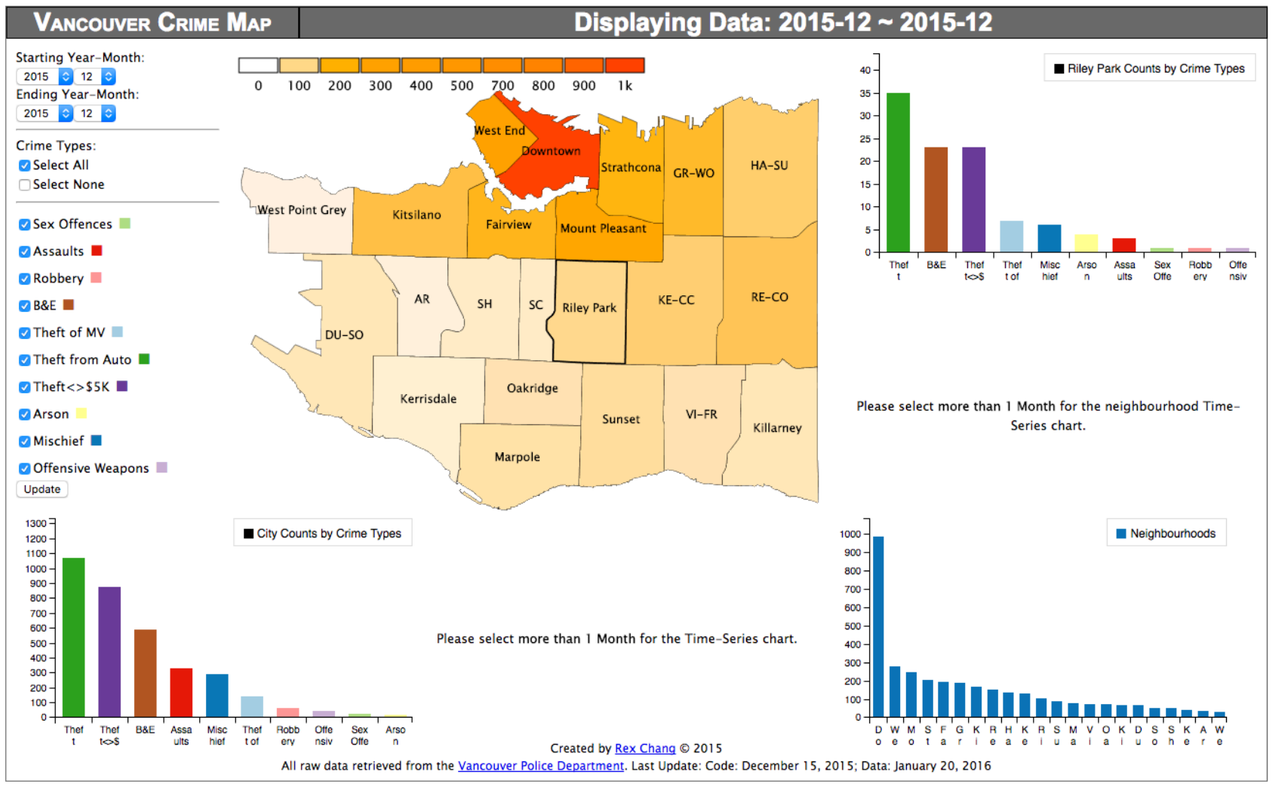
Geometric datasets

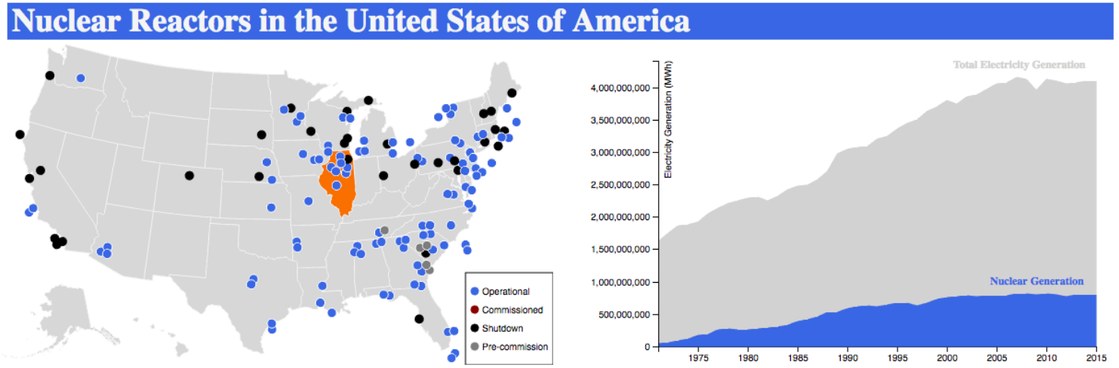
- Shape of items provide the main key for access (spatial key)
- Geometrical objects can be:
  - points (= positions in embedding space)
  - lines (polylines, splines)
  - regions (polygons)
  - surfaces
  - volumes (polyhedra)

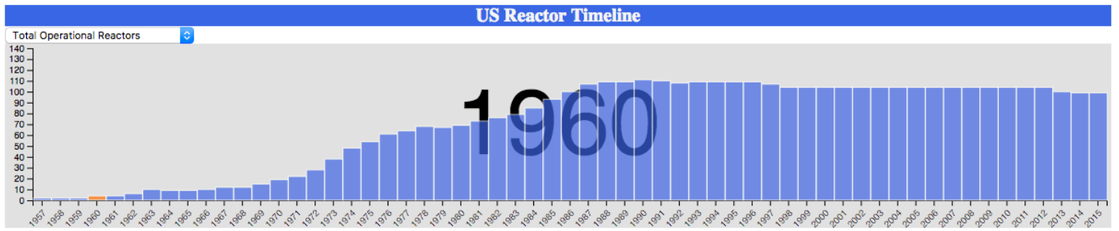
- Most relevant cases in Vis:
  - geographic entities (infovis & scivis)
  - reconstructions from medical data, geoleogical data, etc. (scivis only)

Geometric datasets

- Object-based access:
  - point at item, get shape
- Space-based access:
  - point at shape, get information about item

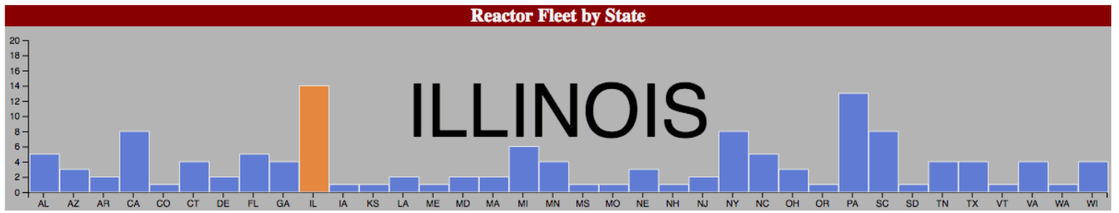






Reactor Information by Time

Grid Connection	Reactor Name	Type	Reactor Status in 2015	Location	Shutdown Date	Unit Power (MW)
1960	DRESDEN-1	BWR	Permanently Shutdown	MORRIS, IL	1978	197
1957	GE VALLECITOS	BWR	Permanently Shutdown	Pleasanton, Sunol, CA	1963	24
1957	SHIPPINGPORT	PWR	Permanently Shutdown	Shippingport, PA	1982	60
1960	YANKEE NPS	PWR	Permanently Shutdown	ROWE, MA	1991	167

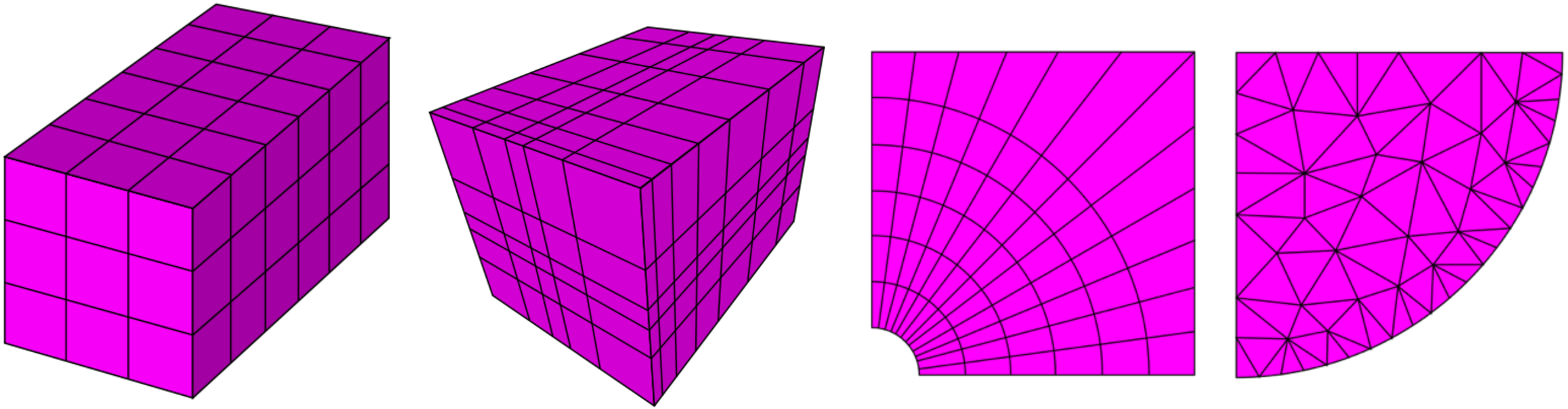


Fields - scivis

- Continuous data $f : \Omega \rightarrow \mathbb{R}^n$
  - Domain Ω usually a subset of some spatial domain: $\mathbb{R}^2$, $\mathbb{R}^3$, or a manifold surface
  - Range $\mathbb{R}^n$ generic for a set of quantitative measures, possibly heterogeneous
- Discrete sampling & reconstruction
  - function f evaluated (measured or simulated) at a finite set of locations in Ω
  - Ω is subdivided into cells to form a *grid*
  - values are assigned to either cells or vertices of the grid
  - values are interpolated on the basis of sampled data
- Examples:
  - Ω a country, f temperature / pressure / wind velocity measured at weather stations
  - NMR: Ω the human body, f density from scan
  - Digital picture: Ω the CCD sensor, f measures intensity of light in primary colors

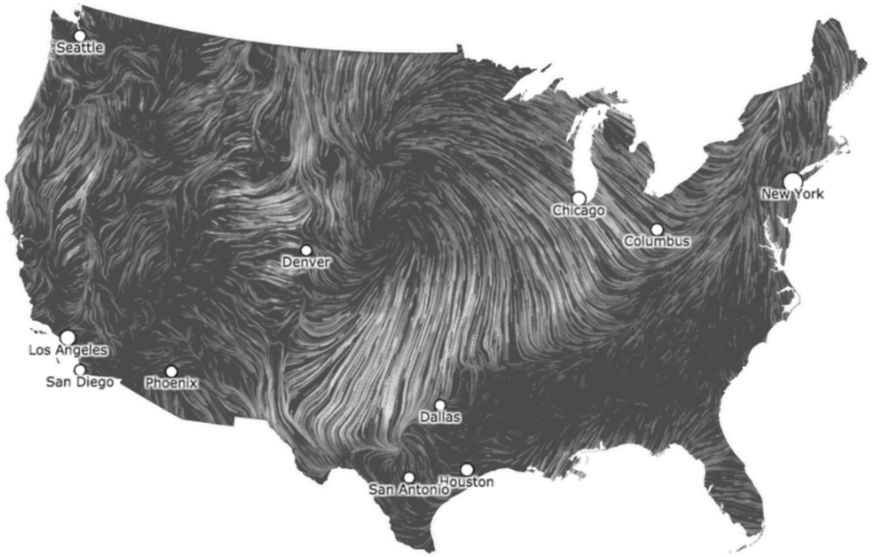
Fields

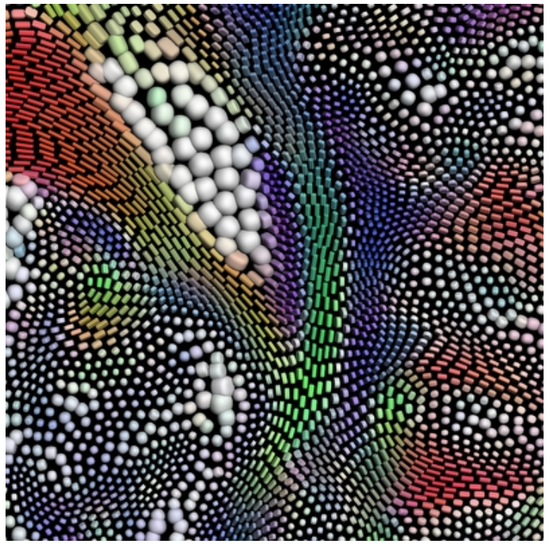
- Grids:
  - Uniform: implicit geometry and topology
  - Rectilinear: implicit topology, rectilinear geometry, non-uniform sampling
  - Structured: implicit topology, curvilinear geometry
  - Unstructured: explicit topology & geometry, arbitrary data distribution



Fields

- *Scalar field*: f measures numbers (integer, real)
  - e.g.: temperature, pressure, X-ray, …
- *Vector field*: f measures vectors (direction & norm)
  - e.g.: velocity, acceleration, wind, electomagnetic field
- *Tensor field*: f measures tensors (one matrix per sample point)
  - e.g., curvature, mechanical stress, NMR diffusion





Unstructured data

- no predefined data model
- text-heavy, interspersed with facts (measures, dates, times, locations)
- video, images

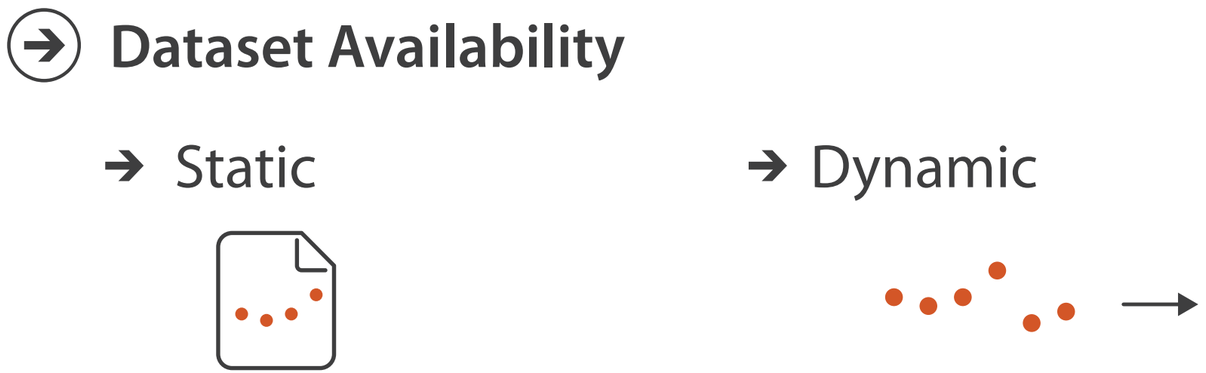
- Translate into structured data
  - requires data analysis and interpretation
  - natural language processing
  - text mining (keywords, concepts categories)

} data analytics / mining / learning

- Beyond scope of this course
  - we consider structured data
  - only easy structuring tasks: cleaning, aggregation, statistics

Dataset availability

- Static data (offline):
  - all data are available at the beginning
  - can be processed offline (data preparation)
  - can be loaded all together
- Dynamic data:
  - data come as a *stream*
  - items can appear / disappear
  - attributes can change their values



Dataset availability

- Time series:
  - data recorded over time: time gives a primary key for access
  - can be static: recorded over a period of time and processed later on
  - can be dynamic: acquired and processed in real time
- Large datasets:
  - do not fit in memory or in the application
  - available as static but processed as dynamic
  - sorted and streamed
  - Vis app can only offer a detailed view on a window (subset of the stream)
  - plus possibly a compendium of the whole

Dataset and data types - summary

Data and Dataset Types

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Data model vs. Conceptual model

- Data Model: Low-level description of the data
  - Set with operations, e.g., floats with +, -, /, *
- Conceptual Model: Mental construction
  - Includes semantics, supports reasoning

Data	Conceptual
1D floats	temperature
3D vector of floats	space

Data model vs. Conceptual model

- From data model...
  - 32.5, 54.0, -17.3, ... (floats)
- using conceptual model...
  - Temperature
- to data type
  - Task: physics experiment - Continuous to 4 significant digits (Quantitative)
  - Task: taking a shower - Hot, warm, cold (Ordinal)
  - Task: toasting bread - Burned vs. Not burned (Nominal/Boolean)

Combinations, derived data

- Networks can have attributes
- Attributes have hierarchies
- Data types can be transformed

Real life is complicated...

Next Time

- to read
  - VAD Ch. 3: Task Abstraction