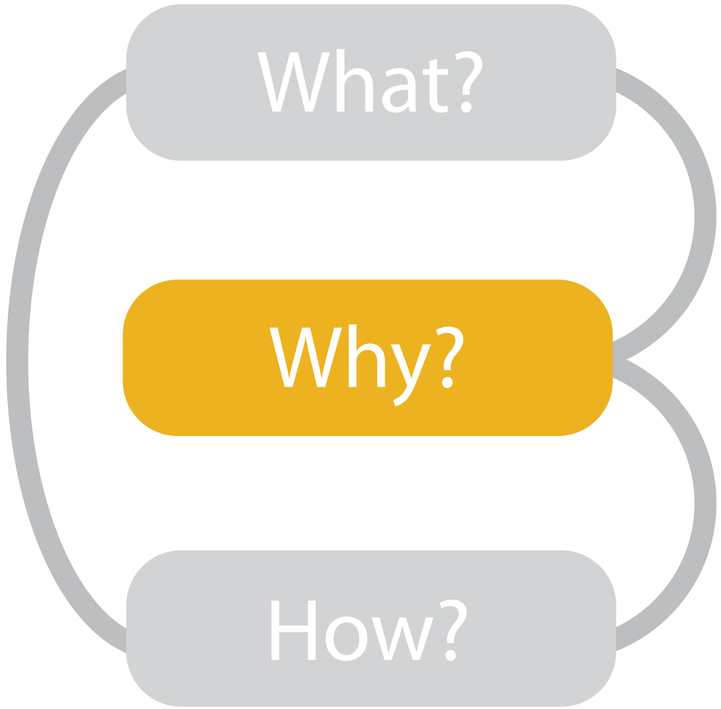
Data Visualization (CSCI 627/490)

D3

Dr. David Koop

Tasks



Why?

Actions

Analyze

- Consume
  - *Discover*
  - *Present*
  - *Enjoy*
- Produce
  - *Annotate*
  - *Record*
  - *Derive*

Search

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

Query

- Identify
- Compare
- Summarize

Targets

All Data

- Trends
- Outliers
- Features

Attributes

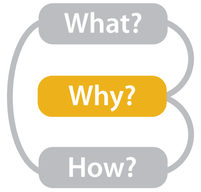
- One
  - *Distribution*
  - *Extremes*
- Many
  - *Dependency*
  - *Correlation*
  - *Similarity*

Network Data

- Topology
  - *Paths*

Spatial Data

- Shape



[Munzner (ill. Maguire), 2014]

Visualization for Consumption

- Discover new knowledge
  - Generate new hypothesis or verify existing one
  - Designer doesn’t know what users need to see
  - "why doesn't dictate how"
- Present known information
  - Presenter already knows what the data says
  - Wants to communicate this to an audience
  - May be static but not limited to that
- Enjoy
  - Similar to discover, but without concrete goals
  - May be enjoyed differently than the original purpose

Actions: Search

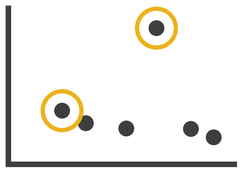
- What does a user know?
  - Lookup: check bearings
  - Locate: find on a map
  - Browse: what's nearby
  - Explore: where to go
    - Patterns

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

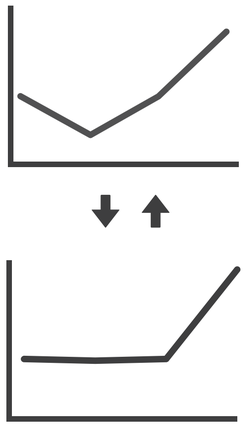
[Munzner (ill. Maguire), 2014]

Query

➔ Identify



➔ Compare



➔ Summarize

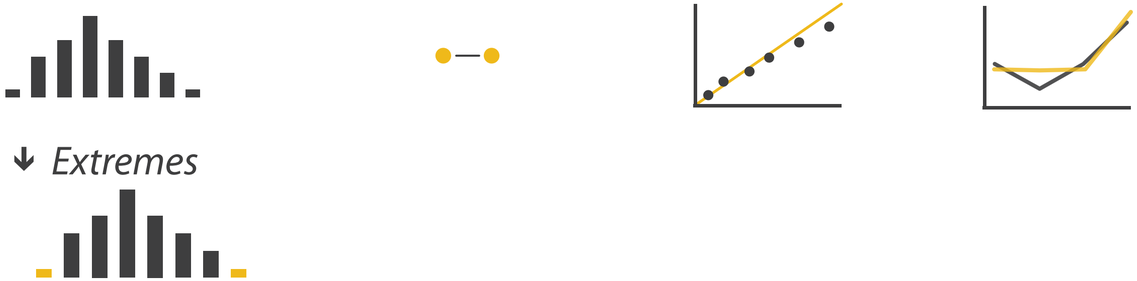
- Number of targets: One, Some (Often 2), or All
- Identify: characteristics or references
- Compare: similarities and differences
- Summarize: overview of everything

[Munzner (ill. Maguire), 2014]

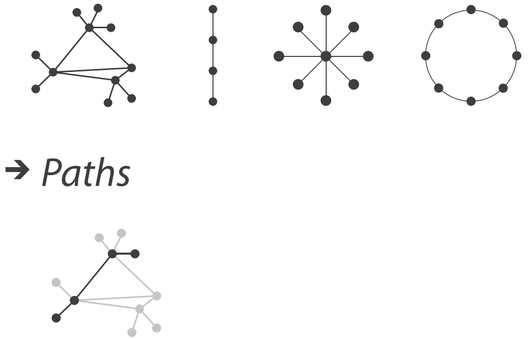
Targets

- ALL DATA
  - Trends
  - Outliers
  - Features

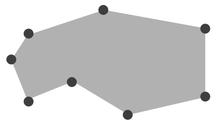
- ATTRIBUTES
  - One
    - *Distribution*
    - *Extremes*
  - Many
    - *Dependency*
    - *Correlation*
    - *Similarity*



- NETWORK DATA
  - Topology
  - *Paths*

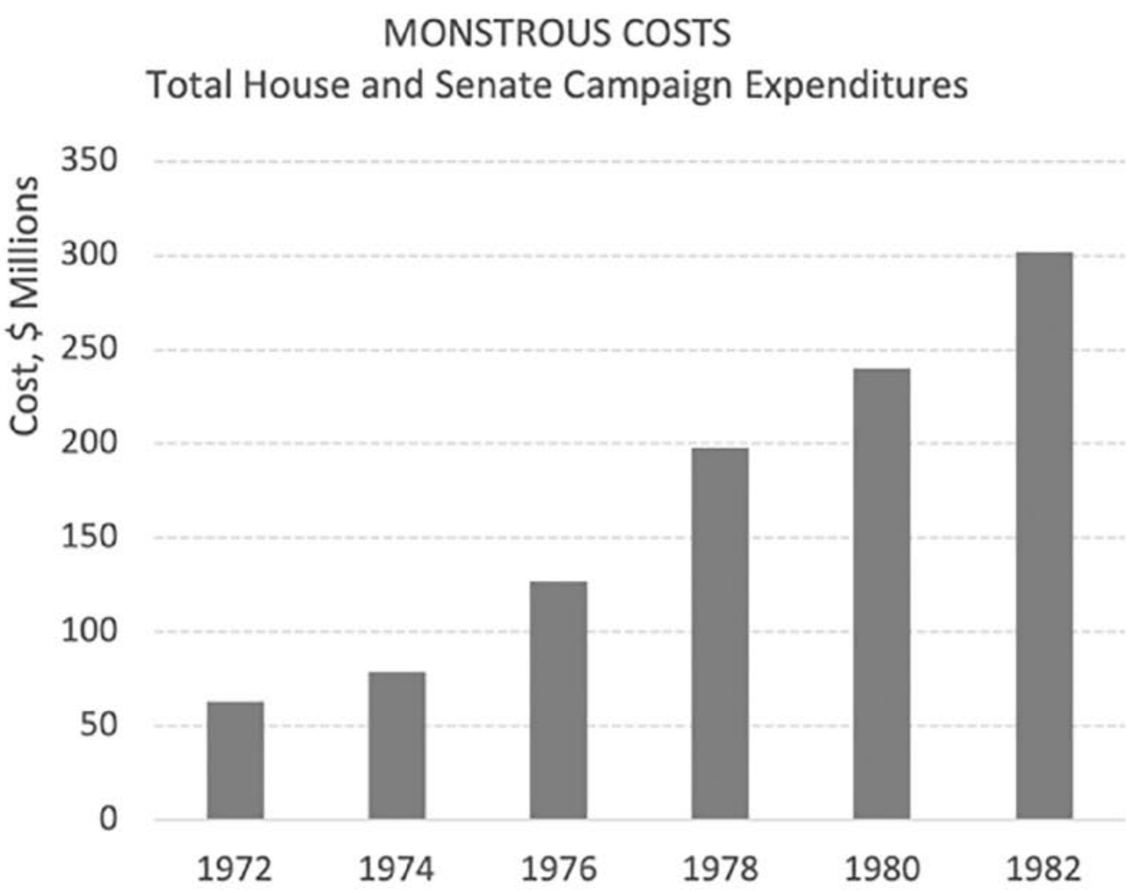


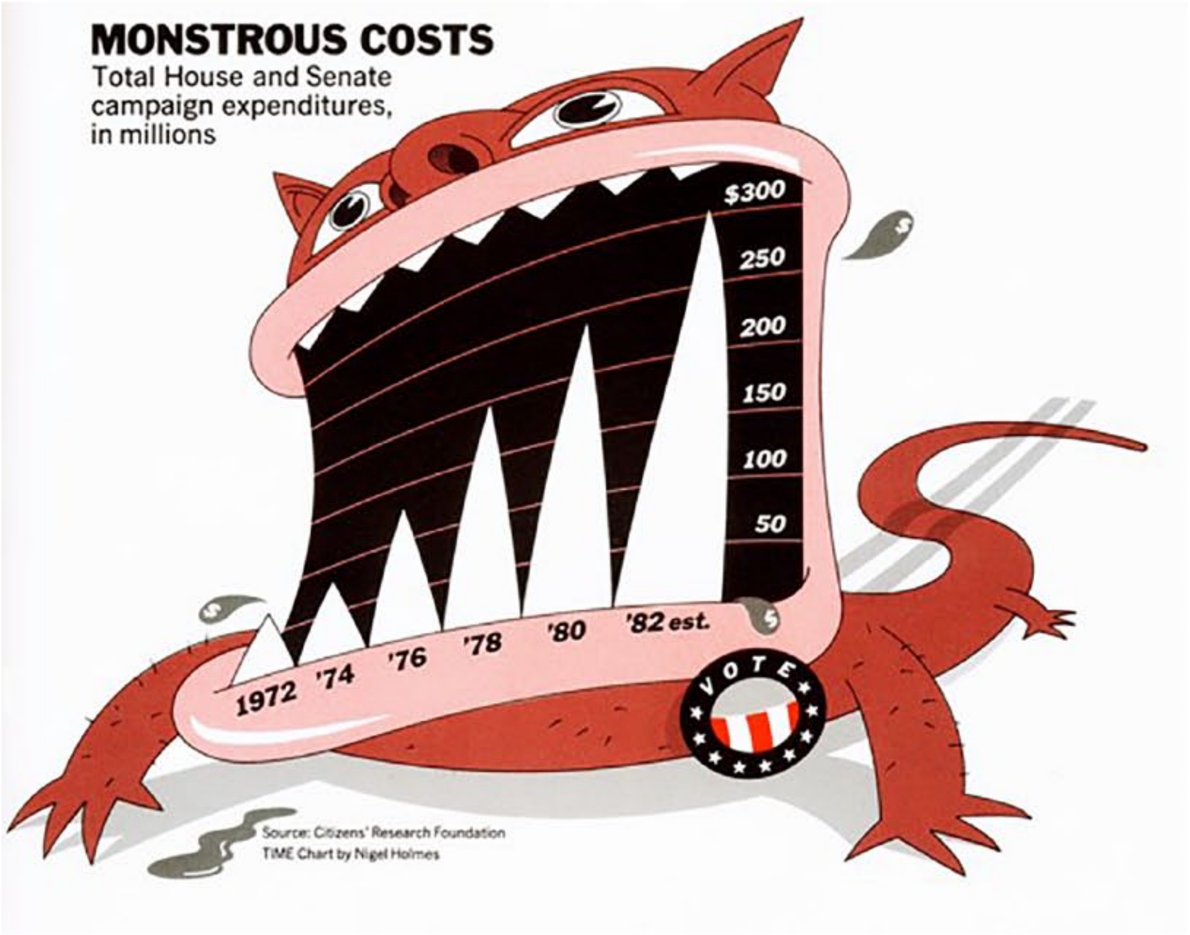
- SPATIAL DATA
  - Shape



[Munzner (ill. Maguire), 2014]

Memorability

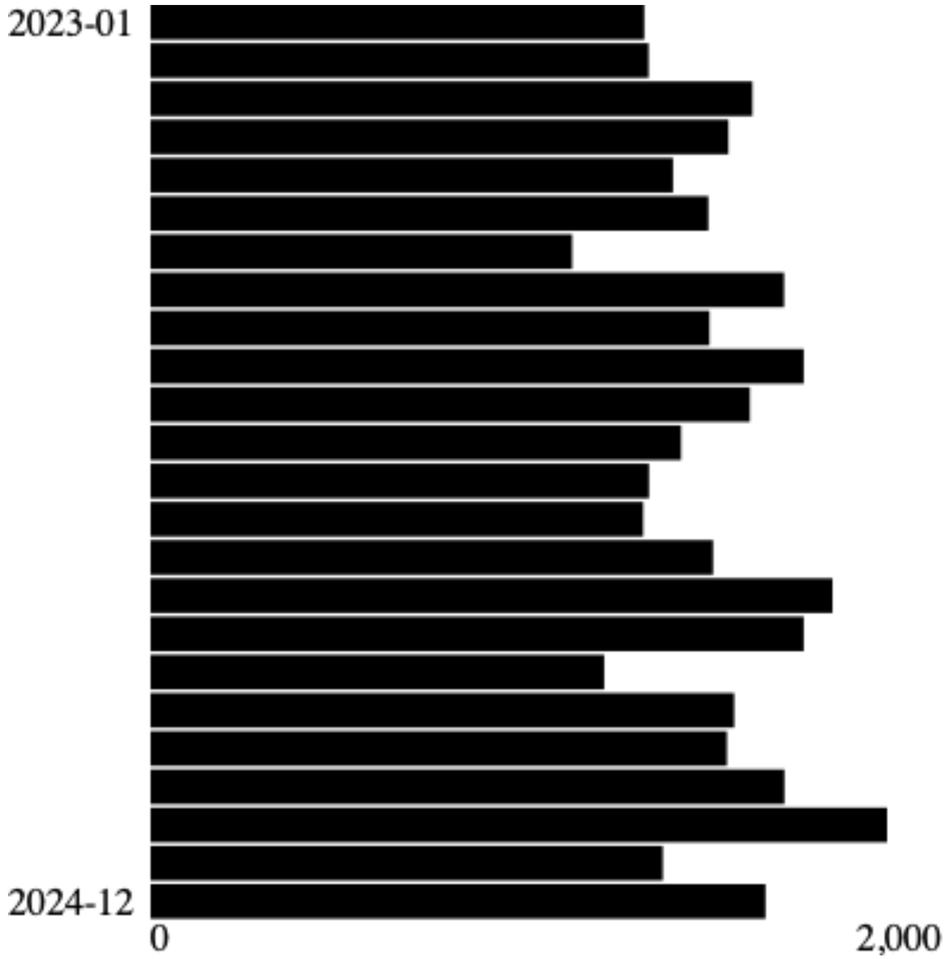




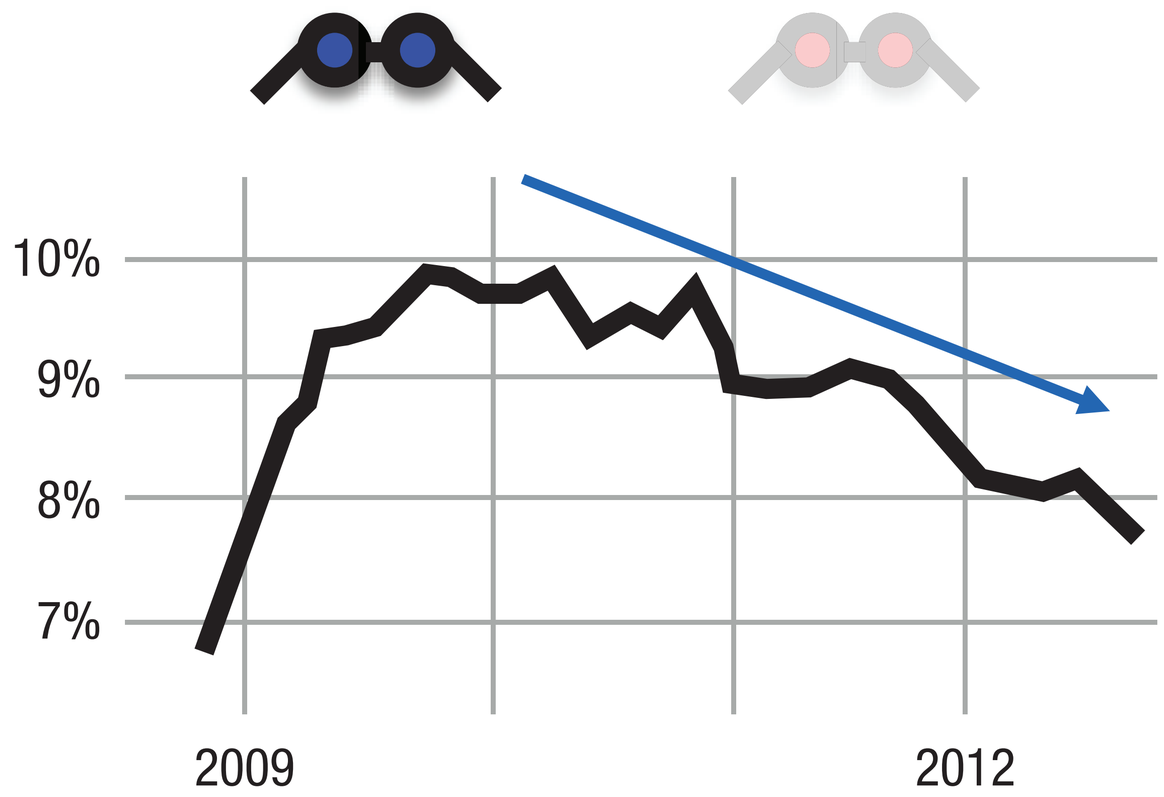
[N. Holmes, 2014] and [S. Franconeri et al., 2021]

Assignment 2

- Due today
- Chicago Food Inspections
- Data Processing in JavaScript
- Create Bar Charts using SVGs and JavaScript
- **Do not sort** the data for Parts 2 & 3
- [CSCI 627] Add Interaction

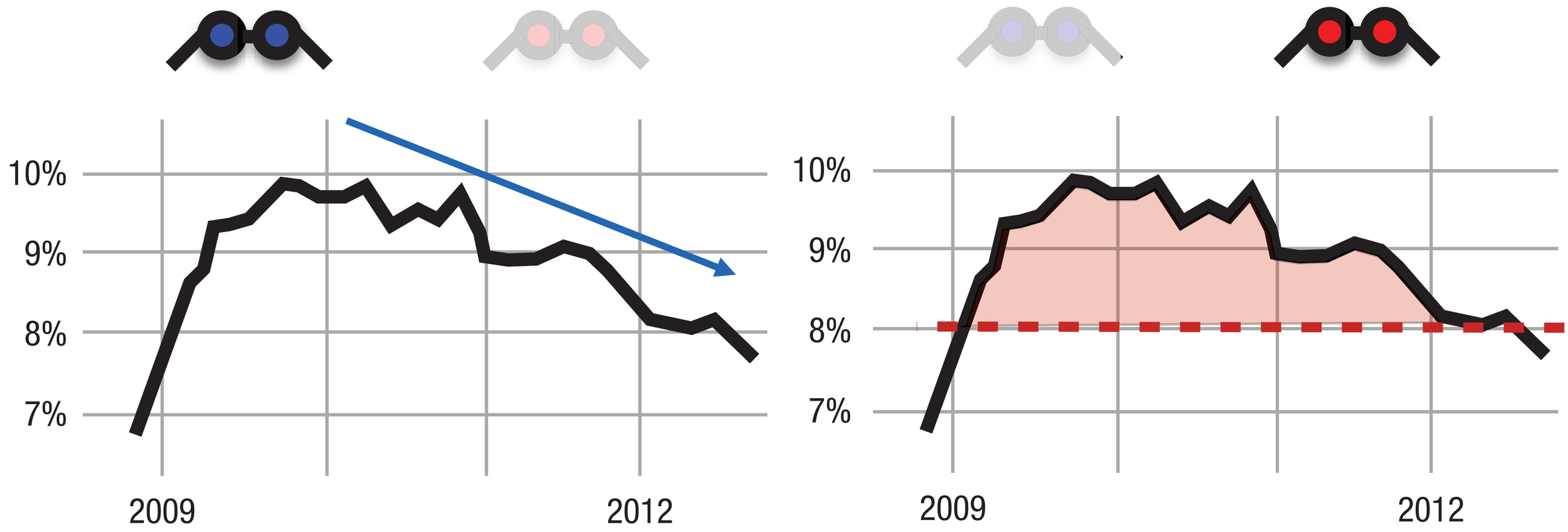


Present to Persuade



[S. Franconeri et al., 2019]

Present to Persuade



[S. Franconeri et al., 2019]

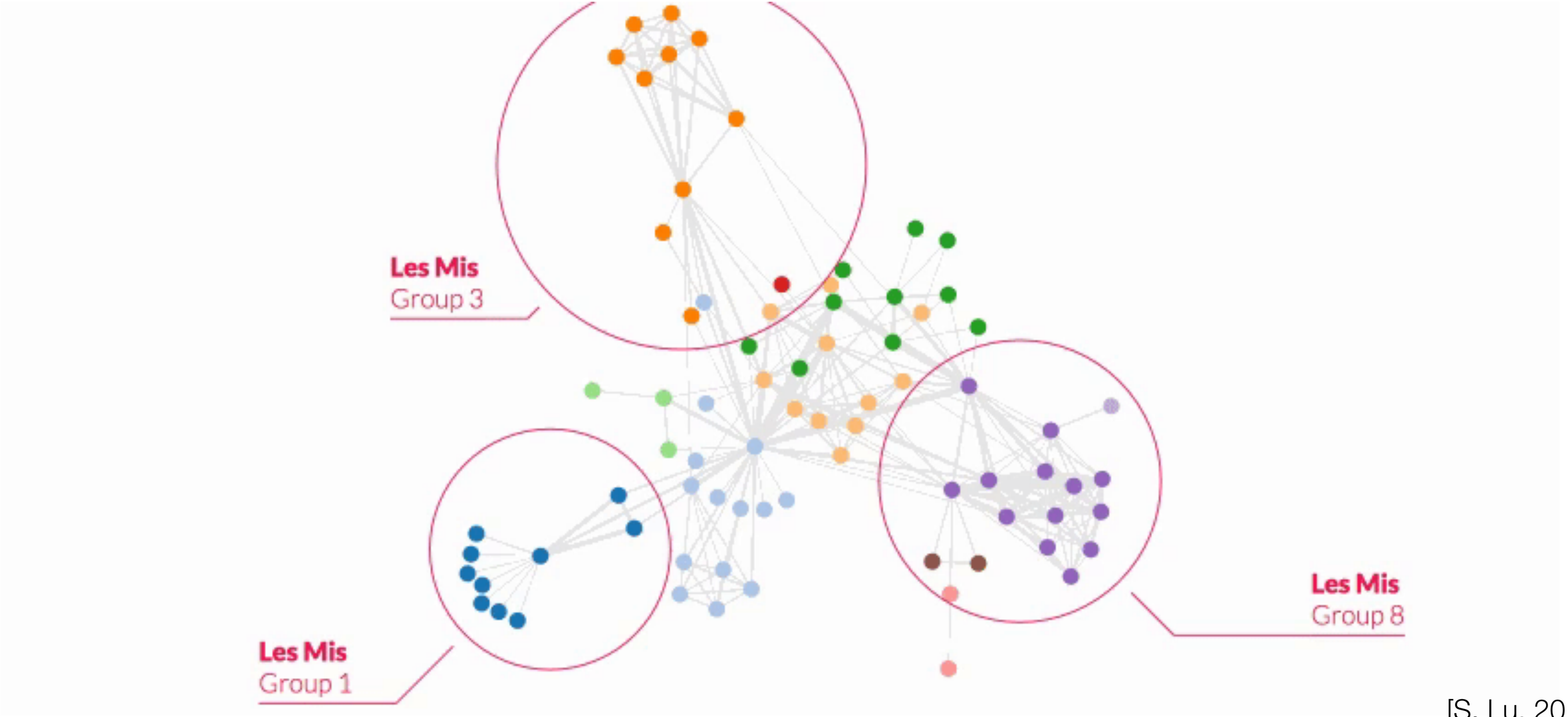
Influencing Messages in Visualizations

- Perception is influenced by existing biases (e.g. unemployment numbers)
- Perception is influenced by visualization's title [Kong et al., 2019]
- Perception can be biased by social influence [Hullman et al., 2011]
- See A. Cairo's books

Visualization for Production

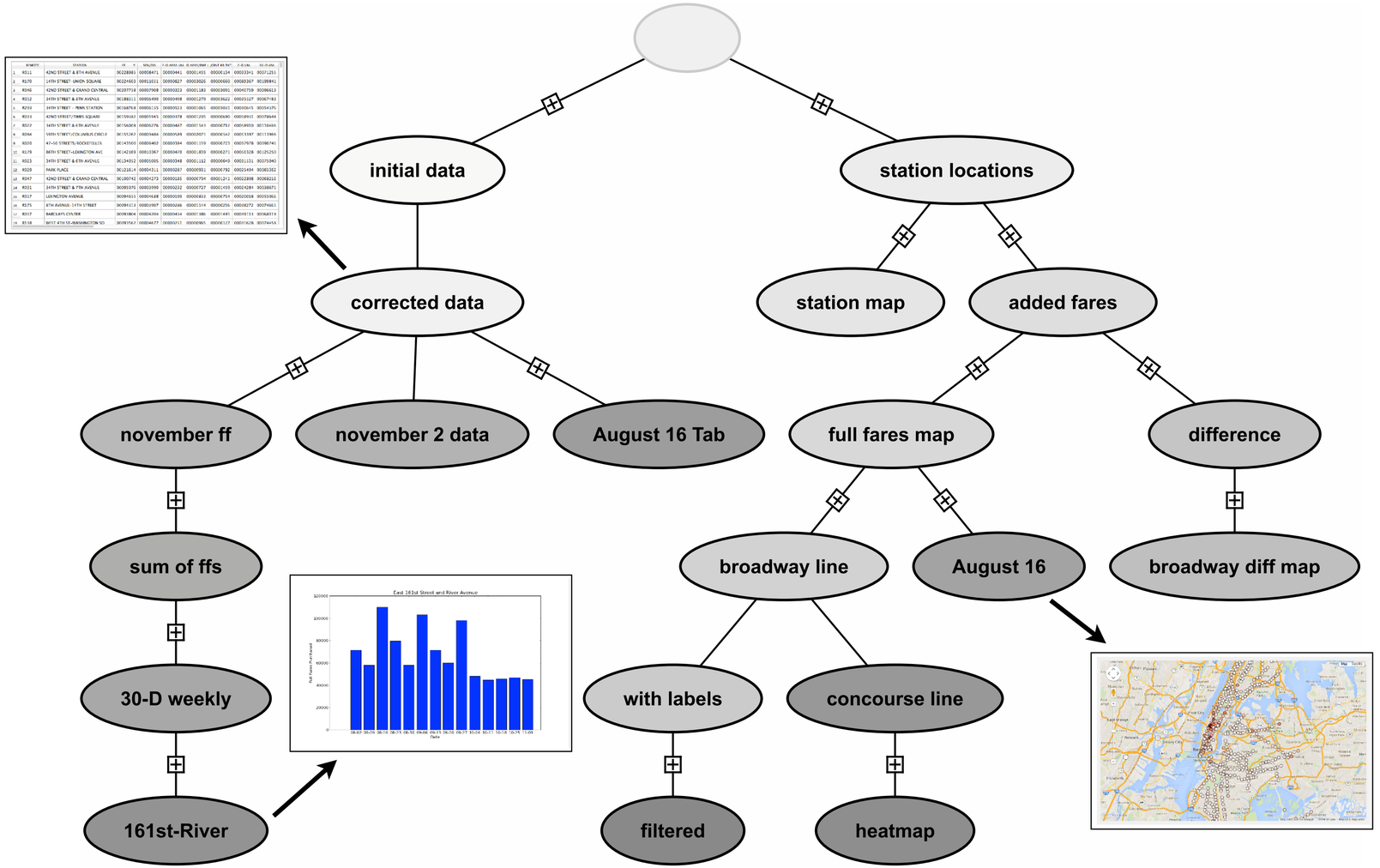
- Generate new material
- Annotate
- Record
- Derive (Transform)

Annotation: Circle Annotations

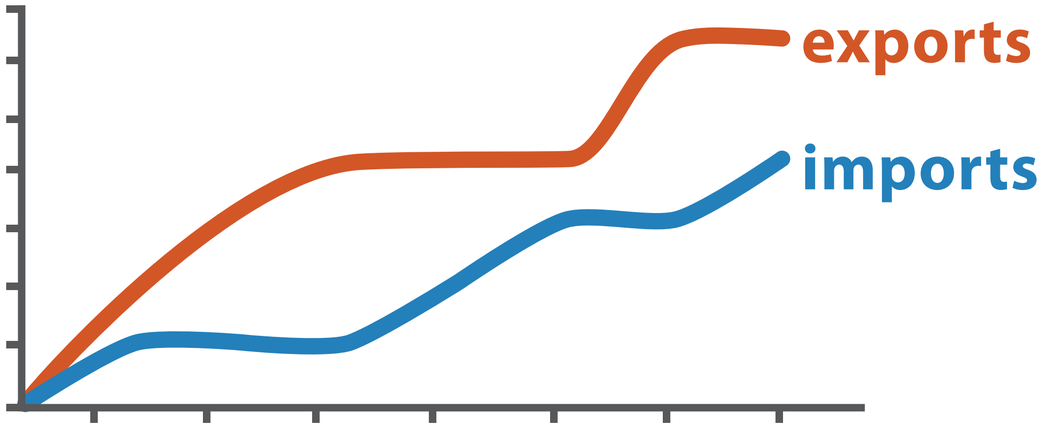


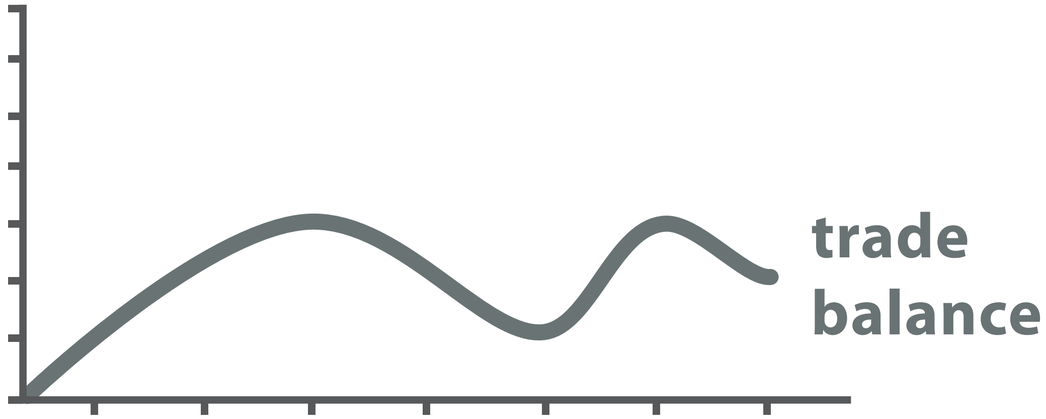
[S. Lu, 2017]

Record: Provenance of MTA Data Exploration



Derived Data





trade balance = *exports* – *imports*

Original Data

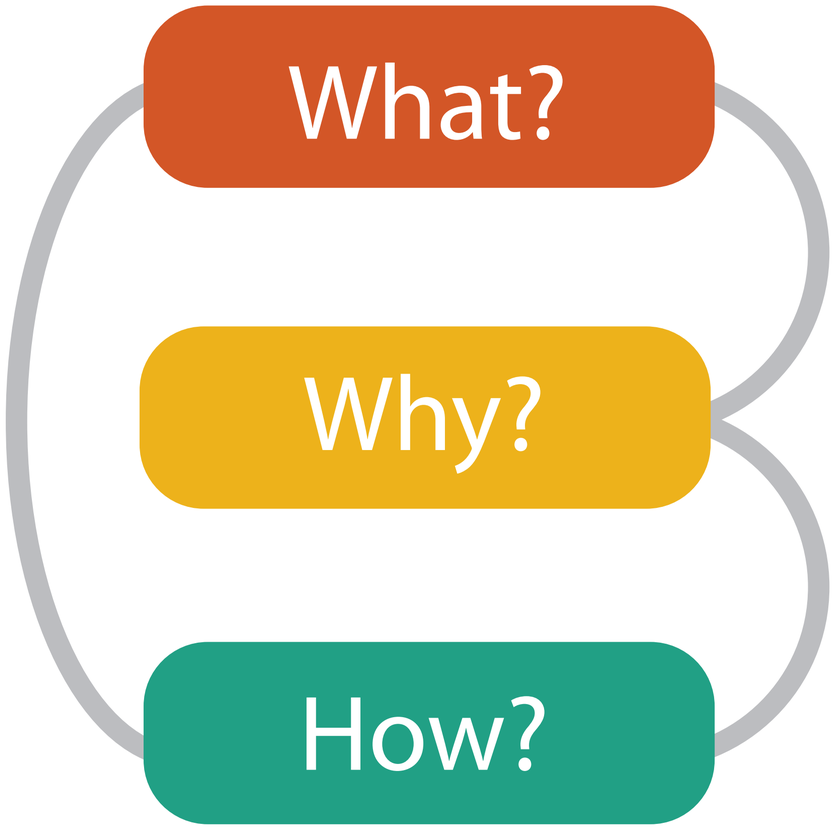
Derived Data

[Munzner (ill. Maguire), 2014]

Visualization for Production

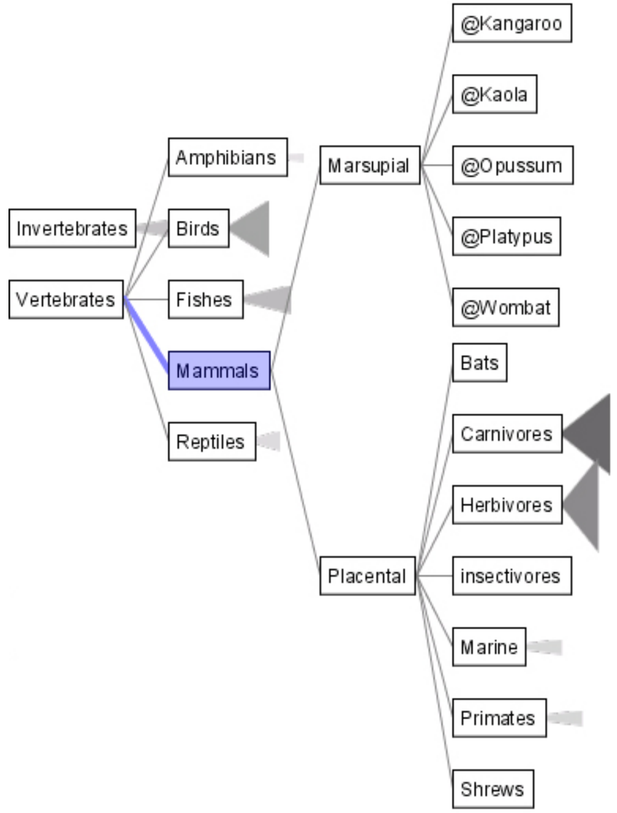
- Generate new material
- Annotate:
  - Add more to a visualization
  - Usually associated with text, but can be graphical
- Record:
  - Persist visualizations for historical record
  - Provenance (graphical histories): how did I get here?
- Derive (Transform):
  - Create new data
  - Create derived attributes (e.g. mathematical operations, aggregation)

Roadmap

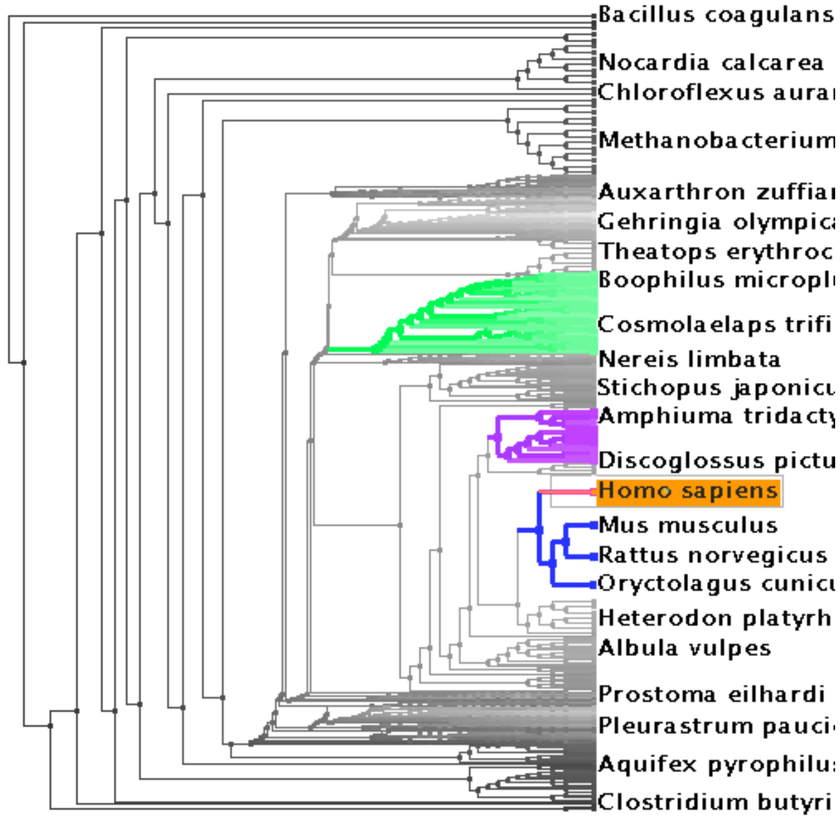


- What? → Data
  - Types
  - Semantics
- Why? → Tasks
  - Actions
  - Targets
- How → Vis Idioms/Techniques
  - Data Representation
  - Visual Encoding
  - Interaction Encoding

Analysis Example: Different “Idioms”



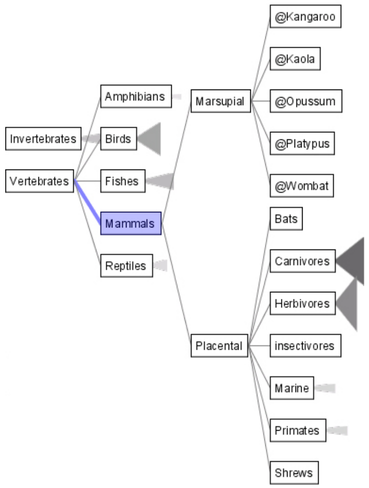
[SpaceTree, Grosjean et al.]



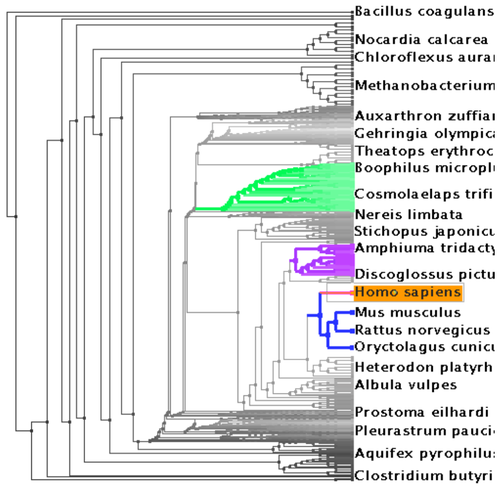
[TreeJuxtaposer, Munzner et al.]

“Idiom” Comparison

SpaceTree

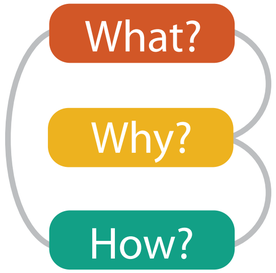


TreeJuxtaposer



[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.]



What?

- Tree

Why?

- Actions
  - Present → Locate → Identify

- Targets
  - Path between two nodes

How?

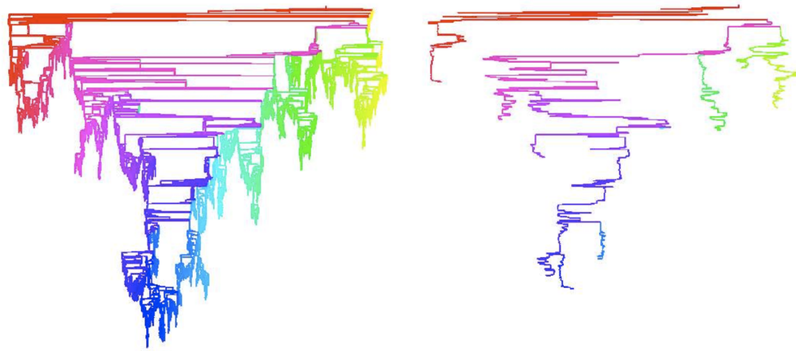
- SpaceTree
  - Encode → Navigate → Select → Filter → Aggregate

- TreeJuxtaposer
  - Encode → Navigate → Select → Arrange

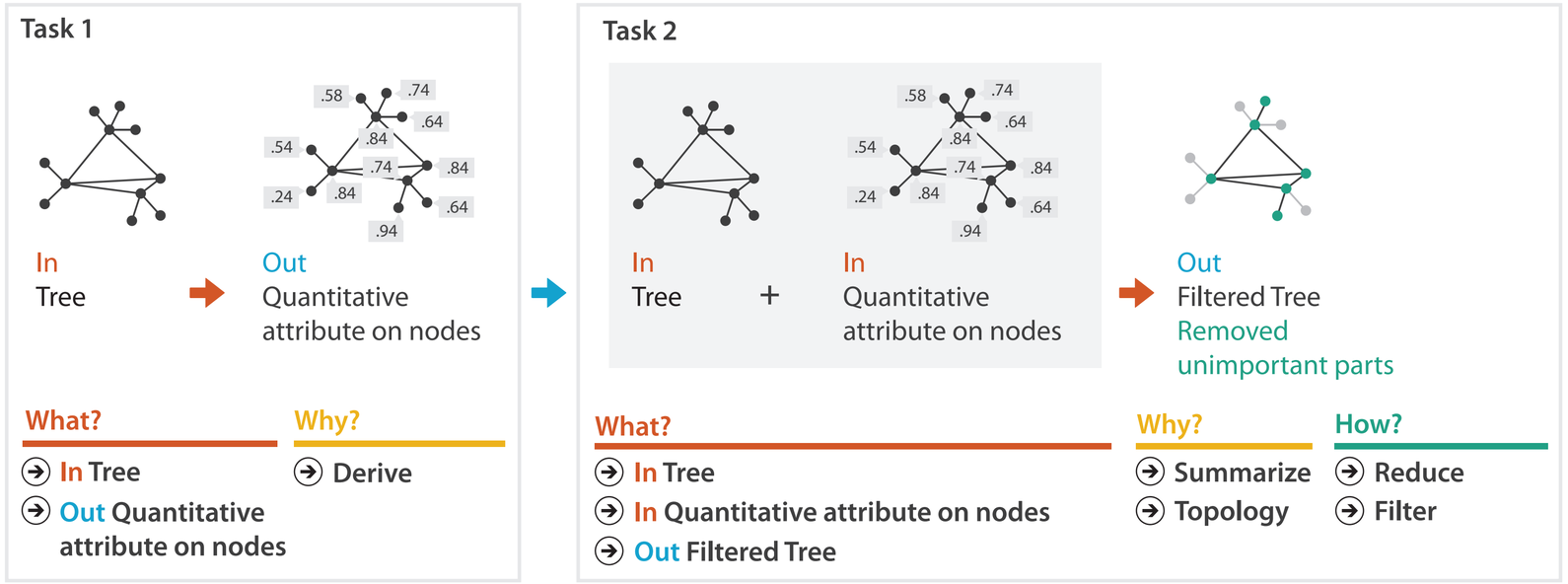
[Munzner (ill. Maguire), 2014]

Analysis Example: Derivation

- Strahler number
  - centrality metric for trees/networks
  - derived quantitative attribute
  - draw top 5K of 500K for good skeleton



[Using Strahler numbers for real time visual exploration of huge graphs. Auber. Proc. Intl. Conf. Computer Vision and Graphics, pp. 56–69, 2002.]



[Munzner (ill. Maguire), 2014]

d3.js

Data-Driven Documents

Data-Driven Documents (D3)

- Open-Source JavaScript Library
- http://d3js.org/
- Original Authors: Mike Bostock, Vadim Ogievestky, and Jeff Heer
- Focus on Web standards, customization, and usability
- Grew from work on Protovis: more standard, more interactive
- By nature, a **low-level** library; you have control over all elements and styles
- A top project on GitHub (over 106,000 stars as of Sept. 2023)
- Lots of impressive examples
  - Bostock was a New York Times Graphics Editor
  - https://observablehq.com/@mbostock

D3 Key Features

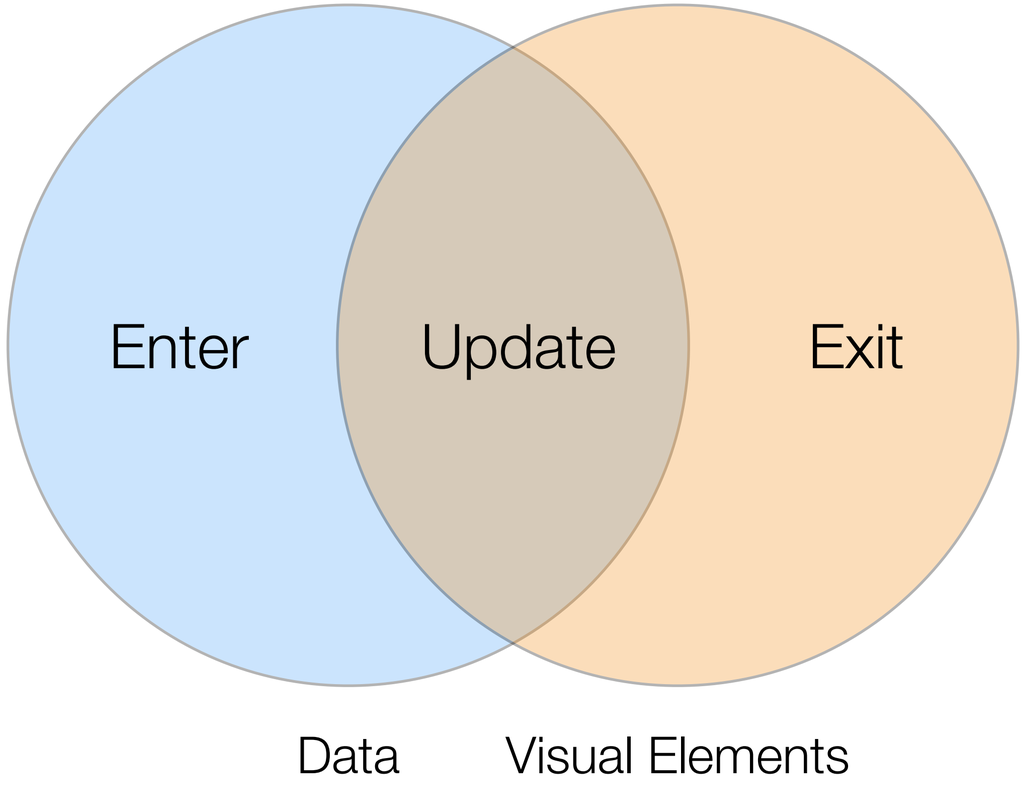
- Supports data as a core piece of Web elements
  - Loading data
  - Dealing with changing data (joins, enter/update/exit)
  - **Correspondence** between data and DOM elements
- Selections (similar to CSS) that allow greater manipulation
- Method Chaining
- Integrated layout algorithms, axes calculations, etc.
- Focus on interaction support
  - Straightforward support for transitions
  - Event handling support for user-initiated changes

D3 Introduction

- Ogievetsky has put together a nice set of interactive examples that show off the major features of D3
- https://observablehq.com/@dakoop/d3-intro
- Standalone version: http://dakoop.github.io/IntroD3/
  - (Updated from original)
- Other references:
  - https://observablehq.com/@d3/learn-d3
  - https://observablehq.com/@d3/gallery
  - Murrary's book on Interactive Data Visualization for the Web
  - The D3 website: d3js.org

D3 Data Joins

- Two groups: data and visual elements
- Three parts of the join between them: enter, update, and exit
- enter: **`s.enter()`**, update: **`s`**, exit: **`s.exit()`**



Merge vs. Join

- Merge creates a new selection that includes the items from **both** selections
  - If you want to update all elements (including those just added via enter), use merge!
  - Useful when enter+update have similar transitions
- Join allows you to modify different parts of the selection in a single statement
  - Also will create the final selection
  - Does enter+append and exit+remove automatically
  - Pass functions to modify the enter, update, and exit parts of the selection
  - Examples: https://observablehq.com/@d3/selection-join

Transitions

- Nested transitions (those that "hang off" of a parent transition) follow immediately after the parent transition