Information Visualization

Intro

Tamara Munzner
Department of Computer Science
University of British Columbia

10 September 2015

http://www.cs.ubc.ca/~tmm/courses/547-15

Audience

- no prerequisites
  - many areas helpful but not required
    - human-computer interaction, computer graphics, cognitive psychology, graphic design, algorithms, machine learning, statistics, ...
- open to non-CS people
  - if no programming background, can do analysis or survey project
- open to advanced undergrads
  - talk to me
- open to informal auditors
  - some or all days of readings/discussion, as you like
    - you'll get out of it what you put into it...

Waitlist

- currently 40 registered and 16 on waitlist
  - wow!
- don't panic, people are still shopping around for classes
- highly likely that all who want to take can be accommodated
  - without schlepping extra chairs each time :-)
- make sure to record your name on signup sheet today
  - with probability of attending, including real vs audit
  - update at end of class today, and start of class
- structure plans thus slightly tenative
  - might tweak depending on final enrollment

Class time

- week 1
  - 1 lecture
- weeks 2-9: Participation [30%]
  - before class: you read chapter+paper, write questions/comments
  - during class: I lecture briefly, we discuss, in-class design exercises, ...
    - week 2, 3
      - guest lectures (Robert Kosara, Matt Brehmer)
    - week 8
      - no class (annual VIS conference)
- weeks 10-13: Presentations [20%]
  - before one of the classes: you each read paper on topic of your choice
  - during class: you present it to everybody else (~10 min)

Readings

- textbook
  - Tamara Munzner. Visualization Analysis and Design. AK Peters Visualization Series. CRC Press, 2014.
    - http://www.cs.ubc.ca/~tmm/vadbook/
  - library has multiple ebook copies
  - to buy yourself, cheapest is amazon.com
- papers
  - links posted on course page
  - if DL links, use library EZproxy from off campus
- readings posted by one week before class
- usually one chapter + one paper per class session

Paper Types

- technique/algorithm
- design studies (problem-driven)
- systems
- evaluation
- model/theory

Participation [30%]

- written questions on reading in advance (18% of total mark)
  - due 1:30pm (30 min before class)
  - 3 total, at least 1 for each reading
  - bring printout or laptop with you, springboard for discussion
- discussion/participation in class (12% of total mark)
- attendance expected
  - tell me in advance if you'll miss class (and why)
  - question credit still possible if submitted in advance
  - tell when you recover if you were ill

Questions

- questions or comments
- fine to be less formal than written report
  - correct grammar and spelling still expected
  - be concise: a few sentences is good, one paragraph max!
- should be thoughtful, show you've read and reflected
  - poor to ask something trivial to look up
  - ok to ask for clarification of genuinely confusing section
- examples on http://www.cs.ubc.ca/~tmm/courses/infovis/structure.html

Projects [50%]

- solo, or group of 2, or group of 3
  - groups highly encouraged; amount of work commensurate with group size
- stages
  - pitches (oral, in class): Oct 22
  - meetings (individual, outside class): through Nov 5
  - proposals (written): Nov 9, 5pm
  - status updates incl related work (written): Nov 23, 5pm
  - final presentations (oral): Dec 15 afternoon (times TBD)
  - final reports (written): Dec 17, 5pm
- resources
  - software, data
  - project ideas
  - guest lecture: Brehmer on toolkits/resources (Sep 29)

Projects

- programming
  - common case
  - I will only consider supervising students who do programming projects
  - three types
    - problem-driven design studies (target specific task/data)
    - technique-driven (explore design choice space for encoding or interaction idiom)
    - algorithm implementation (as described in previous paper)
- analysis
  - use existing tools on dataset
  - detailed domain survey
  - particularly suitable for non-CS students
- survey
  - very detailed domain survey
  - particularly suitable for non-CS students

Projects: Design Studies

- BYOD (Bring Your Own Data)
  - you have your own data to analyze
  - your thesis/research topic (very common case)
  - dovetail with another course (sometime possible but timing can be difficult)
- FDOI (Find Data Of Interest)
  - many existing datasets, see resource page to get started
    - http://www.cs.ubc.ca/group/infovis/resources.shtml

Presentations [20%]

- last several weeks of class
- present, analyze, and critique one paper
  - send me topic choices by Nov 2, I will assign papers accordingly
- expectations
  - slides required
  - summary/description important, but also your own thoughts
    - analysis according to book framework
    - critique of strengths and weaknesses
- timing
  - exact times TBD depending on enrollment
  - likely around 10 minutes each
- topics at http://www.cs.ubc.ca/~tmm/courses/infovis/presentations.html

Marking

- 50% Project
  - 2% Pitches
  - 10% Proposal
  - 6% Status Updates
  - 12% Final Presentation
  - 20% Final Report
  - 50% Content
- 20% Presentations
  - 75% Content: Summary 50%, Analysis 25%, Critique 25%
  - 25% Delivery: Presentation Style 50%, Slide Quality 50%
- 30% Participation
  - 60% Written Questions
  - 40% In-Class Discussion/Exercises
- marking by buckets
  - great 100%
  - good 89%
  - ok 78%
  - poor 67%
  - zero 0%

Course Goals

- twofold goal
  - specific: teach you some infovis
  - generic: teach you how to be a better researcher
- feedback through detailed written comments on writing and presenting
  - both content and style
  - at level of paper review for your final project
  - goal: within a week or so
- fast marking for reading questions
  - great/good/ok/poor/zero
  - goal: turn around before next class
    - one week at most

Finding me

- email is the best way to reach me: tmm@cs.ubc.ca
- office hours Tue right after class (3:30-4:30pm)
  - or by appointment
- X661 (X-Wing of ICICS/CS bldg)
- course page is font of all information
  - don't forget to refresh, frequent updates
  - http://www.cs.ubc.ca/~tmm/courses/547-15

Chapters/Topics

- What's Vis and Why Do It?
- Marks and Channels
- What: Data Abstractions
- Why: Task Abstractions
- Rules of Thumb
- Analysis: Four Levels for Validation
- Arrange Tables
- Arrange Spatial Data
- Arrange Networks
- Map Color and Other Channels
- Manipulate View
- Facet Into Multiple Views
- Reduce Items and Attributes
- Analysis Case Studies

Guest Lectures

- Tue Sep 15 (next time!)
  - Robert Kosara, Tableau
  - Tableau intro/overview demo
- Tue Sep 29
  - Matt Brehmer, UBC
  - resources discussion/demos
  - in both cases, brief intro lecture on readings from me first

Topics Preview

Defining visualization (vis)

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

Why?...

Why have a human in the loop?

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

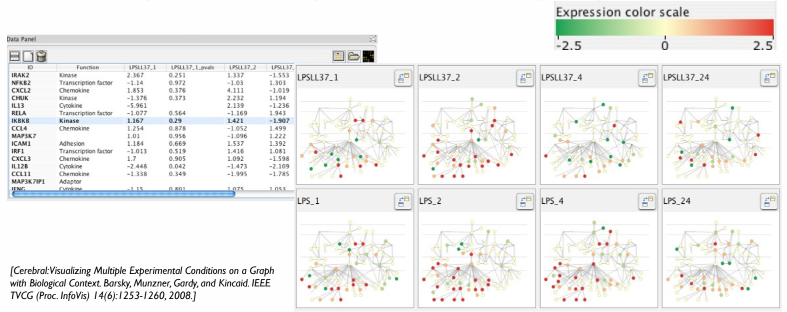
Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

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

Why use an external representation?

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

- external representation: replace cognition with perception

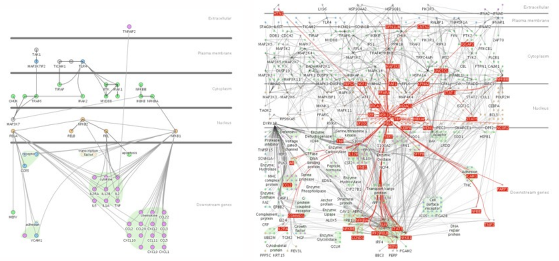


[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE TVCG (Proc. InfoVis) 14(6):1253-1260, 2008.]

Why have a computer in the loop?

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

- beyond human patience: scale to large datasets, support interactivity
  - consider: what aspects of hand-drawn diagrams are important?



[Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. Barsky, Gardy, Hancock, and Munzner. Bioinformatics 23(8):1040-1042, 2007.]

Why depend on vision?

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

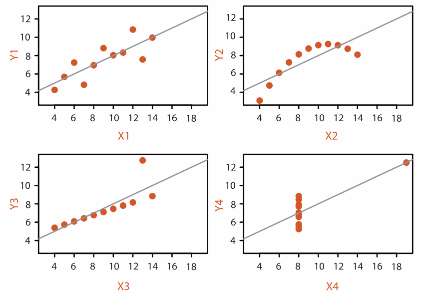
- human visual system is high-bandwidth channel to brain
  - overview possible due to background processing
    - subjective experience of seeing everything simultaneously
    - significant processing occurs in parallel and pre-attentively
- sound: lower bandwidth and different semantics
  - overview not supported
    - subjective experience of sequential stream
- touch/haptics: impoverished record/replay capacity
  - only very low-bandwidth communication thus far
- taste, smell: no viable record/replay devices

Why show the data in detail?

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

Anscombe's Quartet

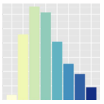
Identical statistics	
x mean	9
x variance	10
y mean	8
y variance	4
x/y correlation	1

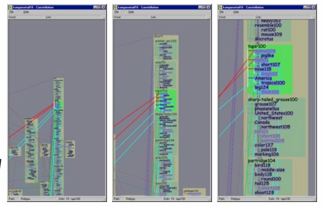


Idiom design space

The design space of possible vis idioms is huge, and includes the considerations of both how to create and how to interact with visual representations.

- **idiom**: distinct approach to creating or manipulating visual representation
  - how to draw it: **visual encoding** idiom
    - many possibilities for how to create
  - how to manipulate it: **interaction** idiom
    - even more possibilities
      - make single idiom dynamic
      - link multiple idioms together through interaction





[A layered grammar of graphics. Wickham. Journal of Computational and Graphical Statistics 19:1 (2010), 3–28.]

[Interactive Visualization of Large Graphs and Networks. Munzner. Ph.D. thesis, Stanford University Department of Computer Science, 2000.]

Why focus on tasks and effectiveness?

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

- tasks serve as constraint on design (as does data)
  - idioms do not serve all tasks equally!
  - challenge: recast tasks from domain-specific vocabulary to abstract forms
- most possibilities ineffective
  - validation is necessary, but tricky
  - increases chance of finding good solutions if you understand full space of possibilities
- what counts as effective?
  - novel: enable entirely new kinds of analysis
  - faster: speed up existing workflows

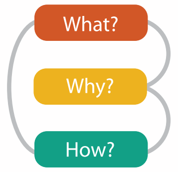
Resource limitations

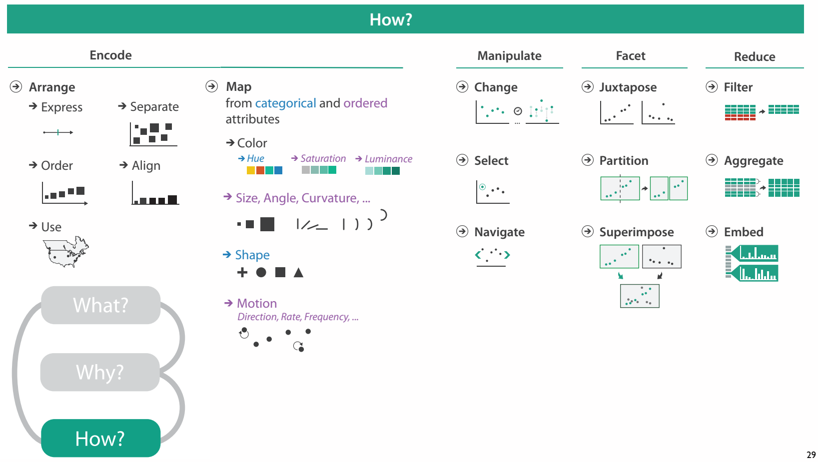
Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.

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

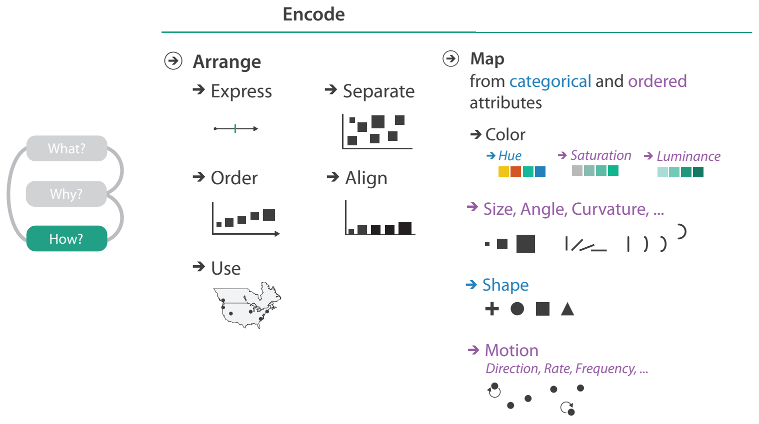
Analysis: What, why, and how

- what is shown?
  - **data** abstraction
- why is the user looking at it?
  - **task** abstraction
- how is it shown?
  - **idiom**: visual encoding and interaction
- abstract vocabulary avoids domain-specific terms
  - translation process iterative, tricky
- what-why-how analysis framework as scaffold to think systematically about design space



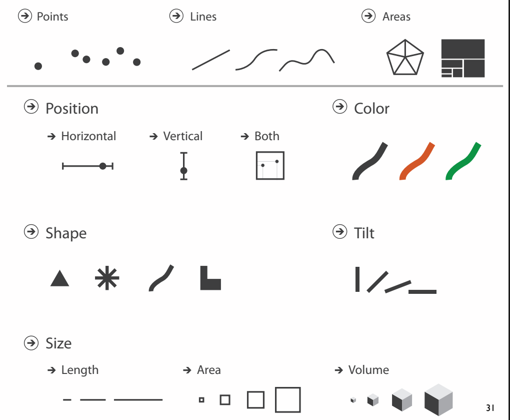


Encode

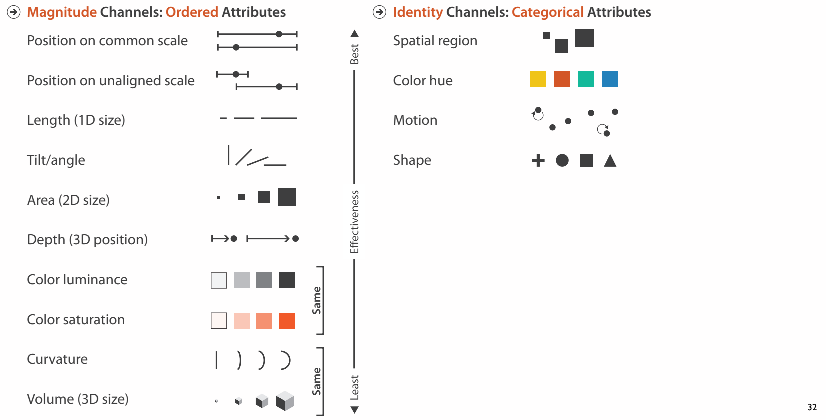


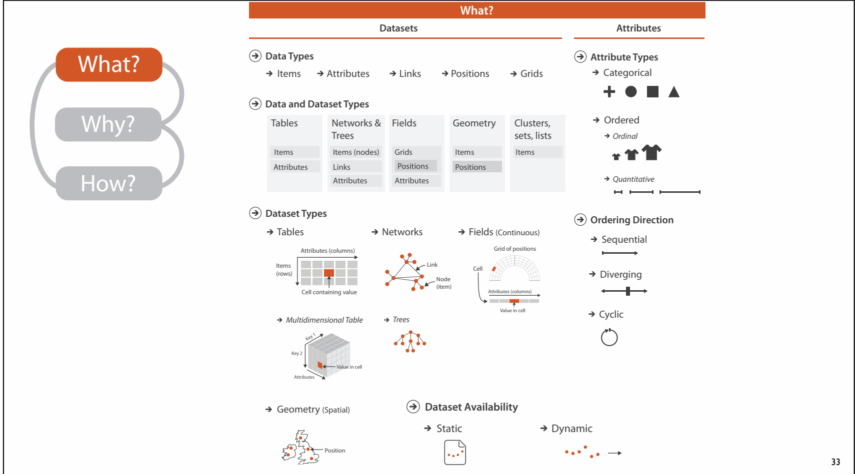
Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks

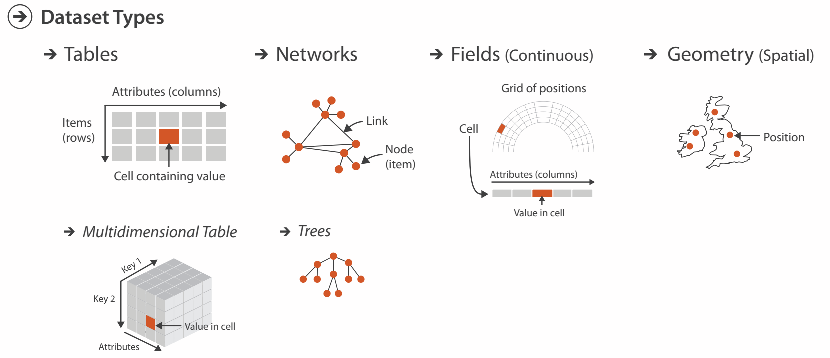


Channels: Expressiveness types and effectiveness rankings

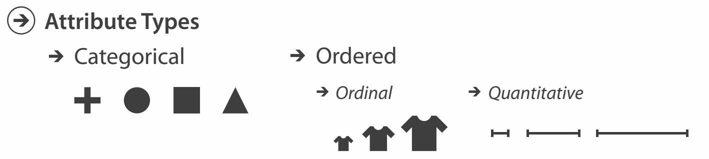


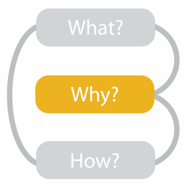


Dataset types

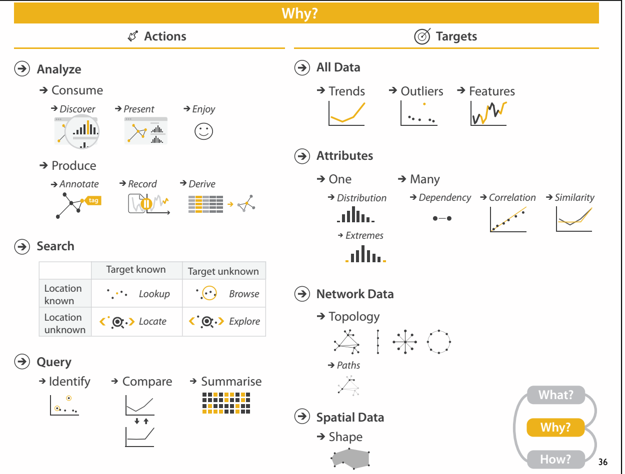


Attribute types



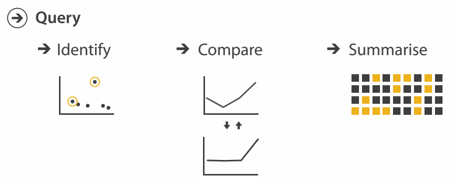


- {action, target} pairs
  - discover distribution
  - compare trends
  - locate outliers
  - browse topology

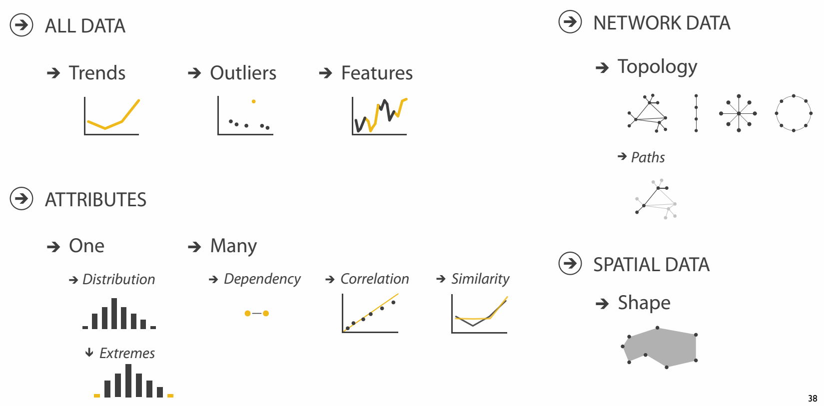


Actions: low-level query

- how much of the data matters?
  - one, some, all



Why: Targets

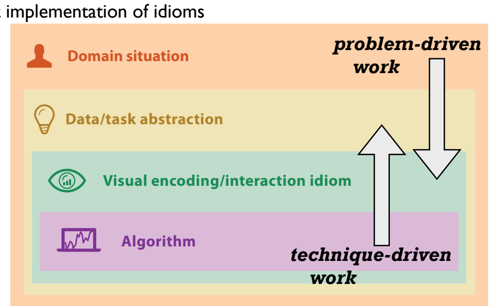


Rules of Thumb

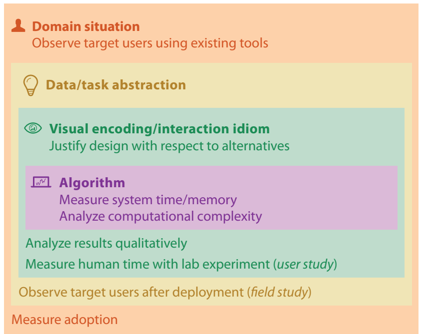
- No unjustified 3D
- Eyes beat memory
- Resolution over immersion
- Overview first, zoom and filter, details on demand
- Function first, form next
- ...

Four Levels of Design

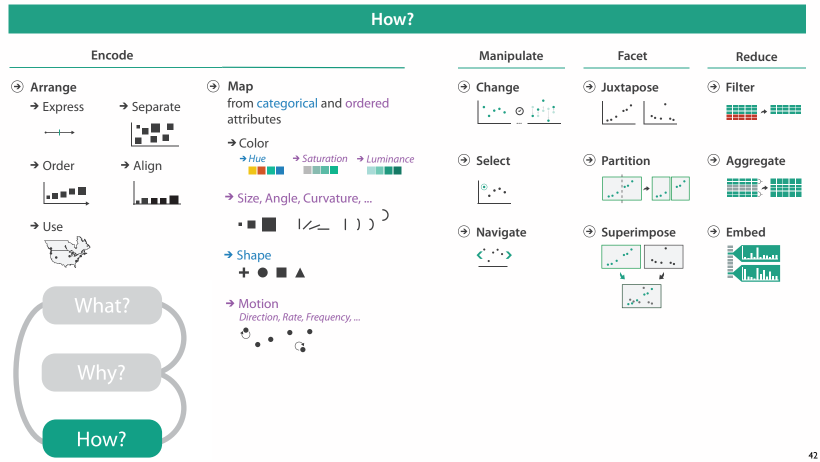
- domain situation: all aspects of user context
- data/task abstraction: why/what
- encoding/interaction idioms: how
- algorithm: efficient implementation of idioms



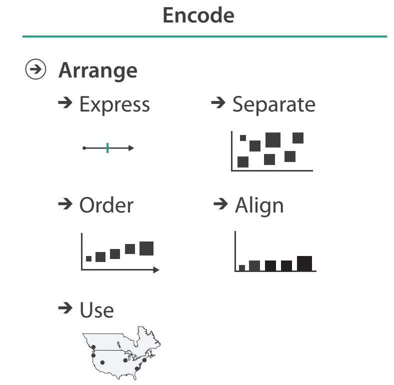
Nested Levels of Design and Validation



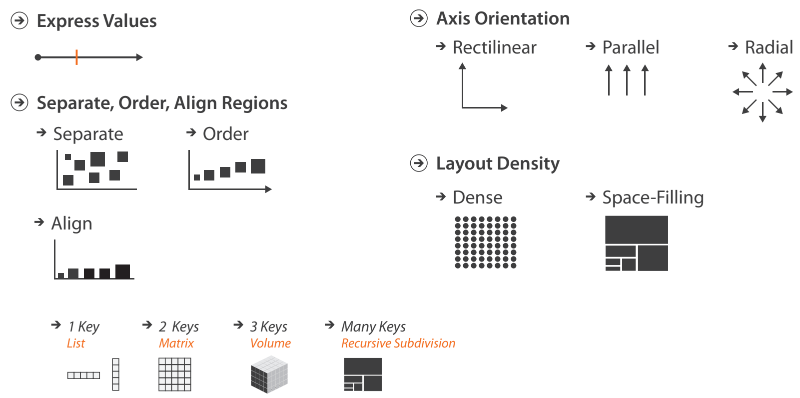
- mismatch: cannot show idiom good with system timings
- mismatch: cannot show abstraction good with lab study



Arrange space



Arrange tables

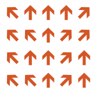


Arrange spatial data

- Use Given
  - Geometry
    - Geographic
    - Other Derived

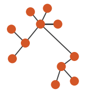
  - Spatial Fields
    - Scalar Fields (one value per cell)
      - Isocontours
      - Direct Volume Rendering

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

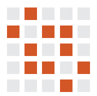


Arrange networks and trees

- Node-link Diagrams
  Connections and Marks
  ✔ NETWORKS ✔ TREES



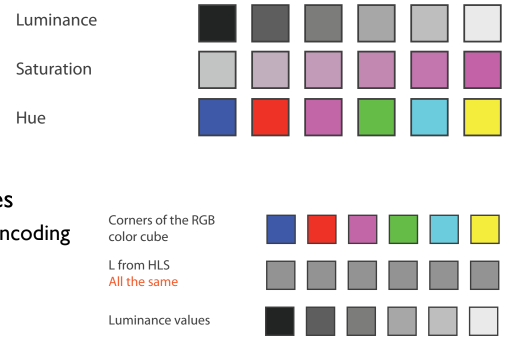
- Adjacency Matrix
  Derived Table
  ✔ NETWORKS ✔ TREES



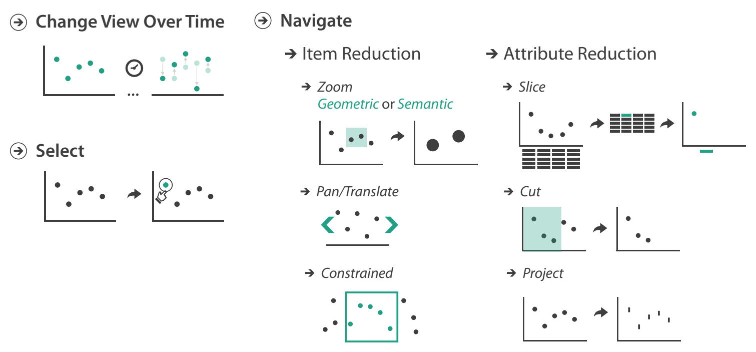
- Enclosure
  Containment Marks
  ✘ NETWORKS ✔ TREES

Color: Luminance, saturation, hue

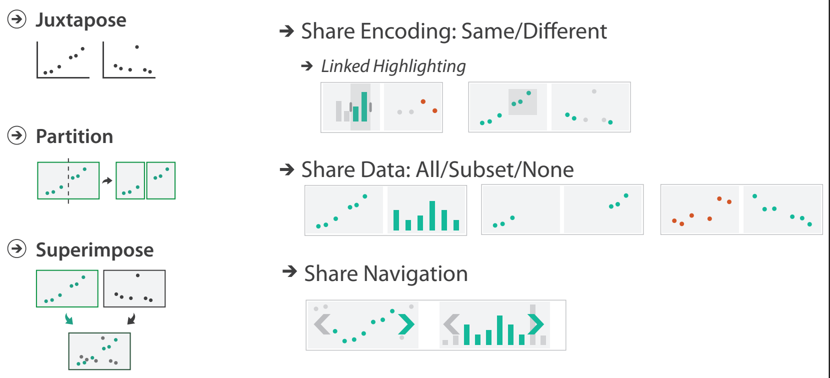
- 3 channels
  - identity for categorical
    - hue
  - magnitude for ordered
    - luminance
    - saturation
- other common color spaces
  - RGB: poor choice for visual encoding
  - HSL: better, but beware
    - lightness ≠ luminance



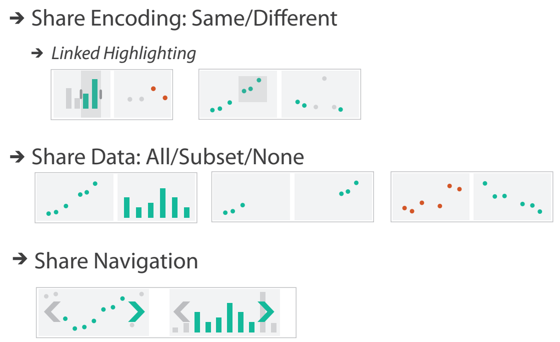
Manipulate



Facet

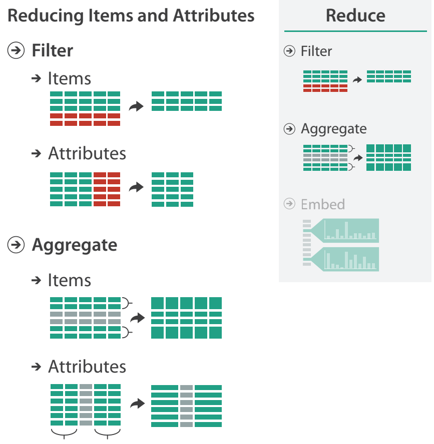


Juxtapose and coordinate views



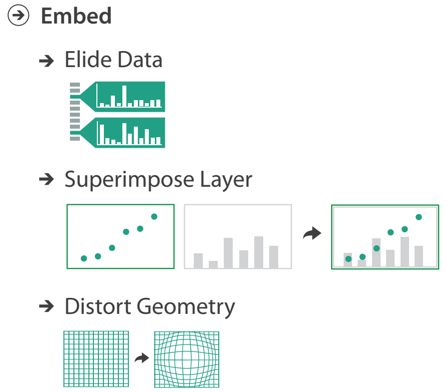
Reduce items and attributes

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



Embed: Focus+Context

- combine information within single view
- elide
  - selectively filter and aggregate
- superimpose layer
  - local lens
- distortion design choices
  - region shape: radial, rectilinear, complex
  - how many regions: one, many
  - region extent: local, global
  - interaction metaphor



Next Time

- to read
  - Book: Marks and Channels (Ch 5)
  - Paper: Polaris
    - academic paper, Tableau is the spinoff company
- guest lecture by Robert Kosara on Tableau