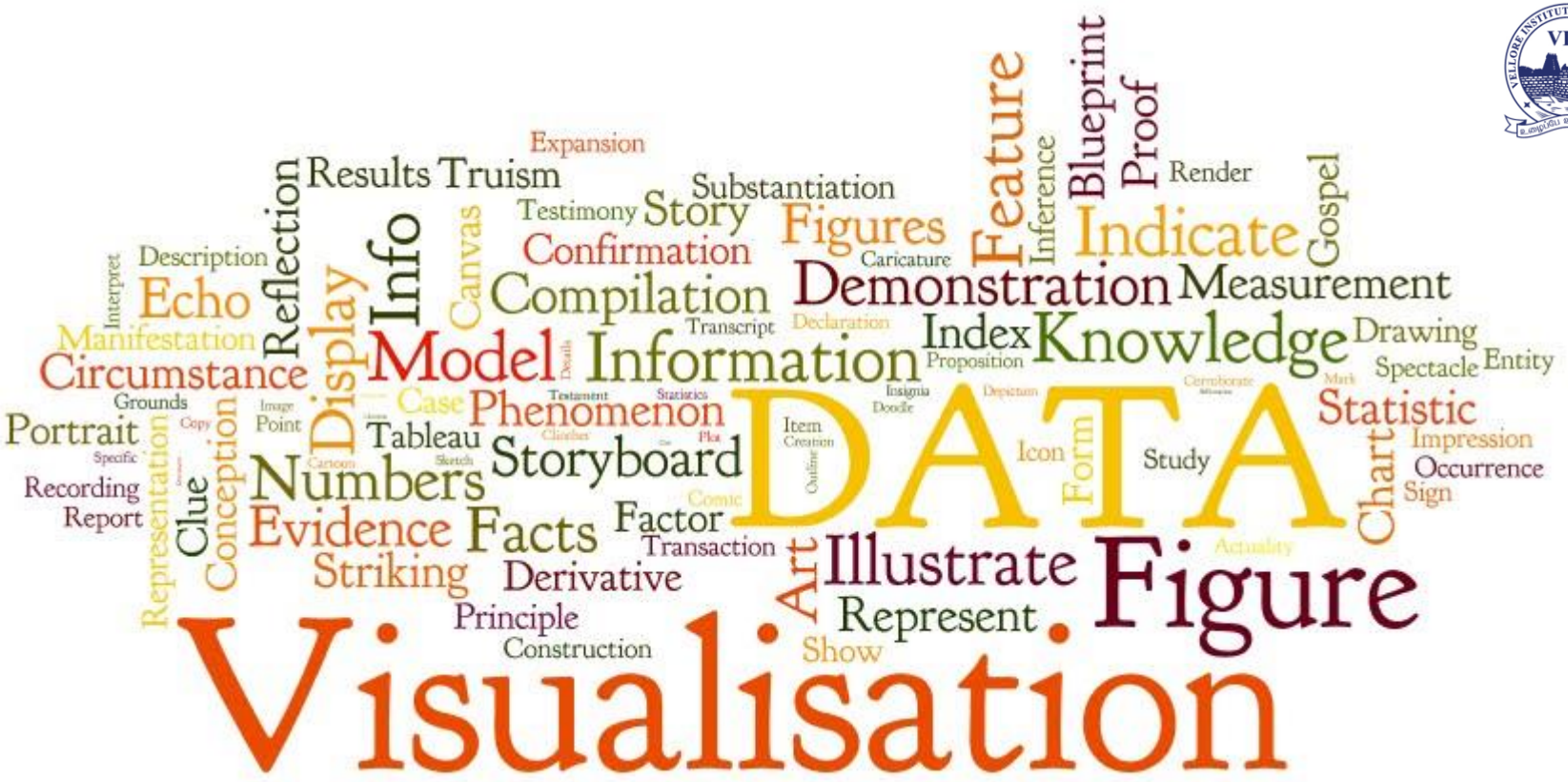


4.Task Abstraction

Prof. Ramesh Ragala

SCSE, VIT, Chennai

Task Abstraction

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

Why Analyze Tasks Abstractly?

- **The highest – level actions are to use vis to consume or produce information**
  - Consuming/Producing: to present, to discover and to enjoy
    - Discover may involve the generating or verifying a hypothesis.
  - Search: it can be classified according to whether the identity and location of target is known or not
  - Queries: three scopes
    - Identify one target, compare some targets and summarize all targets

Why Analyze Tasks Abstractly?

- Targets for **all kinds of data** are finding trends and outliers
- For **attributes**, targets can be one value, the extremes of minimum and maximum values, or the distribution of all values across the entire attribute
- For **multiple attributes**, the target can be dependencies, correlations, or similarities between them.
- The target with network data can be topology in general or paths in particular, and with spatial data the target can be shape.

Why Analyze Tasks Abstractly?

- Vis framework encourages you to consider **tasks** in ***abstract*** form, rather than the ***domain-specific*** way.
- Transforming task descriptions from domain-specific language into abstract form allows you to reason about ***similarities*** and ***differences*** between them.
- Otherwise, it's ***hard to make useful comparisons*** b/w domain situations, because if you don't do this kind of translation then everything just appears to be different. That apparent difference is ***misleading***.

Why Analyze Tasks Abstractly?

- The vis tool might be usable for many different goals.
- It is often useful to consider only one of the user's goals at a time, in order to more easily consider the question of how a particular idiom supports that goal.
- To describe complex activities, you can specify a chained sequence of tasks, where the output of one becomes the input to the next.
- Another important reason to **analyze the task** is to understand whether and how to transform the user's original data into different forms by **deriving new data**. That is, the task abstraction can and should guide the data abstraction.

Who: Designer or User

- Who has a goal or makes a design choice: **the designer of the vis or the end user of the vis.**
- The designer has built many choices.
- **Tools are limited(kinds of data and tasks that they can address),** but their strength is that users are not faced with an overwhelming array of design choices.
- The breadth of choices is both a strength and a limitation:

Who: Designer or User

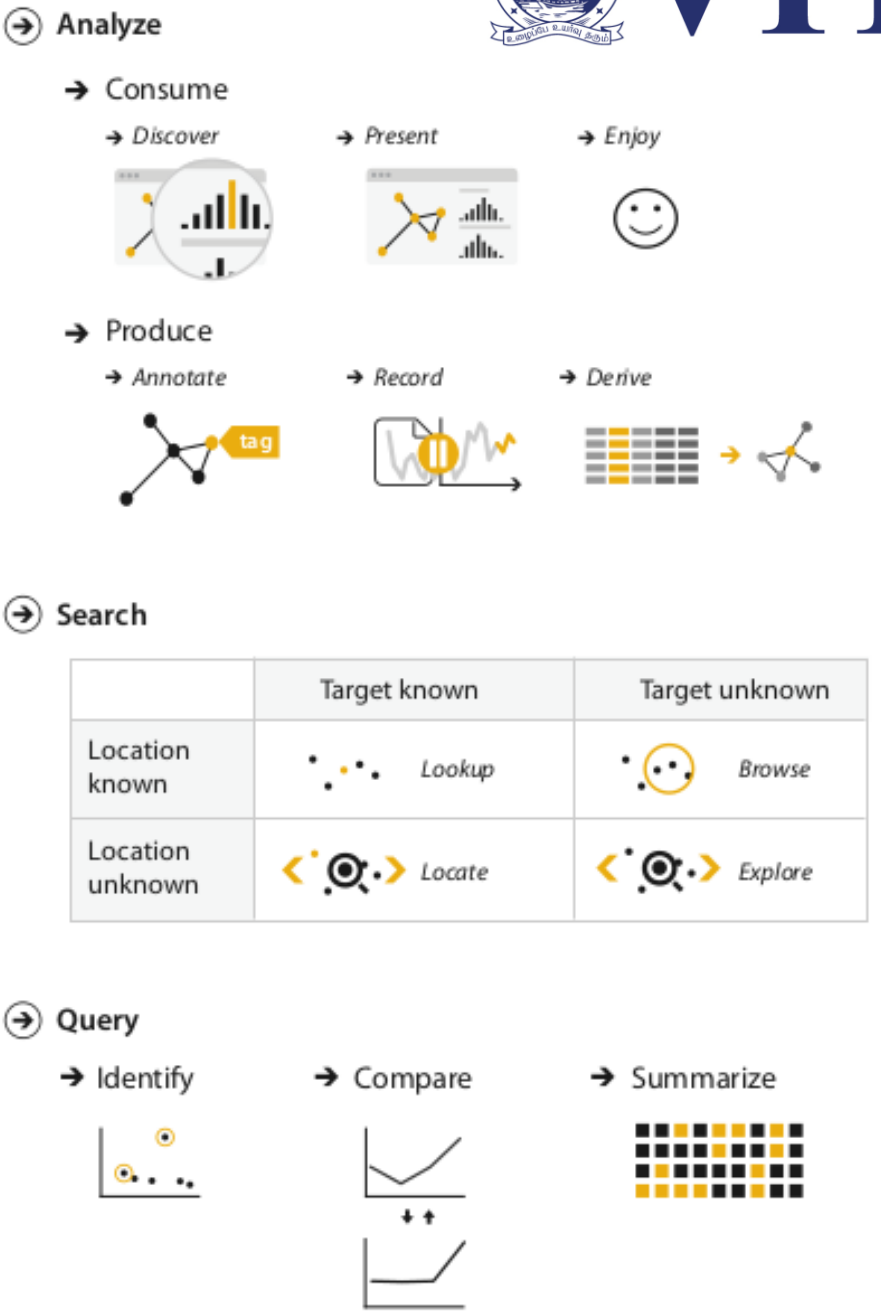
- Users have a lot of power, but they also may make **ineffective choices** if they do not have a **deep understanding** of many vis design issues.

Actions

- Three levels of actions that define user goals.
- The high-level choices describe **how the vis is being used to analyze**, either to **consume existing data** or to also **produce additional data**.
- The mid-level choices cover what **kind of search** is involved, in terms of whether the target and location are known or not.
- The low-level choices pertain to the **kind of query**: does the user need to identify one target, compare some targets, or summarize all of the targets?

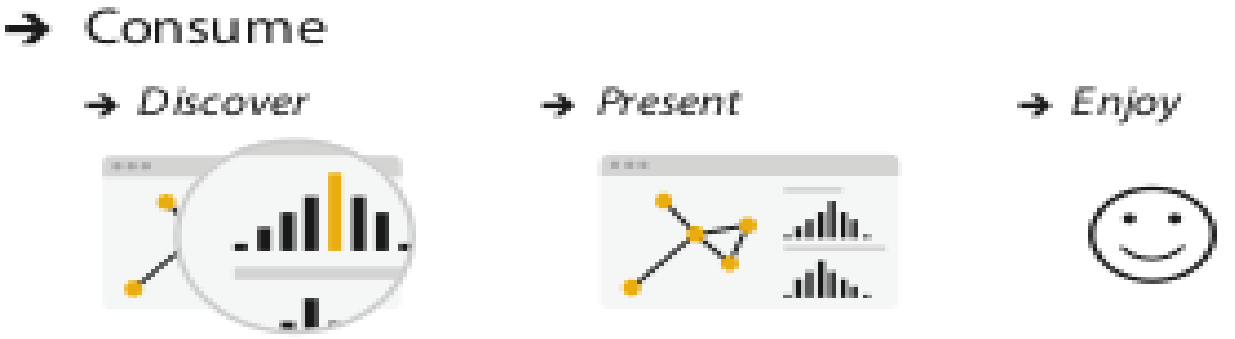
Actions

The choices at each of these ***three levels*** are independent from each other, and it's usually useful to describe actions at all three of them.



Actions - Analyze

- Two possible goals
- 1.Consume existing information(common).
  - **Discover** something new → it arise from serendipitous observation of unexpected phenomena.
  - Analyze information that is already completely understood to third party (**present**) → story telling with data, guiding audience through series of cognitive operations. → decision making, planning, etc
  - For users to enjoy a vis to indulge their casual interests in a topic



Actions – Analyze

- Produce: user is to generate new material
  - The goal with produce is to produce output that is used immediately.
- Two possible goals
- 2.Actively produce new information.
- There are three kinds of produce goals:
  - Annotate
  - Record
  - Derive

Actions – Analyze

- Annotate
  - Refers to the addition of graphical or textual annotations associated with one or more preexisting visualization elements
  - It is a manual action by the user
- Record
  - Its goal is saves or captures visualization elements as persistent artifacts.
  - Examples of artifacts screen shots, list of bookmarked elements or locations, parameter settings, interaction logs etc.
  - The record choice saves a persistent artifact, in contrast to the annotate.

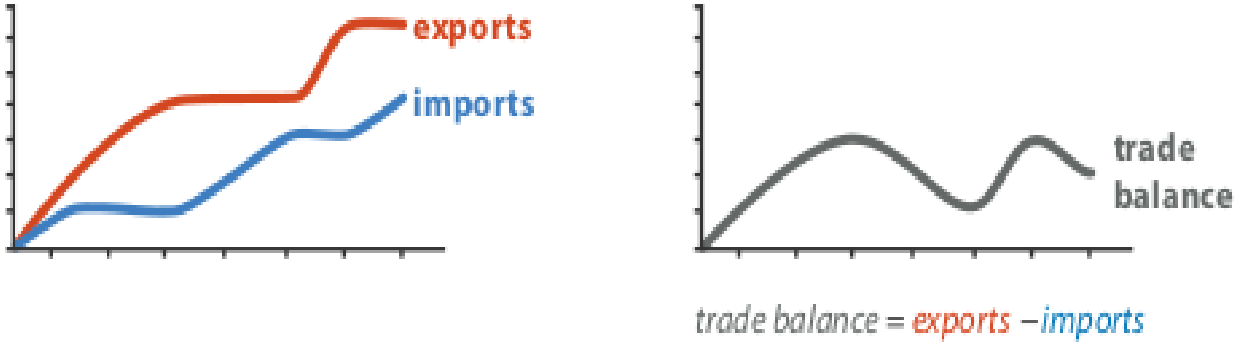
Actions – Analyze

- Output of each task includes static snapshot.

Figure 3.4. Graphical history recorded during an analysis session with Tableau. From [Heer et al. 08, Figure 1].

Actions – Analyze

- Derive:
  - Its goal is to produce new data elements based on existing data elements
  - Data can be transformed from one type to another.
  - It is used by vis designers



Search

- All of the high-level analyze cases require the user to ***search*** for elements of interest within the vis as a mid-level goal.
- The classification of search into four alternatives according to search target is ***already known or not***.

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

Search - Lookup

- If users already know ***both*** what they're looking for and where it is, then the search type is simply lookup.

- Example, a user of a tree vis showing the ancestral relationships between mammal species might want to look up humans, and can get to the right spot quickly by remembering how humans are classified.

Search - Locate

- To find a known target at an unknown location, the search type is locate: that is, find out where the specific object is.
- User might not know where to find rabbits, and would have to look around in a number of places before locating them as lagomorphs (not rodents)!

Search - Browse

- Users don’t know exactly what they’re looking for, but they do have a location in mind of where to look for it, the search type is browse.
- For instance, if a user of a tree vis is searching within a particular subtree for leaf nodes having few siblings, it would be an instance of browse because the location is known in advance, even though the exact identity of the search target isn’t.

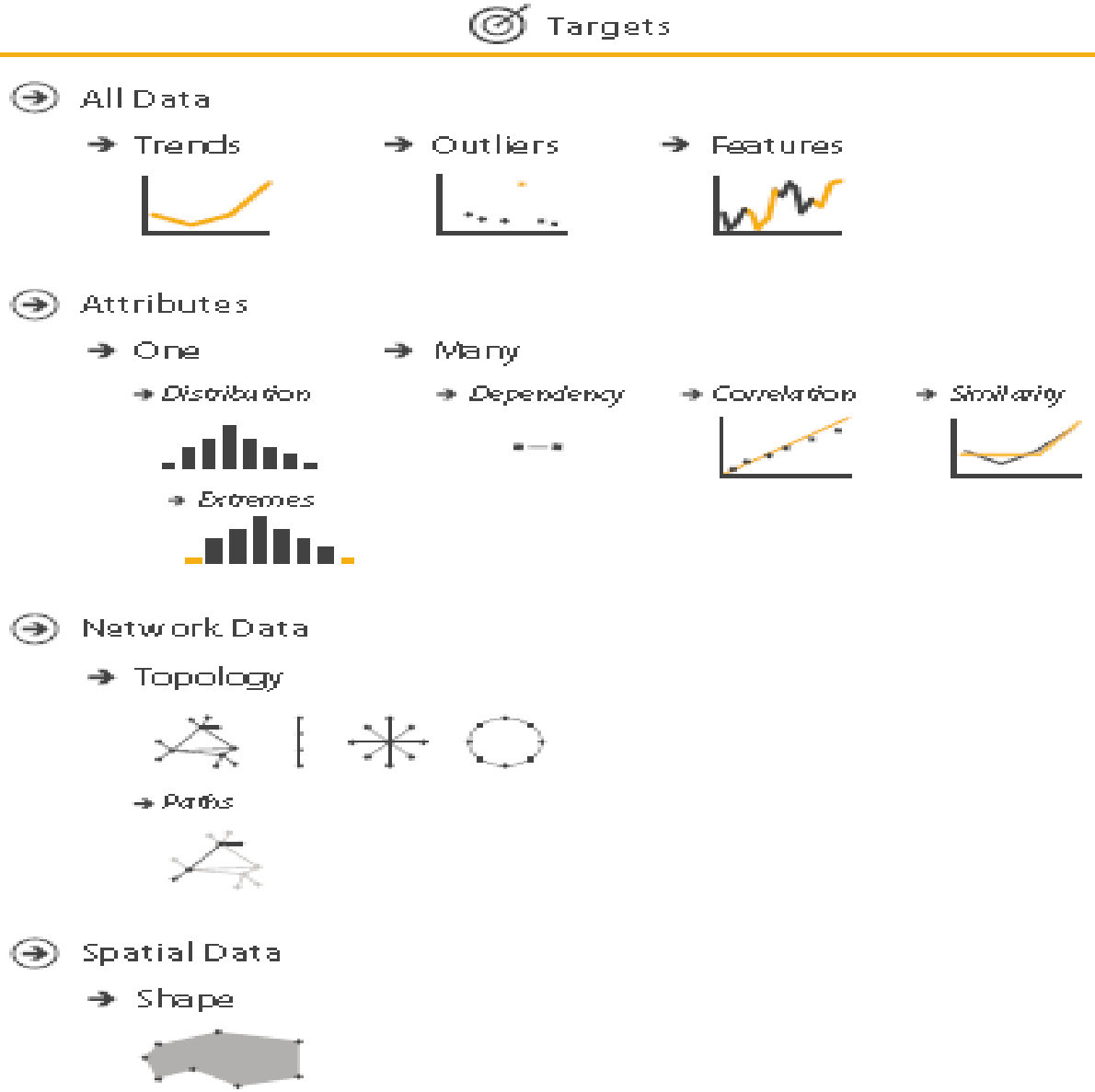
Search - Browse

- When users are not even sure of the location, the search type is explore. It entails searching for characteristics without regard to their location, often beginning from an overview of everything.
- Examples - Searching for outliers in a scatterplot, for anomalous spikes or periodic patterns in a line graph of time-series data.

Query

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

Targets


Targets
All Data
Trends
Outliers
Features
Attributes
One
Many
Distribution
Extremes
Dependency
Correlation
Similarity
Network Data
Topology
Paths
Spatial Data
Shape

Targets

- Four kinds of abstract targets
- Targets → some aspect of the data that is of interest to the user
- Targets are nouns, where as actions are verbs
- The idea of a target is explicit with search and query actions
- Three highest-level of targets are
  - Trends
  - Outliers
  - features

Targets

- Trends
  - It is a high-level characterization of a pattern in the data
  - Examples: trends in increase, decrease, peaks, etc
- Outliers
  - Almost inevitably, some data doesn’t fit well with that backdrop; those elements are the outliers.
- Features:
  - The exact definition of features is task dependent, meaning any particular structures of interest.

Targets

- Attributes:
  - Attributes are specific properties that are visually encoded.
  - The lowest-level target for an attribute is to find an individual value.
  - Finding extreme values: minimum and maximum values across the range
  - Distribution of all values for an attribute

Targets

- Attributes: (Multiple Attributes)
  - Dependencies
  - Correlations
  - Similarities

Targets

- Some of the targets are specific types of datasets
  - Network data
    - Targets → networks topology, path
  - Spatial datasets
    - Targets → geometric shape

Visual Encoding

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

visual encoding

Visual Encoding

- How a visualization idiom can be constructed out of a set of design choices.

Thank You