Visualization for Data Science
Task Abstraction

Office hours start this week

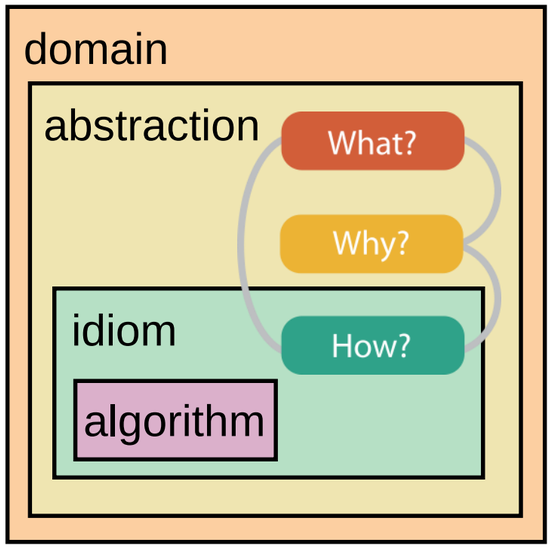
- TA Office Hours
  - Tuesdays in RM 238 at 5pm
  - Wednesdays Online on Zoom at 5pm
  - Saturday Online on Zoom at 2pm
  - Sunday Online on EdStem at 4pm
- Instructor Office Hours
  - **In Person RM 202 from 1 – 2:45pm on Wednesdays**

Individual Mindmap exercise - 10 minutes

Make a list of all the keywords you have been exposed to in the class so far. Create a mind map that connects them together.

Analysis framework: Four levels, three questions

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

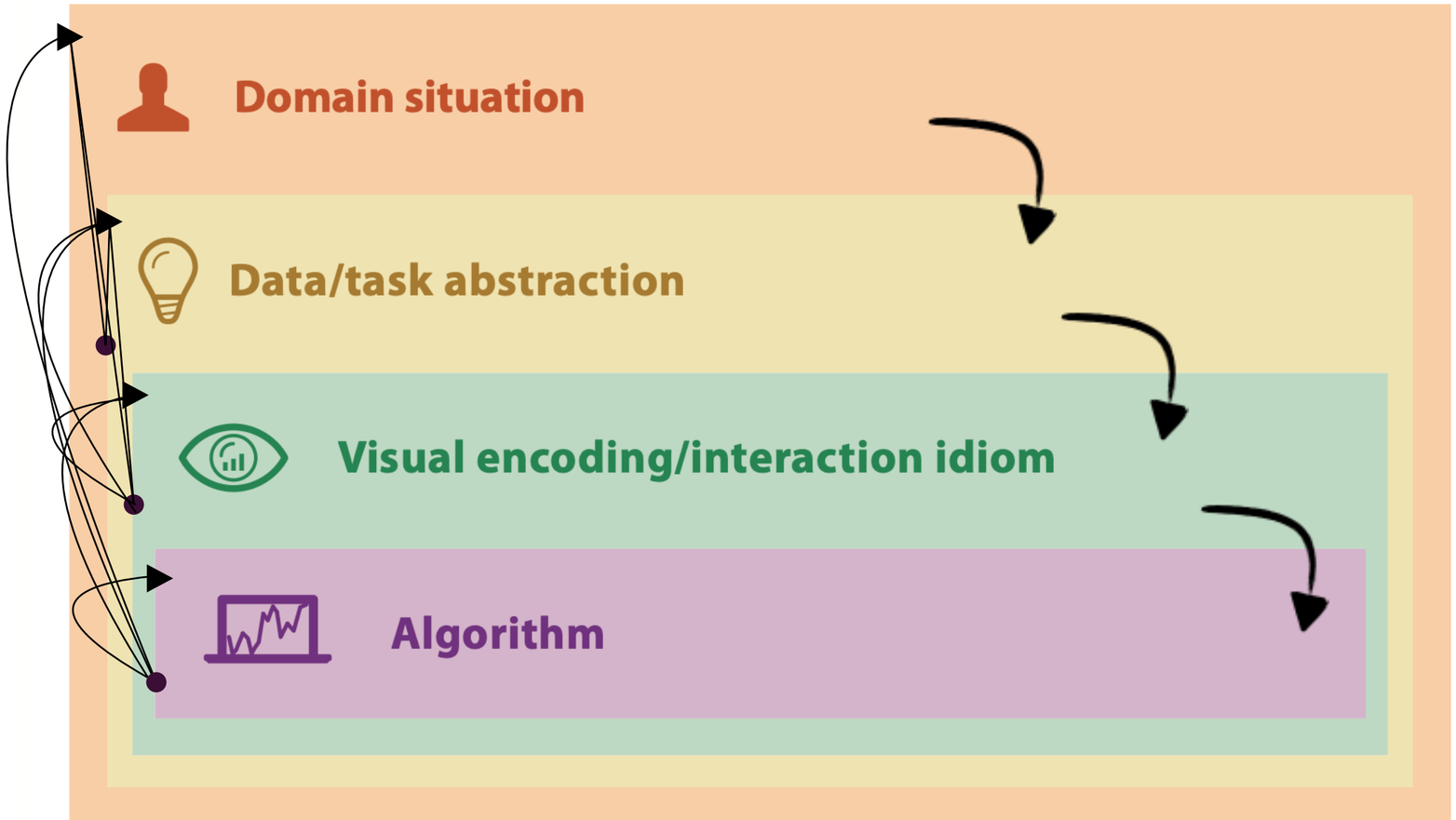


[A Multi-Level Typology of Abstract Visualization Tasks. Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

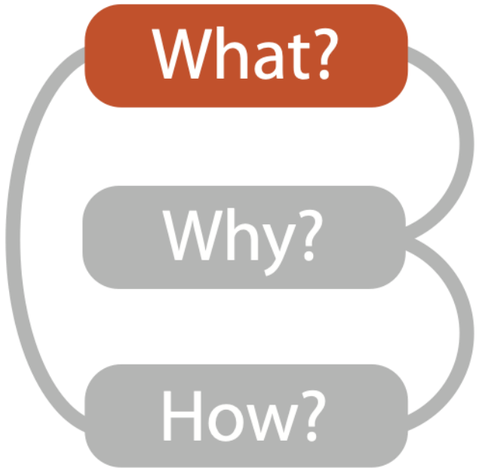
[A Nested Model of Visualization Design and Validation. Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]

Nested model

- downstream: cascading effects
- upstream: iterative refinement



[A Nested Model of Visualization Design and Validation. Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]



What?

Datasets

Data Types

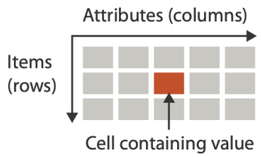
→ Items → Attributes → Links → Positions → Grids

Data and Dataset Types

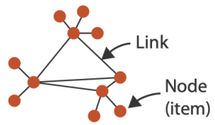
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Types

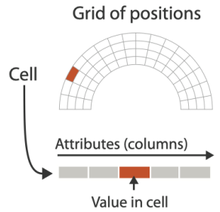
→ Tables



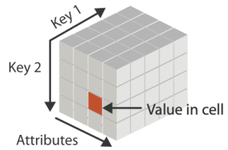
→ Networks



→ Fields (Continuous)



→ *Multidimensional Table*



→ *Trees*

→ Geometry (Spatial)

Dataset Availability

→ Static

→ Dynamic

Attributes

Attribute Types

→ Categorical

→ Ordered

→ *Ordinal*

→ *Quantitative*

Ordering Direction

→ Sequential

→ Diverging

→ Cyclic

Data abstraction: Three operations

- translate from domain-specific language to generic visualization language

- identify dataset type(s), attribute types
- identify cardinality
  - how many items in the dataset?
  - what is cardinality of each attribute?
    - number of levels for categorical data
    - range for quantitative data

- consider whether to transform data
  - guided by understanding of task

Data Abstraction Revisited

City	Condition	Temperature
Calgary	Mainly Clear	-4°C
Charlottetown	Light Snowshower	-6°C
Edmonton		-7°C
Fredericton	Clear	-9°C
Halifax	Light Snow	-6°C
Iqaluit	Clear	-28°C
Montréal	Mainly Clear	-9°C
Ottawa (Kanata - Orléans)	Mainly Clear	-10°C
Prince George	Fog	-6°C
Québec	Mainly Clear	-15°C
Regina	Mist	-13°C
Saskatoon	Mist	-11°C
St. John's	Mostly Cloudy	-5°C
Thunder Bay	Light Snow	0°C
Toronto	Cloudy	-1°C
Vancouver	Mainly Clear	7°C
Victoria	Mainly Clear	5°C
Whitehorse	Mostly Cloudy	-12°C
Winnipeg	Mist	-6°C
Yellowknife	Light Snow	-13°C

Data Model

- Mathematical Abstraction
- Variable data types in programming languages

Conceptual Model

- Mental construction
- Based on an understanding of tasks

Data Model

- floats
- 32.52, 54.06, -14.35, ...

Conceptual Model

- temperature

Data Abstraction

- task: forecasting the weather
  continuous to 2 significant figures: quantitative
- task: deciding if bath water is ready
  hot, warm, cold: ordinal
- task: decide if I should leave the house today
  above freezing, below freezing: categorical

Learning Outcomes

- Describe why task abstraction is important
- Explain the nested design model
- List the various actions and targets
- Analyze a dataset using both the task and data abstraction frameworks presented so far

From domain to abstraction

Epidemiologist studying the spread of a new strain of influence

Task: Contrast the prognosis of patients who were intubated in the ICU more than one more after exposure to patients hospitalized within the first week.

Anthropologist studying ancient African kingdoms

Task: Explore the marriage customs that existed in the Benin and Shona kingdoms using folklore and written traditions reasons why similarities between the marriage ceremonies in the Benin and Shona kingdoms. within the Benin

Pastor studying the applications of the parables of Jesus

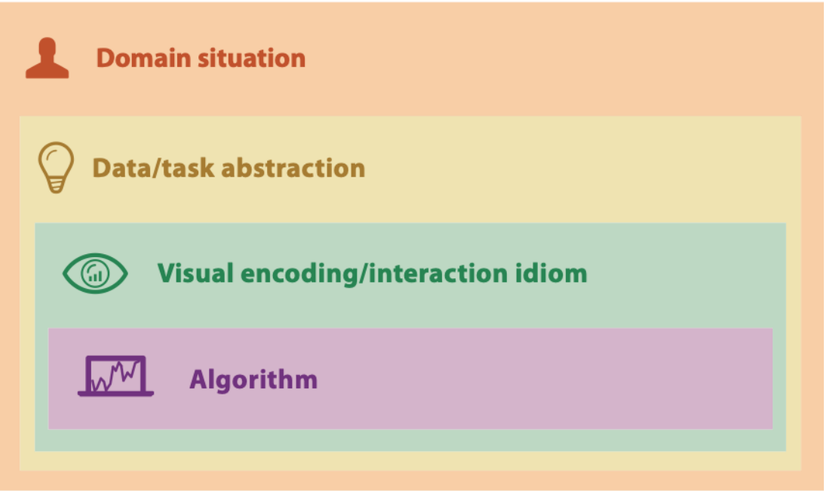
Task: Perform an exegesis that elucidates the different meaning of the parable of the Lost Son in the gospel of Luke

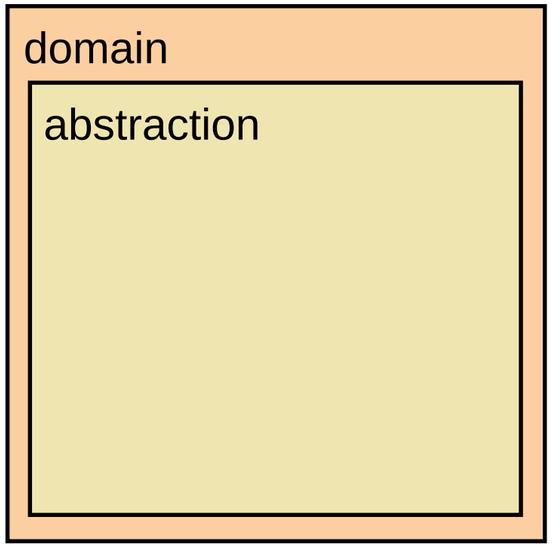
Biologist studying immune system response

Task: See if the results for tissue samples treated with LL-37 match up with the ones without the peptide.

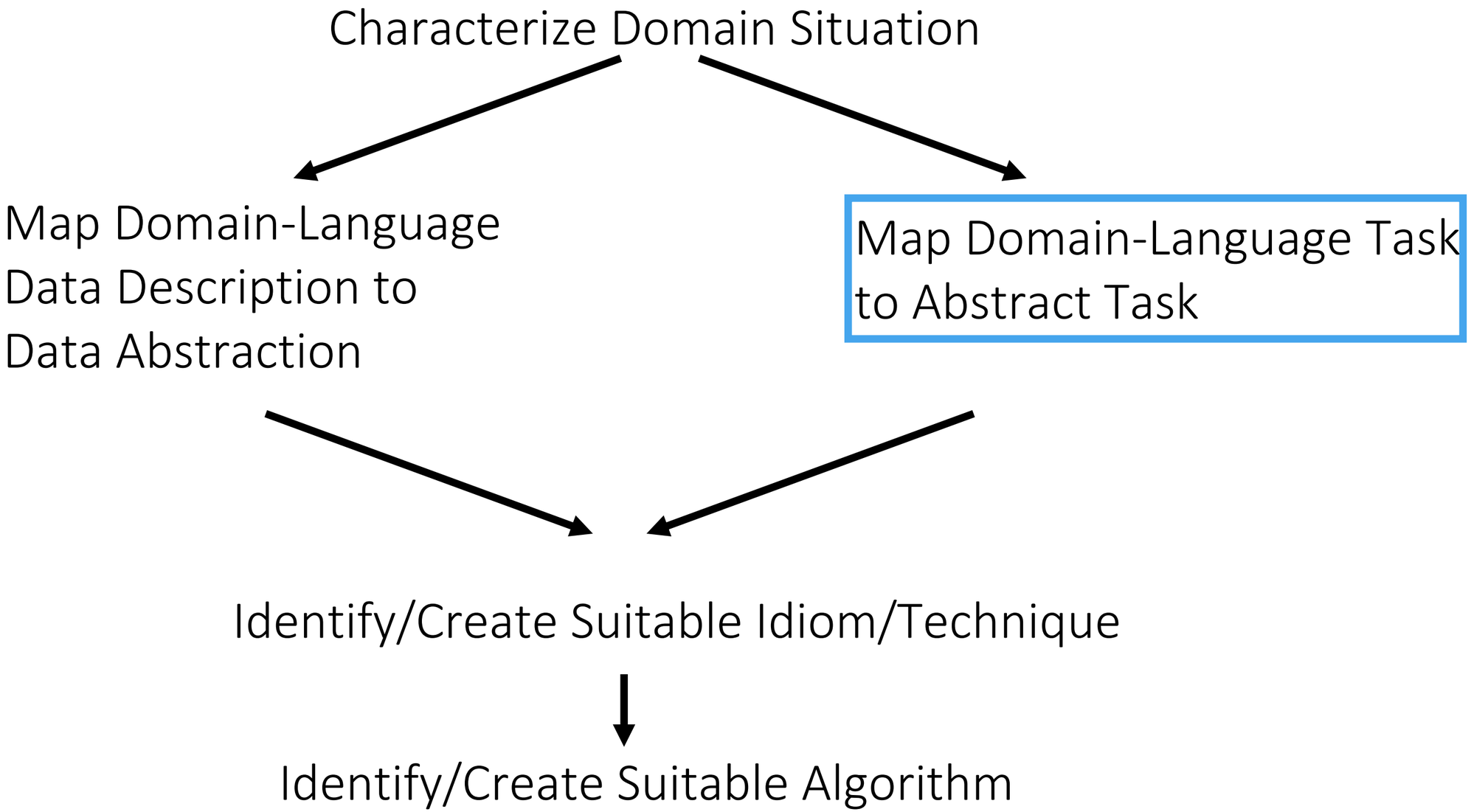
From domain to abstraction

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





Design process



Whose Task: Designer vs. User

From Tasks to Tools – A continuum

- Specific tools
  - Narrow use: typically designed for a specific context
  - Designer has customized the tool in such a way that the user cannot change it
  - Limited design choices
  - High stakes
- General tools
  - Users have a lot of power

Task abstraction: Actions and targets

- very high-level pattern

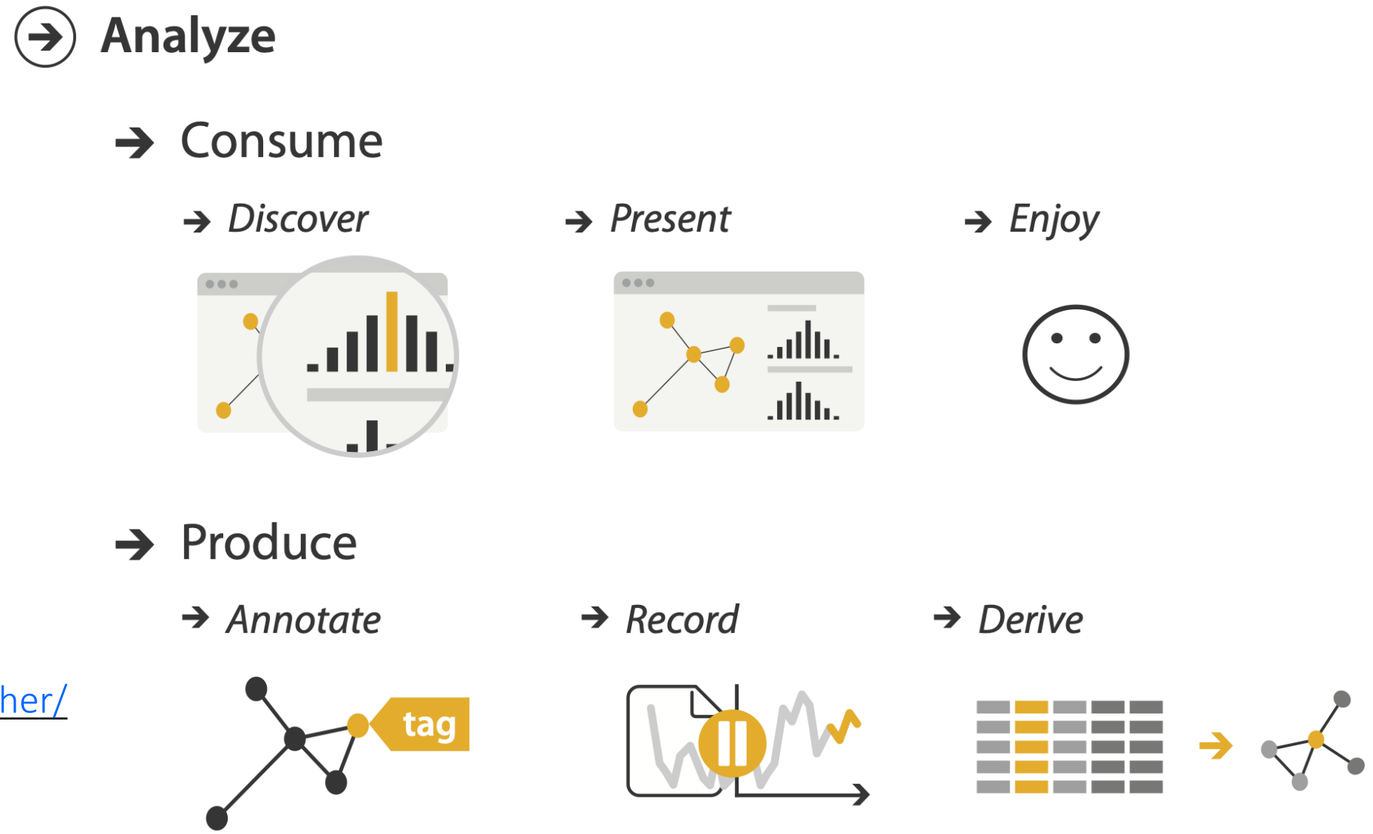
- actions
  - analyze
    - high-level choices
  - search
    - find a known/unknown item
  - query
    - find out about characteristics of item

- targets
  - what is being acted on

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

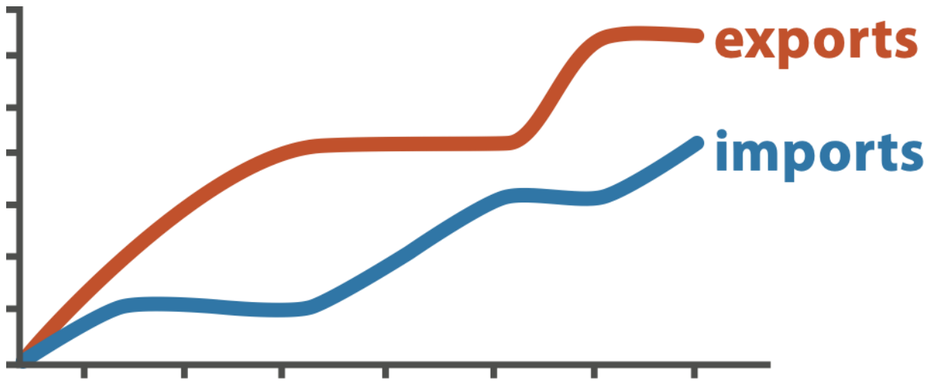
Actions: Analyze

- consume
  - discover vs present
    - classic split
    - aka explore vs explain
  - enjoy
    - newcomer
    - aka casual, social
    - https://namerology.com/baby-name-grapher/
- produce
  - annotate, record
  - derive
    - crucial design choice

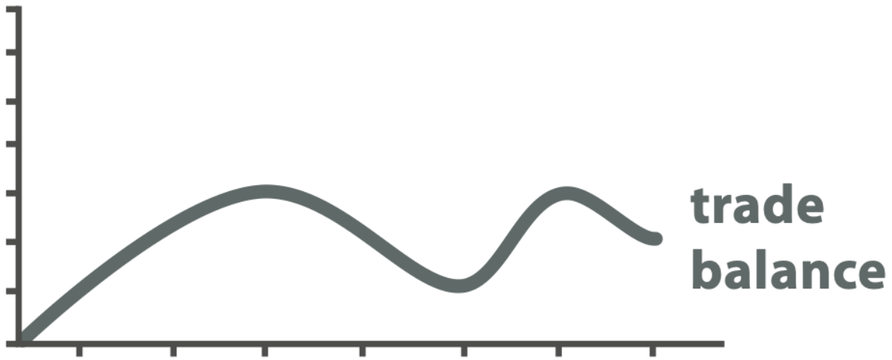


Derived attributes

- derived attribute: compute from originals
  - simple change of type
  - acquire additional data
  - complex transformation



Original Data



trade balance = *exports* – *imports*

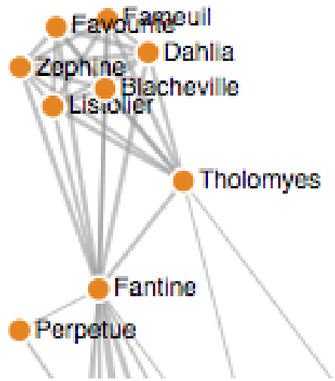
Derived Data

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: find cool neighborhood in new city

Search

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



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

Actions: Query

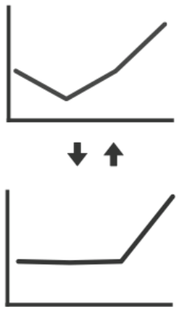
how much of the data matters?

- one: identify
- some: compare
- all: summarize

Query

→ Identify

→ Compare

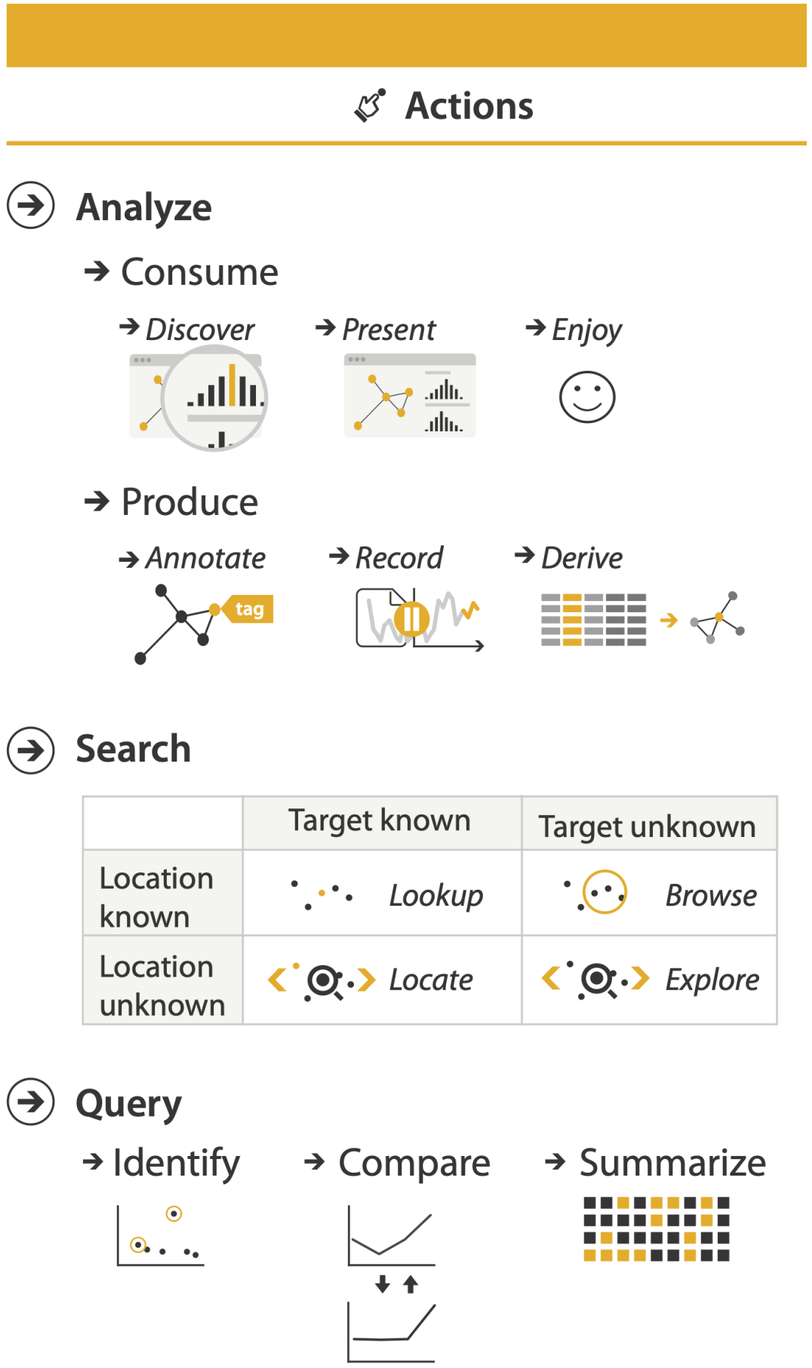


→ Summarize

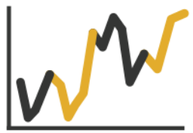
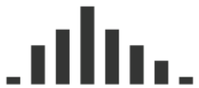
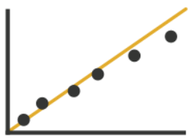
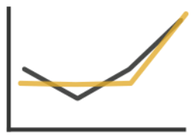
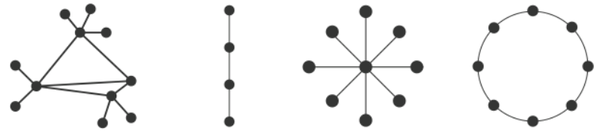
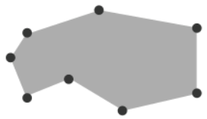
Actions

independent choices for each of these three levels

- analyze, search, query
- mix and match



Task abstraction: Targets

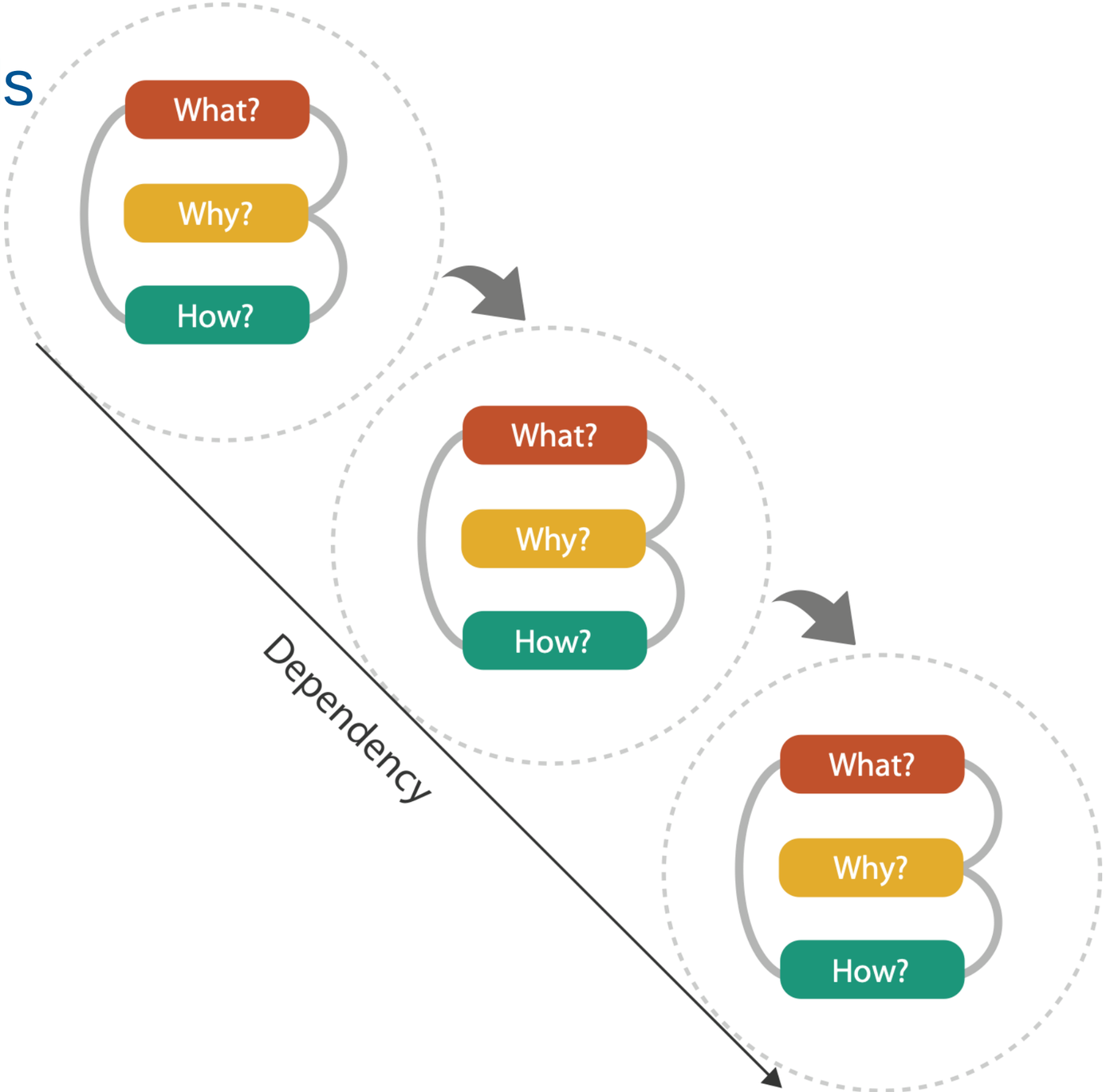
- **All Data**
  - Trends 
  - Outliers 
  - Features 
- **Attributes**
  - One
    - *Distribution* 
    - *Extremes* 
  - Many
    - *Dependency* 
    - *Correlation* 
    - *Similarity* 
- **Network Data**
  - Topology 
  - *Paths* 
- **Spatial Data**
  - Shape 

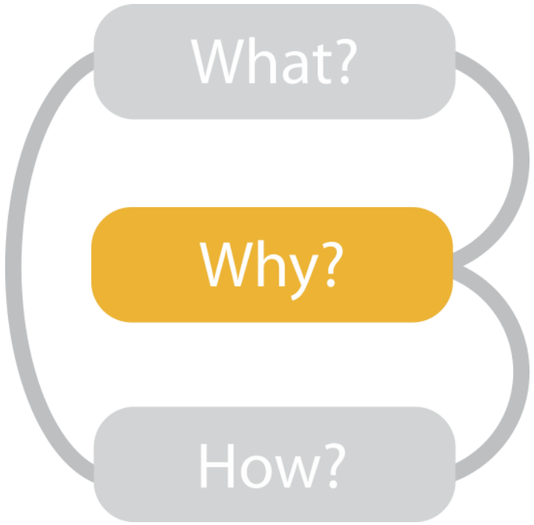
Abstraction

- 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, ...

Means and ends





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

Why?

Actions

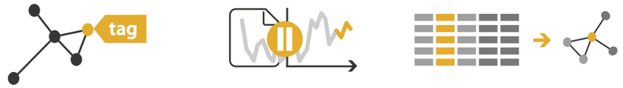
Analyze

→ Consume

→ *Discover* → *Present* → *Enjoy*

→ Produce

→ *Annotate* → *Record* → *Derive*

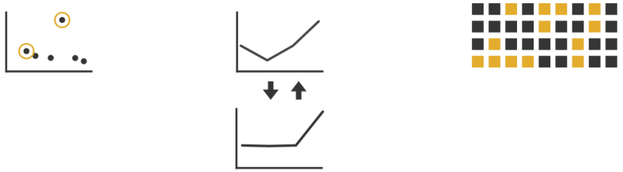


Search

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

Query

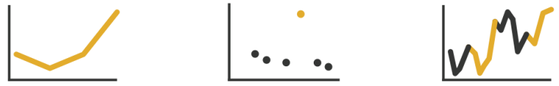
→ Identify → Compare → Summarize



Targets

All Data

→ Trends → Outliers → Features

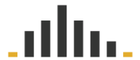


Attributes

→ One → Many

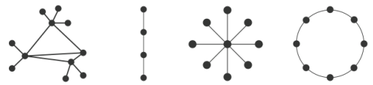
→ *Distribution* → *Dependency* → *Correlation* → *Similarity*

→ *Extremes*



Network Data

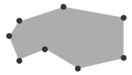
→ Topology



→ *Paths*

Spatial Data

→ Shape



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

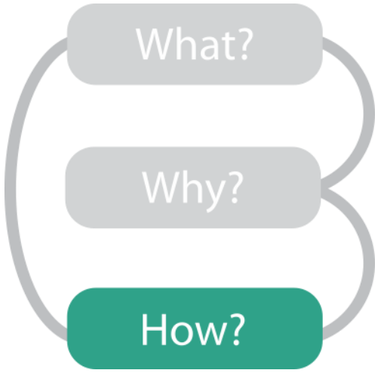
- Change
- Select
- Navigate

Facet

- Juxtapose
- Partition
- Superimpose

Reduce

- Filter
- Aggregate
- Embed



DATA

NYC OPEN DATA WEEK MULTI-PARK SQUIRREL COUNT

The Squirrel Census is a multimedia science, design, and storytelling project focusing on the Eastern gray (*Sciurus carolinensis*). They count squirrels and present their findings to the public.

On March 1, 2020 — with the help of 72 volunteer Squirrel Sighters, as well as NYC Open Data — they performed a sample count in 24 New York City parks, and gathered other material data. Four hundred and thirty-three squirrel sightings were tallied. The methodology was less focused on total squirrel numbers per hectare and more attuned to the stories — of squirrels, humans, and parks.

The data is organized into three sets below.

USER GUIDE

PARK DATA

SQUIRREL DATA

STORIES

Data and Task Abstraction Group Exercise – 20 minutes

- Dataset
- https://www.thesquirrelcensus.com/data

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)*