Nested Model | **Domain > Abstraction > Idiom > Algorithm**

Who are the target users? | **What** is shown → Data Abstraction; **Why** is it shown → Task Abstraction | how is it shown? **how to draw** → visual encoding; **how to manipulate** → interaction | efficient computation

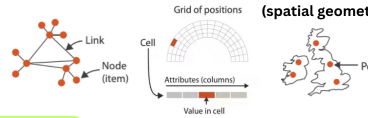
Data Abstraction (WHAT)

Data types: structural / mathematic interpretation of data

Item: Individual entity, discrete. *Ex. person, car, stock (row)*
Attribute: property observed, measured, logged. *Ex. age, name (col)*
Links: Relationship between two items *Ex. friendship on facebook*
Positions: spatial data, location in 2D or 3D *Ex. lat/lon, pixels*
Grids: sampling strategy for continuous data

Dataset Types

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



(spatial geometry)

Attribute types

- Categorical
- Ordered
  - Ordinal
  - Quantitative

* year is quantitative!!!

Ordering Direction: Sequential, Diverging, Cyclic

Dataset Availability: Static, Dynamic

3 operations for data abstraction

1. **identify dataset/attribute types**
2. **identify cardinalities (# of items, # of levels, range)**
3. **consider whether to transform data**

Task Abstraction (WHY)

- **{action, target} pairs**

Actions: Analyze, Search, Query
Targets: What's being acted on

ex.
- **discover distribution**
- **compare trends**
- **locate outliers**
- **browse topology**

Action Explanation

ANALYZE → Consume (→ Discover, → Present, → Enjoy); Produce (→ Annotate, → Record, → Derive)

SEARCH

	Target known	Target unknown
Location known	Lookup	Browse
Location unknown	Locate	Explore

Query: → Identify, → Compare, → Summarize

how much of the data matters?
One → identify
Some → compare
All → summarize

Target Explanation

Targeting ALL DATA: → Trends, → Outliers, → Features

Attributes: → One (→ Distribution, → Extremes); → Many (→ Dependency, → Correlation, → Similarity)

Network Data: → Topology, → Paths

Spatial Data: → Shape

Interactive views (how to handle complexity)

Manipulate: Change, Select, Navigate

Facet: Juxtapose, Partition, Superimpose

Change
- Re-encode
- Change parameters: widgets & controls
- Change order/arrangement (bar chart)
- Animated transition

Reduce: → Filter, → Aggregate, → Embed

Navigate: → Zoom (Geometric), → Pan/Translate, → Constrained

Pros of interactive views
- Flexible, powerful intuitive
- Flexible transitions provide support

Cons of interactive views
- time cost
- remembering previous state
- users may not interact as planned

View coordination design choices

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

Reduce Items and Attributes

Filter **pro**: straightforward, intuitive. **con**: out of sight, out of mind

Embed (combine focus + context info within single view)

Elide Data
Distant Geometry

Aggregate Data

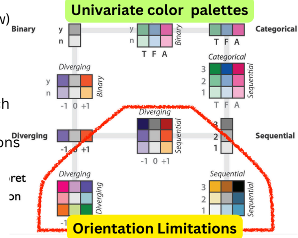
group of elements is represented by a smaller # of derived elements. pro: inform about whole set. con: difficult to avoid losing signal. ex. histogram, static item. aggregation, scented widget. ex fish eye distortion

Colors

diverging (two colors at each end), **categorical**, **sequential (lighter to darker)**

luminance: how bright (b/w)
saturation: how colorful
hue: what color

Univariate color palettes



- rainbow is bad default!
- saturation and luminance not separable from each other, or transparency.
- Small regions need HIGHER saturation, large regions need lower saturation.

Multivariate Tables

→ Express Values; → Axis Orientation (→ Rectilinear, → Parallel, → Radial)
→ Separate, Order, Align Regions (→ Separate, → Order, → Align); → Layout Density (→ Dense)
→ 1 Key *List*; → 2 Keys *Matrix*

- **key**
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - **multidimensional tables**: multiple keys

→ **value**
→ **totals**

query: start with nothing, add in elements
filters: start with everything, remove elements.

Orientation Limitations

Rectilinear: scalability w.r.t to axes (2 axies yes, 3 hard to use, 4 impossible)
Parallel: Unfamiliarity
Radial: Perceptual limits. angular position + radial distance, non-uniform sector size.

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

Diverging:
- When data has "midpoint" use a neutral colour (W, B, Y)
- saturated for endpoints

Sequential: increase lumination or saturation
Cyclic: a cyclic multihue
Bivariate: Be cautious since it's difficult to interpret when there's multiple levels in each direction

Rules of Thumb

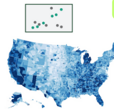
No unjustified 3D –Power of the plane –Disparity of depth –Occlusion hides information –Perspective distortion dangers –Tilted text isn't legible • No unjustified 2D • Eyes beat memory • Resolution over immersion • Overview first, zoom and filter, details on demand • Responsiveness is required • Function first, form next

Maps

- Geographic mark: Poly Marks w/ 2D boundary
- encode: **position/order only**

CANNOT use size, orientation, symbol/shape
Thematic maps: show variability of attribute
Population maps: Most attributes just show where people live, consider when to normalize by population density..

Chloropleth



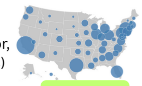
given spatial data, table w/1 quant. attribute/region
- encoding:
  - mark: poly
  - 2D shared boundary
  - position/order

pros: easy to read, well established, data often collected and aggregated by geographical regions

cons: region size affects vis, color has huge influence.

unavailable: size, orientation, shape/symbol

Symbol Map



point mark or glyph, 1 symbol per region, size encoding. **free channels:** symbol/shape, color, orientation (for asymm glyphs) **pros:** somewhat intuitive to read/understand. can mitigate roblems with region size vs data salience. **cons:** possible occlusion/overlap, complex glyphs require explanation training.

Contiguous Cartogram

change all poly marks on map simultaneously.
- poly marks, 2D shared boundaries

encoding: 2D size for quant attr, order, maintain relative poly pos. algo to create new marks, maintain shared & order.

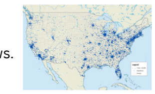
non-contiguous cartogram

point marks, 2D shared boundaries. **encoding:** quant attr. w/ 2D size, relative locations of poly, w/ order. **unavailable:** position, orientation, symbol/shape. **free:** color

contiguous pros/cons

pros: can be intriguing/engaging. **cons:** need familiarity of original dataset + memory. Major distribution is problematic, difficult to extract exact quantities.

Dot density maps



place def point marks where density in region shows.
- 2 mark types at the same level → (point, poly)
- **unavailable:** size, shape, orientation, position
- **pros:** 1-1 straightforward to understand, avoid chloropeth non-uniform region size prob
- **cons:** normalization. difficult to extract quantities. performance: rendering too many dots is too slow. many - one (density). May confuse if random positions assumed to be meaningful.

non-contiguous pros/cons

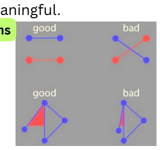
pros: stylized marks easier to understand then distorted contiguous shapes. good base for combining w/ other approaches. equal area approach can mitigate size disparity problem of cartograms **cons:** need familiarity w/ dataset, quasi-geo pos need explanation. equal area version impose distortions if original regions have disparate size.

Network Tasks

Topology based tasks

– find paths – find (topological) neighbors – compare centrality/importance measures – identify clusters / communities

Node-link diagrams



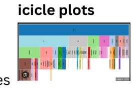
level 1: nodes, point notes
level 2: links, segment connecting marks
unavailable: position, order, length (1D size).

good node-link layouts:
- **minimize:** edge crossing, node overlaps, distance b/w neighbours, total drawing area, edge bends
- **maximize:** angular dist. b/w edges. aspect ratio disparities.
- **emphasize symmetry:** F(layout) = a*[crossing counts] + b*[drawing space used] + ...

Is it a network of a tree?

node link diagrams: network, tree
adjacency matrix: network, tree
implicit (spatial pos): tree
enclosure: tree

icicle plots



Implicit Tree Layouts

treemap - containment, only leaves visible. **sunburst** - position (radial), inner nodes and leaves visible. **icicle plots -** pos (rectilinear) inner nodes and leaves visible.

General limitations:

- Diverging length not aligned
- Hard to make unaligned comparisons over time
- No horizontal / vertical axis
- Difficult to find outliers / extremes

space not used effectively, lots of space
- watch out for occlusion vs stacked chart → requires prior knowledge
- heavy reliance on text to extract info
- overlapping info (like dots & numbers) → hard to compare areas
- overall information density is low, lots of space
- comparing heights is difficult with radial bar shape since does not make use of aligned position channels → high cognitive load
- area differences are hard to accurately decode → high cognitive load
- comparisons for absolute values is difficult b/c small multiples not displayed on either axis → need to mentally divide left & right bars

Analyzing Marks and Channels

Channels: control appearance of marks

→ **Magnitude** Channels: **Ordered attributes**

- 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)

→ **Identity** Channels: **Categorical Attributes**

- Spatial region
- Color hue
- Motion
- Shape

Best ↔ Effectiveness ↔ Least; Same

glyphs more then one mark for each item.

levels

hierarchical structure of marks. shows info on multiple scales

level 1

2 possible coordinate systems

vertical, horizontal, origin, 1D size: length height; Polar / radial, 1D size: width length, angular, radial distance, origin

mutually exclusive, can't use both at once

position: → Horizontal (1D) → Vertical (1D) → 2D

size: → Length /Width (1D) → Area (2D)

examples

scatterplot — points: H&V pos/order; points: H&V pos/order, size (2D); points: H&V pos/order, size (2D), color
bar/lollipop chart — segments: V pos/order/size; segments: V pos/order/size. points: H&V pos/order; segments: H pos/order/size. points: H&V pos/order
line chart — path: H&V pos/order; path: H&V pos/order. points: H&V pos/order; path: H&V pos/order. points: H&V pos/order
filled area chart — boundary (closed path), interior; poly: H&V pos/order; poly: H&V pos/order. points: H&V pos/order

Shared boundary constraints

can't independently change pos, order, size, orientation, symbol

Channels: geometric primitives

1 item	special case of path 2 items	many items
point	segment	path, poly

marks for links

→ Connection: segment
→ Containment: poly

tree map — channel: 2D position shared
matrix/grid — channel: 2D position shared
stacked bar chart — glyphs: multiple segments; channel: vert shared; 1D horizontal shared positions, 1D vertical shared positions

vert shared

Types of Channels

Magnitude Channels: Ordered Attributes
- 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)

Identity Channels: Categorical Attributes
- Spatial region
- Color hue
- Motion
- Shape

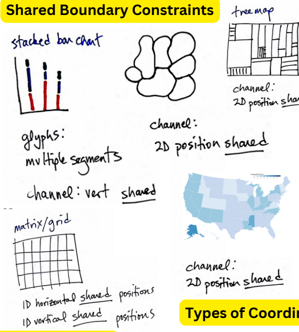
Best ↑ Effectiveness ↓ Least; Same / Same

expressiveness
– match channel and data characteristics

effectiveness
– channels differ in accuracy of perception

distinguishability
– match available levels in channel w/ data

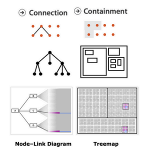
Shared Boundary Constraints



- marks that touch by design: shared boundary constraints
  – cannot independently change position, order, size, orientation, shape/shape
    - moving one would affect others
  – consider separately for each direction (horiz, vert) or intrinsically combined (2D)

Link marks: Connection and containment

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

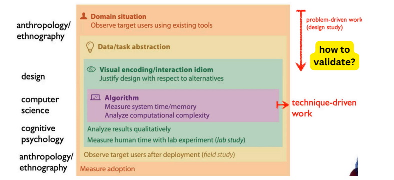


Nested model: Four levels of visualization design

- *domain situation*
  – who are the target users?
- *abstraction*
  – translate from specifics of domain to vocabulary of visualization
    - **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).]

Very iterative process



Aggregation

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

Coordinate views: Design Choice Interaction

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

A) Fully Redundant (same encoding, all data items shared)
B) Overview/Detail - Same Form (same encoding, subset of data items shared)
C) Small Multiples (same encoding, data items partitioned across views)
D) Multiform (different encoding, all data shared)
E) Overview/Detail - Multiform (different encoding, subset of data shared) F) No Linkage (different encoding, no data shared)

Types of Coordinate Symbols

- Node–Link Diagrams — Connection Marks — ✓ NETWORKS ✓ TREES
- Adjacency Matrix — Derived Table — ✓ NETWORKS ✓ TREES
- Enclosure — Containment Marks — ✓ NETWORKS ✓ TREES

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

Task Abstraction

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

{action, target} pair action can be analyze, search, and query

What? Why? How?

{action, target} pair action can be analyze, search, and query

Geographic Map

poly marks with 2D shared boundaries, encode:

- position / order channels, using given spatial data

cannot use other channels to show other attributes, locked down as unavailable:

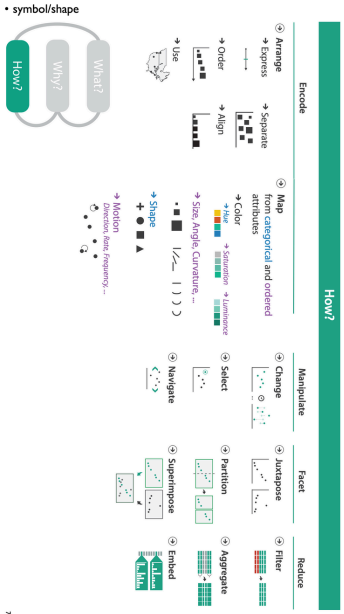
- size
- orientation
- symbol/shape

Filter

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

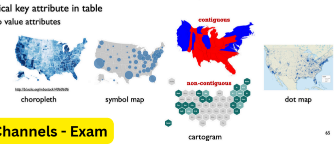
Choropleth map

- **pros** – easy to read and understand – well established visualization (no learning curve) – data is often collected and aggregated by geographical regions • **cons** – most effective visual channel (position) used "just" for geographic location • reasonable if understanding spatial distribution / patterns is a central task – visual salience of color coding depends on region size • not true importance wrt attribute value • large regions appear more important than small ones – color palette choice has a huge influence on the result



Thematic Maps

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



Analyzing Marks & Channels - Exam

Are you plotting every individual record? That's Level 1.
Are you plotting grouped data, like each country's total? That's Level 2.
Are you showing one mark for everything (e.g., global totals)? That's Level 3.

points (0D), lines (1D), areas (2D)

is used across multiple components or views of a visualization?.

What's being encoded as part of position/order? ex (yearly income per person)

What's being encoded horizontally as part of size? ex (crime rate))

1D Size (Length/Height)
Example: A bar chart where each bar's height (or length) is proportional to a data value.
What's Being Encoded: A single dimension of scale—longer bar = larger value, shorter bar = smaller value.
Perception: Humans are good at comparing lengths, so 1D size is perceived accurately.

2D Size (Area)
Example: A bubble chart where the circle area represents a data value, a choropleth map that shades regions with larger or smaller areas.
What's Being Encoded: 2D space—bigger circle = larger value, smaller circle = smaller value.
Perception: People are less precise at comparing areas, especially when shapes differ(circles vs. squares).

Symbol (or Shape)
Definition: The form of the mark (circle, square, triangle, star, etc.).
Typical Uses:
Categorical Distinctions: Assigning different shapes to separate categories. For instance, circles for one product type, squares for another.
• **Limited Number of Categories:** Symbol encoding is most effective if you have only a few categories (usually up to 4–6) so viewers can easily distinguish them.
• **Example:** A scatter plot using different shapes for different experimental groups.

Orientation
Definition: The rotational angle or direction in which a mark is displayed.
Indicating direction or flow (e.g., wind direction arrows).
Distinguishing categories by different rotations if you only have a few categories.
Highlighting trends that have a directional component (e.g., up/down for increases/decreases).
Example: A small arrow rotated to show "increasing" (pointing up) vs. "decreasing" (pointing down).

Symbol Map

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

Contiguous cartogram

pros – stylized marks usually easier to understand than distorted contiguous shapes – can serve as good base for combining with other approaches • especially equal-area versions with uniform shapes – equal-area approach can mitigate size disparities problems of cartograms • **cons** – still requires some familiarity with original dataset – quasi-geographic positioning may need some explanation – equal-area versions impose substantial distortions if original regions have very disparate size

Dot density maps

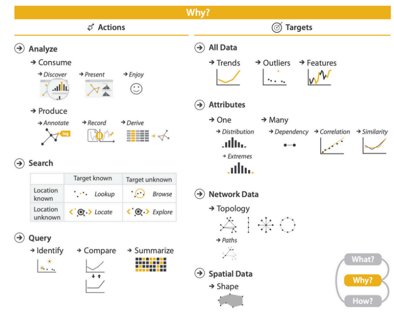
- **pros** – one-to-one is straightforward to understand • **cons** – same non-uniform region size problems as choropleth – challenge: normalization, just like choropleths • show population density (correlated with attribute), not effect of interest – perceptual disadvantage: difficult to extract quantities – performance disadvantage: rendering many dots can be slow – many-to-one (density) may confuse if random positions assumed to be meaningful

Non-contiguous cartogram

- **pros** – stylized marks usually easier to understand than distorted contiguous shapes – can serve as good base for combining with other approaches • especially equal-area versions with uniform shapes – equal-area approach can mitigate size disparities problems of cartograms • **cons** – still requires some familiarity with original dataset – quasi-geographic positioning may need some explanation – equal-area versions impose substantial distortions if original regions have very disparate size

Rules of Thumb

No unjustified 3D –Power of the plane –Disparity of depth –Occlusion hides information –Perspective distortion dangers –Tilted text isn't legible • No unjustified 2D • Eyes beat memory • Resolution over immersion • Overview first, zoom and filter, details on demand • Responsiveness is required • Function first, form next



Spatial Aggregation

MAUP: Modifiable Areal Unit Provlem.
- **changing boundaries of cartographic regions can yield dramatically different results**

Clustering

- classification of items into similar bins
  – based on similiarity measure
    - Euclidean distance, Pearson correlation
  – partitioning algorithms
    - divide data into set of bins
    - # bins (k) set manually or automatically
  – hierarchical algorithms
    - produce "similarity tree" (dendrograms): cluster hierarchy
    - agglomerative clustering: start w/ each node as own cluster, then iteratively merge
- cluster hierarchy: derived data used w/ many dynamic aggregation idioms
  – cluster more homogeneous than whole dataset
    - statistical measures & distribution more meaningful

Color

- first rule of color: do not talk about color!
  – color is confusing if treated as monolithic
- decompose into three channels
  – ordered can show magnitude
    - luminance: how bright
    - saturation: how colorful
  – categorical can show identity
    - hue: what color
- channels have different properties
  – what they convey directly to perceptual system
  – how much they can convey: how many discriminable bins can we use?

Luminance / Saturation / Hue

Channel Effectiveness

cardinality: how many possible values there are.

accuracy: how precisely can we tell the difference between encoded items? **discriminability:** how many unique steps can we perceive? **separability:** is our ability to use this channel affected by another one? **popout:** can things jump out using this channel?

Aggregate

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

Multivariate Tables

Keys & Values

→ Tables

- **key**
  –independent attribute
  –used as unique index to look up items
  –simple tables: 1 key
  –multidimensional tables: multiple keys
- **value**
  –dependent attribute, value of cell
- **classify arrangements by key count**
  –0, 1, 2, ...

Attributes (columns), Items (rows), Cell containing value
→ Multidimensional Table — Key 1, Key 2, Attributes, Value in cell

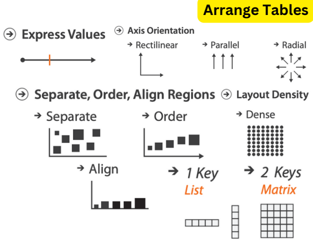
→ 0 Keys → 1 Key List → 2 Keys Matrix
→ Express Values

Regions: Separate, order, align

→ Separate → Order → Align → 1 Key List → 2 Keys Matrix

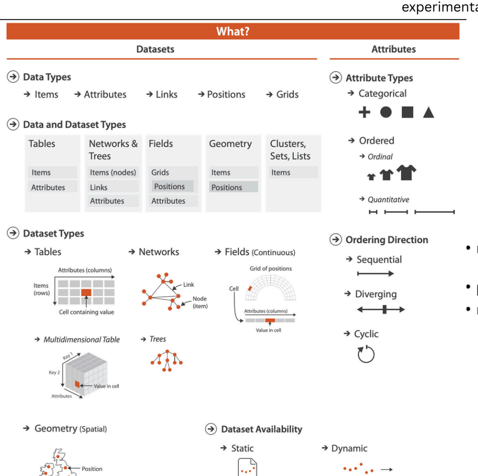
- separate into regions by any attribute: categorical or ordered
  – **regions**: contiguous bounded areas distinct from each other
  – no conflict with expressiveness principle for categorical attributes
  – one glyph/mark per region
- then can use ordered attribute to order and align regions
  – order in one direction (eg horiz), then align other direction (eg vert)
  – align: impose shared coordinate frame so absolute position has meaning
    - difference between value and 0 point along axis

Arrange Tables



Orientations

- rectilinear: scalability wrt #axes
  - 2 axes best, 3 problematic, 4+ impossible
- parallel: unfamiliarity, training time
- radial: perceptual limits
  – polar coordinate asymmetry
    - angular position lower precision than rectilinear position
    - radial distance (length) lower precision than aligned rectilinear position
    - nonuniform sector size (width & area), depending on radial distance
    - frequently problematic -- but sometimes can exploit on purpose!
      – for 2 attribs of unequal importance
  – strength: cyclic representation (when matching data)



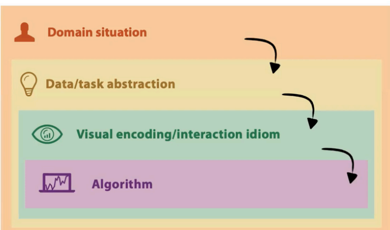
CPSC 447

Week 2: Nested Model

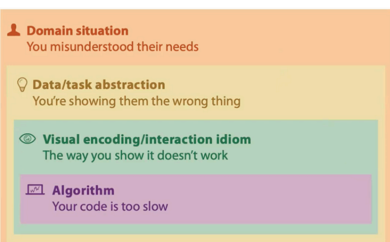
Analysis Framework

- *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

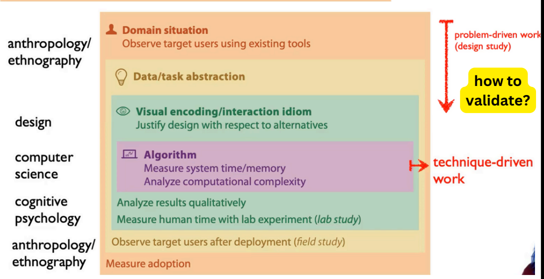
Nested model (Cascading model)



Very iterative process



what's wrong?



how to validate?

Week 2: Data Abstraction

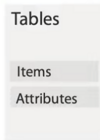
Semantics

- semantics: real-world meaning
- data types: structural or mathematical interpretation of data
  - item, link, attribute, position, (grid)
  - different from data types in programming!

Data types

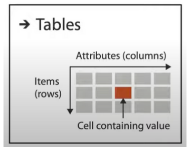
- item: individual entity, discrete
  - eg patient, car, stock, city
  - "independent variable"
- attribute: property that is measured, observed, logged...
  - eg height, blood pressure for patient
  - eg horsepower, make for car
  - "dependent variable"
- links
  - express relationship between two items
  - eg friendship on facebook, interaction between proteins
- positions
  - spatial data: location in 2D or 3D
  - pixels in photo, voxels in MRI scan, latitude/longitude
- grids
  - sampling strategy for continuous data

Dataset Types

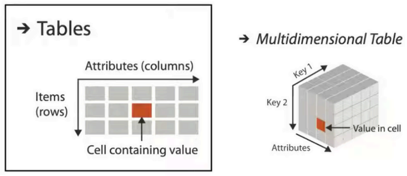


table

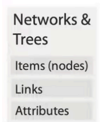
- flat table
  - one item per row
  - each column is attribute
  - cell holds value for item-attribute pair



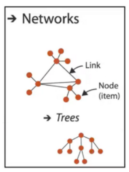
multidimensional table



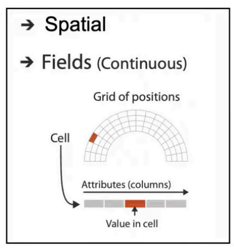
network/graph



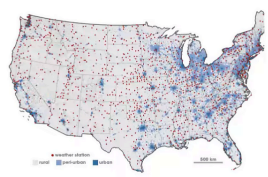
- network/graph
  - nodes (vertices) connected by links (edges)
  - tree is special case: no cycles
    - often have roots and are directed



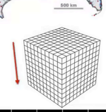
Spatial fields

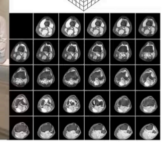


- attribute values associated w/ cells
- cell contains value from continuous domain
  - eg temperature, pressure, wind velocity
- measured or simulated

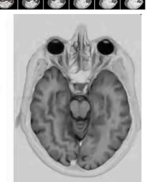


- major concerns
  - sampling: where attributes are measured
  - interpolation: how to model attributes elsewhere
  - grid types
- major divisions
  - attributes per cell: scalar (1), vector (2), tensor (many)

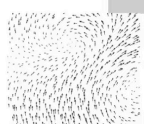




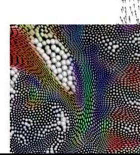
1 attribute — scalar

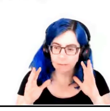


2 attributes — vector



many attributes — tensor

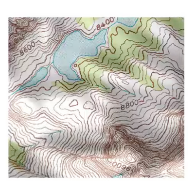




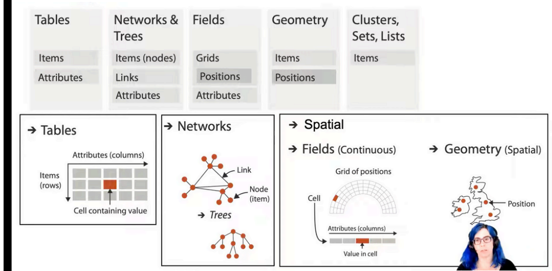
70

Geometry

- shape of items
- explicit spatial positions / regions
  - points, lines, curves, surfaces, volumes
- boundary between computer graphics and visualization
  - graphics: geometry taken as given
  - vis: geometry is result of a design decision



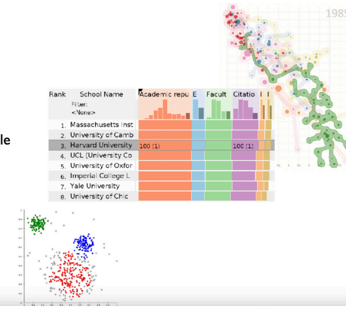
FULL LIST OF DATASET TYPES



Collections

Collections

- how we group items
- sets
  - unique items, unordered
- lists
  - ordered, duplicates possible
- clusters
  - groups of similar items



Data Types

→ Items → Attributes → Links → Positions → Grids

- which classes of values & measurements?
- categorical (nominal)
  - compare equality
  - no implicit ordering
- ordered
  - ordinal
    - less/greater than defined
  - quantitative
    - meaningful magnitude
    - arithmetic possible

Attribute Types

→ Categorical → Ordered

→ *Ordinal* → *Quantitative*

Ordering Direction

→ Sequential → Diverging → Cyclic

Dataset Availability

→ Static → Dynamic

- translate from domain-specific language to generic visualization language
- identify dataset type(s), attribute types
- identify cardinality
  - how many items in the dataset?
  - what is cardinality of each attribute?
    - number of levels for categorical data
    - range for quantitative data
- consider whether to transform data
  - guided by understanding of task

categorical
ordinal
quantitative

Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07
32	7/16/07	2-High	Medium Box	0.6	7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07

Data Model vs Conceptual Models

- data model
  - mathematical abstraction
    - sets with operations, eg floats with * / - +
    - variable data types in programming languages
- conceptual model
  - mental construction (semantics)
  - supports reasoning
  - typically based on understanding of tasks [stay tuned!]
- data abstraction process relies on conceptual model
  - for transforming data if needed

- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready
  - above freezing, below freezing: categorical
    - task: decide if I should leave the house today

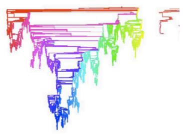
Derived Attribute

They can help reveal patterns or relationships not immediately obvious in the raw data.

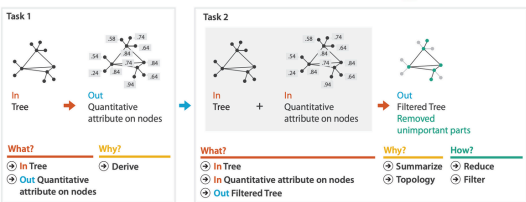
- derived attribute: compute from originals
  - simple change of type
  - acquire additional data
  - complex transformation

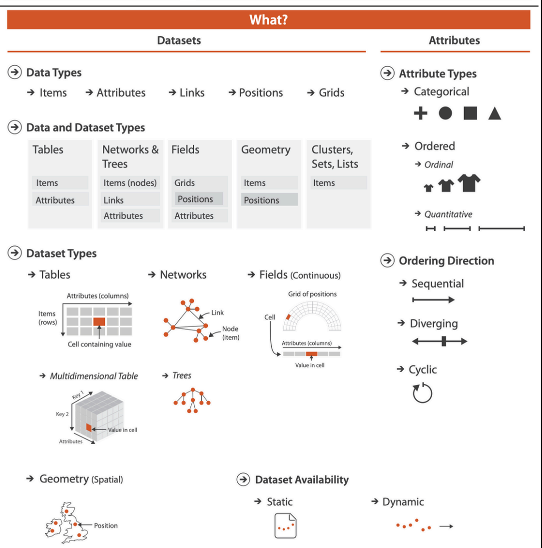
Original Data — Derived Data

- Strahler number
  - centrality metric for trees/networks
  - derived quantitative attribute
  - draw top 5K of 500K for good skeleton



[Using Strahler numbers for real time visual exploration of huge graphs. Auber. Proc. Intl. Conf. Computer Vision and Graphics, pp. 56–69, 2002.]

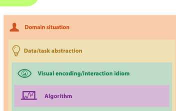


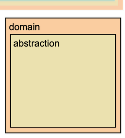


Week 2: Task Abstraction

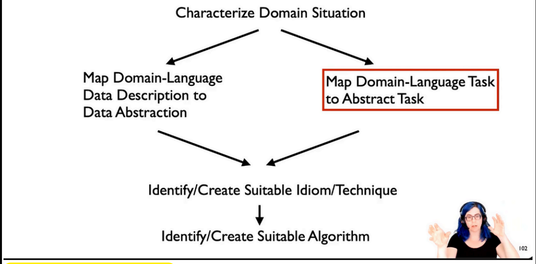
From domain to abstraction

- domain characterization: details of application domain
  - group of users, target domain, their questions & data
    - varies wildly by domain
    - must be specific enough to get traction
  - domain questions/problems
    - break down into simpler abstract tasks
- abstraction: data & task
  - map *what* and *why* into generalized terms
    - identify tasks that users wish to perform, or already do
    - find data types that will support those tasks
      - possibly transform /derive if need be





Design Process



Task abstraction

Task abstraction: Actions and targets

- very high-level pattern
- actions
  - analyze
    - high-level choices
  - search
    - find a known/unknown item
  - query
    - find out about characteristics of item
- {action, target} pairs
  - *discover distribution*
  - *compare trends*
  - *locate outliers*
  - *browse topology*

{action, target} pair: action can be analyze, search, and query

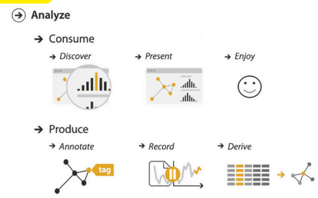
{action, target} pair: action can be analyze, search, and query

- targets
  - what is being acted on

Actions: Analyze

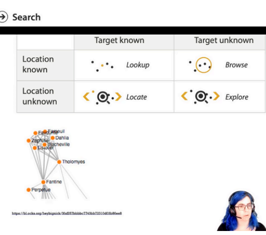
Actions

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



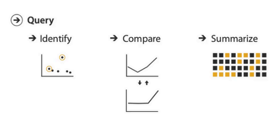
Actions: Search

- what does user know?
  - target, location
- lookup
  - ex: word in dictionary
    - alphabetical order
- locate
  - ex: keys in your house
  - ex: node in network
- browse
  - ex: books in bookstore
- explore
  - ex: find cool neighborhood in new city



Actions: Query

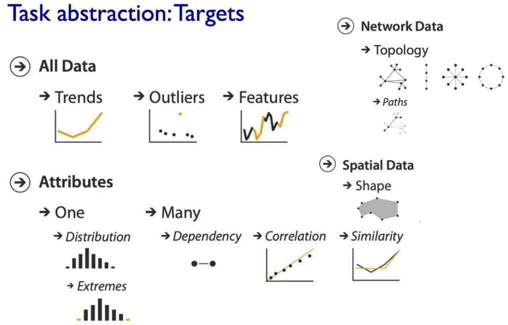
- how much of the data matters?
  - one: identify
  - some: compare
  - all: summarize



Actions

- independent choices for each of these three levels
  - analyze, search, query
  - mix and match

Task abstraction: Targets



Abstraction

- these {action, target} pairs are good starting point for vocabulary
  - but sometimes you'll need more precision!
- rule of thumb
  - systematically remove all domain jargon
- interplay: task and data abstraction
  - need to use data abstraction within task abstraction
    - to specify your targets!
    - but task abstraction can lead you to transform the data
  - iterate back and forth
    - first pass data, first pass task, second pass data, ...

Week 3: Marks and Channels

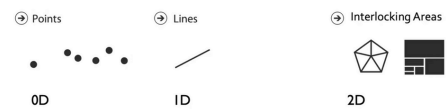
Visual encoding

- how to systematically analyze idiom structure?

- marks & channels
  - marks: represent items or links
  - channels: change appearance of marks based on attributes

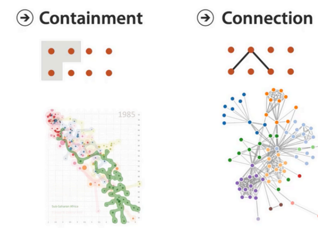
Marks for items

- basic geometric elements

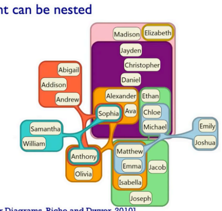


- 3D mark: volume, rarely used

Marks for links



Containment can be nested

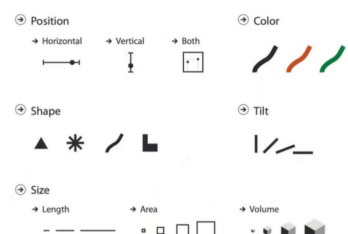


[Untangling Euler Diagrams, Riche and Dwyer, 2010]

Channels

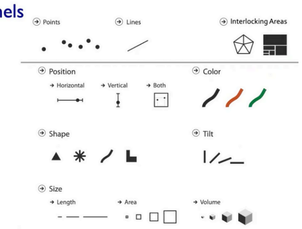
Channels

- control appearance of marks
  - proportional to or based on attributes
- many names
  - **visual channels**
  - visual variables
  - retinal channels
  - visual dimensions
  - ...



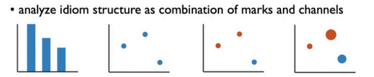
Definitions: Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks
- channel properties differ
  - type & amount of information that can be conveyed to human perceptual system



Visual encoding

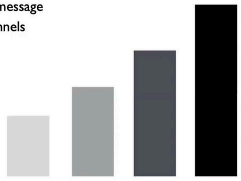
- analyze idiom structure as combination of marks and channels



1:	2:	3:	4:
vertical position	vertical position horizontal position	vertical position horizontal position color hue	vertical position horizontal position color hue size (area)
mark: line	mark: point	mark: point	mark: point

Redundant encoding

- multiple channels
  - sends stronger message
  - but uses up channels



Length and Luminance

Marks as constraints

- math view: geometric primitives have dimensions

- constraint view: mark type constrains what else can be encoded
  - points: 0 constraints on size, can encode more attributes w/ size & shape
  - lines: 1 constraint on size (length), can still size code other way (width)
  - interlocking areas: 2 constraints on size (length/width), cannot size or shape code
    - interlocking: size, shape, position
- quick check: can you size-code another attribute
  - or is size/shape in use?

Scope of analysis

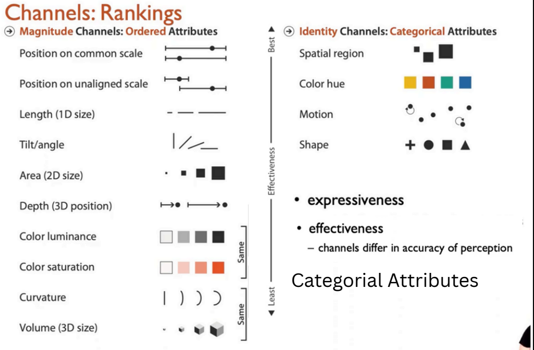
- simplifying assumptions: one mark per item, single view

- later on
  - multiple views
  - multiple marks in a region (glyph)
  - some items not represented by marks (aggregation and filtering)

expressiveness
match channel type to data type

effectiveness
some channels are better than others

Channels: Rankings



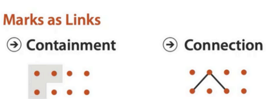
- expressiveness
- effectiveness
  - channels differ in accuracy of perception

Categorial Attributes

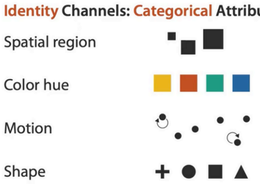
Magnitude Channels Above

Grouping

- containment
- connection



- proximity
  - same spatial region
- similarity
  - same values as other categorical channels



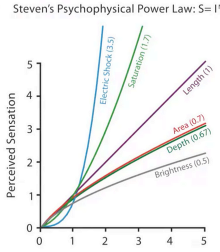
Channel effectiveness

- accuracy: how precisely can we tell the difference between encoded items?
- discriminability: how many unique steps can we perceive?
- separability: is our ability to use this channel affected by another one?
- popout: can things jump out using this channel?

Accuracy: Fundamental theory

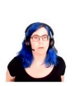
- length is accurate: linear
- others magnified or compressed
  - exponent characterizes

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

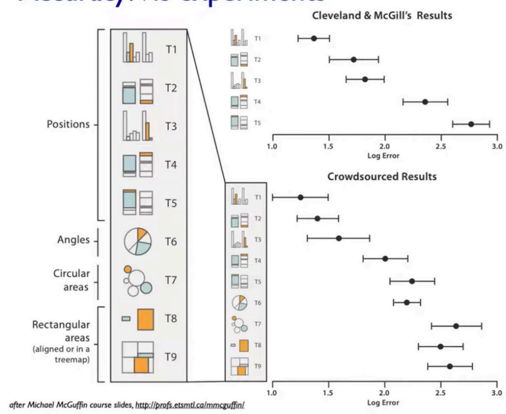


S = sensation

I = intensity



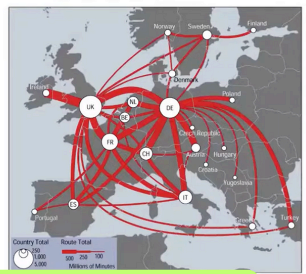
Accuracy: Vis experiments



after Michael McGuffin course slides, http://profs.etsmtl.ca/mmcguffin/

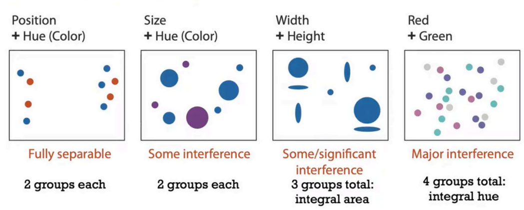
Discriminability: How many usable steps?

- must be sufficient for number of attribute levels to show
  - linewidth: few bins



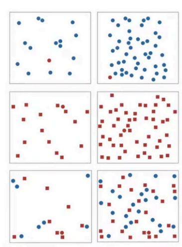
Week 3: Marks and Channels II

Separability vs. Integrality

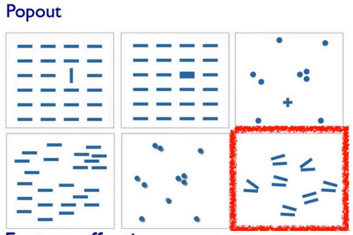


Popout

- find the red dot
  - how long does it take?
- parallel processing on many individual channels
  - speed independent of distractor count
  - speed depends on channel and amount of difference from distractors
- serial search for (almost all) combinations
  - speed depends on number of distractors



Popout



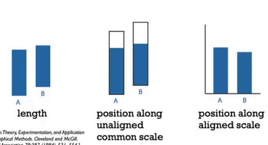
- many channels
  - tilt, size, shape, proximity, shadow direction, ...
- but not all!
  - parallel line pairs do not pop out from tilted pairs

Factors affecting accuracy

- alignment
- distractors
- distance
- common scale / alignment

Relative vs. absolute judgements

- perceptual system mostly operates with relative judgements, not absolute
  - that's why accuracy increases with common frame/scale and alignment

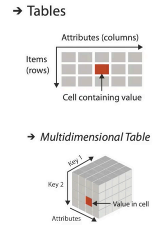


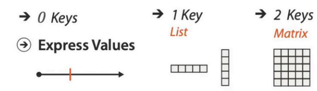
after [Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Cleveland and McGill. Journ. American Statistical Association 79:387 (1984), 531–554.]

Week 4: Tables I and II

Keys and values

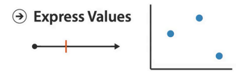
- key
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - multidimensional tables: multiple keys
- value
  - dependent attribute, value of cell
- 0, 1, 2, ...

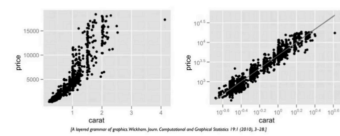




Idiom: **scatterplot**

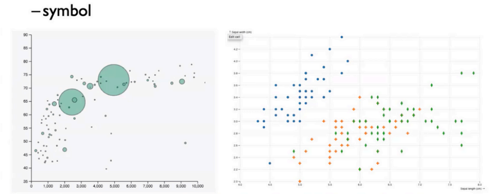
- **express** values (magnitudes)
  - quantitative attributes
- no keys, only values
  - data
    - 2 quant attribs
  - mark: points
  - channels
    - horiz position/order
    - vert position/order





Scatterplots: Encoding more channels

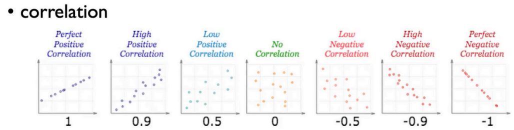
- additional channels viable since using point marks
  - color
  - 2D size (1 quant attribute, used to control 2D area)
    - note radius would mislead, take square root since area grows quadratically
  - symbol



https://www.d3-graph-gallery.com/graph/bubble_basic.html https://observablehq.com/@d3/scatterplot-with-shapes

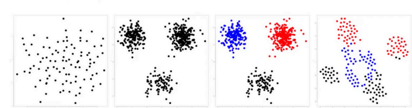
Scatterplot tasks

- correlation

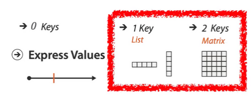


https://www.mathsisfun.com/data/scatter-xy-plots.html

- clusters/groups, and clusters vs classes

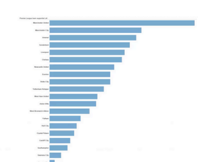


https://www.cs.ubc.ca/labs/imager/tr/2014/DRVisTasks/



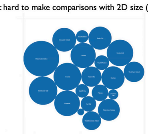
- separate into regions by any attribute: categorical or ordered
  - **regions**: contiguous bounded areas distinct from each other
  - no conflict with expressiveness principle for categorical attributes
  - one glyph/mark per region
- then can use ordered attribute to order and align regions
  - order in one direction (eg horiz), then align in other direction (eg vert)
  - align: impose shared coordinate frame so absolute position has meaning
    - difference between value and 0 point along axis

- best case



Separated but not aligned or ordered

- limitation: hard to make comparisons with 2D size (vs aligned 1D position)



Separated and aligned but not ordered

- limitation: hard to know rank. what's 4th? what's 7th?