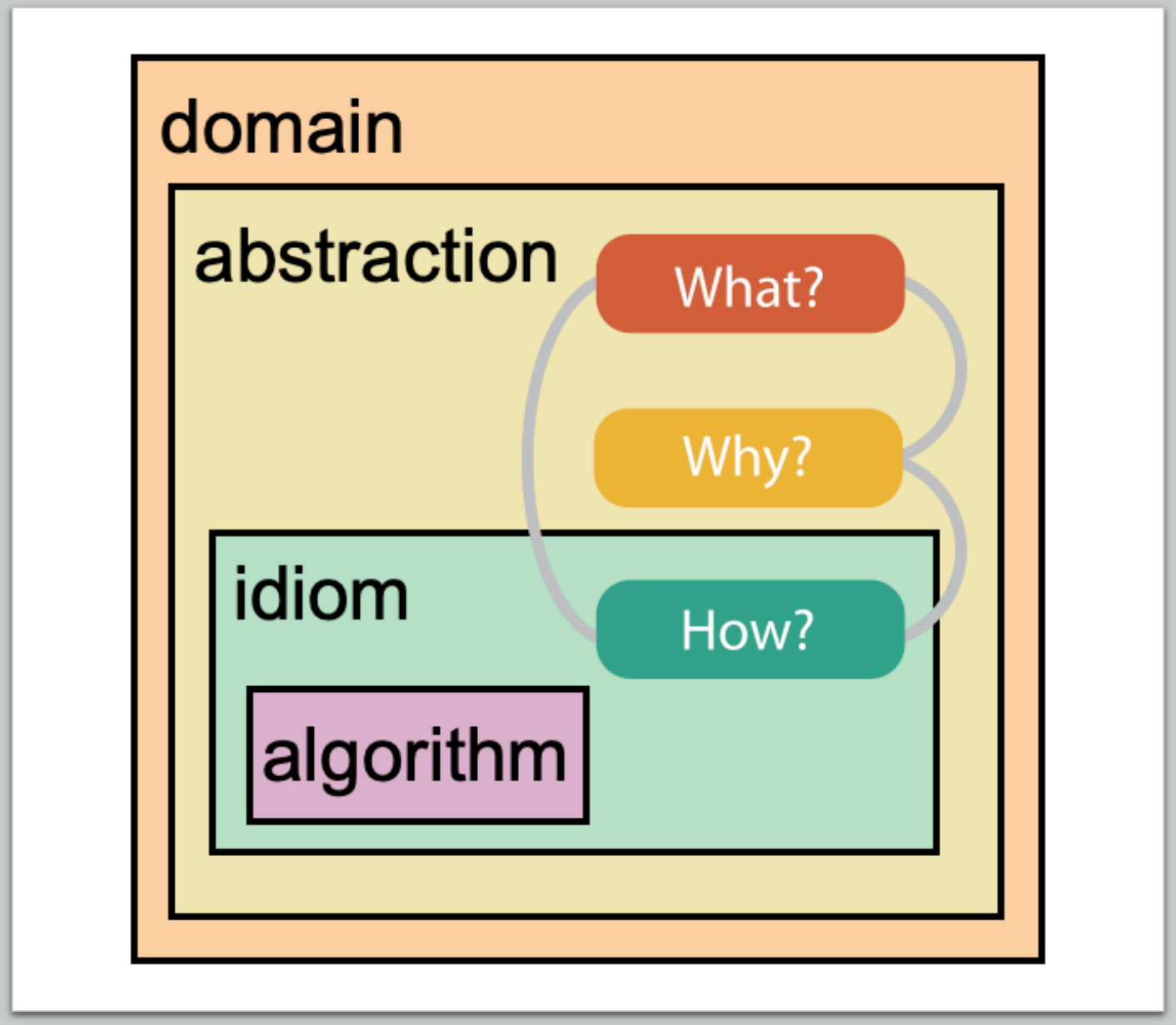

I DON'T ALWAYS STAY ON TASK
BUT WHEN I
quickmeme.com

Task Abstraction

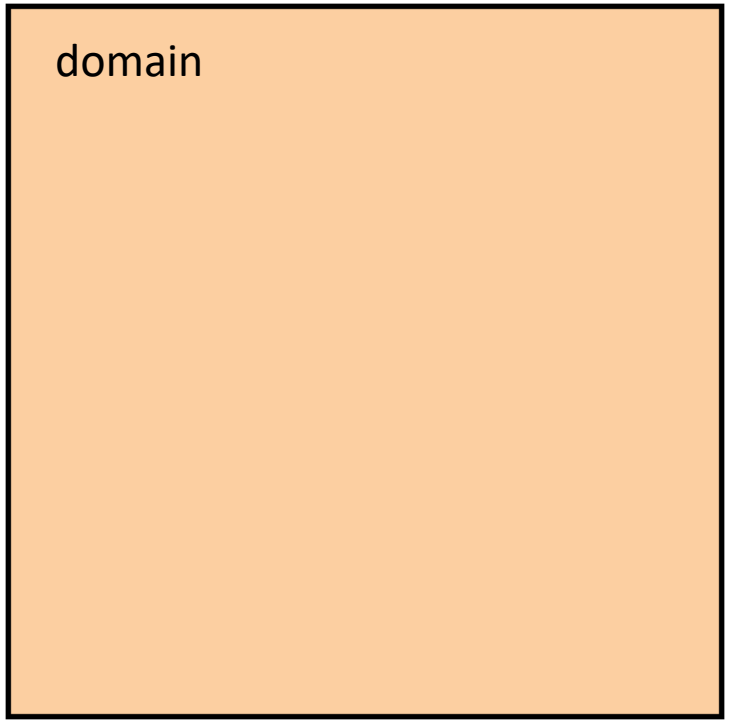
Munzner's Four levels of visualization design

- *domain situation*
  - who are the target users?
- *abstraction*
  - translate from specifics of domain to vocabulary of visualization
    - **what** is shown? **data** abstraction
    - **why** is the user looking at it? **task** abstraction
      - often must transform data, guided by task
- *idiom*
  - **how** is it shown?
    - **visual encoding** idiom: how to draw
    - **interaction** idiom: how to manipulate
- *algorithm*
  - efficient computation

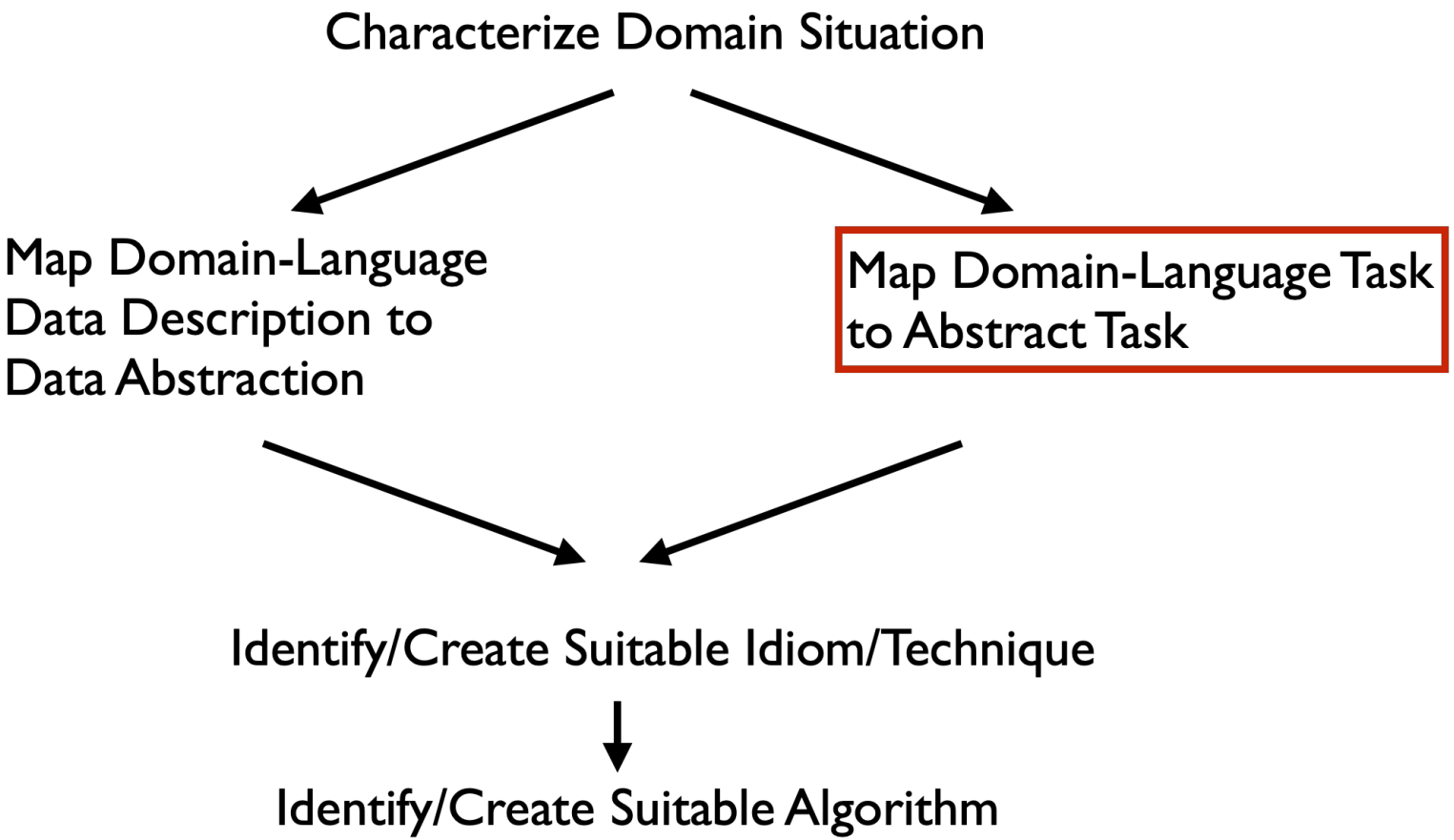


Domain Characterization

- details of an application domain
- group of users, target domain, their questions, & their data
  - varies wildly by domain
  - must be specific enough to get traction
- domain questions/problems
  - break down into simpler abstract tasks



Design Process

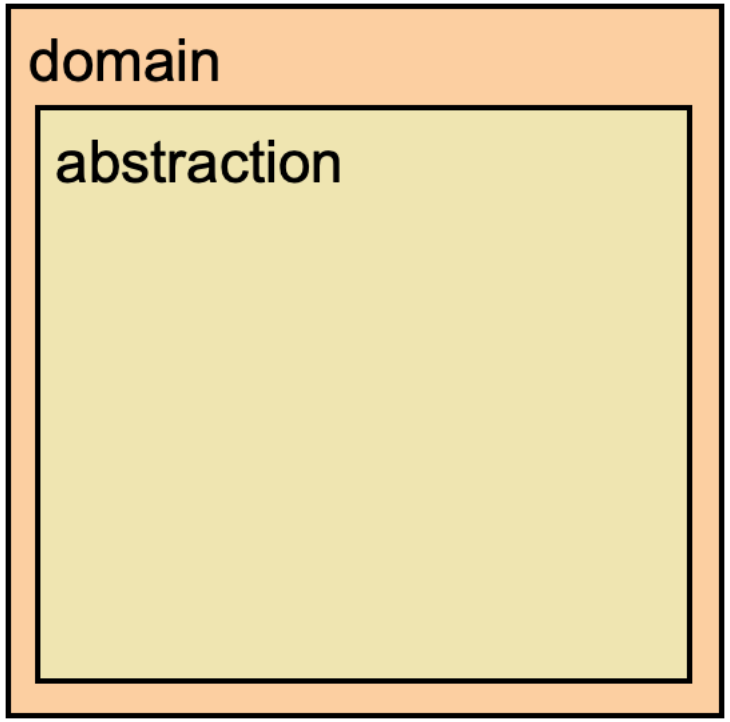


Example: Find Good Movies

- identify good movies in genres I like
- domain:
  - general population, movie enthusiasts

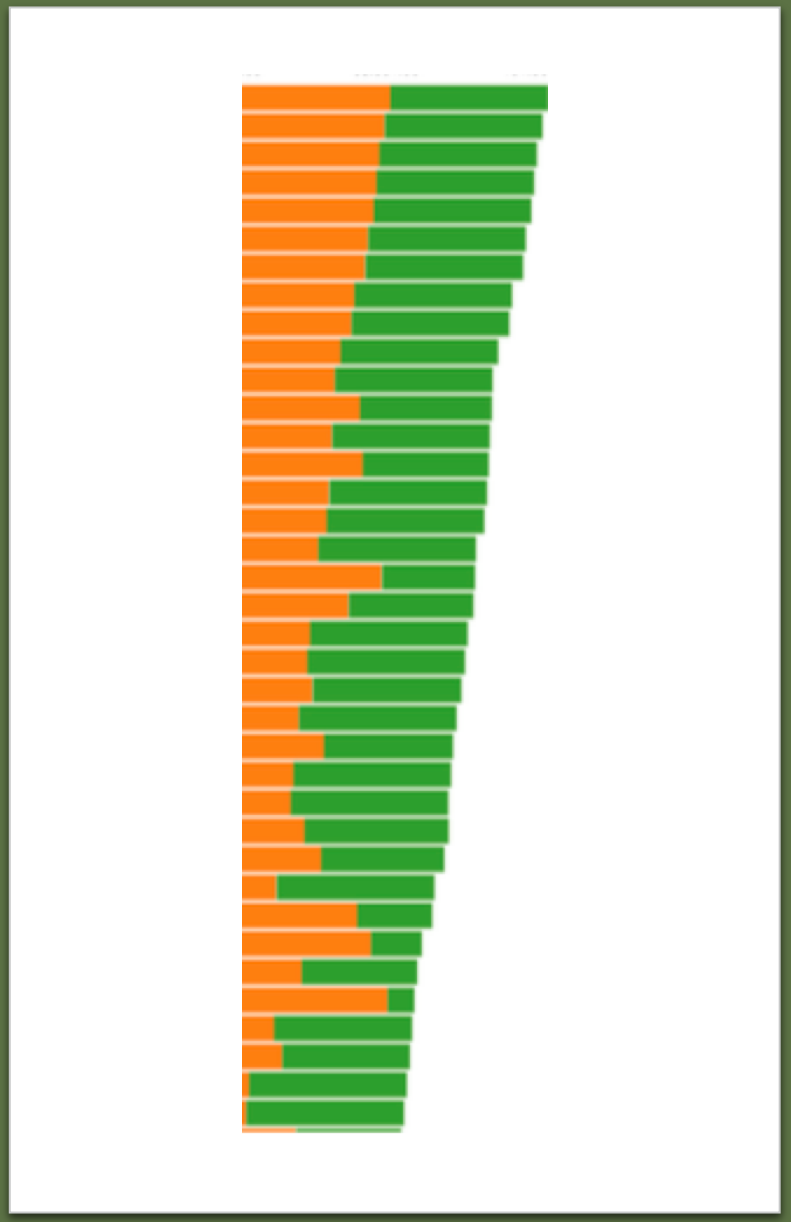
Abstraction: Data and Task

- map *what* and *why* into generalized terms
  - identify tasks that users wish to perform, or already do
  - find data types that will support those tasks
    - possibly transform /derive if need be



Example: Find Good Movies

- identify good movies in genres I like
- domain:
  - general population, movie enthusiasts
- task: what is a good movie for me?
  - highly rated by critics?
  - highly rated by audiences?
  - successful at the box office?
  - similar to movies I liked?
  - matches specific genres?
- data: (is it available?)
  - yes! data sources IMDB, Rotten Tomatoes...



Possible idiom : Stacked Bar Chart for ratings

Example: Find Good Movies

- one possible choice for data and tasks, in domain language
  - data: combine audience ratings and critic ratings
  - task: find high-scoring movies for specific genre
- abstractions?
  - attribute: audience & critic ratings
    - ordinal
      - levels: 3 or 5 or 10...
  - attribute: genre
    - categorical
      - levels: < 20
  - items: movies
    - items: millions
  - task: find high values?

Example: Horrified

- same task: high-score movies
- slightly different data
  - 14K rated horror movies from IMDB
- very different visual encoding idiom
  - circle per item (movie)
  - circle area = popularity
  - stroke width/opacity = avg rating
  - year made = vertical position
- interaction idiom
  - lines connect movies w/ same director, on mouseover

https://www.alhadaqa.com/wp-content/uploads/2020/04/horrified.html

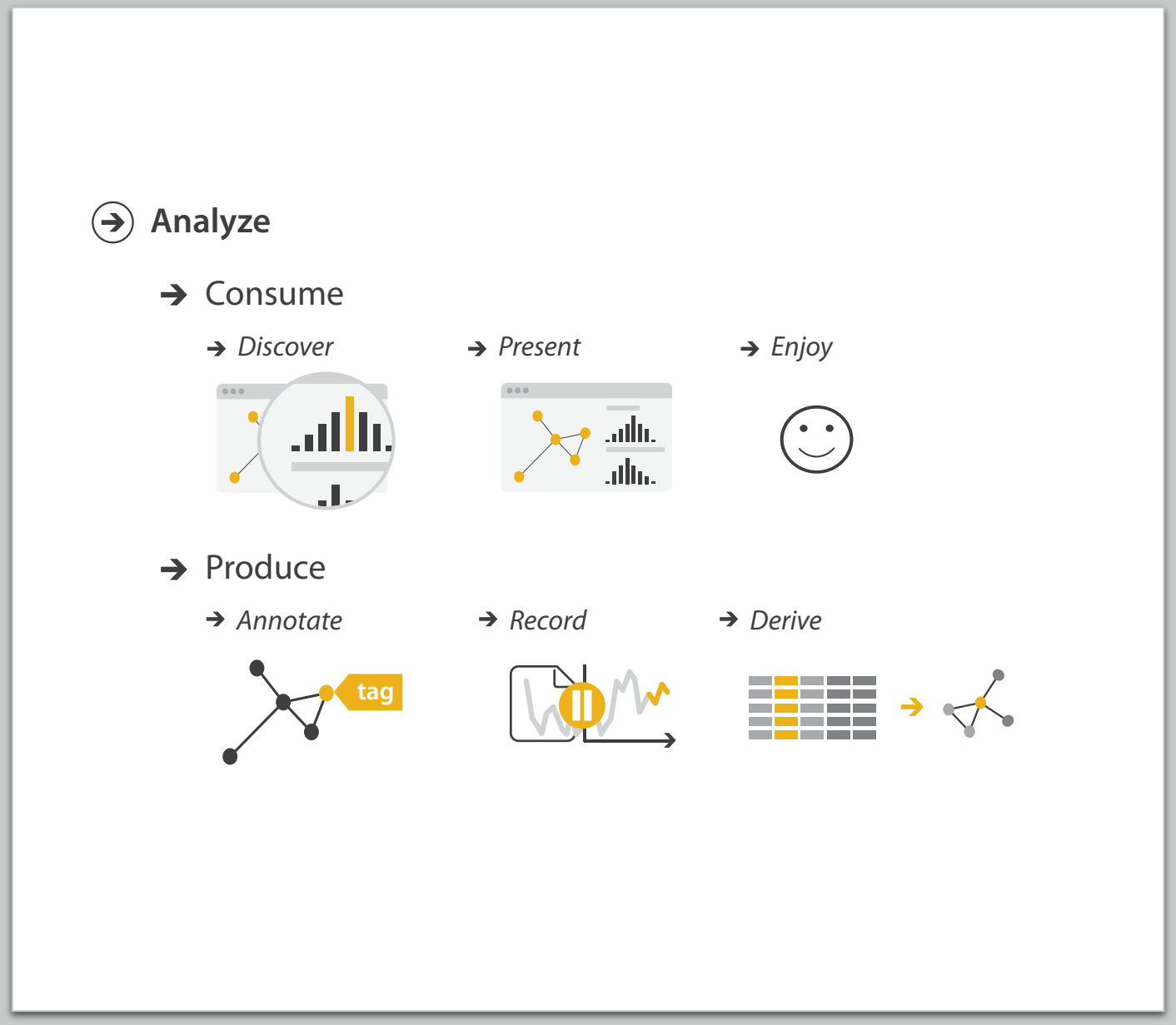
Task Abstraction: Actions and Targets

- very high-level pattern
- actions
  - analyze
    - high-level choices
  - search
    - find a known/unknown item
  - query
    - find out about characteristics of item

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

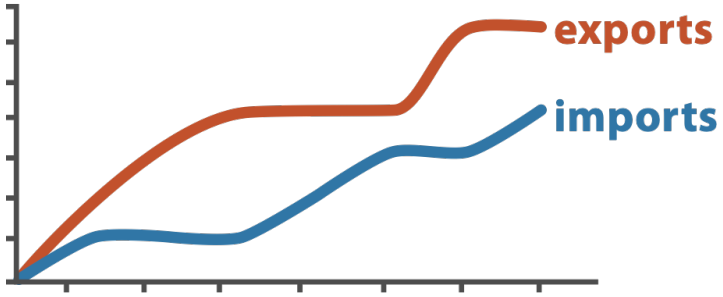
Actions: Analyze

- consume
  - discover vs present
    - classic split
    - aka explore vs explain
  - enjoy
    - newcomer
    - aka casual, social
- produce
  - annotate, record
  - derive
    - crucial design choice

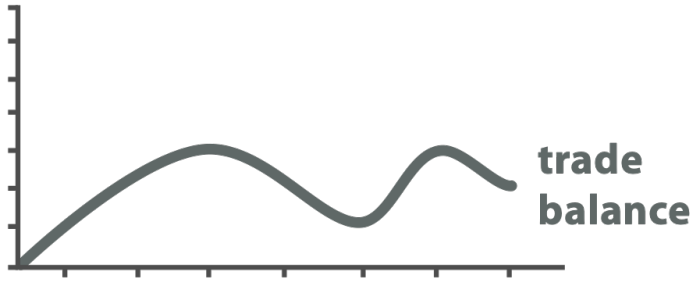


Derive

- don’t just draw what you’re given!
  - decide what the right thing to show is
  - create it with a series of transformations from the original dataset
  - draw that
- *one of the four major strategies for handling complexity*



Original Data

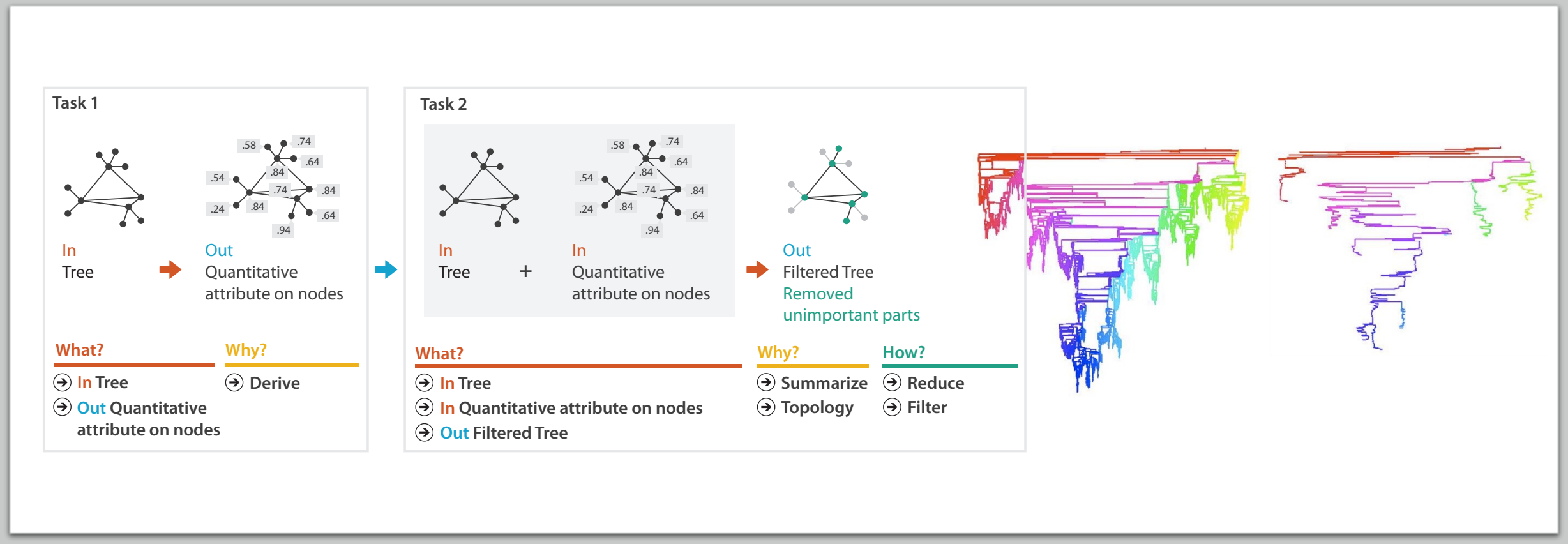


trade balance = exports – imports

Derived Data

Analysis example: Derive one attribute

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



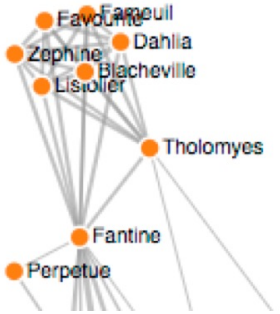
Actions: Search

- what does user know?
  - target, location
- lookup
  - ex: word in dictionary
    - alphabetical order
- locate
  - ex: keys in your house
  - ex: node in network
- browse
  - ex: books in bookstore
- explore
  - ex: cool neighborhood in new city

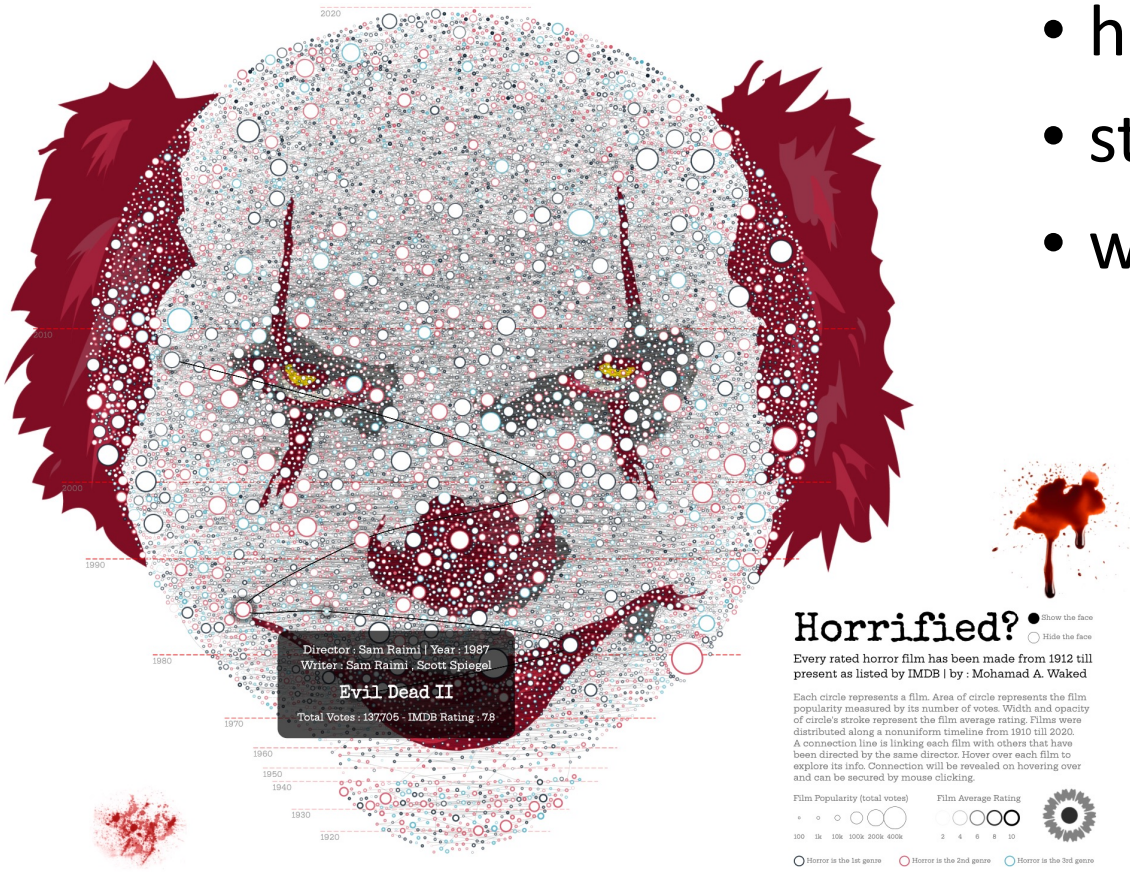
Search

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

https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8



Example: Horrified vs stacked bars



- horrified: browse/explore
- stacked bars: locate/lookup
- which is better?
  - depends on goals / task
    - enjoy, social context, lots of time
    - find 2nd-best rated movie of all time
      - Jeopardy call, < 10 seconds to respond!

Actions: Search, query

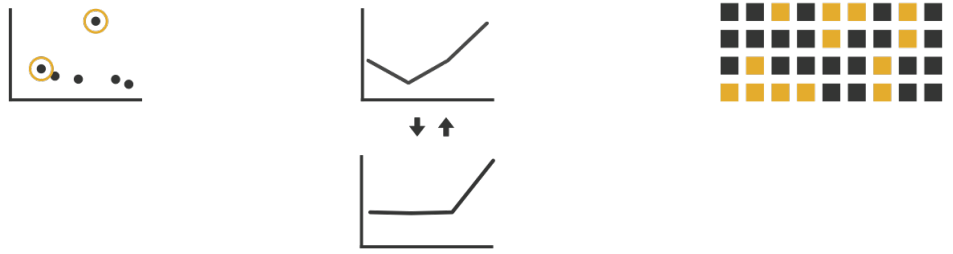
- what does user know?
  - target, location
- how much of the data matters?
  - one, some, all
- independent choices for each of these three levels
  - analyze, search, query
  - mix and match

Search

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

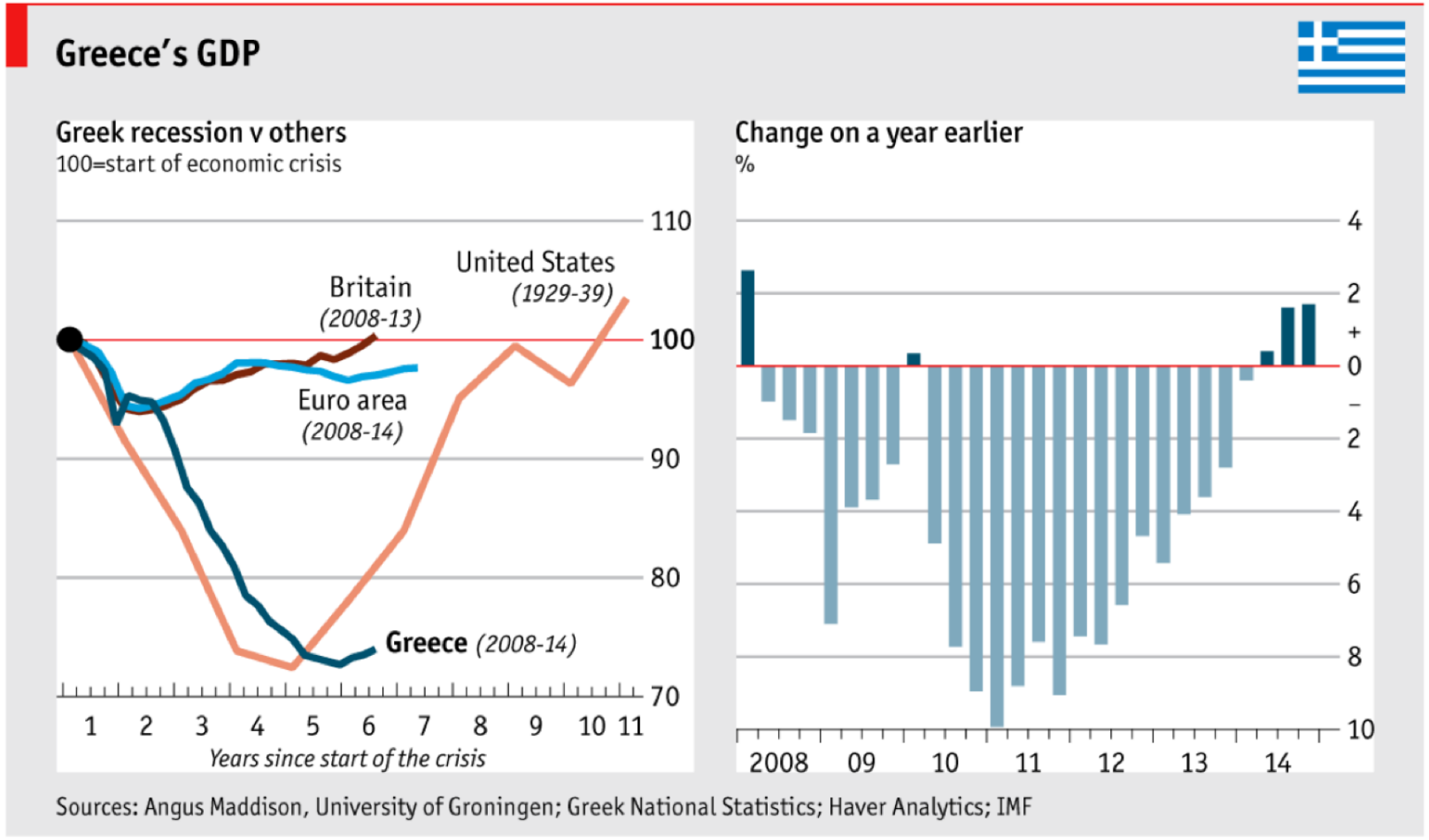
Query

→ Identify → Compare → Summarize

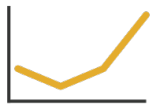
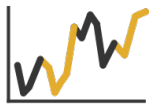
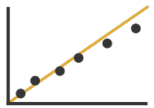
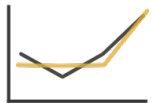
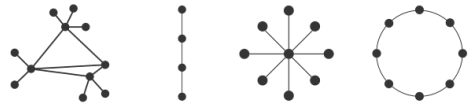
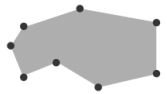


Example: Economics

- task: compare and derive
- data: derive change



Task abstraction: Targets

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

Abstraction

these {action, target} pairs are good starting point for vocabulary

but sometimes you'll need more precision!

rule of thumb

systematically remove all domain jargon

interplay: task and data abstraction

need to use data abstraction within task abstraction

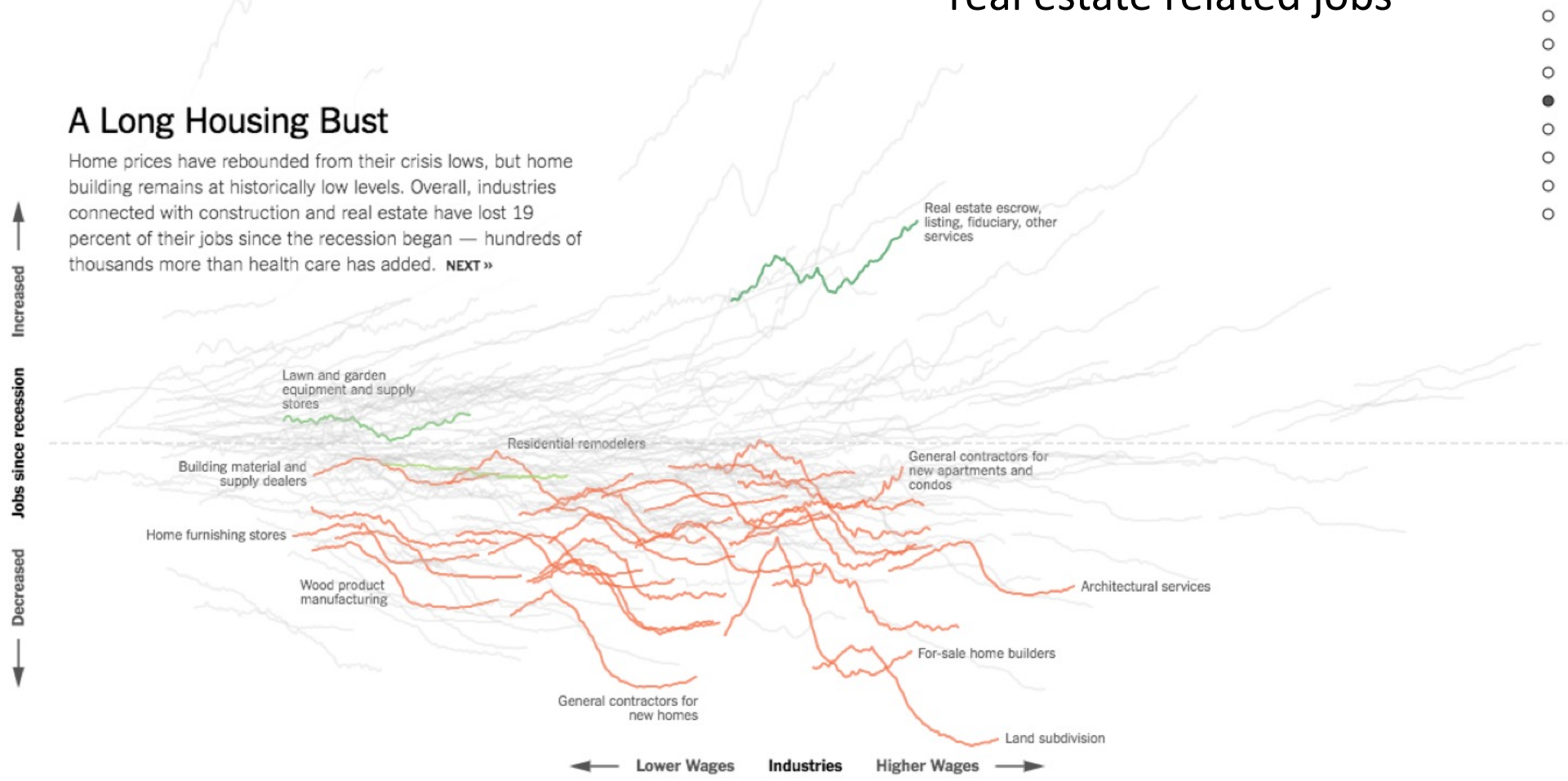
- to specify your targets!
- but task abstraction can lead you to transform the data

iterate back and forth

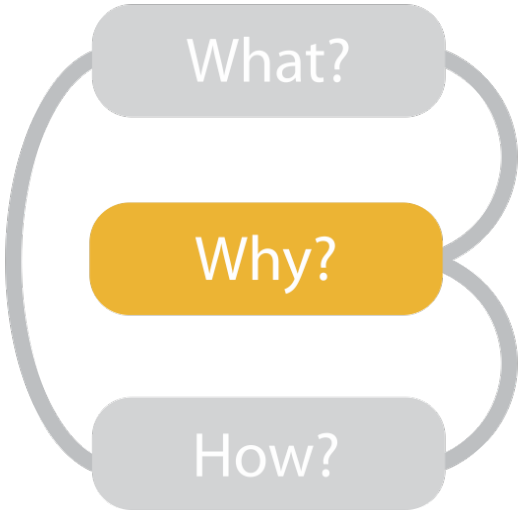
- first pass data, first pass task, second pass data, ...

Examples: Job market

- trends
  - how did job market develop since recession overall?
- outliers
  - real estate related jobs



https://www.nytimes.com/interactive/2014/06/05/upshot/how-the-recession-reshaped-the-economy-in-255-charts.html



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

Why?

Actions

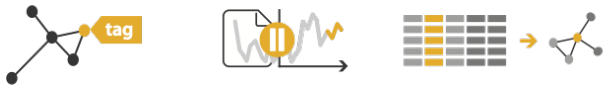
Analyze

→ Consume

→ *Discover* → *Present* → *Enjoy*

→ Produce

→ *Annotate* → *Record* → *Derive*

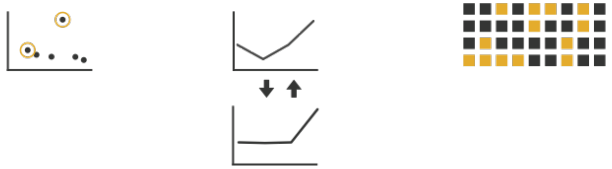


Search

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

Query

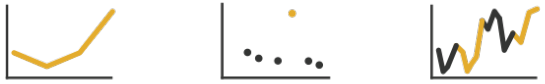
→ Identify → Compare → Summarize



Targets

All Data

→ Trends → Outliers → Features



Attributes

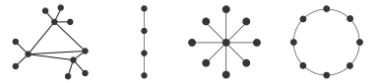
→ One → Many

→ *Distribution* → *Dependency* → *Correlation* → *Similarity*

→ *Extremes*

Network Data

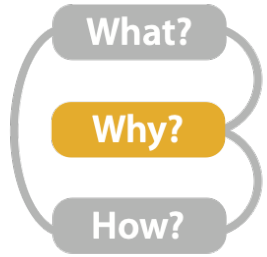
→ Topology



→ *Paths*

Spatial Data

→ Shape



Carbon Emissions

- https://flowingdata.com/2021/01/21/car-cost-vs-emissions/
- https://www.carboncounter.com/#!/explore