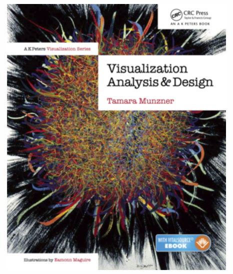
Visualization Analysis & Design



Tamara Munzner
Department of Computer Science
University of British Columbia

University of Washington, Data Science Seminar
September 30 2015, Seattle WA

http://www.cs.ubc.ca/~tmm/talks.html#vad15uw **@tamaramunzner**

Defining visualization (vis)

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

Why?...

Why have a human in the loop?

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

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

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

Why use an external representation?

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

- external representation: replace cognition with perception

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

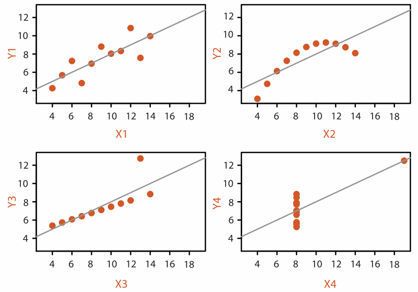
Why represent all the data?

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

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

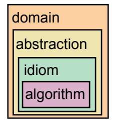
Anscombe's Quartet

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

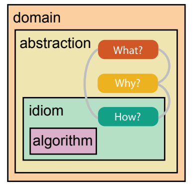


Analysis framework: Four levels, three questions

- *domain* situation
  - who are the target users?
- *abstraction*
  - translate from specifics of domain to vocabulary of vis
- **what** is shown? **data abstraction**
  - often don't just draw what you're given: transform to new form
- **why** is the user looking at it? **task abstraction**
- *idiom*
  - **how** is it shown?
    - **visual encoding idiom**: how to draw
    - **interaction idiom**: how to manipulate
- *algorithm*
  - efficient computation

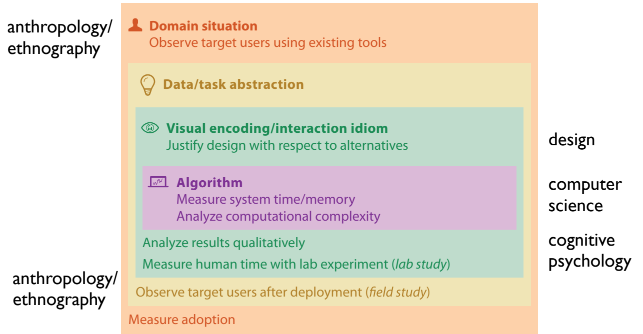


[A Nested Model of Visualization Design and Validation. Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]



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

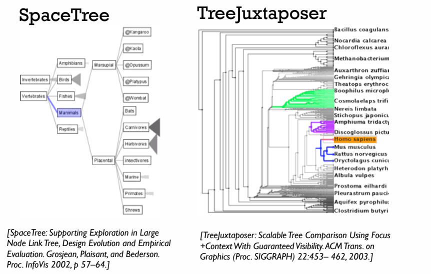
Validation methods from different fields for each level



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

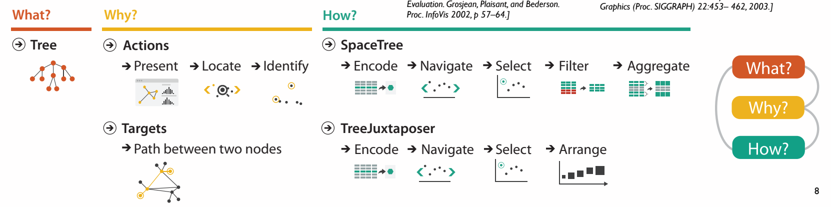
Why analyze?

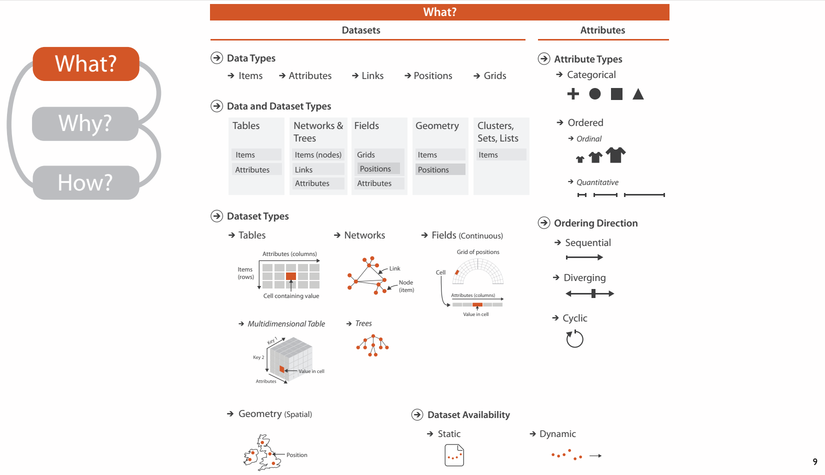
- imposes a structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new



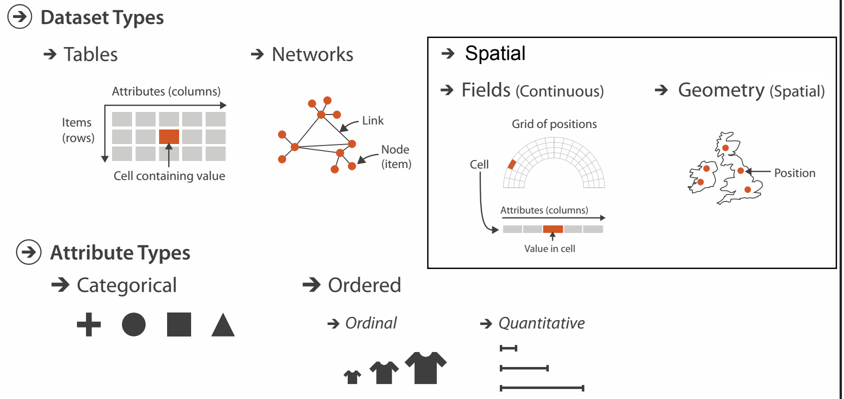
[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

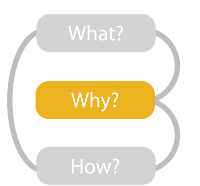
[TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]



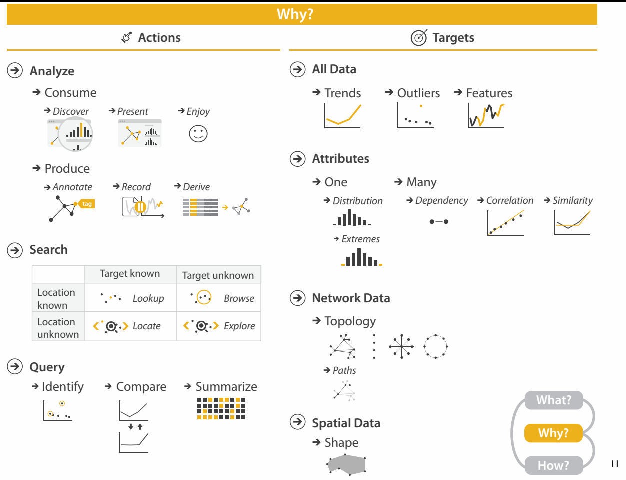


Types: Datasets and data



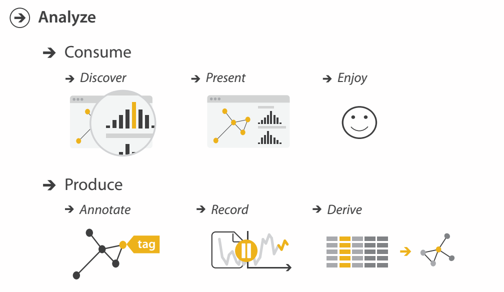


- {action, target} pairs
  - *discover distribution*
  - *compare trends*
  - *locate outliers*
  - *browse topology*



Actions I: Analyze

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



Actions II: Search

- what does user know?
  - target, location

Search

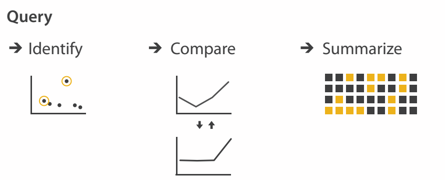
	Target known	Target unknown
Location known	*Lookup*	*Browse*
Location unknown	*Locate*	*Explore*

Actions III: Query

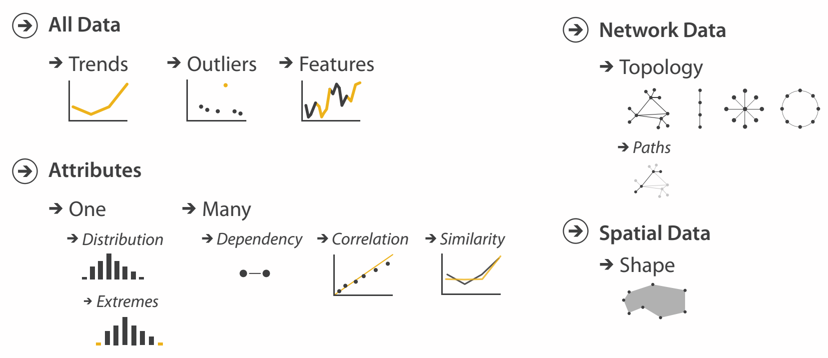
- what does user know?
  - target, location
- how much of the data matters?
  - one, some, all
- analyze, search, query
  - independent choices for each

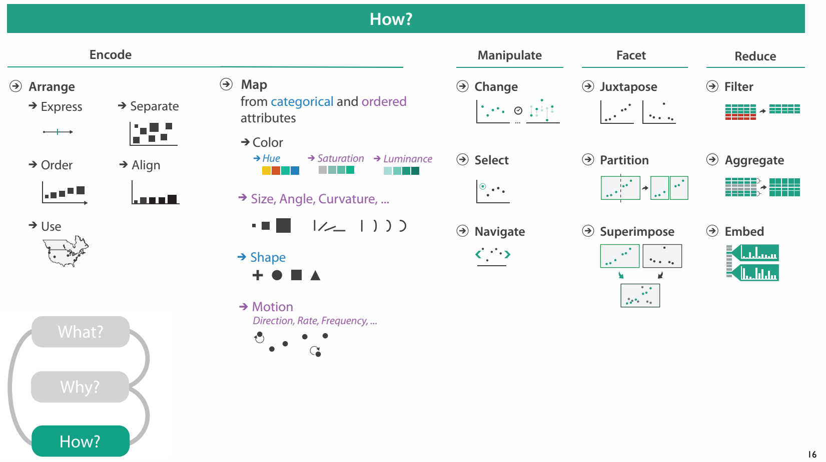
Search

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

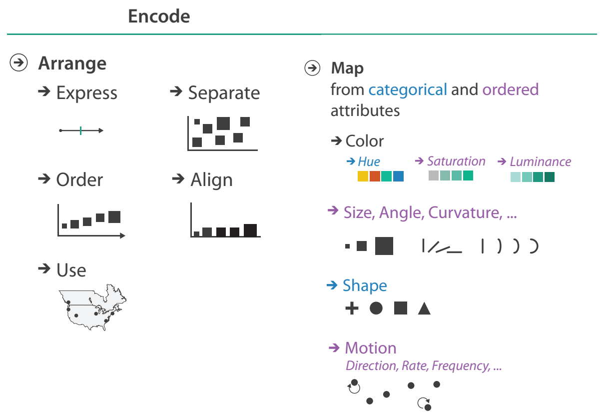


Targets



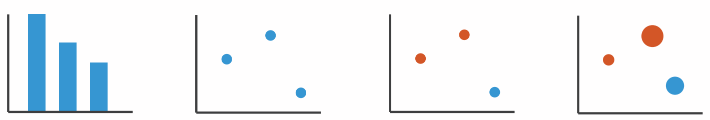


How to encode: Arrange space, map channels



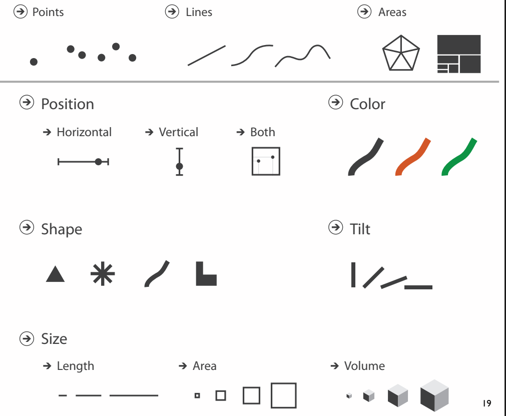
Encoding visually

- analyze idiom structure



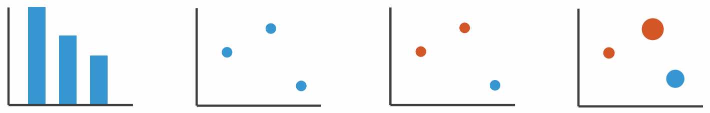
Definitions: Marks and channels

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



Encoding visually with marks and channels

- analyze idiom structure
  - as combination of marks and channels



1: vertical position

mark: line

2: vertical position, horizontal position

mark: point

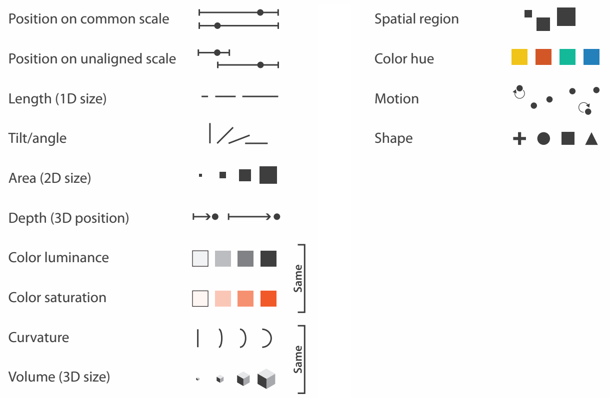
3: vertical position, horizontal position, color hue

mark: point

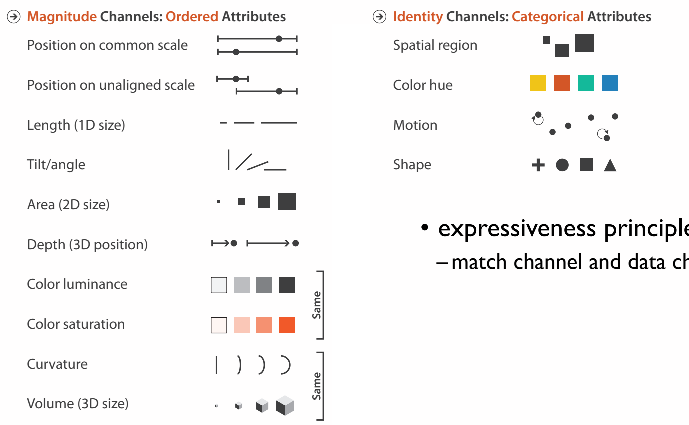
4: vertical position, horizontal position, color hue, size (area)

mark: point

Channels

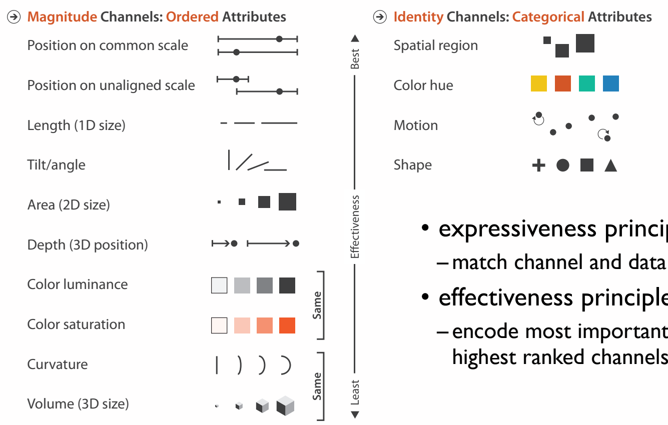


Channels: Matching Types

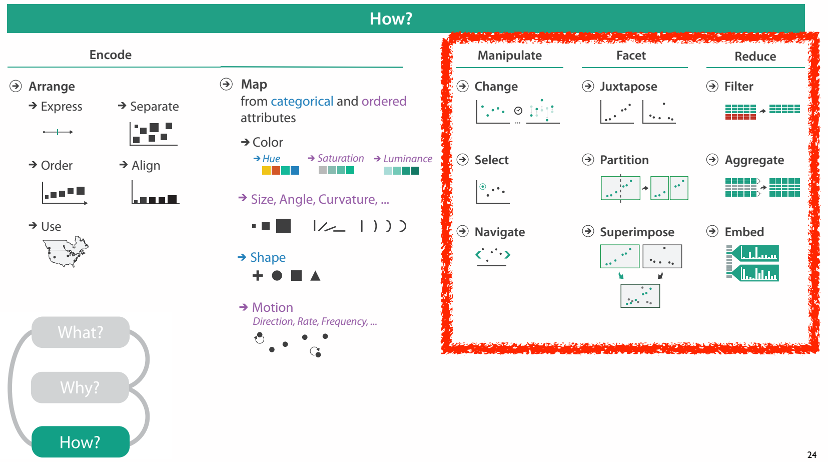


- expressiveness principle
  - match channel and data characteristics

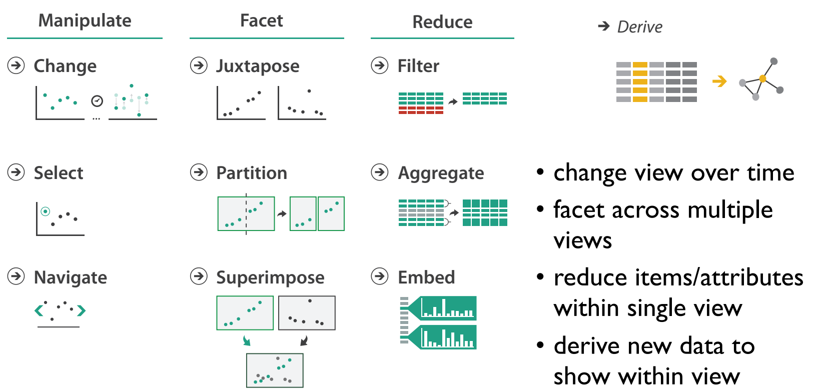
Channels: Rankings



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

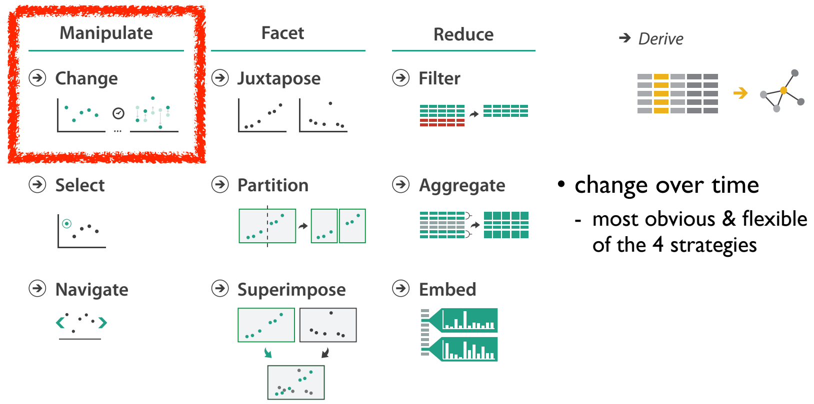


How to handle complexity: 3 more strategies + 1 previous



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

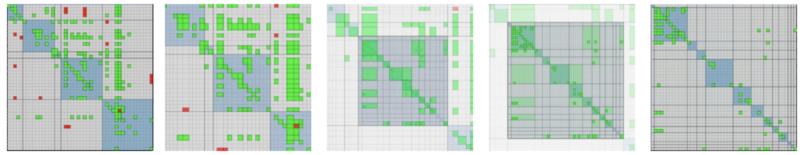
How to handle complexity: 3 more strategies + 1 previous



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

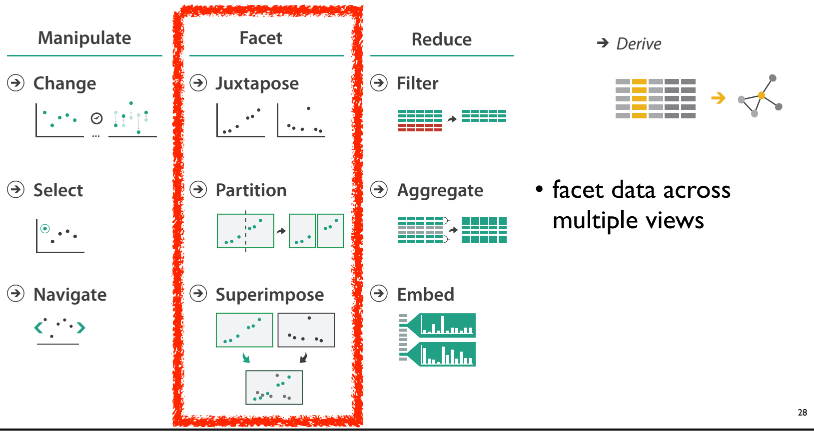
Idiom: **Animated transitions**

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



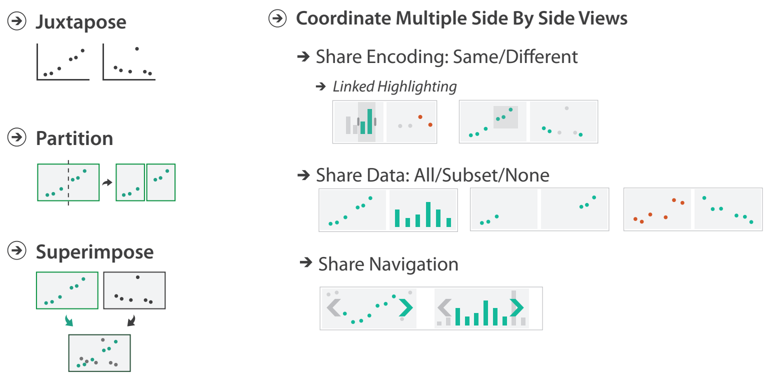
[Using Multilevel Call Matrices in Large Software Projects. van Ham. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 227–232, 2003.]

How to handle complexity: 3 more strategies + 1 previous



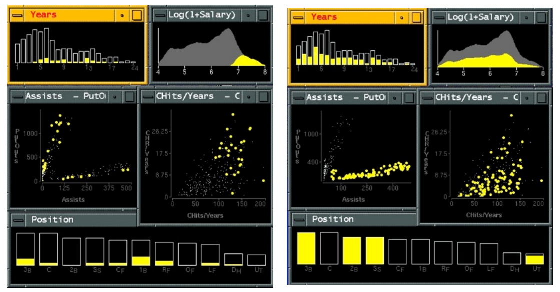
- facet data across multiple views

Facet



Idiom: **Linked highlighting** System: **EDV**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom
- encoding: different
  - ***multiform***
- data: all shared



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

Idiom: **bird's-eye maps** System: **Google Maps**

- encoding: same
- data: subset shared
- navigation: shared
  - bidirectional linking
- differences
  - viewpoint
  - (size)
- ***overview-detail***

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

Idiom: **Small multiples** System: **Cerebral**

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

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

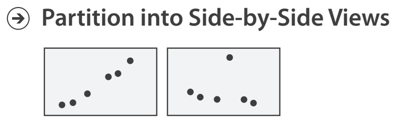
Coordinate views: Design choice interaction

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

- why juxtapose views?
  - benefits: eyes vs memory
    - lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - costs: display area, 2 views side by side each have only half the area of one view

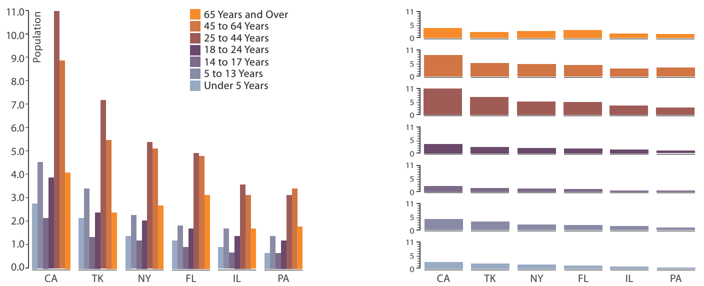
Partition into views

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



Partitioning: List alignment

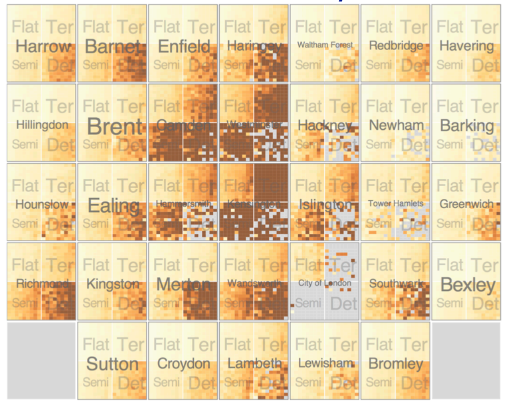
- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages
- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states



Partitioning: Recursive subdivision

System: **HIVE**

- split by neighborhood
- then by type
- then time
  - years as rows
  - months as columns
- color by price
- neighborhood patterns
  - where it's expensive
  - where you pay much more for detached type



[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

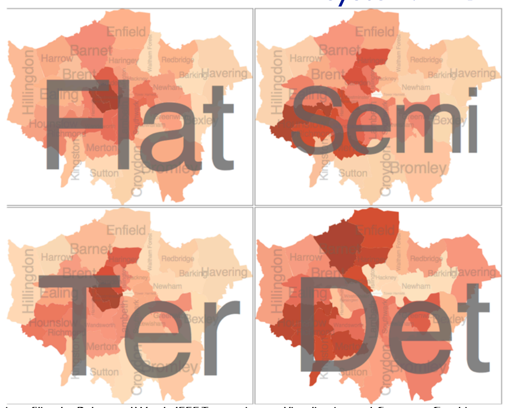
- switch order of splits
  - type then neighborhood
- switch color
  - by price variation
- type patterns
  - within specific type, which neighborhoods inconsistent

[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

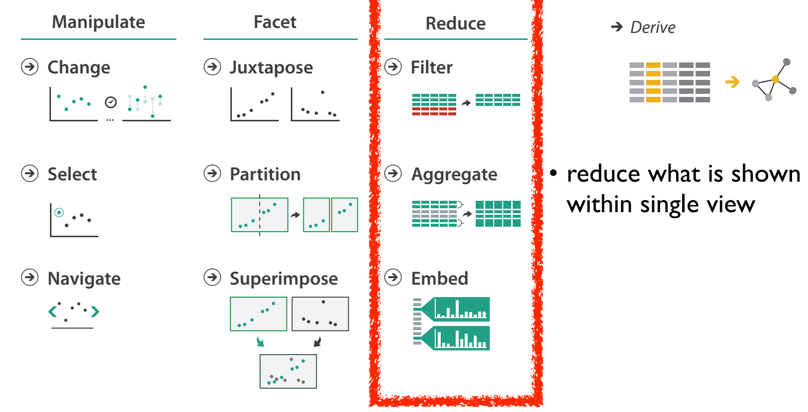
System: **HIVE**

- different encoding for second-level regions
  - choropleth maps



[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

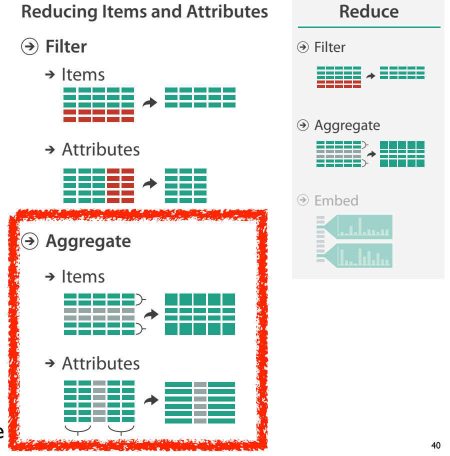
How to handle complexity: 3 more strategies + 1 previous



- reduce what is shown within single view

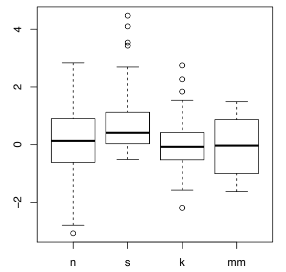
Reduce items and attributes

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



Idiom: **boxplot**

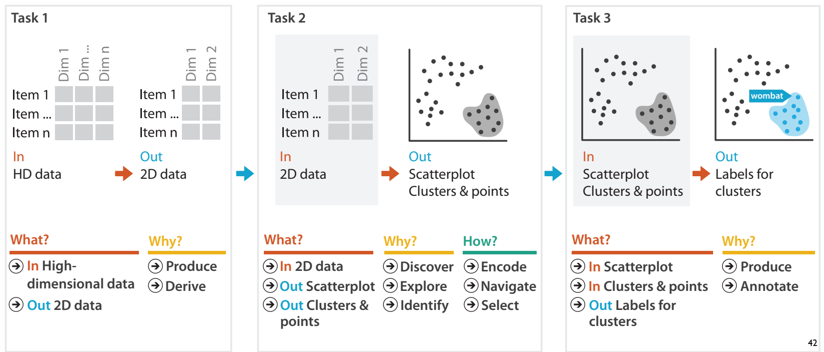
- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown

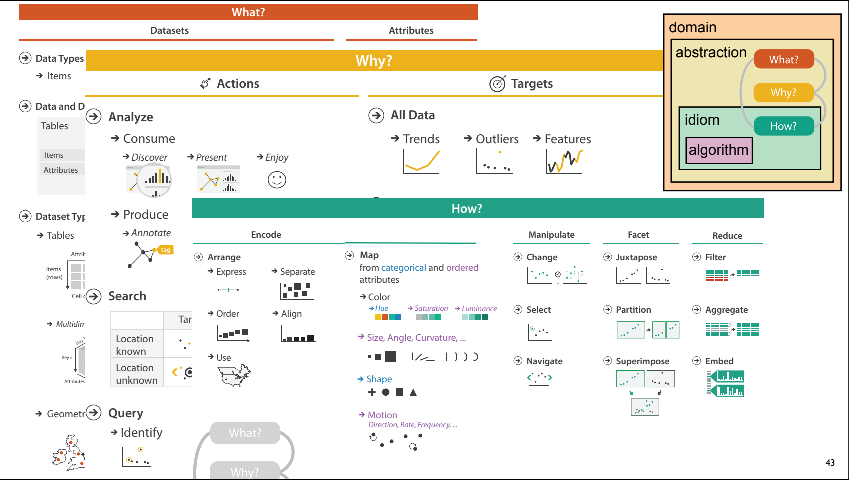


[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]

Idiom: **Dimensionality reduction for documents**

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

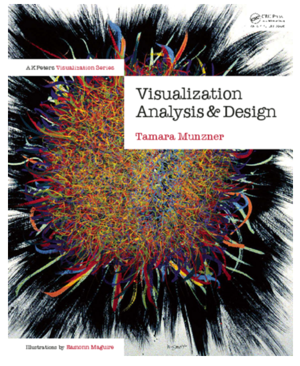




More Information

@tamaramunzner

- this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad15uw
- book page (including tutorial lecture slides)
  http://www.cs.ubc.ca/~tmm/vadbook
  - 20% promo code for book+ebook combo: HVN17
  - http://www.crcpress.com/product/isbn/9781466508910
  - illustrations: Eamonn Maguire
- papers, videos, software, talks, full courses
  http://www.cs.ubc.ca/group/infovis
  http://www.cs.ubc.ca/~tmm



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