University of BRISTOL

Visual Analytics: Task Abstraction

Ian Nabney

ian.nabney@bristol.ac.uk

bristol.ac.uk

University of BRISTOL

Overview

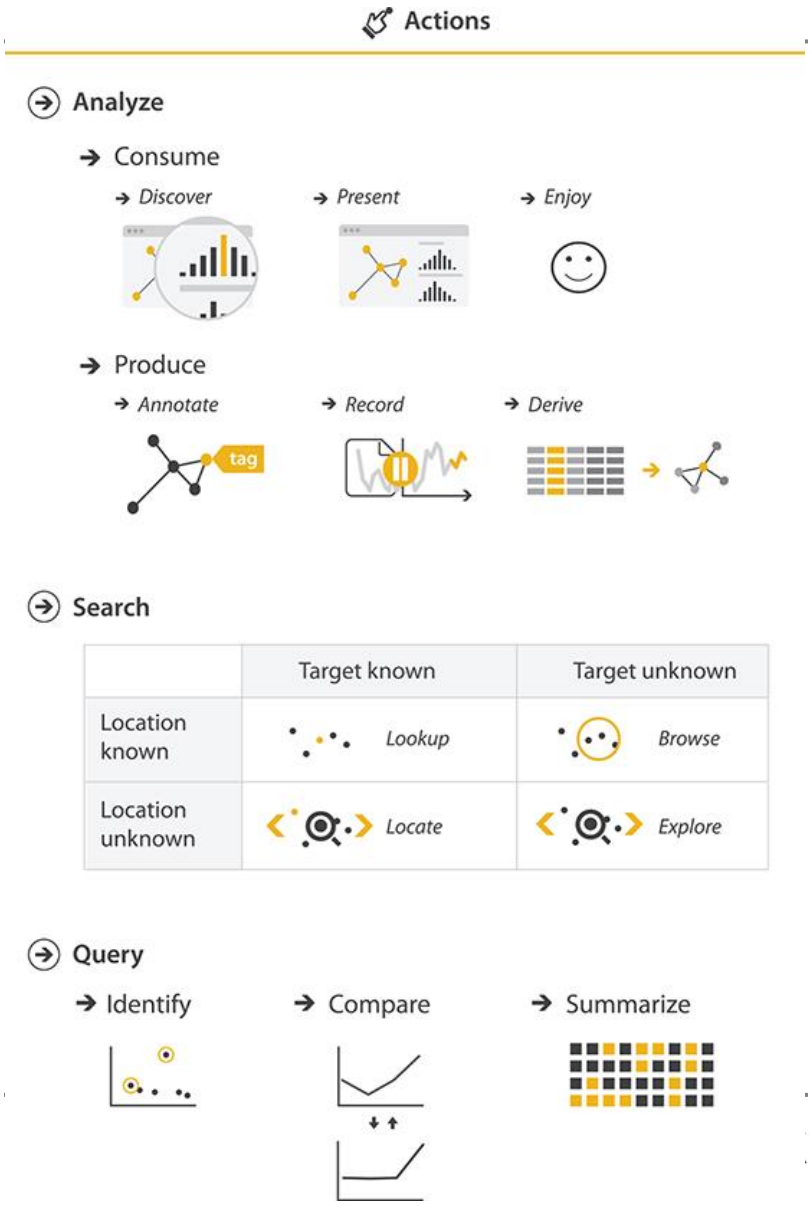
- Reading: Chapter 3 of Munzner (e-book that can be downloaded from UoB library website)
- Understand why task analysis is important to create good visualisations
- Analyse a task in terms of **actions** and **targets**
- Apply this framework to a case study

Why analyse tasks abstractly?

- Transforming task descriptions from domain-specific language into abstract form allows you to reason about similarities and differences between them
- In particular, it helps you to design a visualisation by focusing on what works in a generic way for certain types of task
  - an epidemiologist studying the spread of a new strain of influenza might initially describe her task as “contrast the prognosis of patients who were intubated in the ICU more than one month after exposure to patients hospitalized within the first week”
  - a biologist studying immune system response might use language such as “see if the results for the tissue samples treated with LL-37 match up with the ones without the peptide”
  - “compare values between two groups”
- Why are people using vis:
  - Verbs describing actions
  - Nouns describing targets/goals

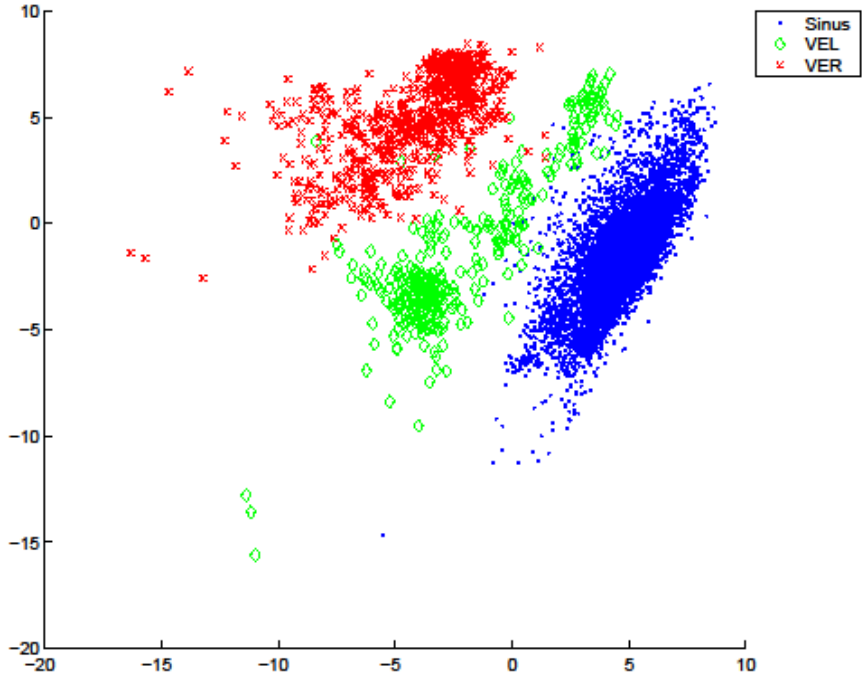
Actions

- Analyse to consume existing data or produce new data
- Search (target known or unknown)
- Query



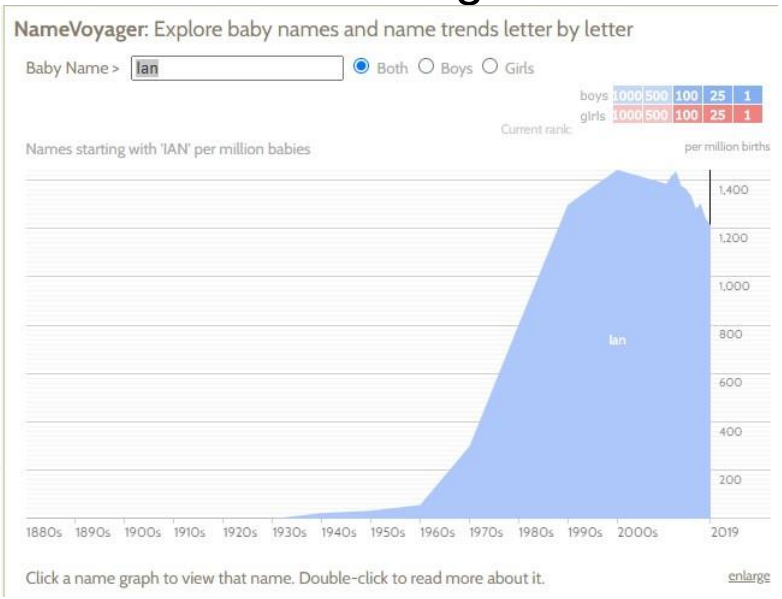
Analyse: discover

- Find new knowledge through visualisation. May be
  - serendipitous observation
  - motivated by theories or models
- This can be
  - generation of a new hypothesis (explore)
  - verify a theory (explain)
- Example – identify outliers



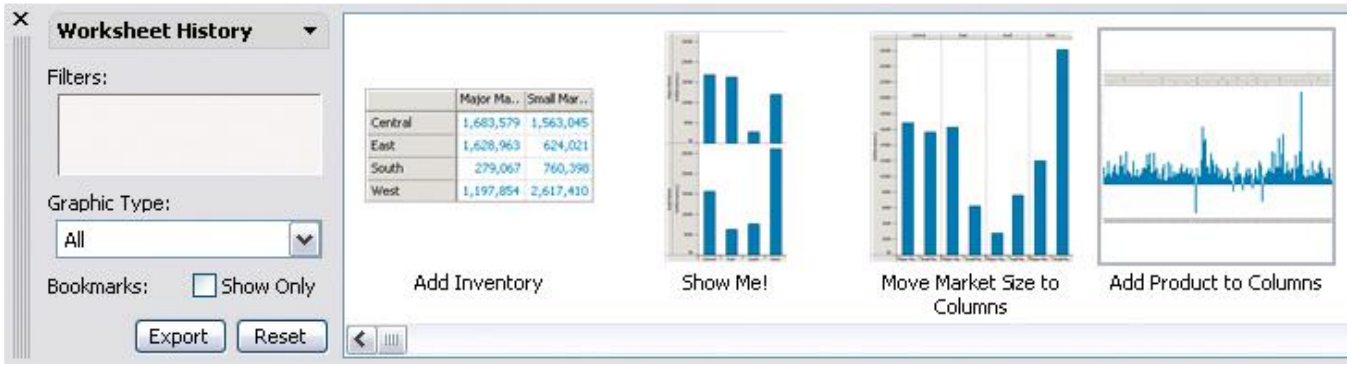
Analyse: present and enjoy

- Present:
  - use of vis for the succinct communication of information, for telling a story with data, or guiding an audience through a series of cognitive operations
  - knowledge presented is known in advance
- Enjoy:
  - user driven by curiosity
  - this may not have been the original intention of the designer!
  - I was born in 1963
  - ‘Ian’ was much less popular then



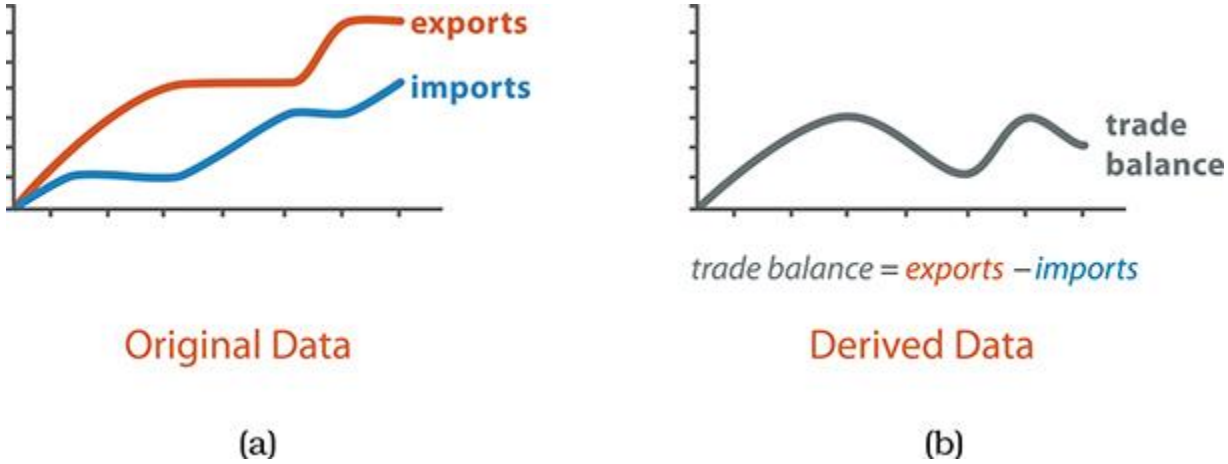
Produce: annotate and record

- The annotate goal refers to the addition of graphical or textual annotations associated with one or more pre-existing visualization elements, typically as a manual action by the user.
- When an annotation is associated with data items, it is effectively a new attribute for them.
- The record goal saves or captures visualization elements as persistent artifacts.
- A graphical history includes a static snapshot of the view showing its current state, and these snapshots accumulate in a branching meta-visualization for the user's entire session.



Produce: derive

- Produce new data elements based on existing data elements
- **Transform** the data with new attributes based on existing ones
  - Sometimes this is a simple change of type (e.g. discretising a variable)
  - Sometimes it means a mathematical calculation (e.g. linear combination of existing variables – see dimensionality reduction later in the course)



Search

- The analyse use cases nearly all involve the user in searching for elements of interest
- Four alternatives categorised in table

Search

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

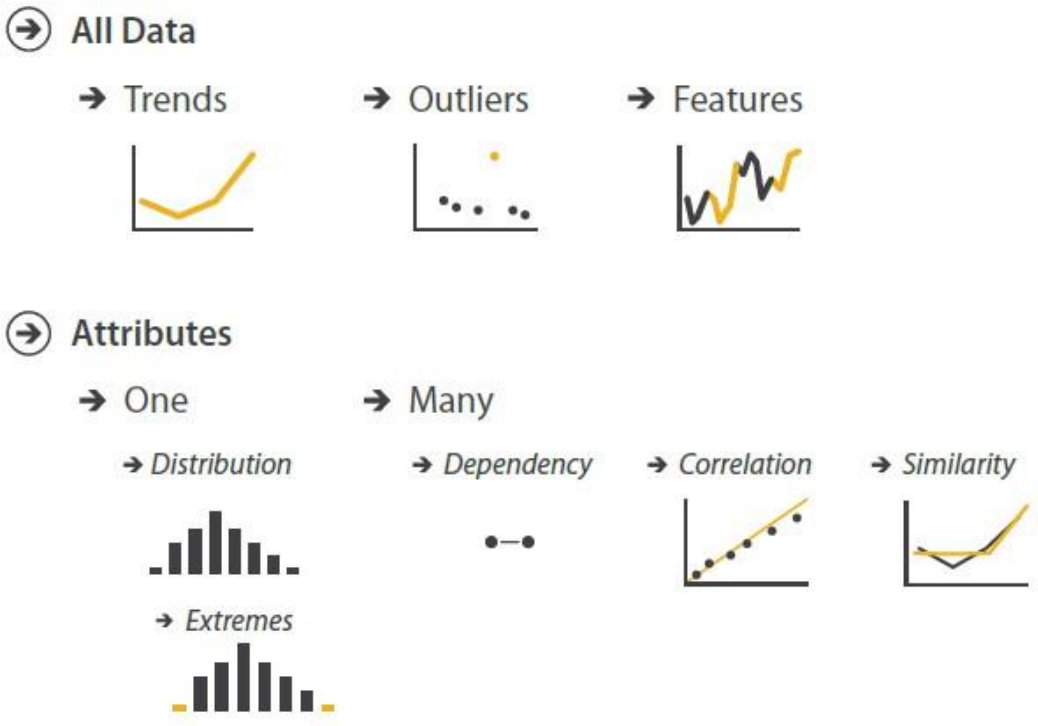
Query

- **Identify** – characteristics of a single target
- **Compare** – multiple targets which require more sophisticated vis idioms to support the user
- **Summarise** – an overview of all possible targets

- Broadly speaking: produce, search, query correspond to the three parts of Shneiderman's mantra: overview first, (zoom and) filter, details on demand

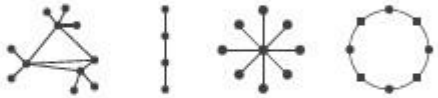
Targets

- The goal of the user encompasses an action based on a target
- Trends are high-level patterns in the data
- Outliers are data points that don't follow trends
- Features are 'everything else', perhaps more local trends, clustering



Targets: networks and spatial data

- **Network Data**
  - Topology

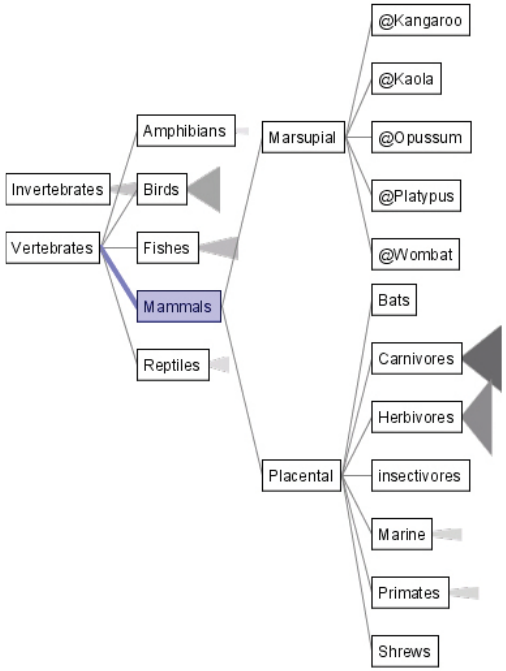


  - *Paths*

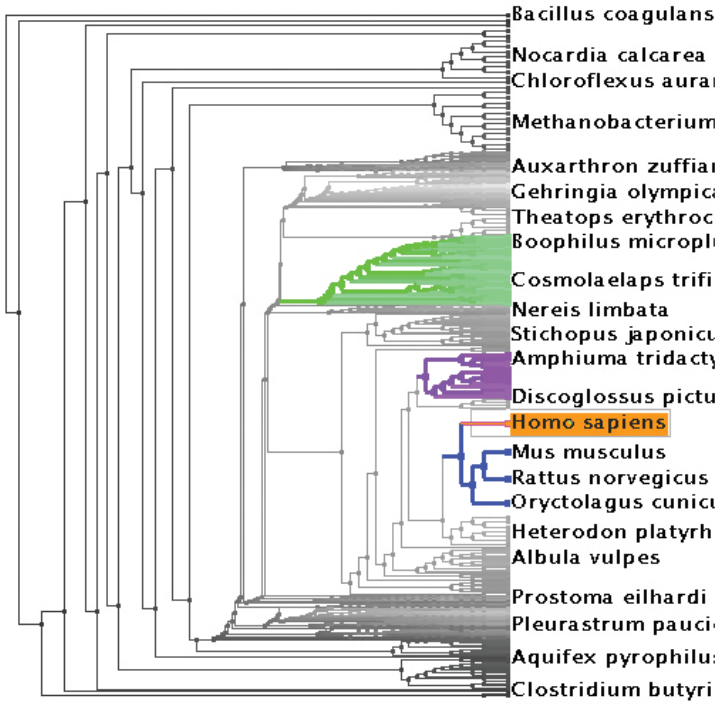
- **Spatial Data**
  - Shape

Case study: comparison of tree visualisers

- Comparing two vis tools that differ in **how** the idiom works when used for exactly the same context of **what** and **why** at abstraction level



SpaceTree



TreeJuxtaposer

Analysis

- Same input data (**what**): a large tree composed of nodes and links.
- **Why** these tools are being used is for the same goal: to present a path traced between two nodes of interest
- Some aspects of idioms (**how**) are the same: both systems allow the user to navigate and to select a path, so it's encoded differently from the non-selected paths through highlighting.
- The systems differ in how elements of the visualization 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

Comparative analysis

What?

- Tree

Why?

- Actions
  - → Present → Locate → Identify



- Targets
  - → Path between two nodes

How?

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





- TreeJuxtaposer
  - → Encode → Navigate → Select → Arrange




University of BRISTOL

Summary

- Understand why task analysis is important to create good visualisations
- Analyse a task in terms of **actions** and **targets**
- Apply this framework to a case study