Chapter 4

Task Abstraction

Overview

- **The Big Picture:**
  - **Why a vis tool is being used?**
  - **Break down into Actions and Targets**
    - **Action: defines user goals**
    - **Target: actions refer to a target, some aspect of the data that is of interest to the user**
- **Why analyze task abstractly?**
- **From Specific to General Vis Tools**
- **Actions**
- **Targets**
- **How: A Preview**
- **Analyzing and Deriving: Examples**

The Big Picture
Why a vis tool is being used?

Why analyze task abstractly?

- **Make your tasks in abstract form and describe your goals as generic as possible**
  - **Transforming task description from domain-specific language into abstract form allows you to reason about similarities and differences between them**
- **The task abstraction can and should guide the data abstraction**
  - **To understand whether and how to transform the original data into different form by deriving new data**

Who: Designer or User

- **It sometimes useful to augment an analysis instance specification by indicating who has a goal or makes a design choice:**
  - **the designer of the vis or the end user**
- **Vis tools fall somewhere along a continuum from specific to general**
  - **Specific**
    - **Tools are narrow**
      - **The designer has built many choices for specific purpose into the design of the tool itself in a way that the user cannot override**
      - **Are limited in the kinds of data and tasks that the tools can address**
      - **Strength: users are not faced with many design choices**

Who: Designer or User

- **General**
  - **Tools are flexible and users have many choices to make.**
  - **Both a strength and limitation**
    - **Users have a lot of power**
    - **They may make ineffective choices if they are not familiar with many vis design issues**

Who: Designer or User

- **Specialized vis tools**
  - **Designed for specific contexts with a narrow range of data configurations**
    - **Often an interesting mix of complex combinations of and special cases of the basic data, and a mix of original and derived data**
- **General vis tools**
  - **Designed to handle a wide range of data in a flexible way**

Actions

- **Three levels of actions that define user goals**
  - **Analyze**
    - **How the vis is used to analyze, either to consume existing data or to also produce additional data**
  - **Search**
    - **What kind of search is involved**
  - **Query**
    - **Does the user need to identify one target, compare some targets, or summarize all of the targets?**

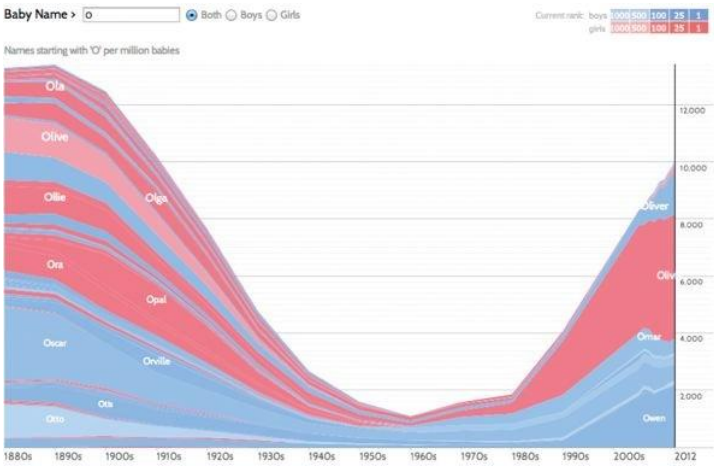
The choices at each level are independent from each other. Usually useful to describe actions at all levels.

Analyze - Consume

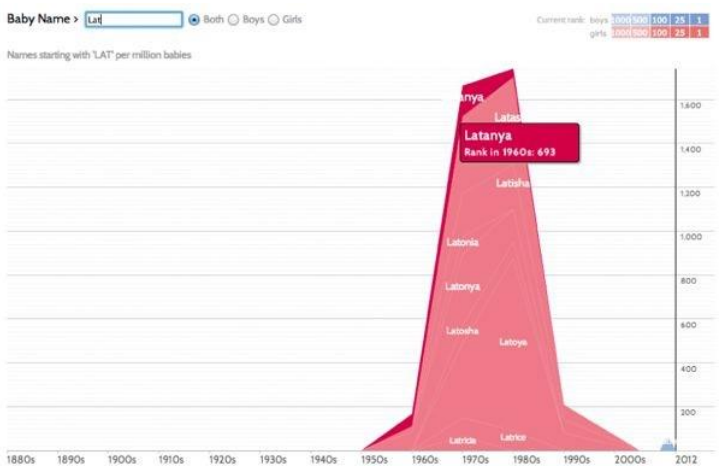
- **Discover**
  - Use vis to find new knowledge (haven't known)
    - Generate a new hypothesis, or
    - Verify an existing hypothesis
- **Present**
  - Use of vis for information communication, for telling a story with data, or guiding an audience through a series of cognitive operations
  - Vis is used to communicate something specific and already understood to an audience
- **Enjoy**
  - Casual encounters with vis, motivated by user's own enjoyment rather than a pressing need

Analyze - Consume

- **Enjoy example**
  - **Name Voyager, a vis tool originally intended for parents focused deciding on what to name their expected baby, ended up being used by many nonparents to analyze historical trends for their own enjoyment**



Names starting with "O" had a notable dip in popularity in the middle of the century



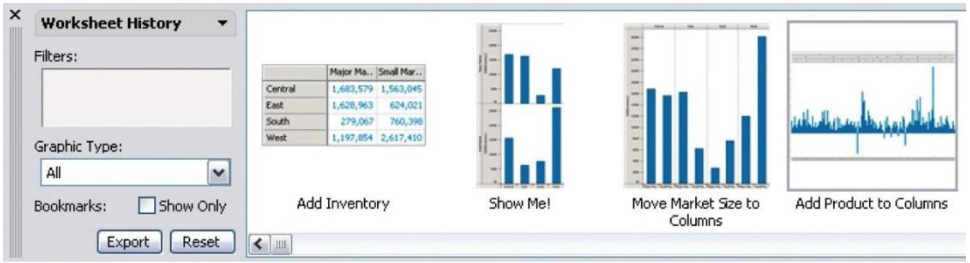
Names starting with "LAT" show a trend of the 1970s

Brightness: popularity color: gender

Analyze - Produce

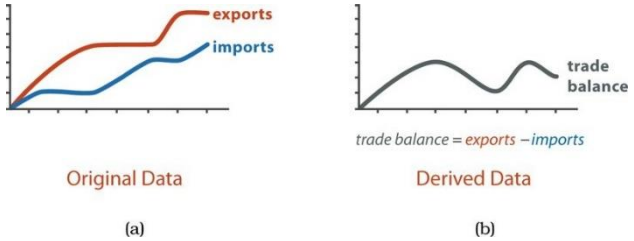
Produce output that is used immediately, as input to a next task

- **Annotate**
  - **The addition of graphical or texture annotations associated with existing visualization elements, typically as a manual action by the user**
    - **The annotation could be thought of as a new attribute for the elements**
    - **For example, the user could annotate all of the points within a cluster with a text label.**
- **Record**
  - **Saves or captures visualization elements as persistent artifacts.**
    - **Artifacts: screen shots, elements or locations, parameter settings, annotations…**
  - **One example of a record goal: Assemble a graphical history, in which the output of each task includes a static snapshot of the view showing its current state.**
  - **Recording the user's entire session of using the vis tool.**



Analyze - Produce

- **Derive**
  - **Produce new data elements based on existing data elements.**
  - **Two ways to think,**
    1. **Derive new attributes from the information contained within existing ones using arithmetic, logical, or statistical operations**
       - **For example, derive new attributes trade balance, the difference between imports and exports.**



    2. **Transform the dataset from one type to another**
       - **For example, for deciding whether water is an appropriate temperature for a shower, quantitative attribute might be transformed into a new derived attribute that is ordered: hot, warm, or cold.**

Analyze - Produce

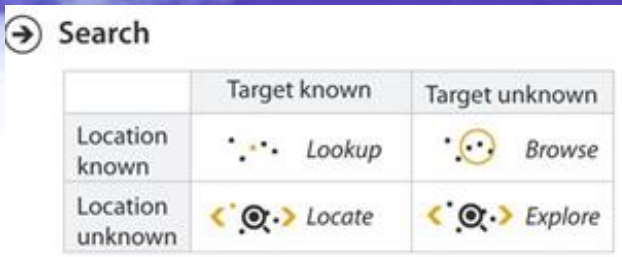
- **Derive (cont.)**
  - **There is a strong relationship between the form of data – the attribute and types- and what kind of vis idoms are effective at displaying it**
    - **Don't just draw what you're given; decide what the right thing to show is, and draw it.**
  - **"Derive" greatly expands the design space of possible vis idoms that you can use to display**
    - **Complex real-world cases frequently needs more complex data abstraction based on deriving new attributes and types**

Analyze - Produce

- **Derive (cont.)**
  - **Type transformation process may involve multiple stages of transformations**

  An example:
  - **VxInsight system transforms a table of genomics data into a network**
    - **Genomics data: a table of 6000 rows of yeast genes, and 18 column containing measurements of the gene activity level.**
    - **Creates a similarity attribute**
      - **Similarity score defined between each gene pair was computed using statistical processing**
    - **Creates a network with links only between the most similar items**

Search



All high-level analyze cases require the user to search for elements of interest as a mid-level goal.

Four kinds of search according to whether the identity and location of the search target is known or not.

- **Lookup**
  - **Users already know what they're looking for (target) and where it is (location).**
- **Locate**
  - **To find a known target at an unknown location, find out where the specific object is.**

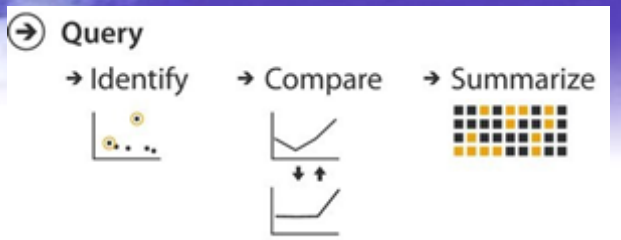
Search

- **Browse**
  - **The exact identity of a search target might not be known in advance; it might be specified based on characteristics. Users are searching for one or more items that fit the specification.**
  - **Users don't know exactly what they're looking for, but they do have a location in mind of where to look for it. (Finding something in the specific range.)**

Search

- **Explore**
  - **Both target and location are not known, the search type is explore.**
  - **Entails searching for characteristics without regard to their location, often beginning from an overview of everything.**
  - **Ex.**
    - **Search for outliers in a scatterplot**
    - **Search for anomalous spikes or periodic patterns in a line graph of time-series data**
    - **Search for unanticipated spatially dependent patterns in a choropleth map**

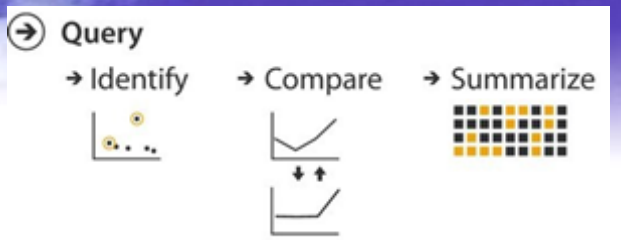
Query



Once a target or a set of targets for a search has been found, a low-level user goal is to query these targets at one of three scopes:

- **Identify**
  - **Identify refers to a single target**
- **Compare**
  - **Compare refers to multiple targets.**
- **Summarize**
  - **Summarize refers to the full set of possible targets.**

Query

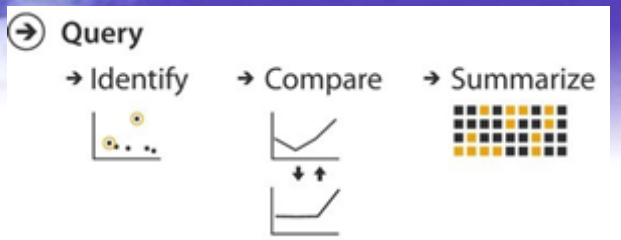


Identify

- **If a search returns known targets, either by lookup or locate, then identify returns their characteristics**

- **If a search returns targets matching particular characteristics, either by browse or explore, then identify returns specific references**

Query



Compare

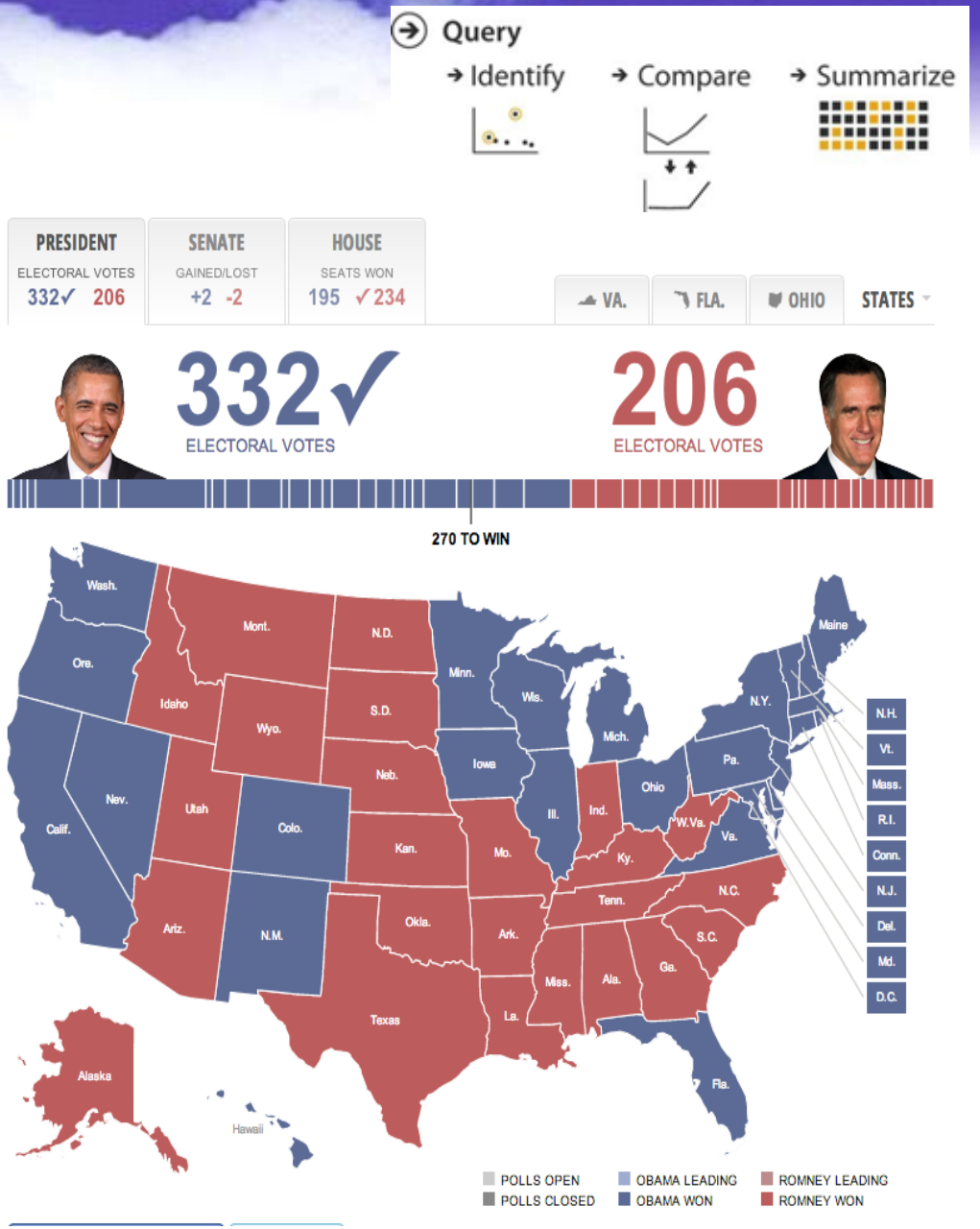
- **Typically more difficult than identify tasks and require more sophisticated idioms to support the user**

Summarize

- **Is also called "overview"**
  - **Verb: to provide a comprehensive view of everything**
  - **Noun: a summary display of everything**

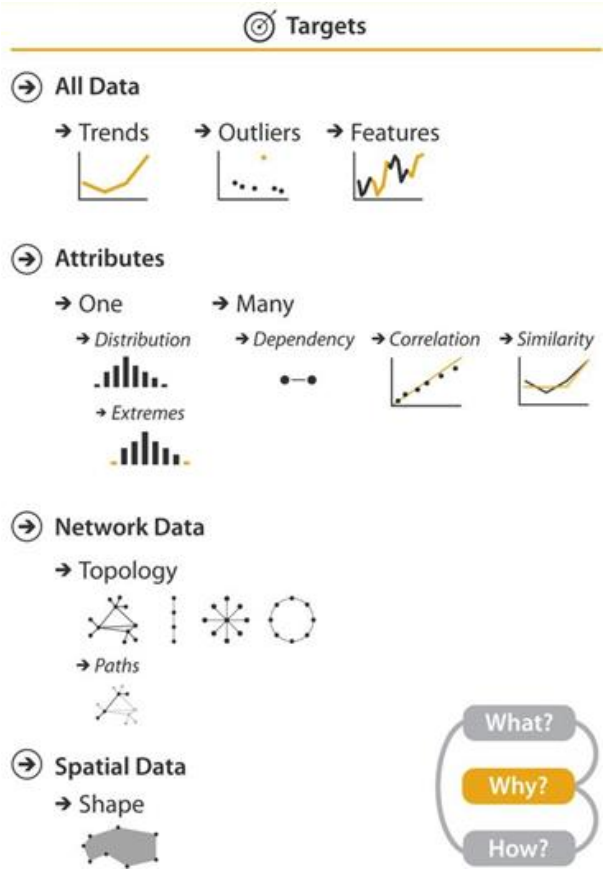
Query

Identify the election results for one state.
Compare the election results of one state to another.
Summarize the election results.



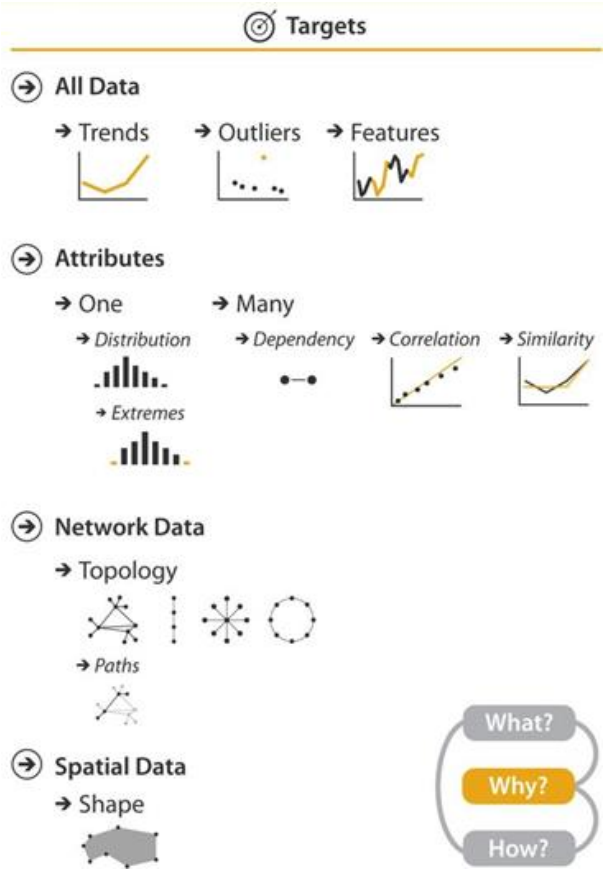
Targets

- **Some aspects of the data that the user is interested in.**
- **All kind of data**
  - **Trend**
    - **A high-level characterization of a pattern in the data**
    - **Increase, decrease, peak, trough**
  - **Outliers**
    - **data doesn't fit well.**
  - **Feature**
    - **Definition is task dependent**
    - **Any particular structure of interest**



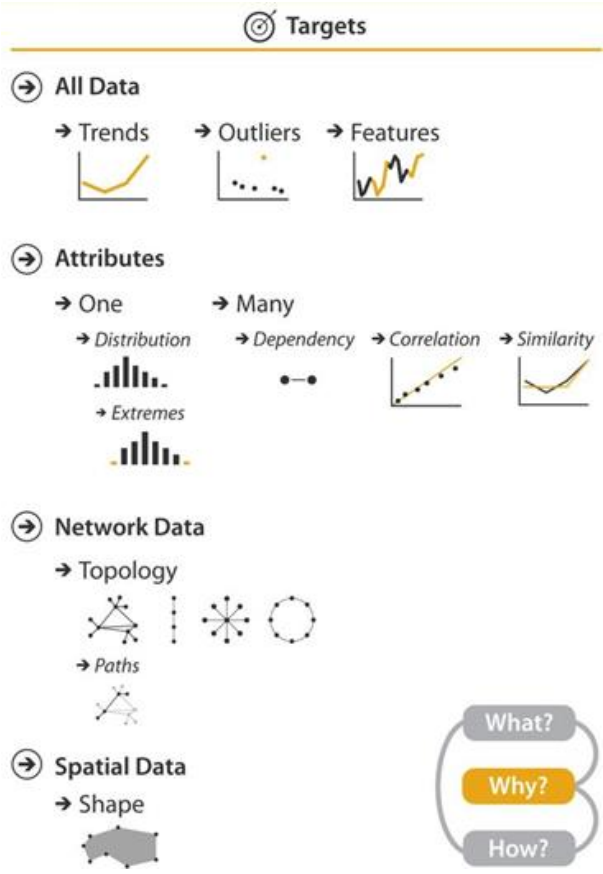
Targets

- **Attributes**
  - **Are specific properties that are visually encoded**
  - **One attribute**
    - **Find an individual value**
    - **Find its extremes**
    - **Find its distribution**
  - **Multiple attributes**
    - **See their dependency, correlation, or similarity**



Targets

- **Network data**
  - **Understand the structure of the interconnections; i.e., network topology**
  - **Find a path that connects two nodes**
- **Spatial data**
  - **Understand and compare the geometric shape**



How: A Preview

- **How a vis idiom can be constructed out of a set of design choice**
- **Four major classes**
  - **Encode**
    - **How to encode data within a view**
  - **Manipulate**
    - **How to change any aspect within a view**
  - **Facet**
    - **How to facet data between views**
  - **Reduce**
    - **How to reduce data**

How?

Encode

- **Arrange**
  - Express
  - Separate
  - Order
  - Align
  - Use

- **Map**
  from categorical and ordered attributes
  - Color
    - *Hue*
    - *Saturation*
    - *Luminance*
  - Size, Angle, Curvature, ...
  - Shape
  - Motion
    Direction, Rate, Frequency, ...

Manipulate

- **Change**

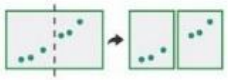
- **Select**

- **Navigate**

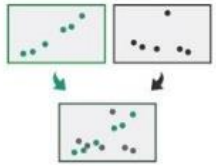
Facet

- **Juxtapose**

- **Partition**



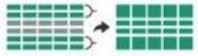
- **Superimpose**



Reduce

- **Filter**

- **Aggregate**



- **Embed**

How: A Preview

Encode: encode data within a view

- How to arrange data spatially
  - Express values
  - Separate, order, and align regions
  - Use given spatial data
- How to map data with all of the nonspatial visual channels
  - Color, size, angle, curvature
  - Shape
  - Motion

How: A Preview

Manipulate: How to change any aspect within a view

- **Select elements from within the view**
- **Navigate to change the viewpoint within the view**

How: A Preview

Facet: How to facet data between views

- How to juxtapose and coordinate multiple views
- How to partition dada between views
- How to superimpose layers on top of each other

Reduce: How to reduce data

- how to filter data away
- How to aggregate many data elements together
- How to embed focus and context information together with a single view

Analyzing and Deriving: Examples

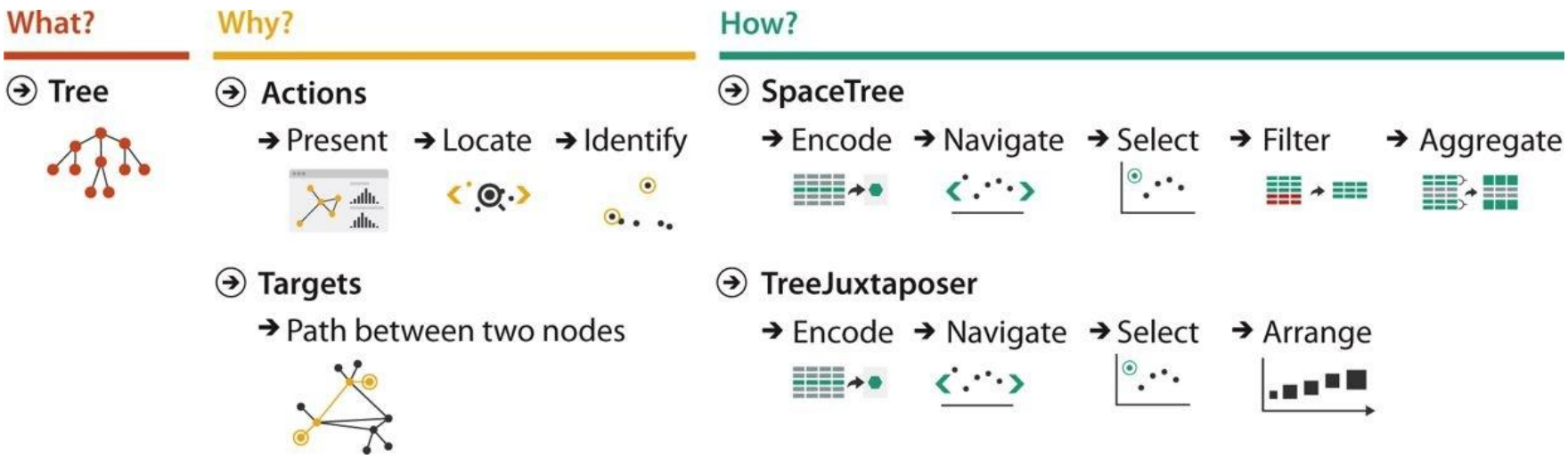
- To give a taste of how this what–why–how framework can be used right away
  - Case 1: Comparing two idioms
  - Case 2: Deriving one attribute
  - Case 3: Deriving many new attributes

Case 1: Comparing Two Idioms

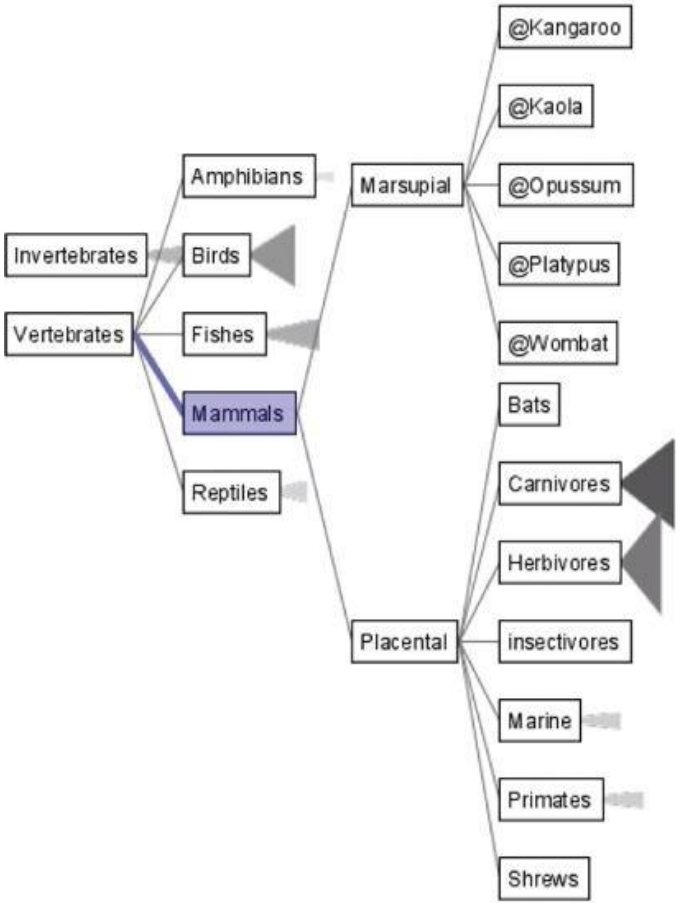
- **Two systems are different in how elements of the visualization are manipulated and arranged**
  - **For same context of why and what at the abstract level**
    - **Same input**
    - **Same goal: to present a path traced between two nodes of interest**
  - **Same aspect of idioms**
    - **Both allow the user to navigate and to select a path, with the result that it's encoded differently from the nonselected paths through highlighting**

Case 1: Comparing Two Idioms

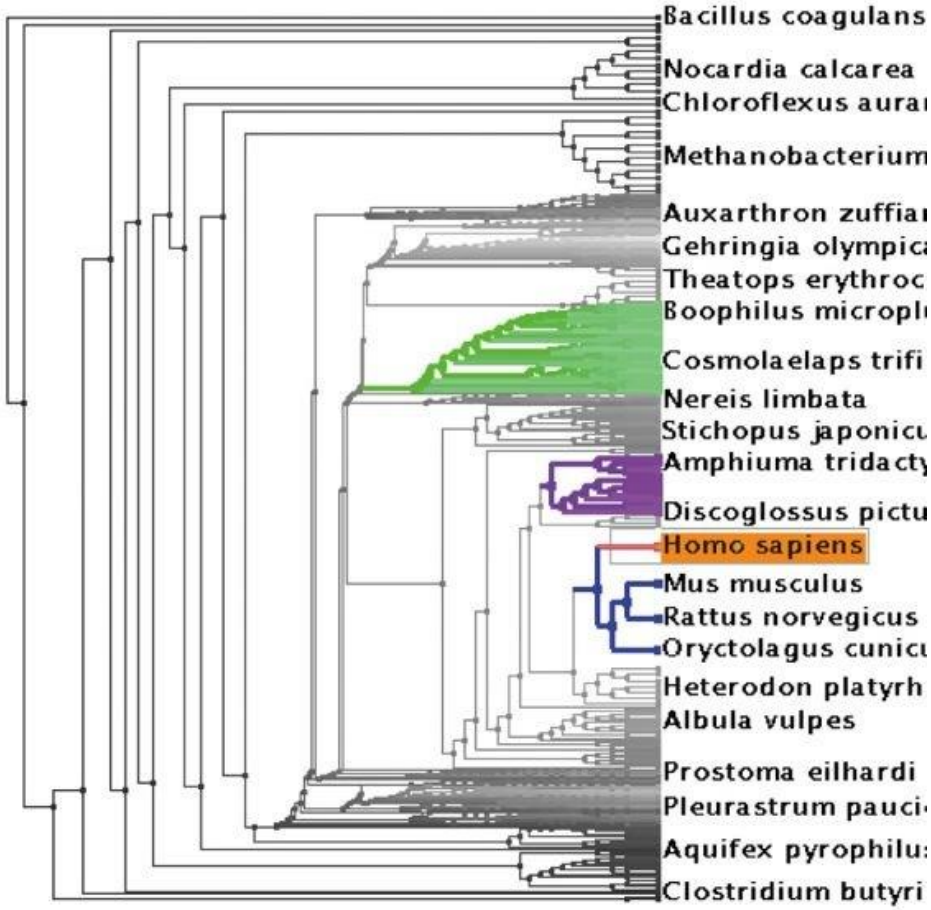
- **Differ in how elements of the vis are manipulated and arranged**
  - **SpaceTree ties the act of selection to a change of what is shown by automatically aggregating and filtering the unselected items.**
  - **TreeJuxtaposer allows the user to arrange areas of the tree to ensure visibility for areas of interest.**



Case 1: Comparing Two Idioms



(a)



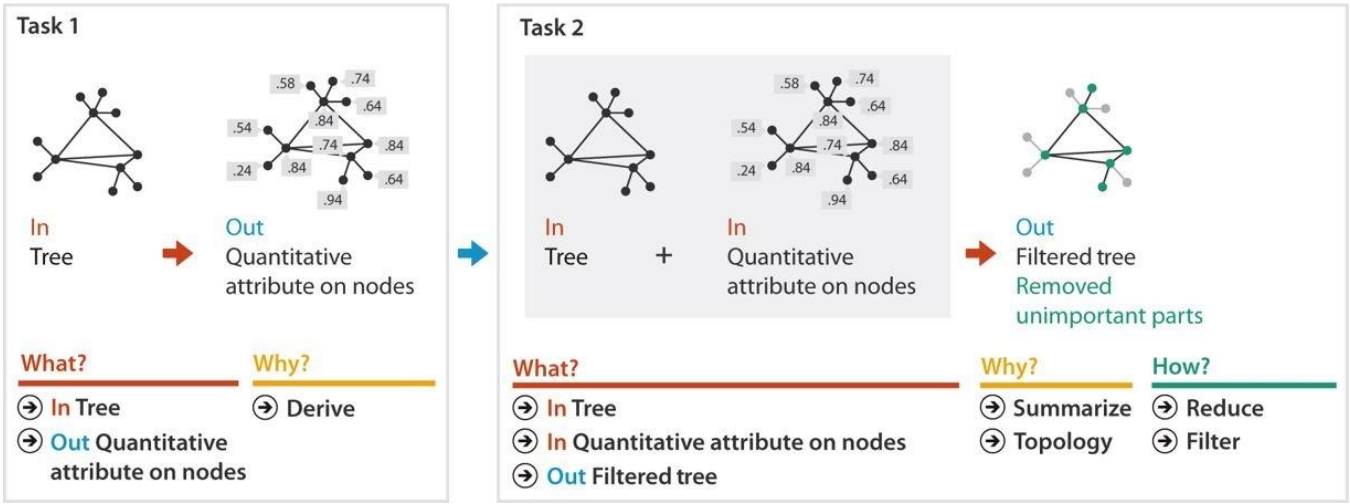
(b)

Case 2: Deriving One Attribute

- **Task 1: Derive new attributes from inputs, use Strahler Numbers.**
- **Task 2: Use Task1 output as input, and filter away the unimportant parts of a tree, in support of the task of summarizing the tree's overall topology.**

Case 2: Deriving One Attribute

- **Task 1: Derive new attributes from inputs, use Strahler Numbers.**
- **Task 2: Use Task1 output as input, and filter away the unimportant parts of a tree, in support of the task of summarizing the tree's overall topology.**



Case 2: Deriving One Attribute