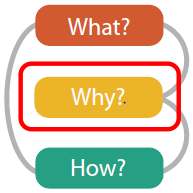
Information Visualization and Visual Analytics (M1522.000500)

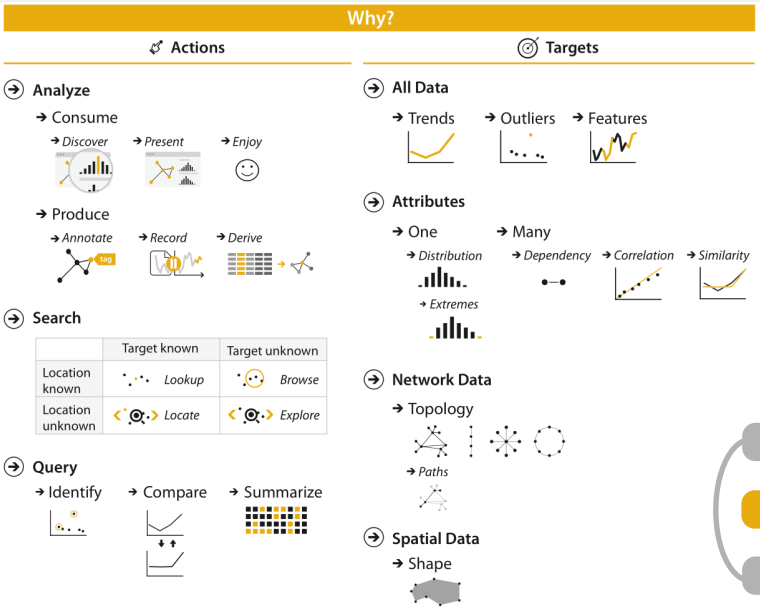
Why: Task Abstraction

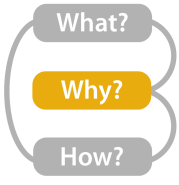
Jinwook Seo, Ph. D.
Professor, Dept. of Computer Science and Engineering
Seoul National University



The Big Picture

Why People Are Using Vis? (for What Tasks?)





Task Abstraction

- Consider tasks in **abstract form**, rather than **domain-specific way**
  - Otherwise, hard to make useful comparisons between domain situations
- domain-specific task descriptions

"Contrast the prognosis of patients who were intubated in the ICU more than one month after exposure to patients hospitalized within the first week"

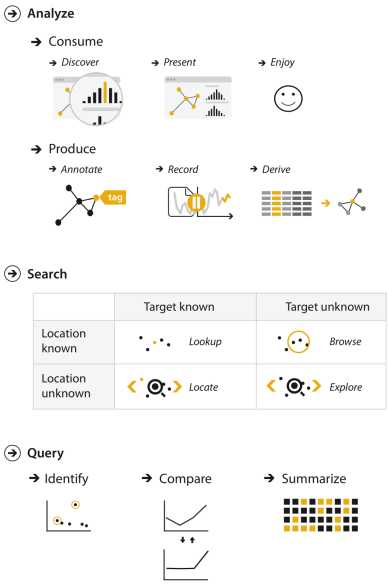
"See if the results for the tissue samples treated with LL-37 match up with the ones without the peptide"

- Abstract form: *"compare values between two groups"*
- **Task abstraction** should guide **data abstraction**

Actions define User **Goals**

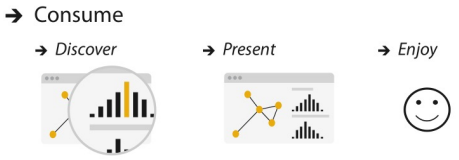
- High-level: **Analyze**
  - Consume, Produce
- Mid-level: **Search**
  - Lookup, Browse, Locate, Explore
- Low-level: **Query**
  - Identify, Compare, Summarize

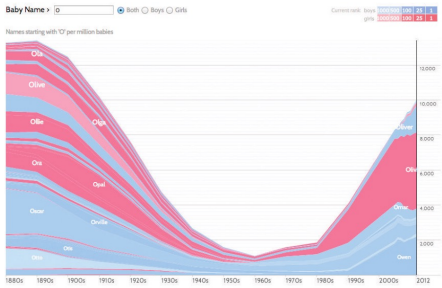
- Choices at each level are independent
- Describe all of actions at all three levels



High-Level: Analyze – Consume (existing information)

- **Discover** (= explore)
  - Find new knowledge that was *not previously known*
    - **generate** a new hypothesis or **verify** an existing hypothesis
  - Often for scientific inquiry
- **Present** (= explain)
  - The communication of information that is *specific and already understood*
  - e.g., infographic (static information graphics)
  - output of a discover session -> input to a present session
- **Enjoy**
  - Motivated by users' enjoyment
  - Casual encounters with vis for enjoyment
  - Users are driven by Curiosity stimulated and satisfied by vis
  - e.g. Name Voyager



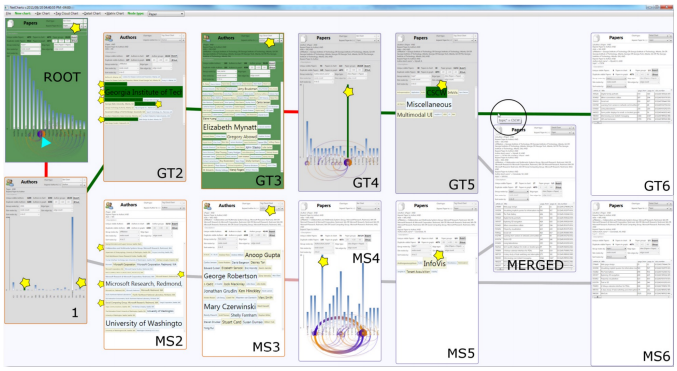


High-level: Analyze – Produce (new information as input to a next instance)

- **Annotate** (~ tag)
  - Attaches temporary info to existing elements
- **Record**
  - To save or capture vis elements as persistent artifacts
  - e.g., screen shots, interaction logs, etc.
- **Derive** (= transform)
  - To produce new data elements (= derived attributes) based on existing elements
    - Could expand the design space of possible vis idioms
  - Changing types of data
  - Transform with additional info
  - Using arithmetic/logical/statistical operations

High-level: Analyze – Produce - Record

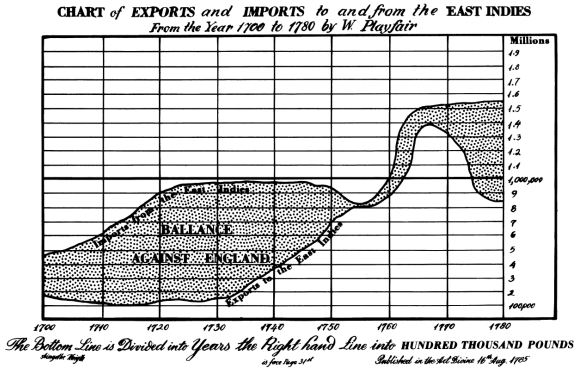
- Record
  - graphical history of visual exploration



A GraphTrail analysis showing two parallel exploration paths

High-level: Analyze – Produce - Derive

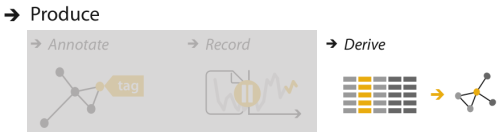
- Do not just draw what you are given
  - Decide what the right thing to show is
  - Create it by transformations
  - and draw it!

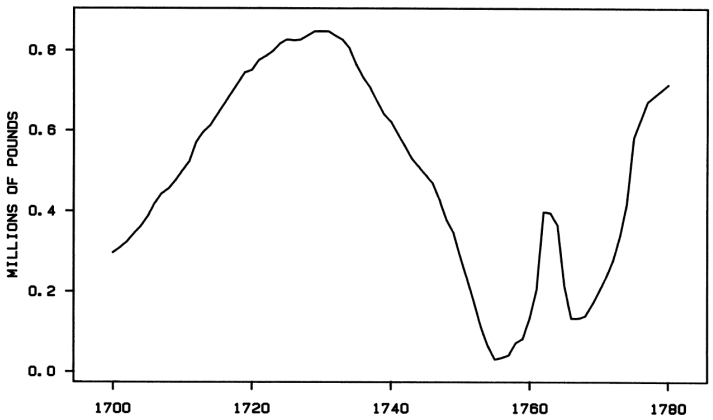


judging the difference in height

High-level: Analyze – Produce - Drive

- Do not just draw what you are given
  - Decide what the right thing to show is
  - Create it by transformations
  - and draw it!
  - detail is aggregated away, but





Mid-level: Search (Find with successful outcome)

- High-level [**analyze** action] cases require the user to **search for elements of interest**

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

hci lab SEOUL NATIONAL UNIVERSITY

Mid-level: **Search** (**Find** with successful outcome)

- e.g. a tree vis for Tree of Life
  - **Lookup**
    - *Looking up* human (target O)
      knowing it belongs to mammals (location O)
  - **Locate**
    - *Locating* rabbits (target O)
      not knowing where it belongs to (location X)
  - **Browse**
    - *Browsing* all leaves (target X)
      of the mammal subtree (location O)
  - **Explore**
    - *Exploring* for a family having
      the largest number of species (target X, location X)

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

http://tolweb.org/tree/home.pages/aboutoverview.html

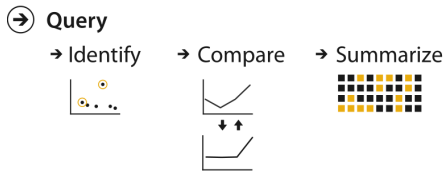
hci lab SEOUL NATIONAL UNIVERSITY

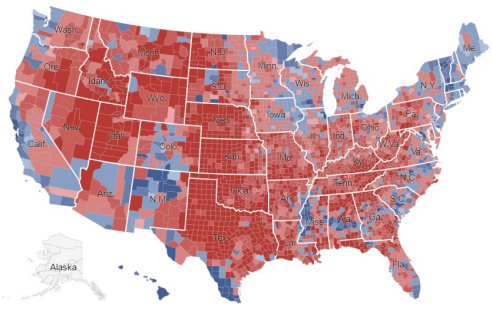
Low-level: Query

- Once targets for a search has found,

  → QUERY the targets

- e.g. a choropleth map of US election results
  - **Identify** (a single target)
    - *identify* the election results for one state
  - **Compare** (multiple targets)
    - *compare* the election results of one state to another
  - **Summarize / Overview** (all possible targets)
    - *summarize* the election results across all states
      to determine how many favored one candidate





Things that actions refer to

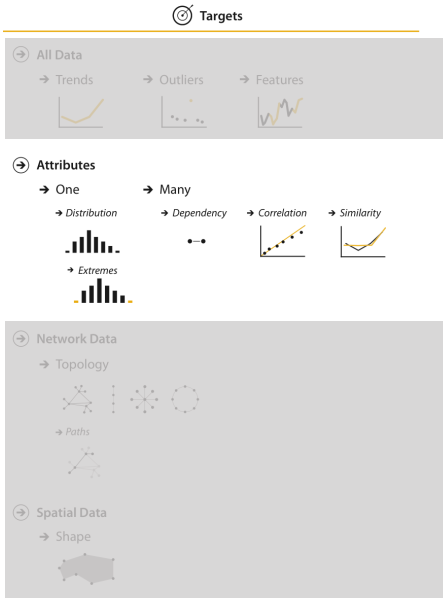
- Some aspect of the data that is interest of the user

Targets – All Data Level

- All Data level
  - **Trends** (= pattern)
    - a high-level characterization of a pattern in the data
    - e.g. increases, decreases, peaks, regression fit etc.
  - **Outliers** (= anomalies, novelties, deviants, surprises)
  - **Features**
    - Any particular structures of interest
    - Task-dependent definition
      - clusters in cluster analysis

Targets – Attributes Level

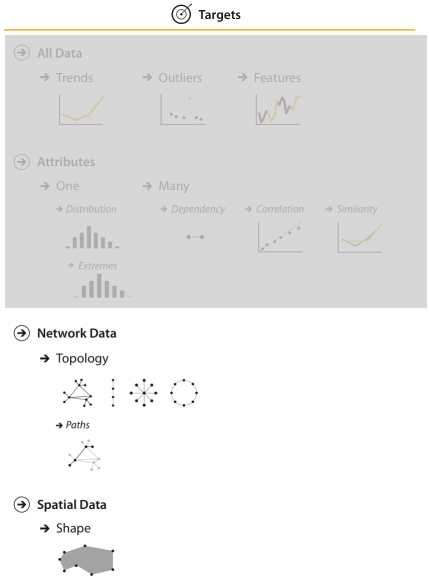
- Attributes level
  - One attribute
    - **An individual value**
    - **extremes** (maximum, minimum)
  - Multiple attributes
    - **Dependency**
      - An attribute depends on another attribute
    - **Correlation**
      - Tendency of values of two attributes are tied
    - **Similarity**
      - Quantitative measurement of how values of two attributes are similar



Targets

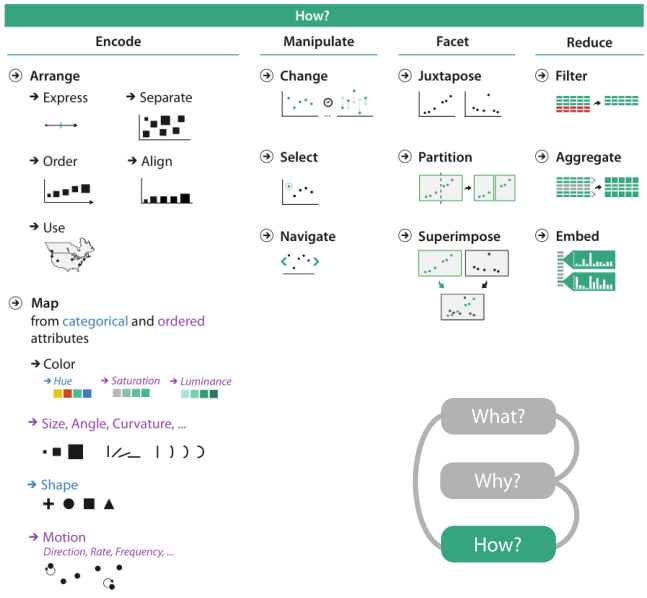
- Targets for specific types of datasets
  - Network data
    - **Topology**
      - paths
  - Spatial data
    - **Shape**
  - And more..

✓ These targets can be expressed in domain-specific terms,
But should always recognize these abstractions



How: A Preview

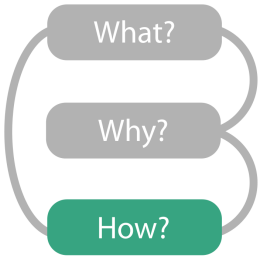
How a vis idiom is constructed? – a set of design choices



How: A Preview

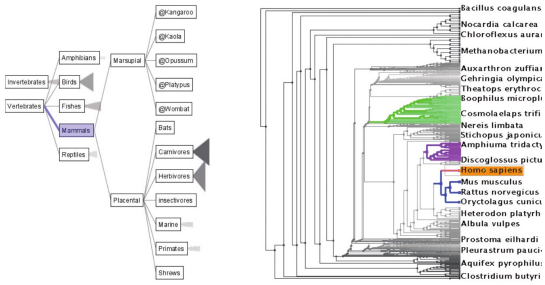
How a vis idiom is constructed? – a set of design choices

- How to encode data (chapter 7, 8, 9, 10)
  - how to **arrange** data **spatially**
  - how to **map** data with **nonspatial** visual channels
- How to manipulate the view (chapter 11)
  - change aspect of view, select element in a view, navigate within the view
- How to facet data between views (chapter 12)
  - juxtapose, partition, superimpose
- How to reduce data (chapter 13)
  - filter, aggregate, embed (chapter 14)



hci lab SEOUL NATIONAL UNIVERSITY

Comparing two idioms (comparative analysis)



(a) SpaceTree (b) TreeJuxtaposer

What?

- Tree

Why?

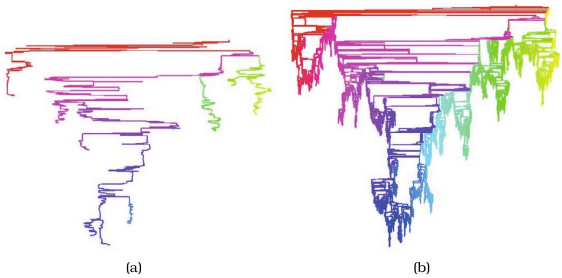
- Actions
  - Present → Locate → Identify
- Targets
  - Path between two nodes

How?

- SpaceTree
  - Encode → Navigate → Select → Filter → Aggregate
- TreeJuxtaposer
  - Encode → Navigate → Select → Arrange

hci lab SEOUL NATIONAL UNIVERSITY

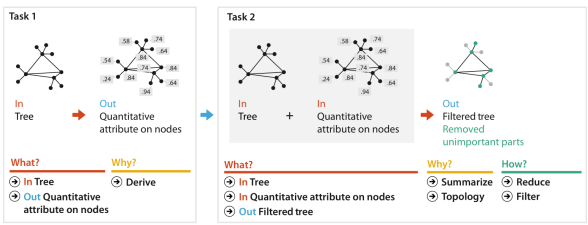
Deriving one attribute



(a) filtered by Strahler number (centrality measure)

(b) original graph

시험문제에 이게 나오면



Note

- Questions?