

Analisi e Visualizzazione di Reti Complesse

DV02 Nested Model
Data Abstraction
Task Abstraction

Prof. Rossano Schifanella



Nested Model

How to evaluate a visualization: So many methods, how to pick?

- Computational benchmarks?
 - quant: system performance, memory
- User study in lab setting?
 - quant: (human) time and error rates, preferences
 - qual: behavior/strategy observations
- Field study of deployed system?
 - quant: usage logs
 - qual: interviews with users, case studies, observations
- Analysis of results?
 - quant: metrics computed on result images
 - qual: consider what structure is visible in result images
- Justification of choices?
 - qual: perceptual principles, best practices

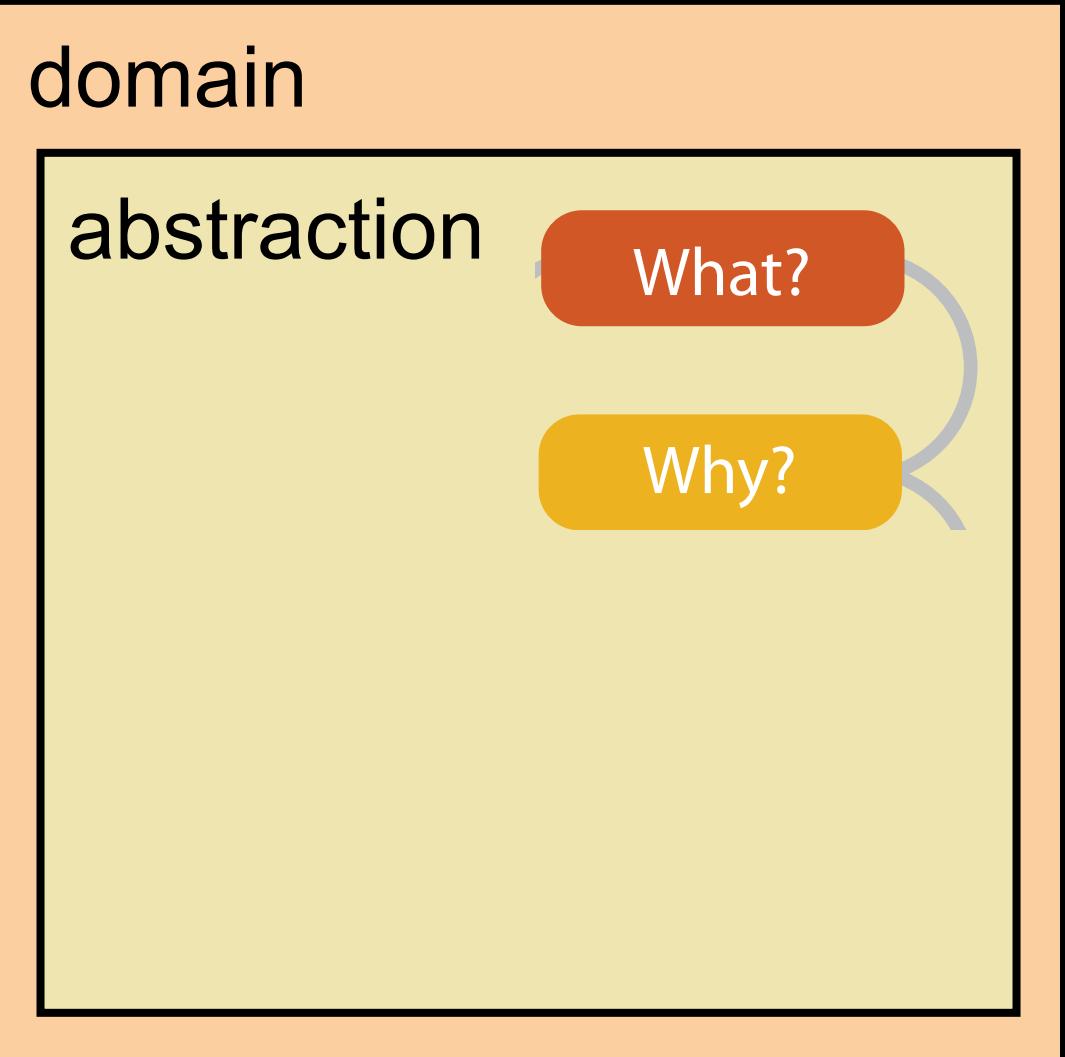
Analysis framework: Four levels, three questions

- **domain situation**
 - who are the target users?

domain

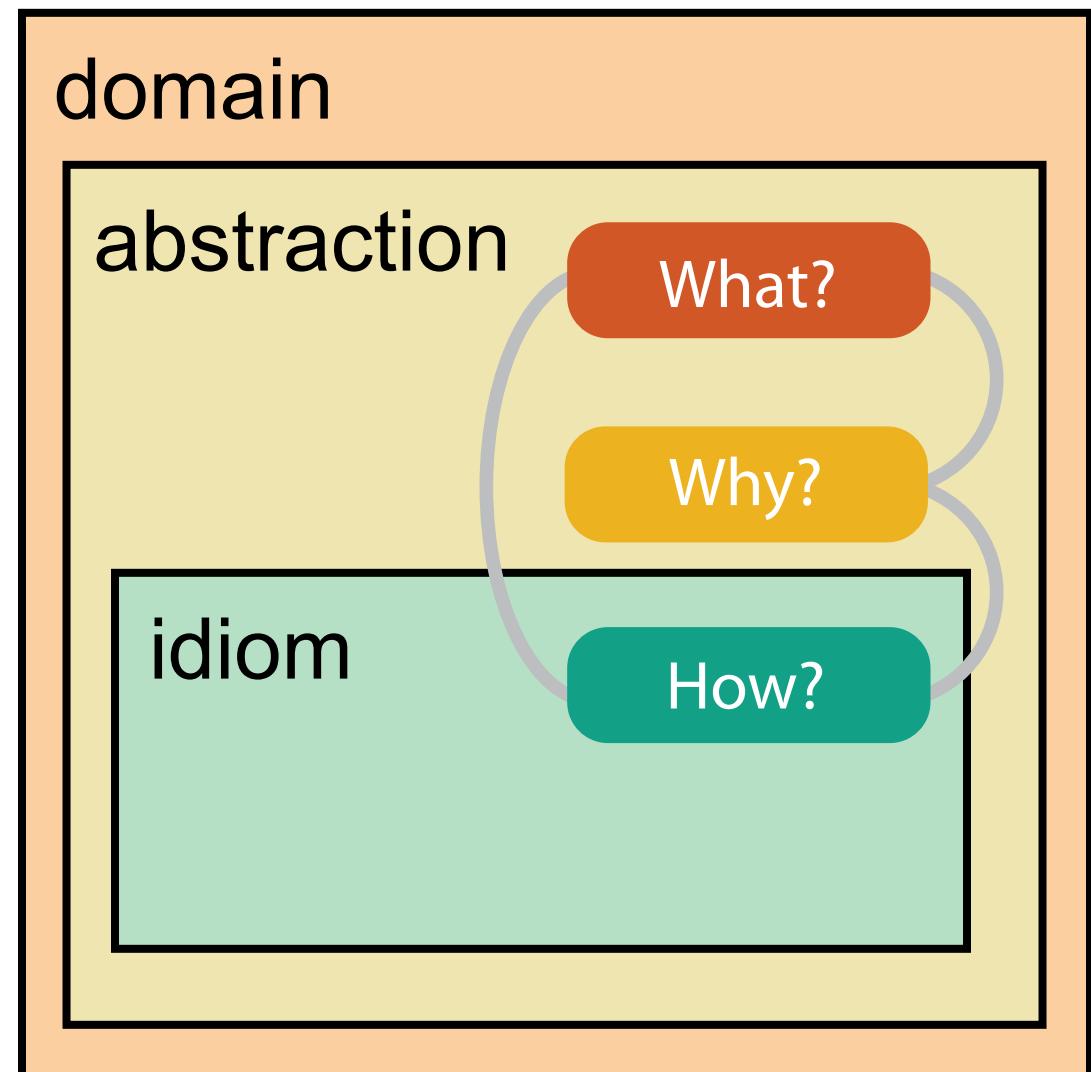
Analysis framework: Four levels, three questions

- **domain situation**
 - who are the target users?
- **abstraction**
 - translate from specifics of a domain to the vocabulary of vis
 - what is shown? data abstraction
 - why is the user looking at it? task abstraction



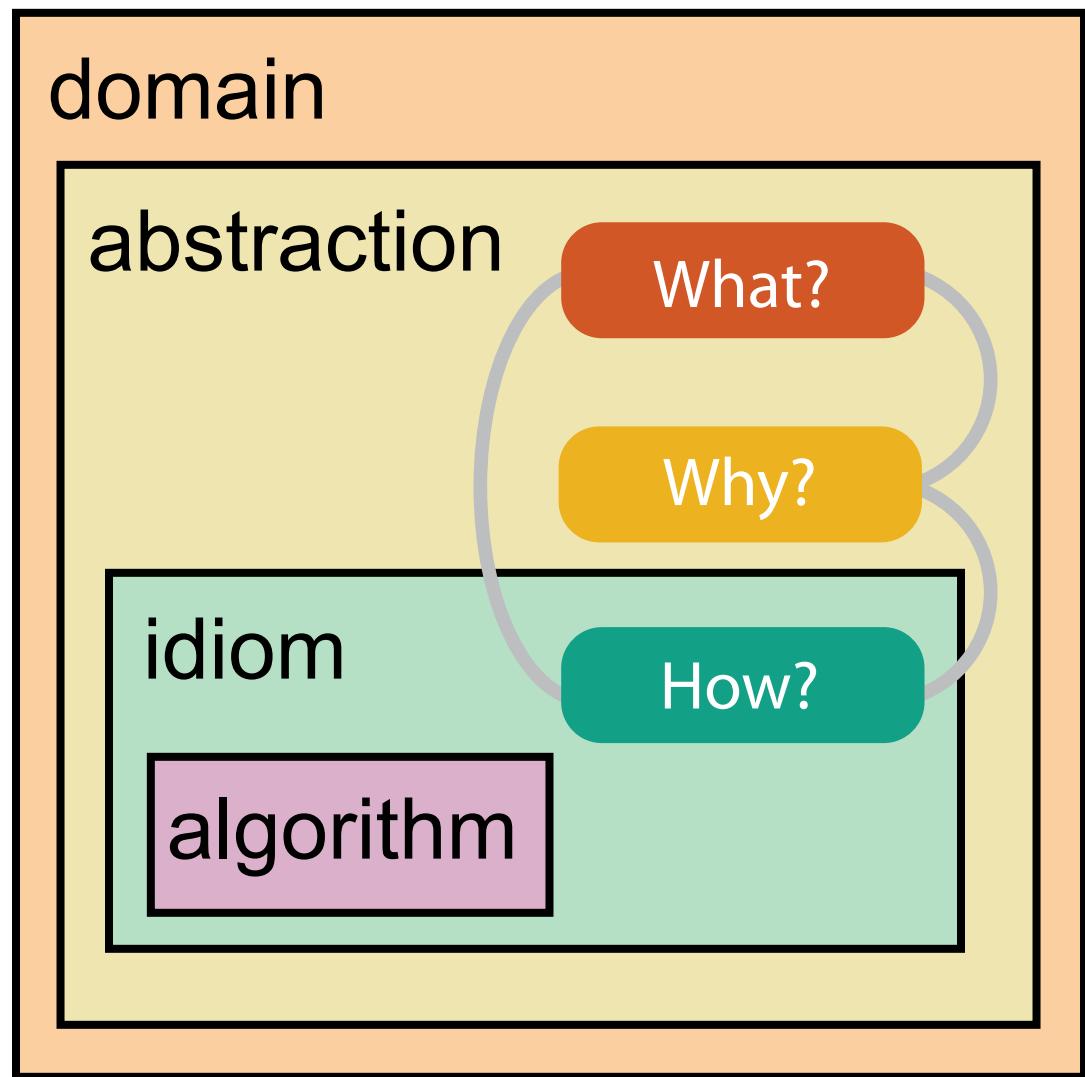
Analysis framework: Four levels, three questions

- **domain situation**
 - who are the target users?
- **abstraction**
 - translate from specifics of a domain to the 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



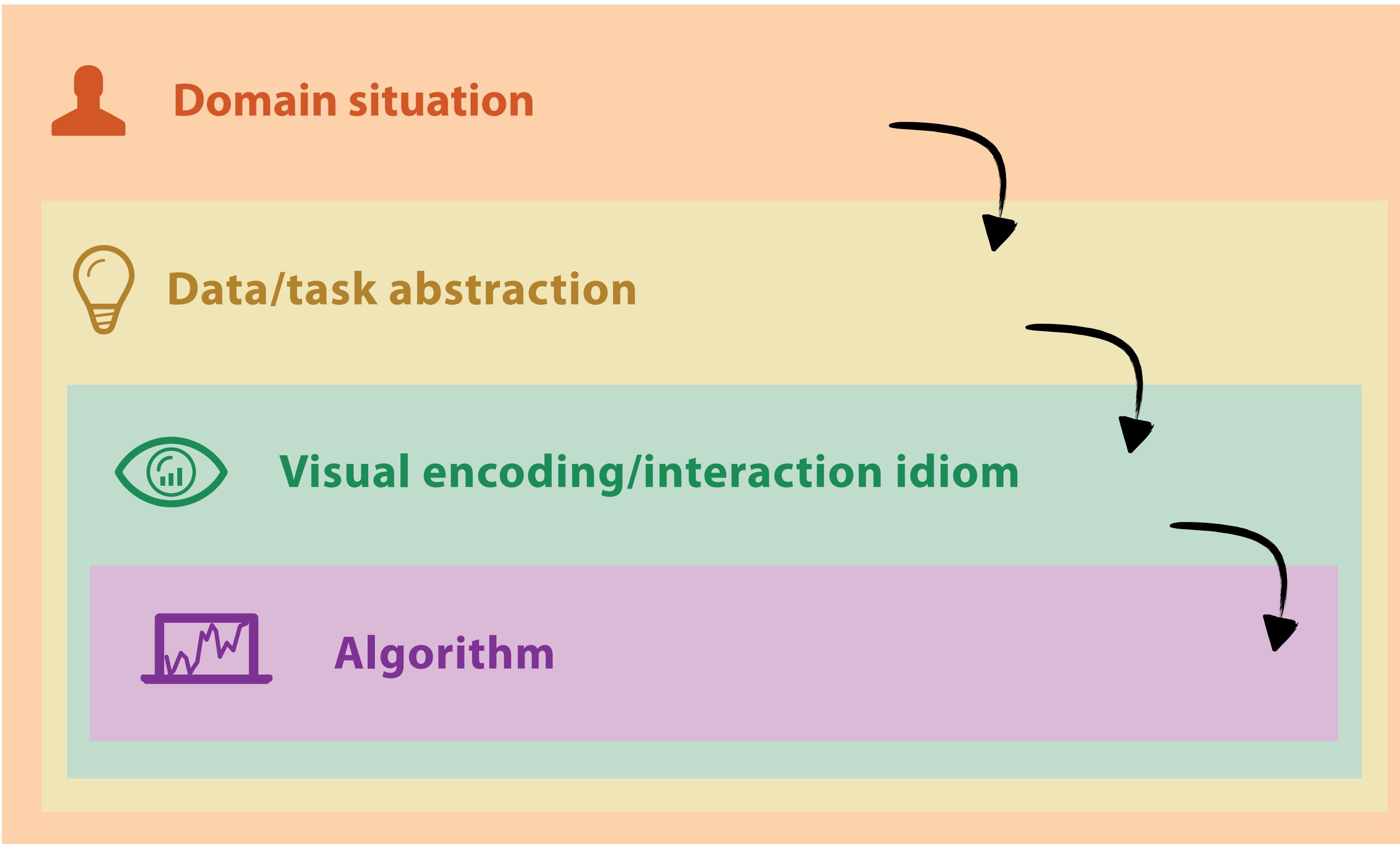
Analysis framework: Four levels, three questions

- **domain situation**
 - who are the target users?
- **abstraction**
 - translate from specifics of a domain to the 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



Nested model

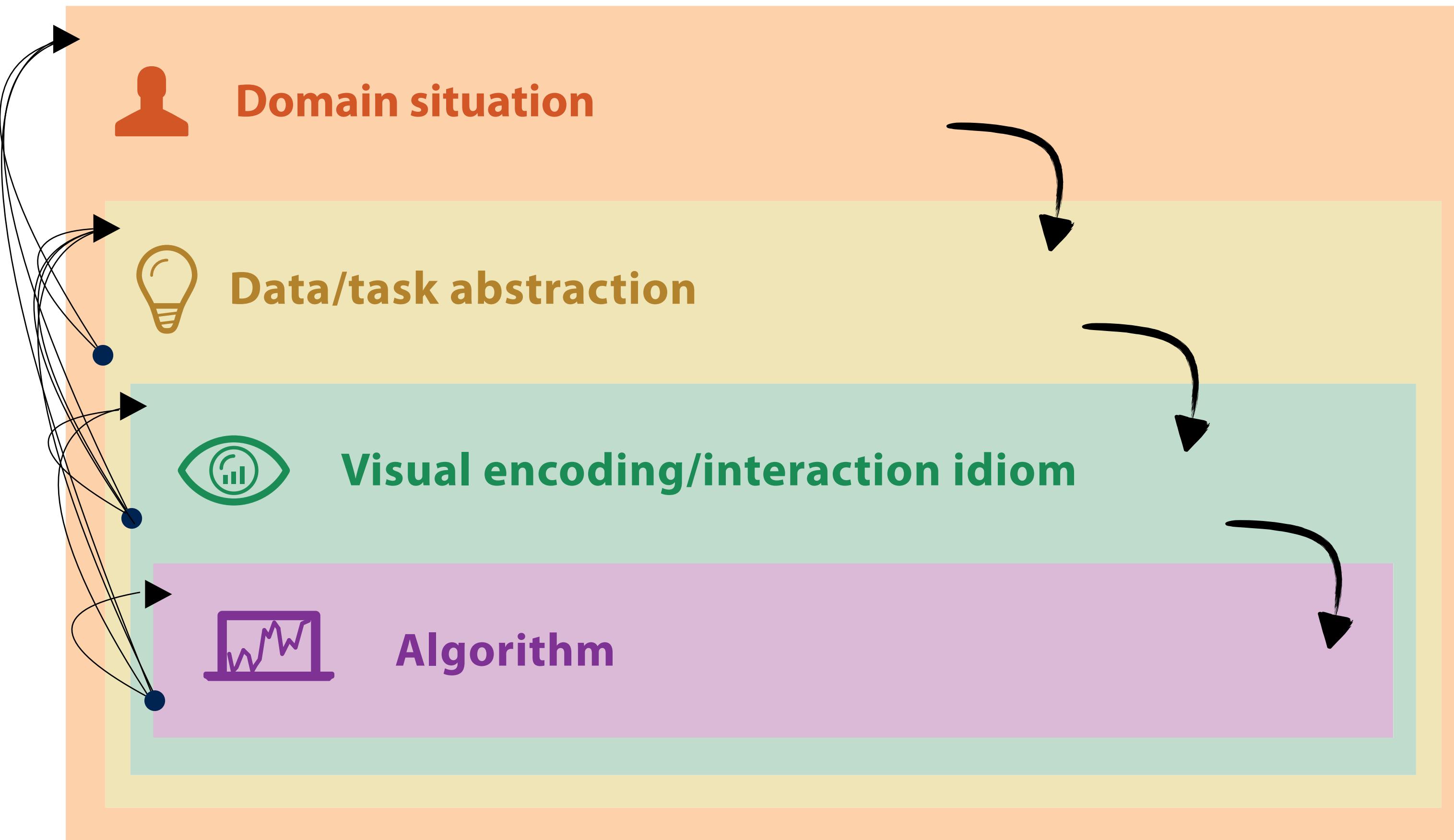
downstream: cascading effects



Nested model

downstream: cascading effects

upstream: iterative refinement



Why is validation difficult?

- different ways to get it wrong at each level



Domain situation

You misunderstood their needs



Data/task abstraction

You're showing them the wrong thing



Visual encoding/interaction idiom

The way you show it doesn't work



Algorithm

Your code is too slow

Why is validation difficult?

- solution: use methods from different fields at each level



Algorithm

Measure system time/memory

Analyze computational complexity

Why is validation difficult?

- solution: use methods from different fields at each level

computer
science



Algorithm

Measure system time/memory

Analyze computational complexity



technique-driven
work

Why is validation difficult?

- solution: use methods from different fields at each level

design

computer
science

cognitive
psychology



Visual encoding/interaction idiom

Justify design with respect to alternatives



Algorithm

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)



**technique-driven
work**

Why is validation difficult?

- solution: use methods from different fields at each level

anthropology/
ethnography

design

computer
science

cognitive
psychology

anthropology/
ethnography

👤 Domain situation

Observe target users using existing tools

💡 Data/task abstraction

👁️ Visual encoding/interaction idiom

Justify design with respect to alternatives

📈 Algorithm

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

➡ technique-driven
work

Why is validation difficult?

- solution: use methods from different fields at each level

anthropology/
ethnography

design

computer
science

cognitive
psychology

anthropology/
ethnography

👤 Domain situation

Observe target users using existing tools

💡 Data/task abstraction

👁️ Visual encoding/interaction idiom

Justify design with respect to alternatives

📈 Algorithm

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption



problem-driven work
(design study)

technique-driven
work

Avoid mismatches

👤 Domain situation

Observe target users using existing tools

💡 Data/task abstraction

👁️ Visual encoding/interaction idiom

Justify design with respect to alternatives

📈 Algorithm

Measure system time/memory

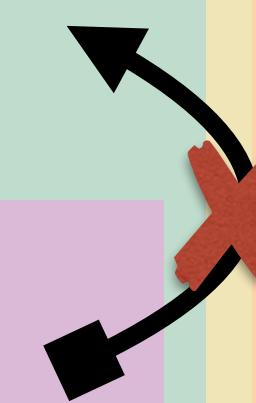
Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

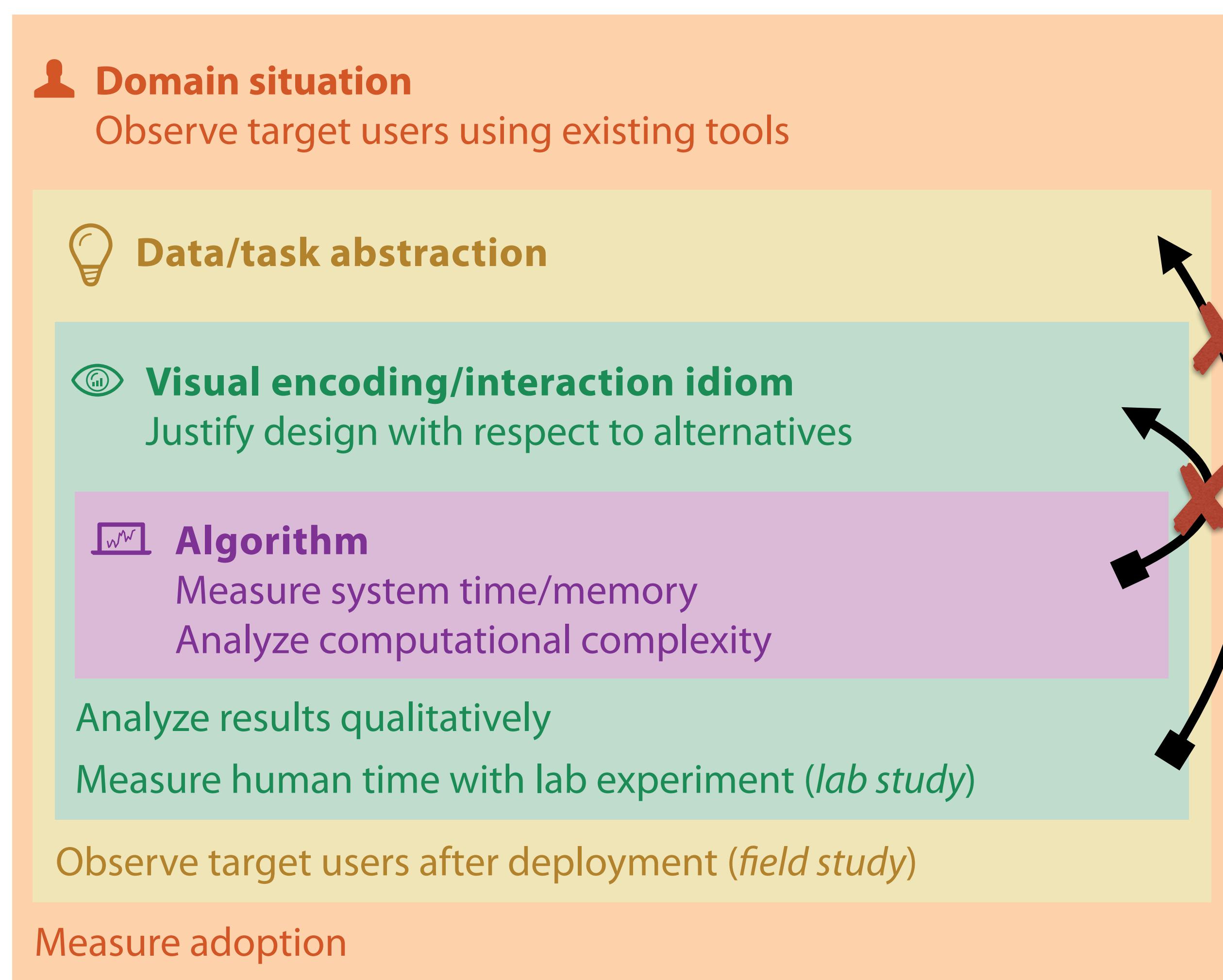
Observe target users after deployment (*field study*)

Measure adoption



computational benchmarks
do not confirm idiom design

Avoid mismatches



lab studies do not confirm task abstraction
computational benchmarks do not confirm idiom design

Reading Material

[dv3] Chapter 4 - Analysis: Four Levels for Validation

Data Abstraction

What does data mean?

14, 2.6, 30, 30, 15, 100001

- What does this sequence of six numbers mean?
 - two points far from each other in 3D space?
 - two points close to each other in 2D space, with 15 links between them, and a weight of 100001 for the link?
 - something else?

Basil, 7, S, Pear

- What about this data?
 - food shipment of produce (basil & pear) arrived in satisfactory condition on the 7th day of the month
 - Basil Point neighborhood of the city had 7 inches of snow cleared by the Pear Creek Limited snow removal service
 - lab rat Basil made 7 attempts to find a way through the south section of the maze, these trials used pear as reward food

Now what?

- **semantics:** real-world meaning
- **data types:** structural or mathematical interpretation of data
 - item, link, attribute, position, (grid)
 - different from data types in programming!

Name	Age	Shirt Size	Favorite Fruit
Amy	8	S	Apple
Basil	7	S	Pear
Clara	9	M	Durian
Desmond	13	L	Elderberry
Ernest	12	L	Peach
Fanny	10	S	Lychee
George	9	M	Orange
Hector	8	L	Loquat
Ida	10	M	Pear
Amy	12	M	Orange

Items & Attributes

- **item:** individual entity, discrete
 - eg, patient, car, stock, city
 - "independent variable"
- **attribute:** property that is measured, observed, logged...
 - eg, height, blood pressure for a patient
 - eg, horsepower, make for car
 - "dependent variable"

attributes: name, age, shirt size, fave fruit

Name	Age	Shirt Size	Favorite Fruit
Amy	8	S	Apple
Basil	7	S	Pear
Clara	9	M	Durian
Desmond	13	L	Elderberry
Ernest	12	L	Peach
Fanny	10	S	Lychee
George	9	M	Orange
Hector	8	L	Loquat
Ida	10	M	Pear
Amy	12	M	Orange

item: person

Other data types

- **links**
 - express the relationship between two items
 - e.g., friendship on Facebook, interaction between proteins
- **positions**
 - spatial data: location in 2D or 3D
 - pixels in photo, voxels in MRI scan, latitude/longitude
- **grids**
 - sampling strategy for continuous data

Dataset types

Tables

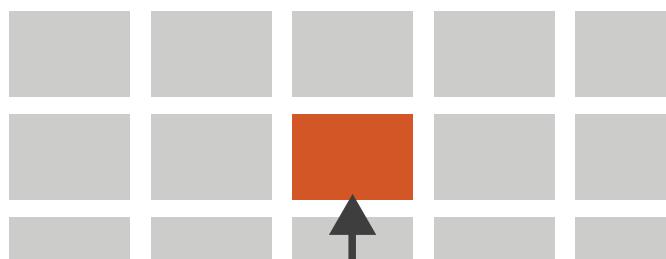
Items

Attributes

→ Tables

Attributes (columns)

Items
(rows)



Cell containing value

flat table

- one item per row
- each column is attribute
- cell holds value for item-attribute pair
- unique key (could be implicit)

attributes: name, age, shirt size, fave fruit

Name	Age	Shirt Size	Favorite Fruit
Amy	8	S	Apple
Basil	7	S	Pear
Clara	9	M	Durian
Desmond	13	L	Elderberry
Ernest	12	L	Peach
Fanny	10	S	Lychee
George	9	M	Orange
Hector	8	L	Loquat
Ida	10	M	Pear
Amy	12	M	Orange

item: person

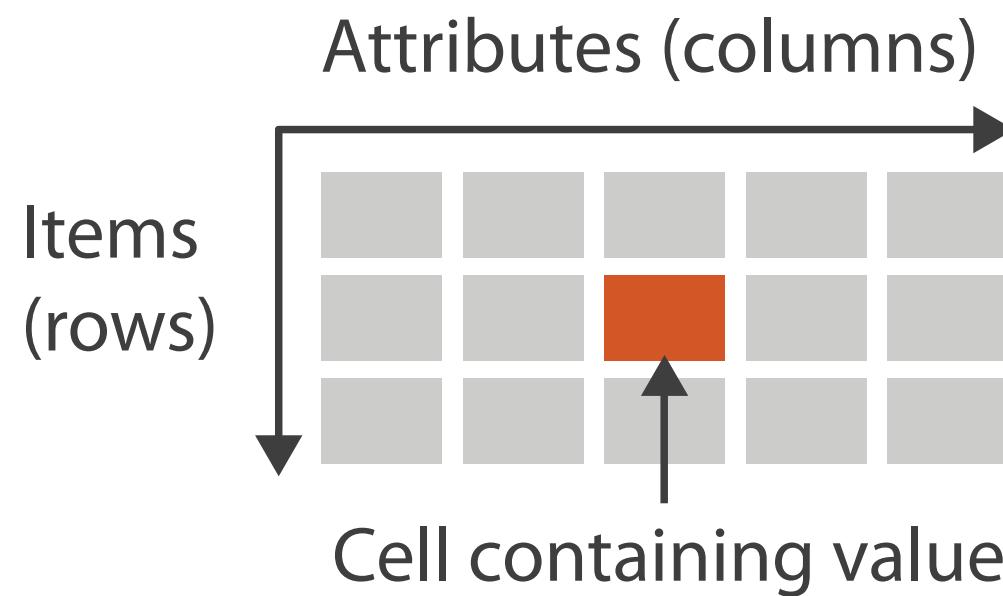
Dataset types

Tables

Items

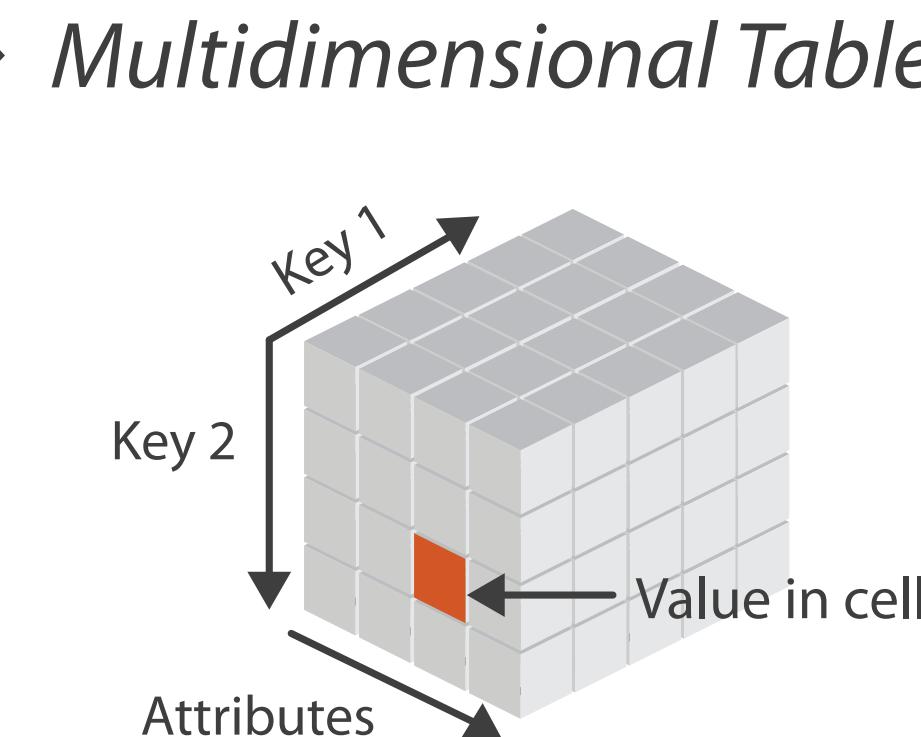
Attributes

→ Tables



multidimensional tables

- indexing based on multiple keys
 - eg, genes, patients



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																															
1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																														
2	1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																													
3	2	1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																												
4	3	2	1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																											
5	4	3	2	1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z																																																																																																																																																																										
6	5	4	3	2	1	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	RP11-35N6.1	TCGA-02-0003-01A-01R-0177-01	TCGA-02-0004-01A-01R-0298-01	TCGA-02-0005-01A-01R-0301-01	TCGA-02-0006-01A-01R-0302-01	TCGA-02-0007-01A-01R-0303-01	TCGA-02-0008-01A-01R-0304-01	TCGA-02-0009-01A-01R-0305-01	TCGA-02-0010-01A-01R-0306-01	TCGA-02-0011-01A-01R-0307-01	TCGA-02-0012-01A-01R-0308-01	TCGA-02-0013-01A-01R-0309-01	TCGA-02-0014-01A-01R-0310-01	TCGA-02-0015-01A-01R-0311-01	TCGA-02-0016-01A-01R-0312-01	TCGA-02-0017-01A-01R-0313-01	TCGA-02-0018-01A-01R-0314-01	TCGA-02-0019-01A-01R-0315-01	TCGA-02-0020-01A-01R-0316-01	TCGA-02-0021-01A-01R-0317-01	TCGA-02-0022-01A-01R-0318-01	TCGA-02-0023-01A-01R-0319-01	TCGA-02-0024-01A-01R-0320-01	TCGA-02-0025-01A-01R-0321-01	TCGA-02-0026-01A-01R-0322-01	TCGA-02-0027-01A-01R-0323-01	TCGA-02-0028-01A-01R-0324-01	TCGA-02-0029-01A-01R-0325-01	TCGA-02-0030-01A-01R-0326-01	TCGA-02-0031-01A-01R-0327-01	TCGA-02-0032-01A-01R-0328-01	TCGA-02-0033-01A-01R-0329-01	TCGA-02-0034-01A-01R-0330-01	TCGA-02-0035-01A-01R-0331-01	TCGA-02-0036-01A-01R-0332-01	TCGA-02-0037-01A-01R-0333-01	TCGA-02-0038-01A-01R-0334-01	TCGA-02-0039-01A-01R-0335-01	TCGA-02-0040-01A-01R-0336-01	TCGA-02-0041-01A-01R-0337-01	TCGA-02-0042-01A-01R-0338-01	TCGA-02-0043-01A-01R-0339-01	TCGA-02-0044-01A-01R-0340-01	TCGA-02-0045-01A-01R-0341-01	TCGA-02-0046-01A-01R-0342-01	TCGA-02-0047-01A-01R-0343-01	TCGA-02-0048-01A-01R-0344-01	TCGA-02-0049-01A-01R-0345-01	TCGA-02-0050-01A-01R-0346-01	TCGA-02-0051-01A-01R-0347-01	TCGA-02-0052-01A-01R-0348-01	TCGA-02-0053-01A-01R-0349-01	TCGA-02-0054-01A-01R-0350-01	TCGA-02-0055-01A-01R-0351-01	TCGA-02-0056-01A-01R-0352-01	TCGA-02-0057-01A-01R-0353-01	TCGA-02-0058-01A-01R-0354-01	TCGA-02-0059-01A-01R-0355-01	TCGA-02-0060-01A-01R-0356-01	TCGA-02-0061-01A-01R-0357-01	TCGA-02-0062-01A-01R-0358-01	TCGA-02-0063-01A-01R-0359-01	TCGA-02-0064-01A-01R-0360-01	TCGA-02-0065-01A-01R-0361-01	TCGA-02-0066-01A-01R-0362-01	TCGA-02-0067-01A-01R-0363-01	TCGA-02-0068-01A-01R-0364-01	TCGA-02-0069-01A-01R-0365-01	TCGA-02-0070-01A-01R-0366-01	TCGA-02-0071-01A-01R-0367-01	TCGA-02-0072-01A-01R-0368-01	TCGA-02-0073-01A-01R-0369-01	TCGA-02-0074-01A-01R-0370-01	TCGA-02-0075-01A-01R-0371-01	TCGA-02-0076-01A-01R-0372-01	TCGA-02-0077-01A-01R-0373-01	TCGA-02-0078-01A-01R-0374-01	TCGA-02-0079-01A-01R-0375-01	TCGA-02-0080-01A-01R-0376-01	TCGA-02-0081-01A-01R-0377-01	TCGA-02-0082-01A-01R-0378-01	TCGA-02-0083-01A-01R-0379-01	TCGA-02-0084-01A-01R-0380-01	TCGA-02-0085-01A-01R-0381-01	TCGA-02-0086-01A-01R-0382-01	TCGA-02-0087-01A-01R-0383-01	TCGA-02-0088-01A-01R-0384-01	TCGA-02-0089-01A-01R-0385-01	TCGA-02-0090-01A-01R-0386-01	TCGA-02-0091-01A-01R-0387-01	TCGA-02-0092-01A-01R-0388-01	TCGA-02-0093-01A-01R-0389-01	TCGA-02-0094-01A-01R-0390-01	TCGA-02-0095-01A-01R-0391-01	TCGA-02-0096-01A-01R-0392-01	TCGA-02-0097-01A-01R-0393-01	TCGA-02-0098-01A-01R-0394-01	TCGA-02-0099-01A-01R-0395-01	TCGA-02-0100-01A-01R-0396-01	TCGA-02-0101-01A-01R-0397-01	TCGA-02-0102-01A-01R-0398-01	TCGA-02-0103-01A-01R-0399-01	TCGA-02-0104-01A-01R-0400-01	TCGA-02-0105-01A-01R-0401-01	TCGA-02-0106-01A-01R-0402-01	TCGA-02-0107-01A-01R-0403-01	TCGA-02-0108-01A-01R-0404-01	TCGA-02-0109-01A-01R-0405-01	TCGA-02-0110-01A-01R-0406-01	TCGA-02-0111-01A-01R-0407-01	TCGA-02-0112-01A-01R-0408-01	TCGA-02-0113-01A-01R-0409-01	TCGA-02-0114-01A-01R-0410-01	TCGA-02-0115-01A-01R-0411-01	TCGA-02-0116-01A-01R-0412-01	TCGA-02-0117-01A-01R-0413-01	TCGA-02-0118-01A-01R-0414-01	TCGA-02-0119-01A-01R-0415-01	TCGA-02-0120-01A-01R-0416-01	TCGA-02-0121-01A-01R-0417-01	TCGA-02-0122-01A-01R-0418-01	TCGA-02-0123-01A-01R-0419-01	TCGA-02-0124-01A-01R-0420-01	TCGA-02-0125-01A-01R-0421-01	TCGA-02-0126-01A-01R-0422-01	TCGA-02-0127-01A-01R-0423-01	TCGA-02-0128-01A-01R-0424-01	TCGA-02-0129-01A-01R-0425-01	TCGA-02-0130-01A-01R-0426-01	TCGA-02-0131-01A-01R-0427-01	TCGA-02-0132-01A-01R-0428-01	TCGA-02-0133-01A-01R-0429-01	TCGA-02-0134-01A-01R-0430-01	TCGA-02-0135-01A-01R-0431-01	TCGA-02-0136-01A-01R-0432-01	TCGA-02-0137-01A-01R-0433-01	TCGA-02-0138-01A-01R-0434-01	TCGA-02-0139-01A-01R-0435-01	TCGA-02-0140-01A-01R-0436-01	TCGA-02-0141-01A-01R-0437-01	TCGA-02-0142-01A-01R-0438-01	TCGA-02-0143-01A-01R-0439-01	TCGA-02-0144-01A-01R-0440-01	TCGA-02-0145-01A-01R-0441-01	TCGA-02-0146-01A-01R-0442-01	TCGA-02-0147-01A-01R-0443-01	TCGA-02-0148-01A-01R-0444-01	TCGA-02-0149-01A-01R-0445-01	TCGA-02-0150-01A-01R-0446-01	TCGA-02-0151-01A-01R-0447-01	TCGA-02-0152-01A-01R-0448-01	TCGA-02-0153-01A-01R-0449-01	TCGA-02-0154-01A-01R-0450-01	TCGA-02-0155-01A-01R-0451-01	TCGA-02-0156-01A-01R-0452-01	TCGA-02-0157-01A-01R-0453-01	TCGA-02-0158-01A-01R-0454-01	TCGA-02-0159-01A-01R-0455-01	TCGA-02-0160-01A-01R-0456-01	TCGA-02-0161-01A-01R-0457-01	TCGA-02-0162-01A-01R-0458-01	TCGA-02-0163-01A-01R-0459-01	TCGA-02-0164-01A-01R-0460-01	TCGA-02-0165-01A-01R-0461-01	TCGA-02-0166-01A-01R-0462-01	TCGA-02-0167-01A-01R-0463-01	TCGA-02-0168-01A-01R-0464-01	TCGA-02-0169-01A-01R-0465-01	TCGA-02-0170-01A-01R-0466-01	TCGA-02-0171-01A-01R-0467-01	TCGA-02-0172-01A-01R-0468-01	TCGA-02-0173-01A-01R-0469-01	TCGA-02-0174-01A-01R-0470-01	TCGA-02-0175-01A-01R-0471-01	TCGA-02-0176-01A-01R-0472-01	TCGA-02-0177-01A-01R-0473-01	TCGA-02-0178-01A-01R-0474-01	TCGA-02-0179-01A-01R-0475

Dataset types

Tables

Items

Attributes

Networks &
Trees

Items (nodes)

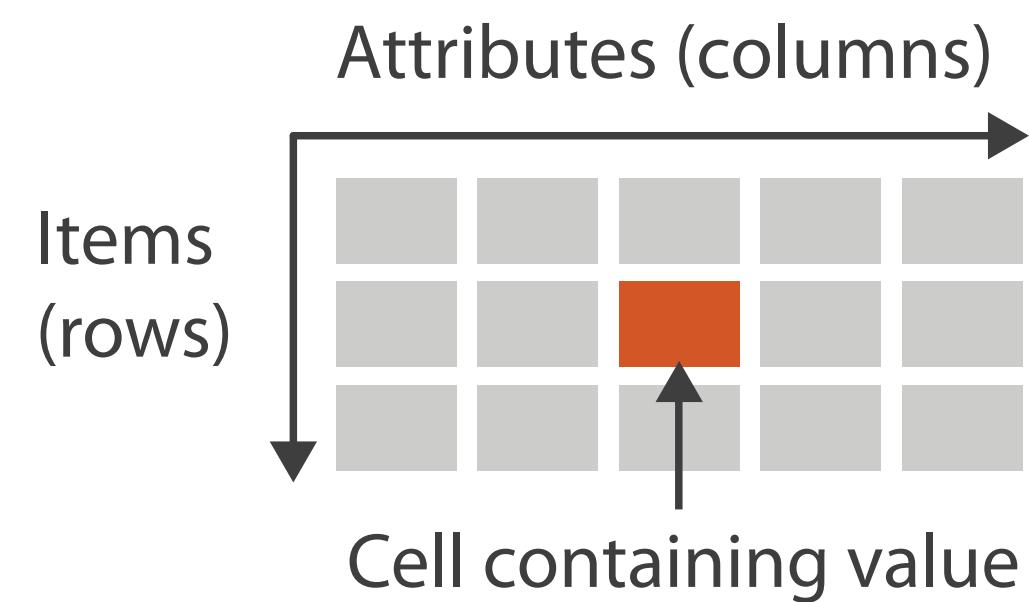
Links

Attributes

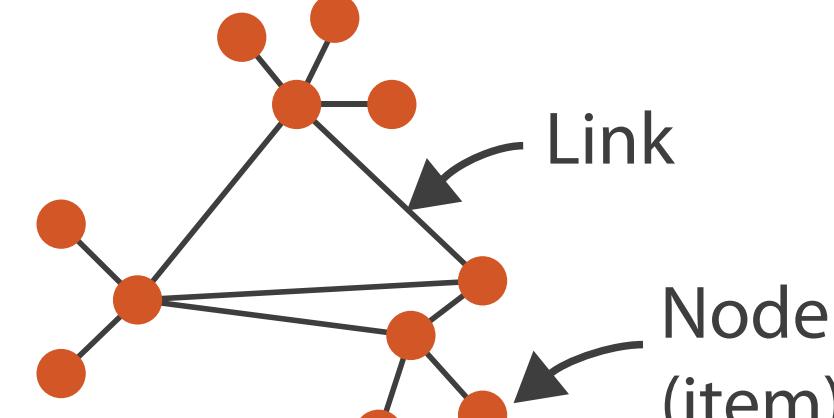
network/graph

- nodes (vertices) connected by links (edges)
- tree is a special case: no cycles
 - often have roots and are directed

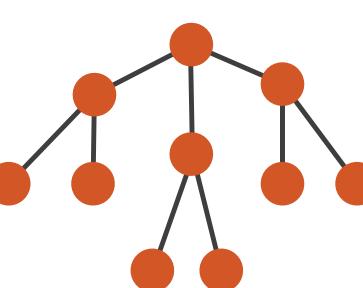
→ Tables



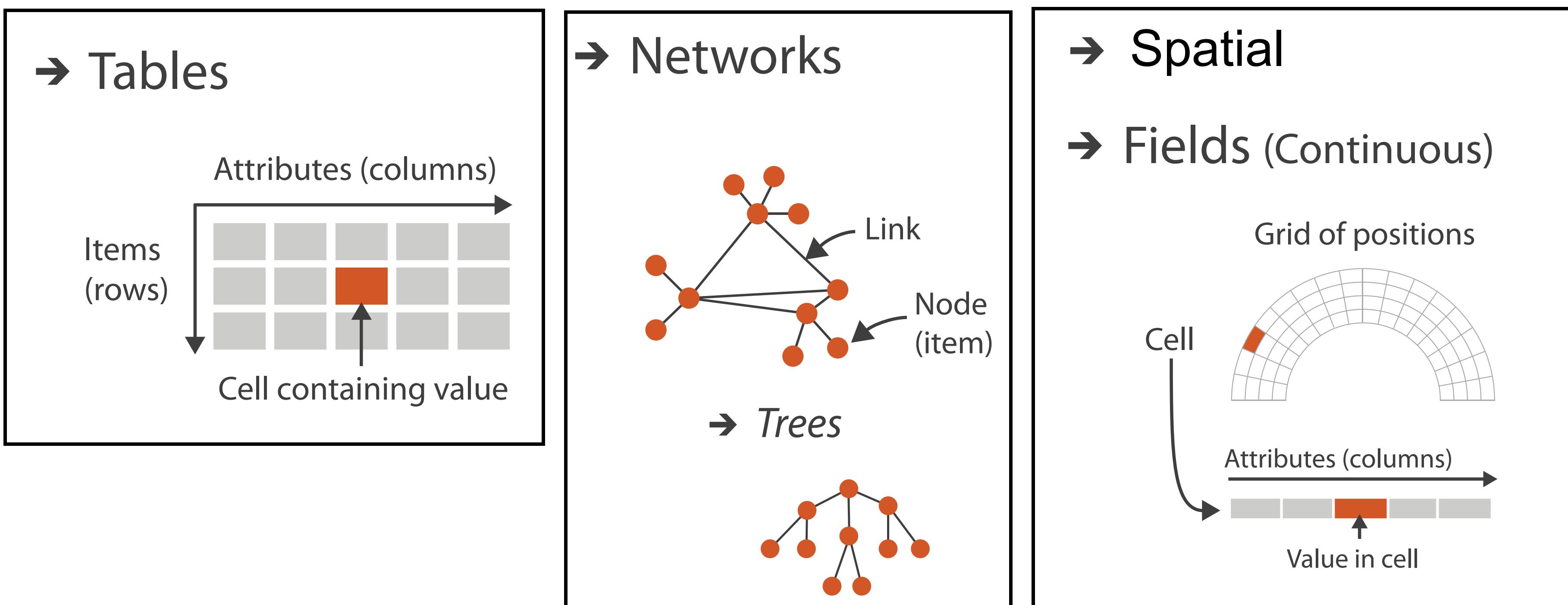
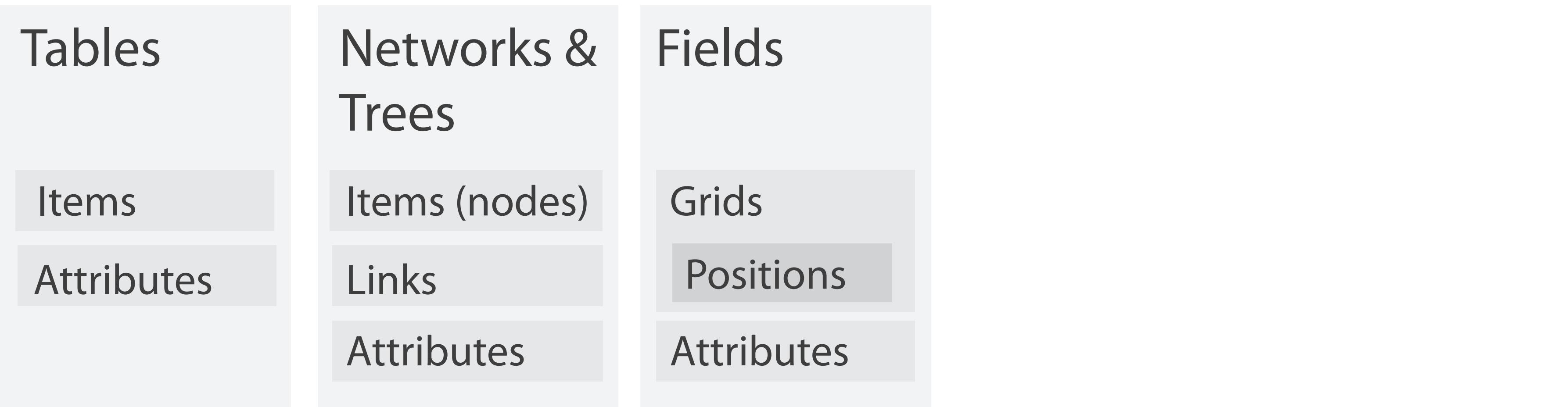
→ Networks



→ Trees

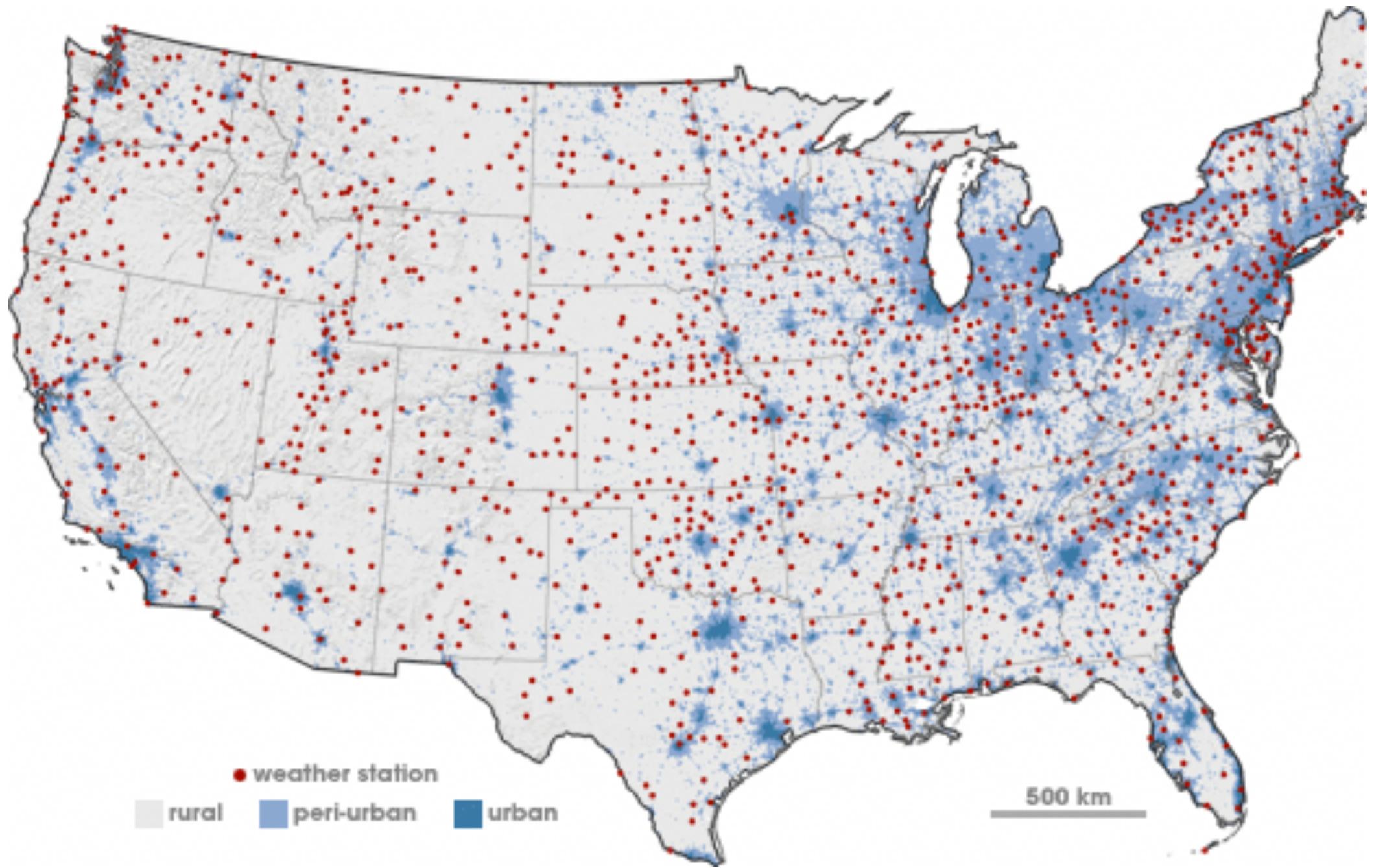
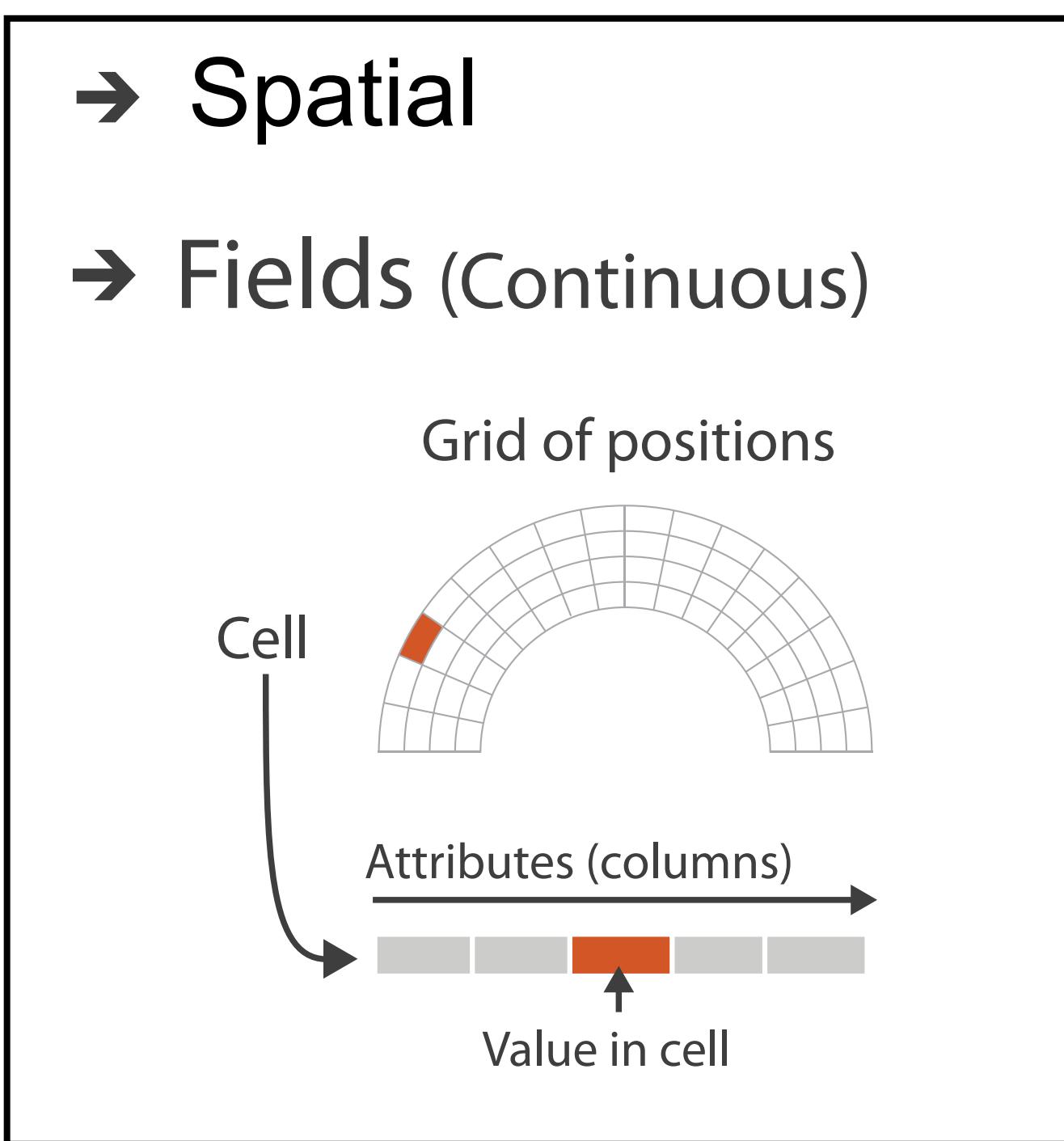


Dataset types



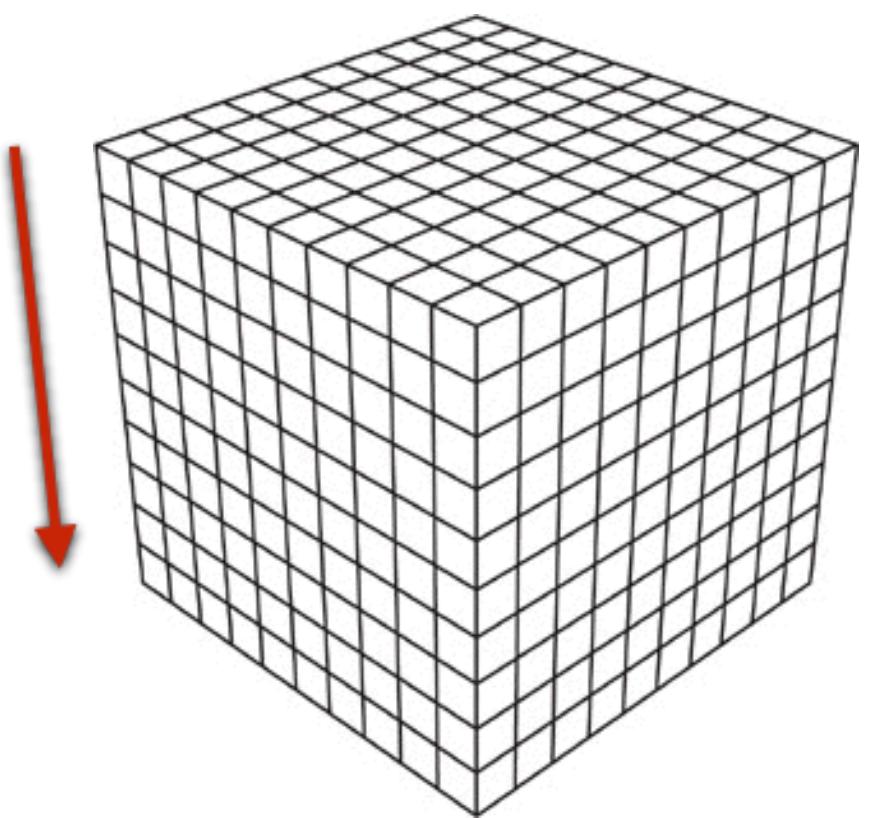
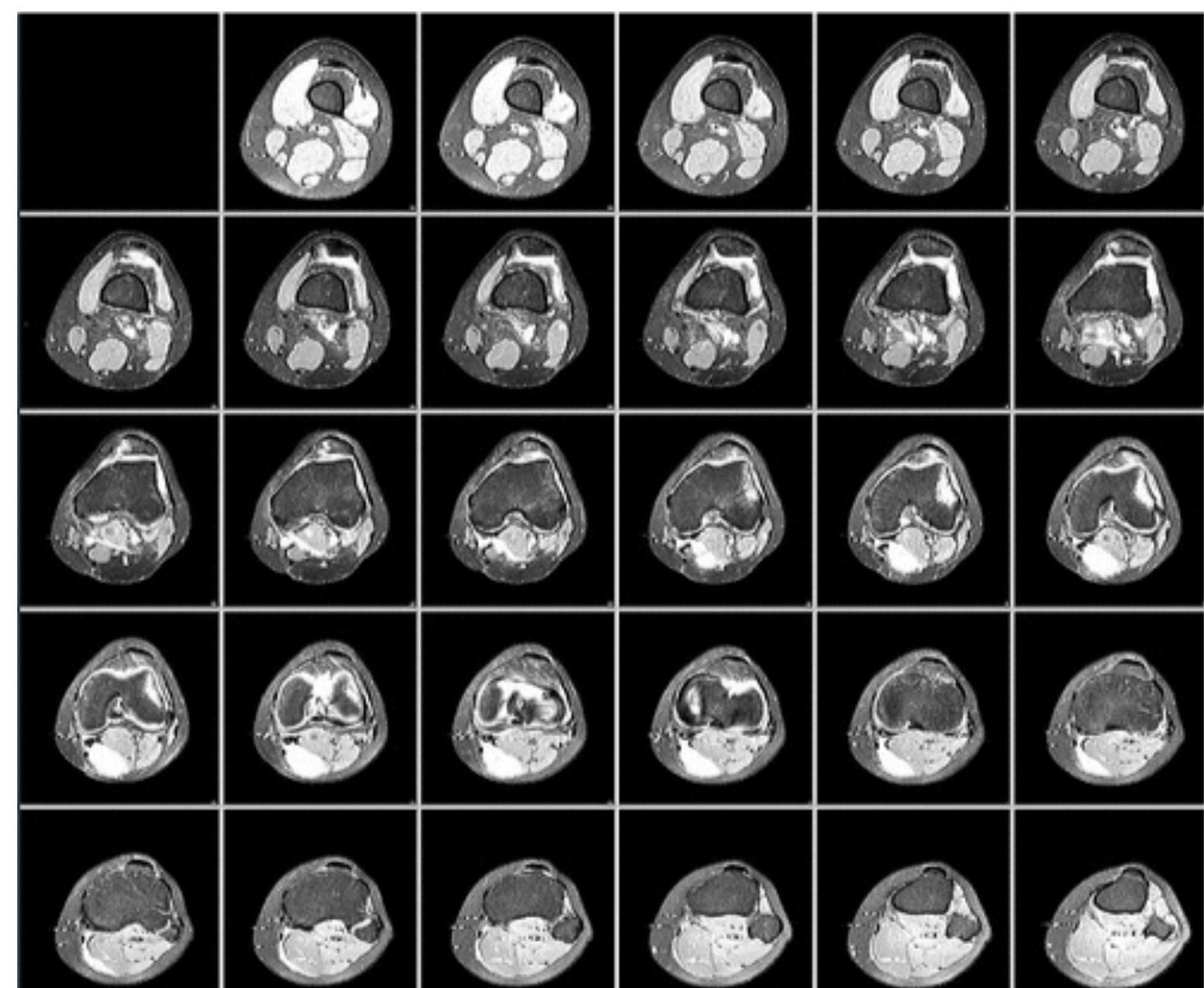
Spatial fields

- attribute values associated w/ cells
- cell contains value from continuous domain
 - eg temperature, pressure, wind velocity
- measured or simulated



Spatial fields

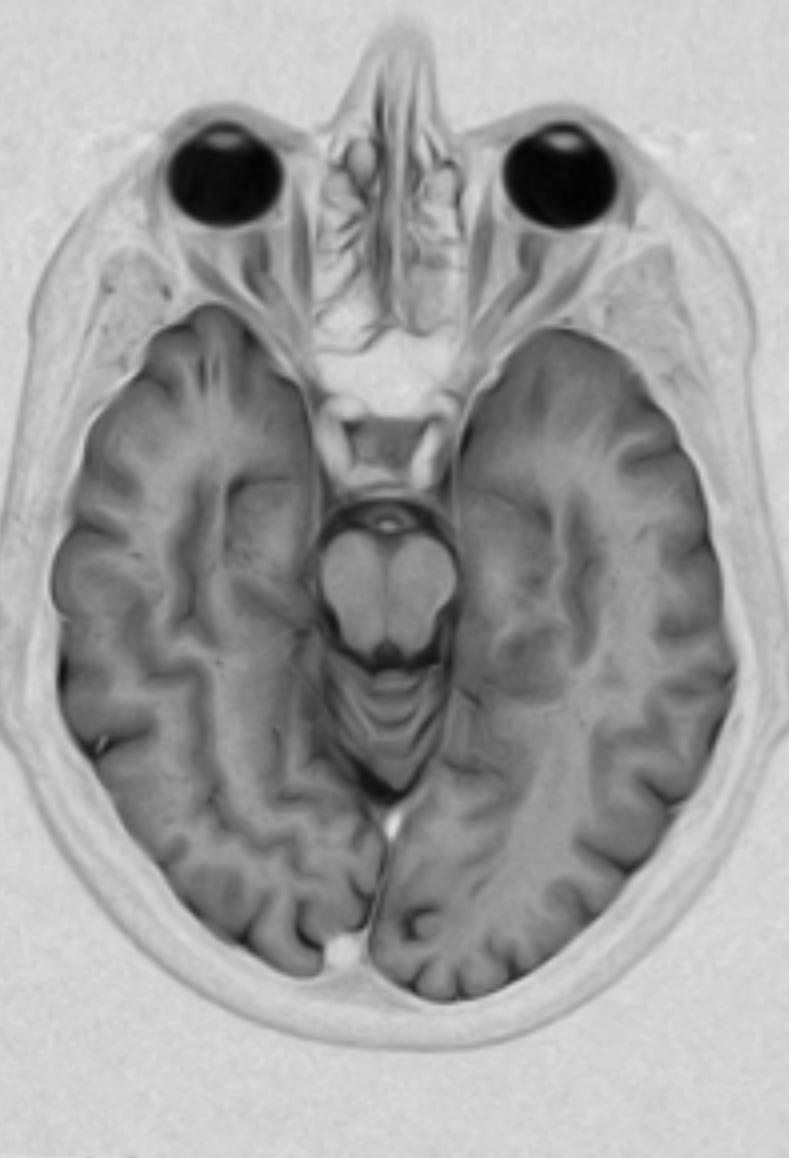
- attribute values associated w/ cells
- cell contains value from continuous domain
 - eg temperature, pressure, wind velocity
- measured or simulated
- major concerns
 - sampling:
where attributes are measured
 - interpolation:
how to model attributes elsewhere
 - grid types



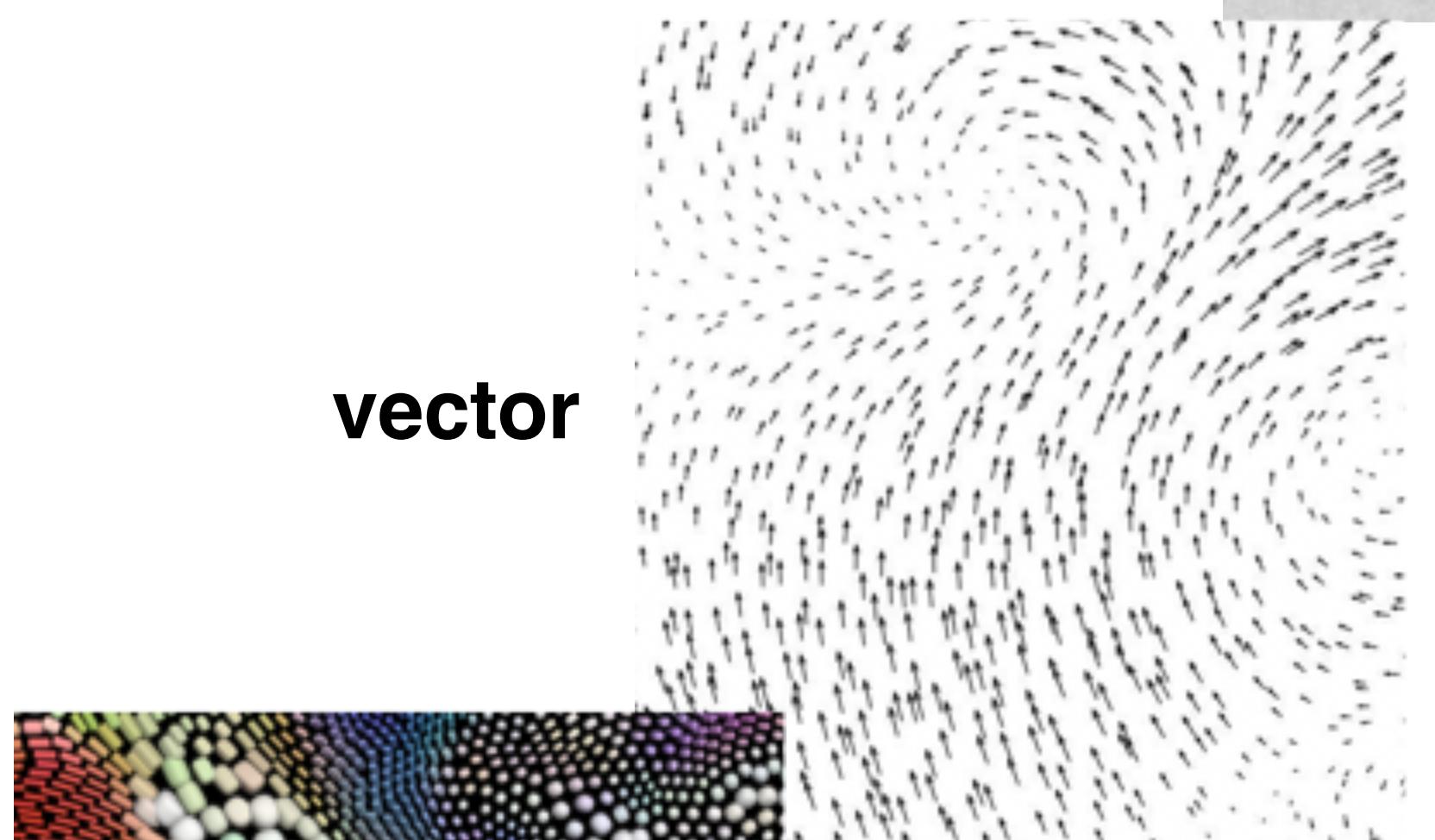
Spatial fields

- attribute values associated w/ cells
- cell contains value from continuous domain
 - eg temperature, pressure, wind velocity
- measured or simulated
- major concerns
 - sampling:
where attributes are measured
 - interpolation:
how to model attributes elsewhere
 - grid types
- major divisions
 - attributes per cell:
scalar (1), vector (2), tensor (many)

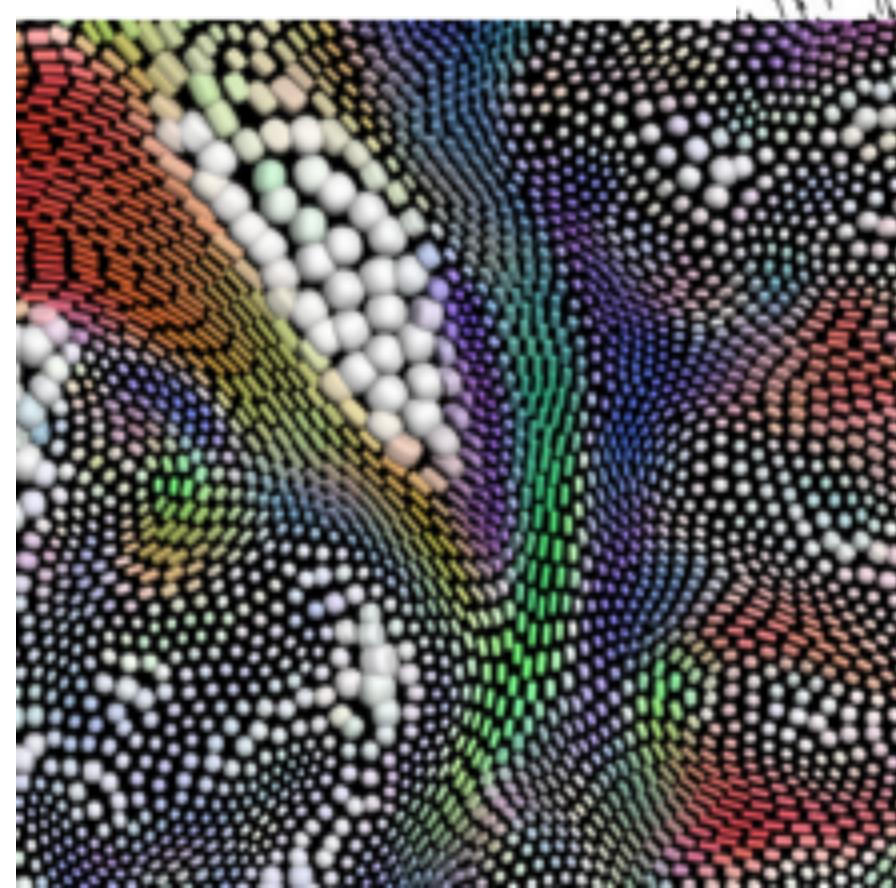
scalar



vector

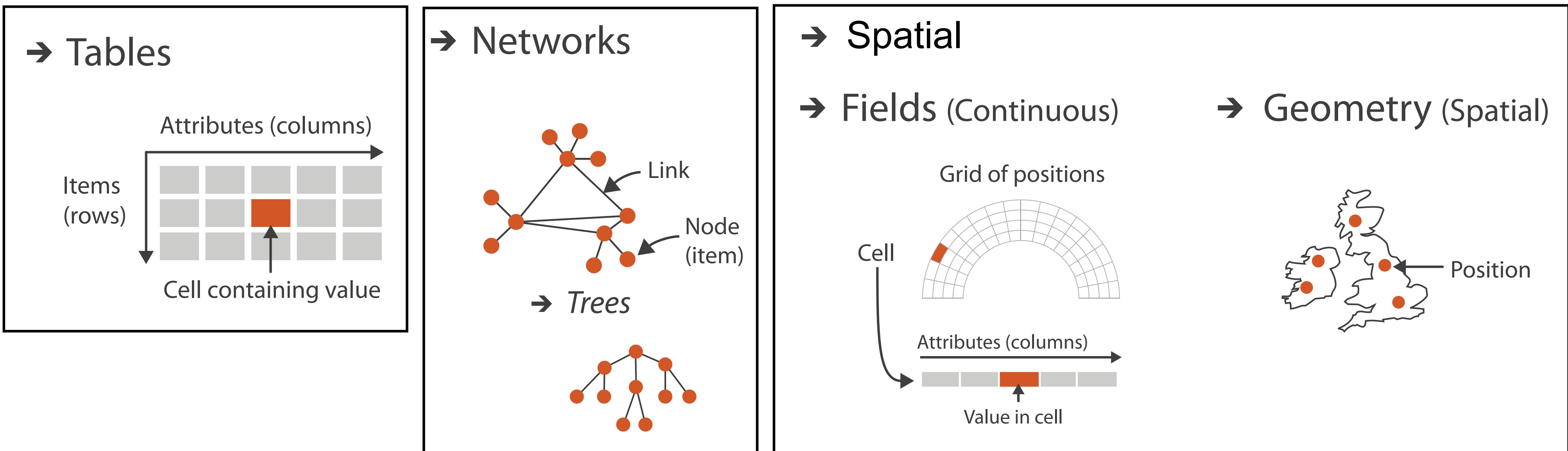


tensor



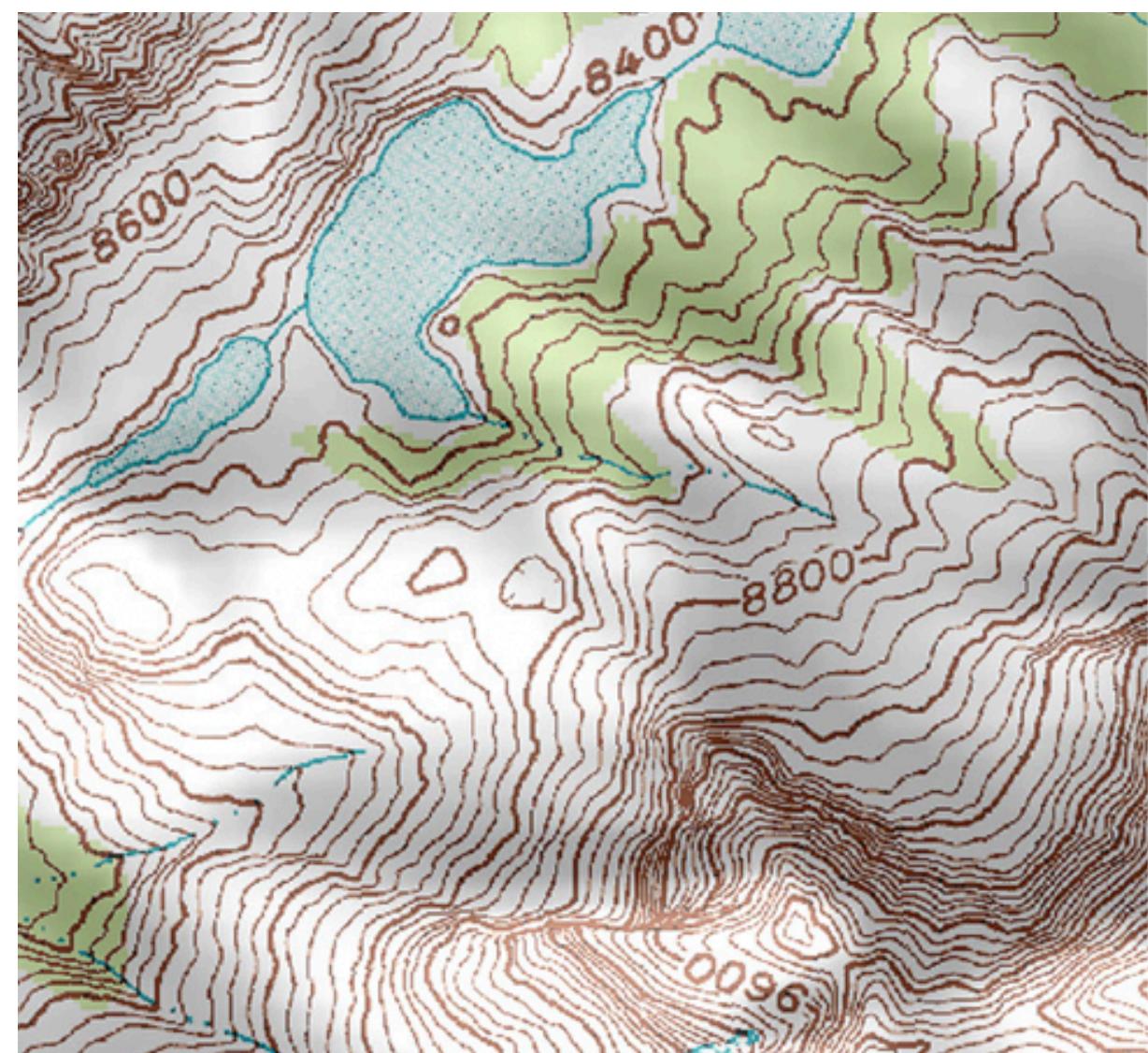
Dataset types

Tables	Networks & Trees	Fields	Geometry
Items	Items (nodes)	Grids	Items
Attributes	Links	Positions	Positions
	Attributes	Attributes	



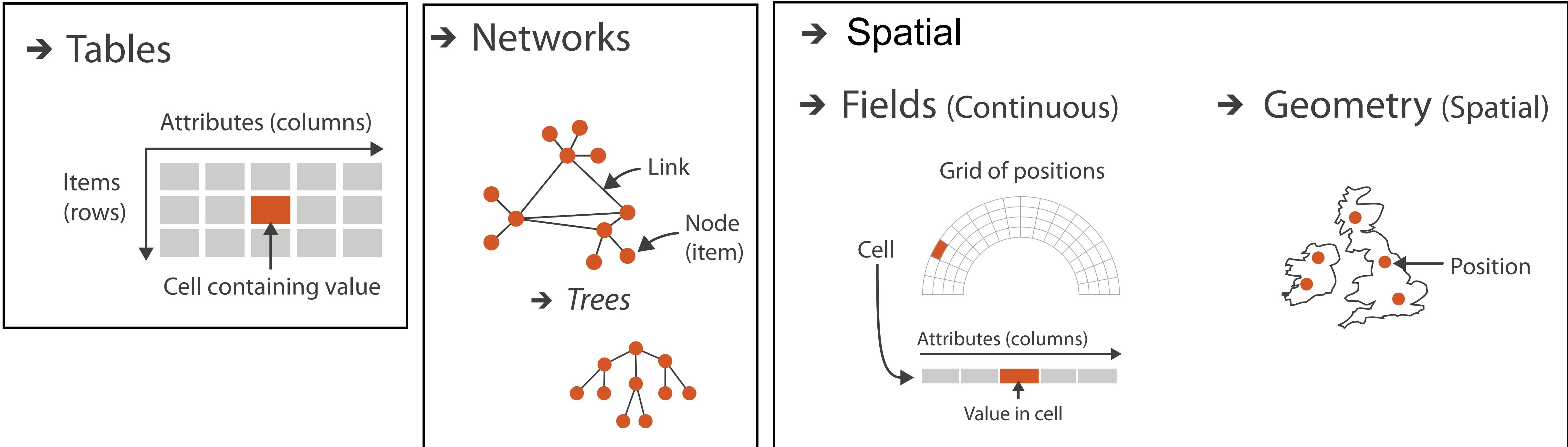
Geometry

- shape of items
- explicit spatial positions / regions
 - points, lines, curves, surfaces, volumes
- boundary between computer graphics and visualization
 - graphics: geometry taken as given
 - vis: geometry is result of a design decision



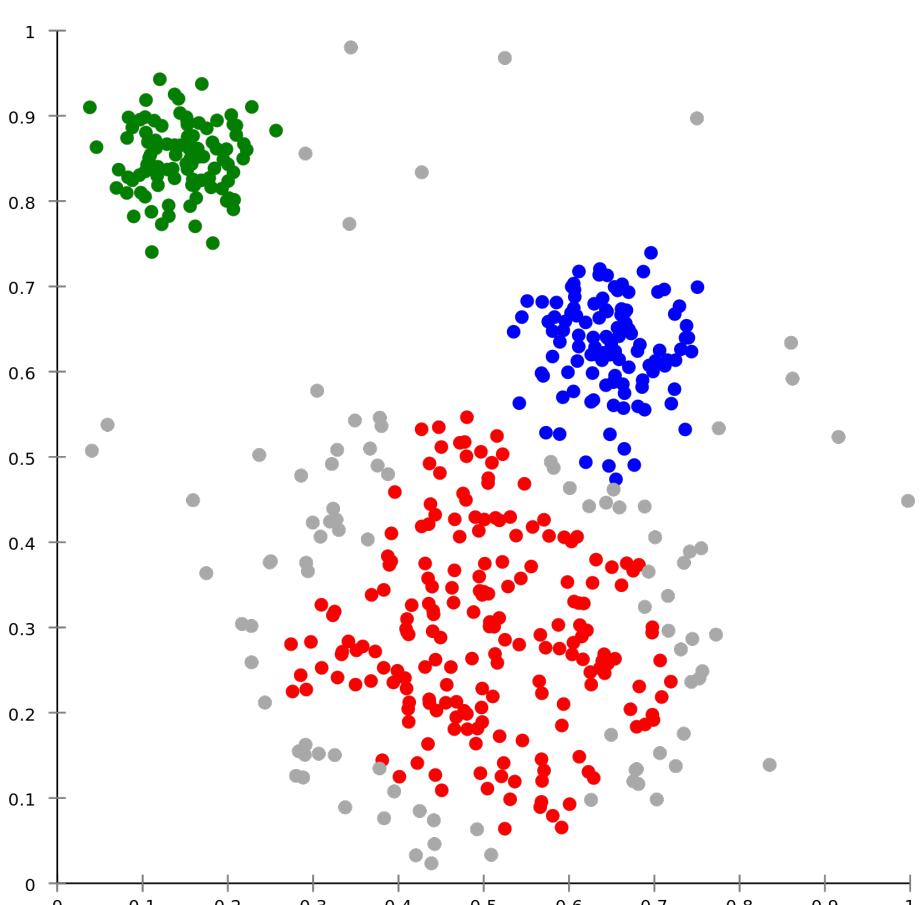
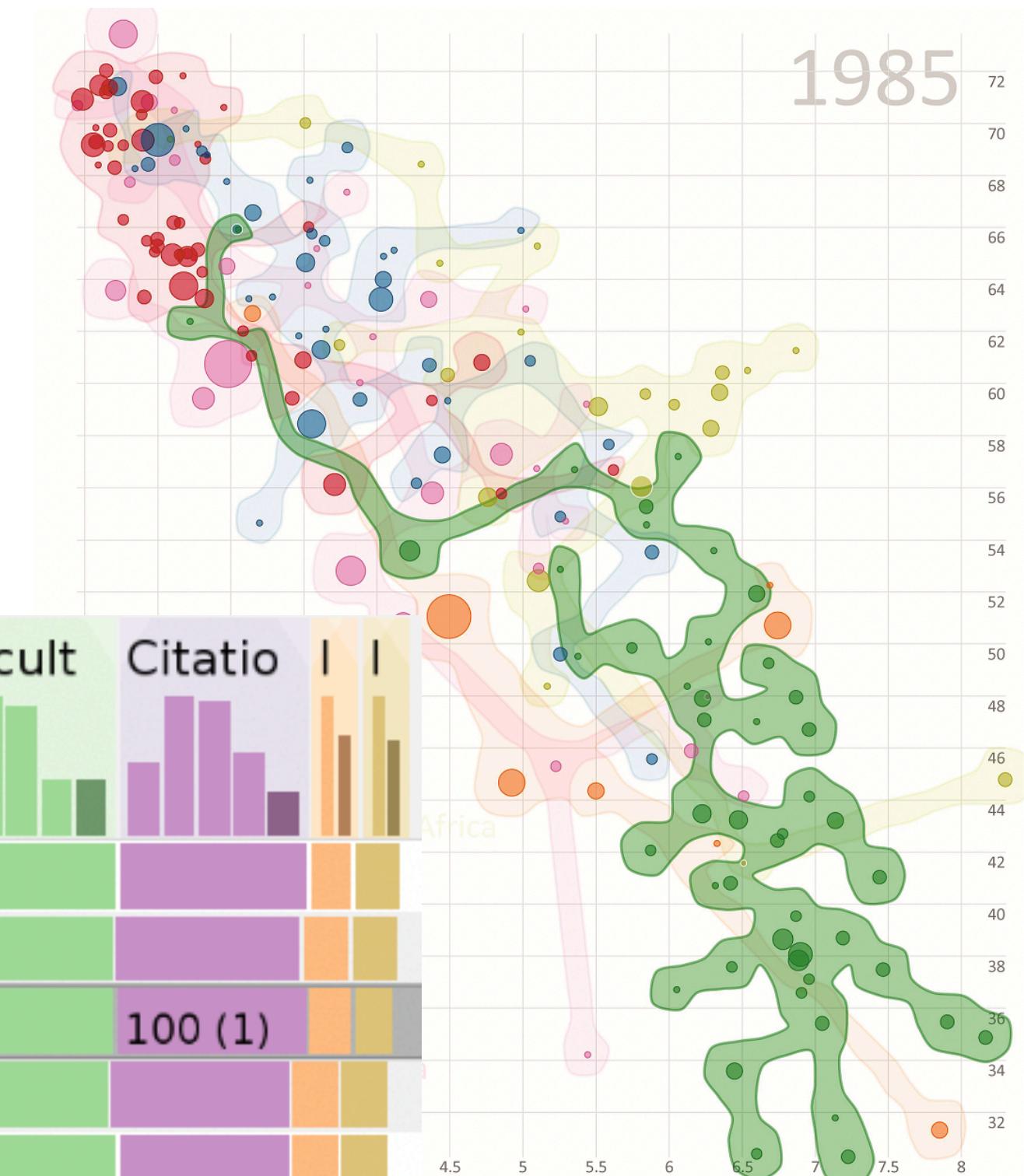
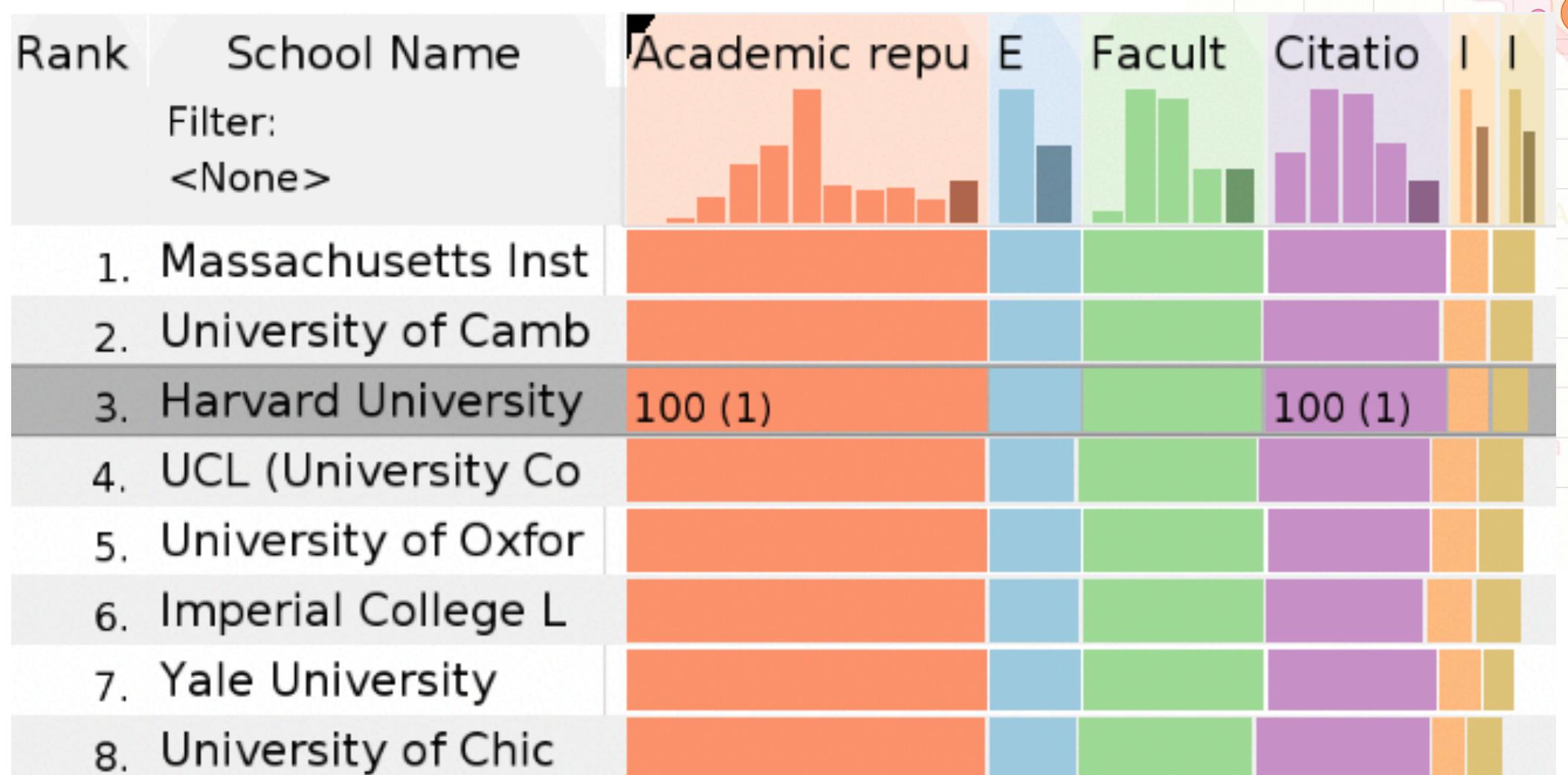
Dataset types

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



Collections

- how we group items
- sets
 - unique items, unordered
- lists
 - ordered, duplicates possible
- clusters
 - groups of similar items



→ Data and Dataset Types

Tables

Items

Attributes

Networks & Trees

Items (nodes)

Links

Attributes

Fields

Grids

Positions

Attributes

Geometry

Items

Positions

Clusters, Sets, Lists

Items

→ Data Types

→ Items

→ Attributes

→ Links

→ Positions

→ Grids

Attribute types

- which classes of values & measurements?
- categorical (nominal)
 - compare equality
 - no implicit ordering
- ordered
 - ordinal
 - less/greater than defined
 - quantitative
 - meaningful magnitude
 - arithmetic possible

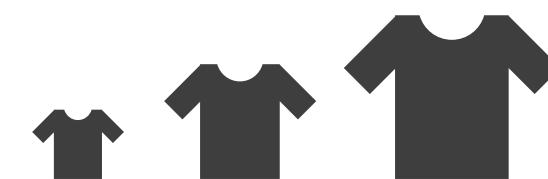
→ Attribute Types

→ Categorical

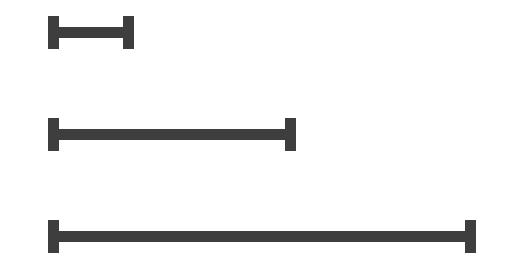


→ Ordered

→ *Ordinal*



→ *Quantitative*



categorical
ordinal
quantitative

A	B	C	D	E	F	G
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date	
3	10/14/06	5-Low	Large Box	0.8	10/21/06	
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08	
32	7/16/07	2-High	Small Pack	0.79	7/17/07	
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07	
32	7/16/07	2-High	Medium Box	0.6	7/18/07	
32	7/16/07	2-High	Medium Box	0.65	7/18/07	
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07	
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07	
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07	
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07	
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05	
69	6/4/05	4-Not Specified	Small Pack	0.44	6/6/05	
69	6/4/05	4-Not Specified	Wrap Bag	0.6	6/6/05	
70	12/18/06	5-Low	Small Box	0.59	12/23/06	
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06	
96	4/17/05	2-High	Small Box	0.55	4/19/05	
97	1/29/06	3-Medium	Small Box	0.38	1/30/06	
129	11/19/08	5-Low	Small Box	0.37	11/28/08	
130	5/8/08	2-High	Small Box	0.37	5/9/08	
130	5/8/08	2-High	Medium Box	0.38	5/10/08	
130	5/8/08	2-High	Small Box	0.6	5/11/08	
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06	
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06	
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08	
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07	
166	9/12/07	2-High	Small Box	0.55	9/14/07	
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06	

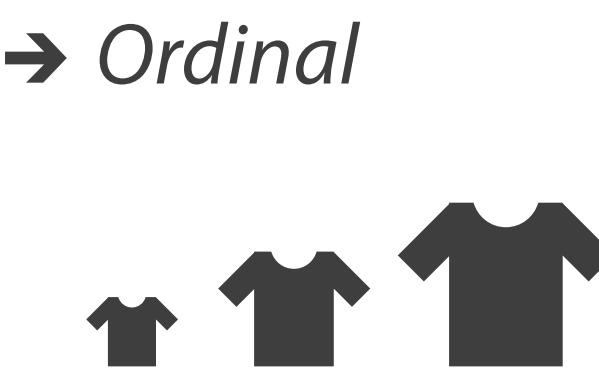
Other data concerns

Attribute Types

→ Categorical

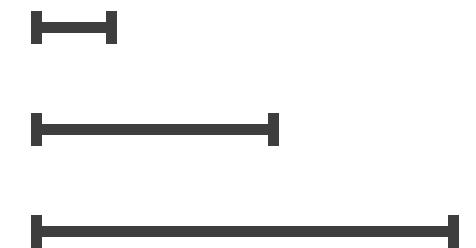


→ Ordered



→ *Ordinal*

→ *Quantitative*



Ordering Direction

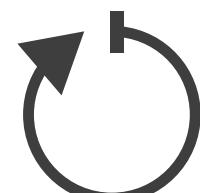
→ Sequential



→ Diverging



→ Cyclic



Dataset Availability

→ Static



→ Dynamic



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 vs conceptual models

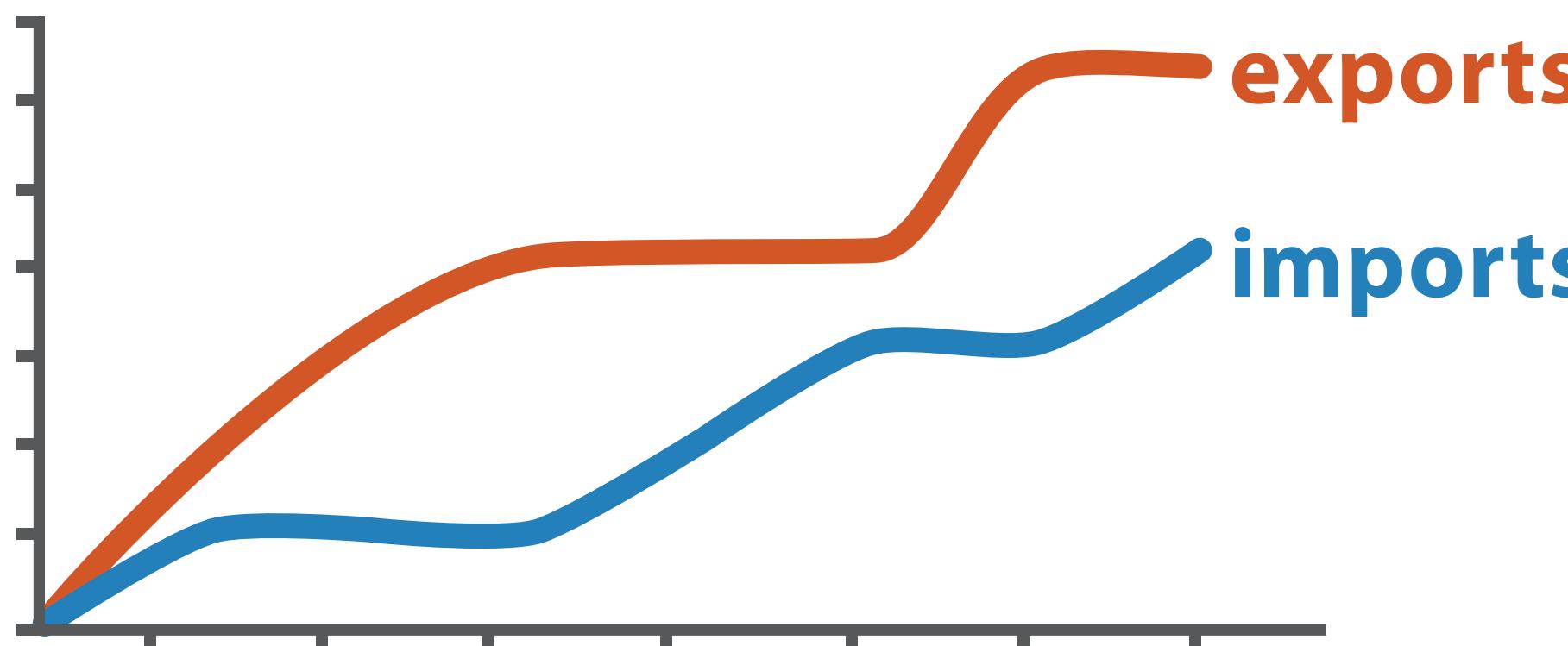
- data model
 - mathematical abstraction
 - sets with operations, eg floats with * / - +
 - variable data types in programming languages
 - conceptual model
 - mental construction (semantics)
 - supports reasoning
 - typically based on understanding of tasks [stay tuned!]
- data abstraction process relies on conceptual model
 - for transforming data if needed

Data vs conceptual model, example

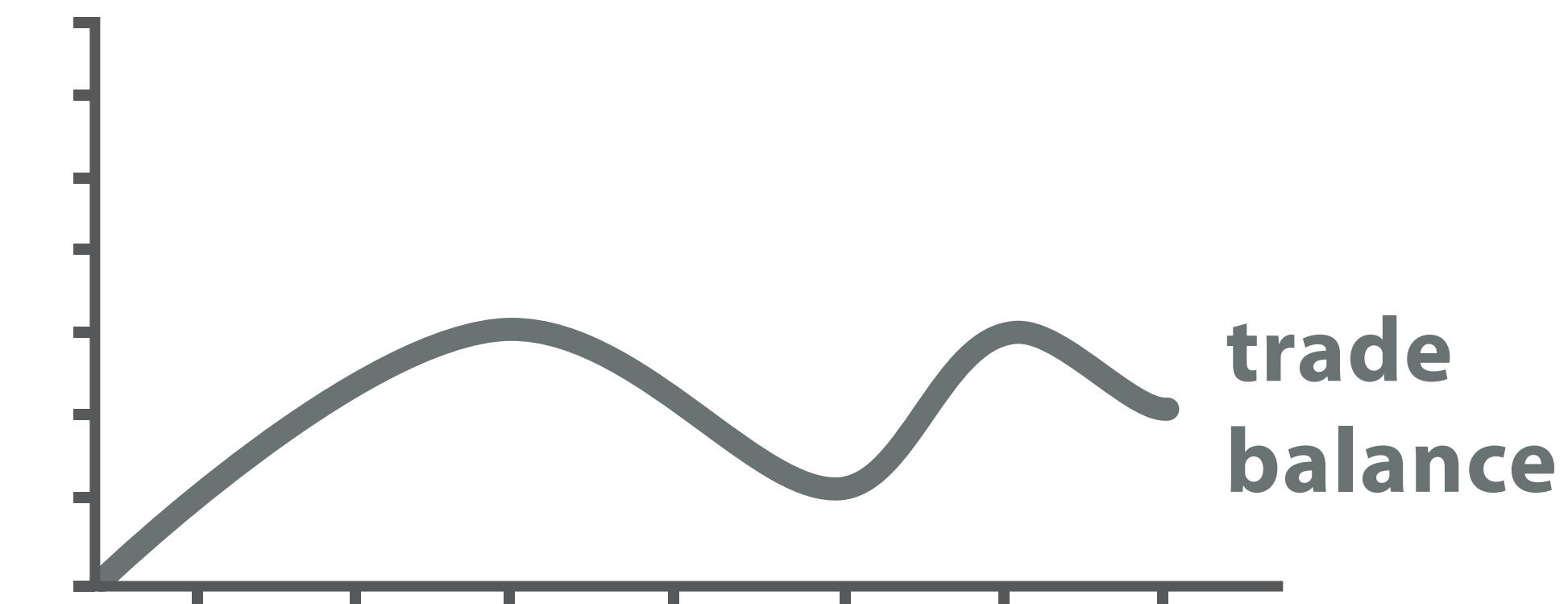
- data model: floats
 - 32.52, 54.06, -14.35, ...
- conceptual model
 - temperature
- multiple possible data abstractions
 - continuous to 2 significant figures: quantitative
 - task: forecasting the weather
 - hot, warm, cold: ordinal
 - task: deciding if bath water is ready
 - above freezing, below freezing: categorical
 - task: decide if I should leave the house today

Derived attributes

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



Original Data



$$\text{trade balance} = \text{exports} - \text{imports}$$

Derived Data

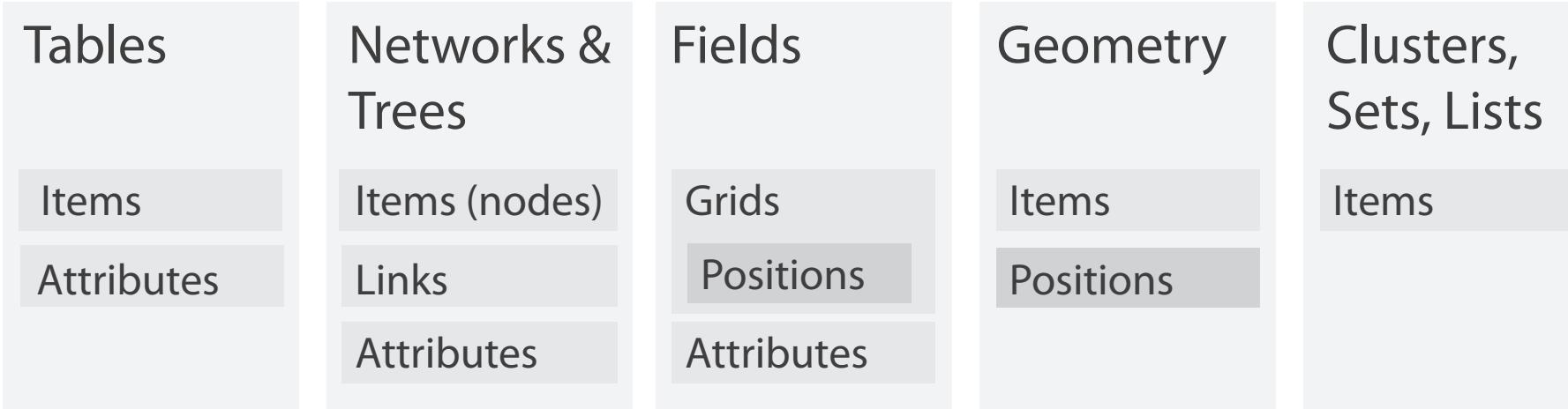
What?

Datasets

→ Data Types

- Items
- Attributes
- Links
- Positions
- Grids

→ Data and Dataset Types



What?

Why?

How?

Attributes

→ Attribute Types

- Categorical



- Ordered

- Ordinal



- Quantitative



→ Ordering Direction

- Sequential



- Diverging

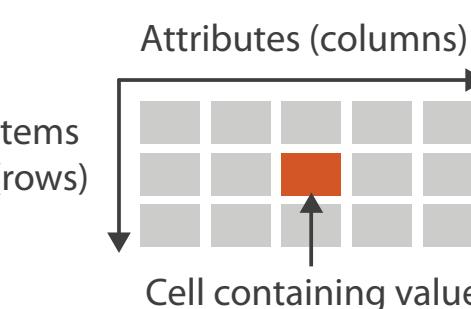


- Cyclic

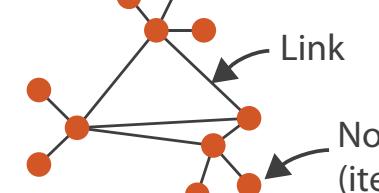


→ Dataset Types

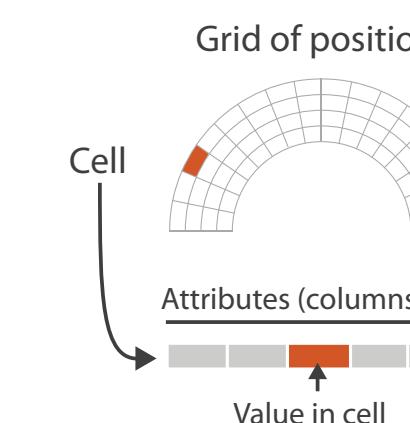
→ Tables



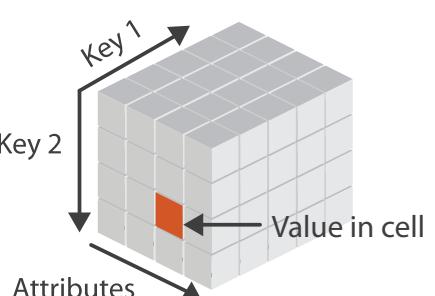
→ Networks



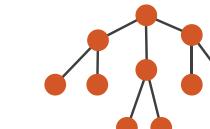
→ Fields (Continuous)



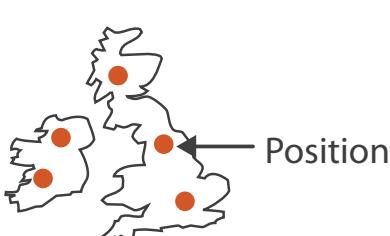
→ Multidimensional Table



→ Trees



→ Geometry (Spatial)



→ Dataset Availability

→ Static



→ Dynamic



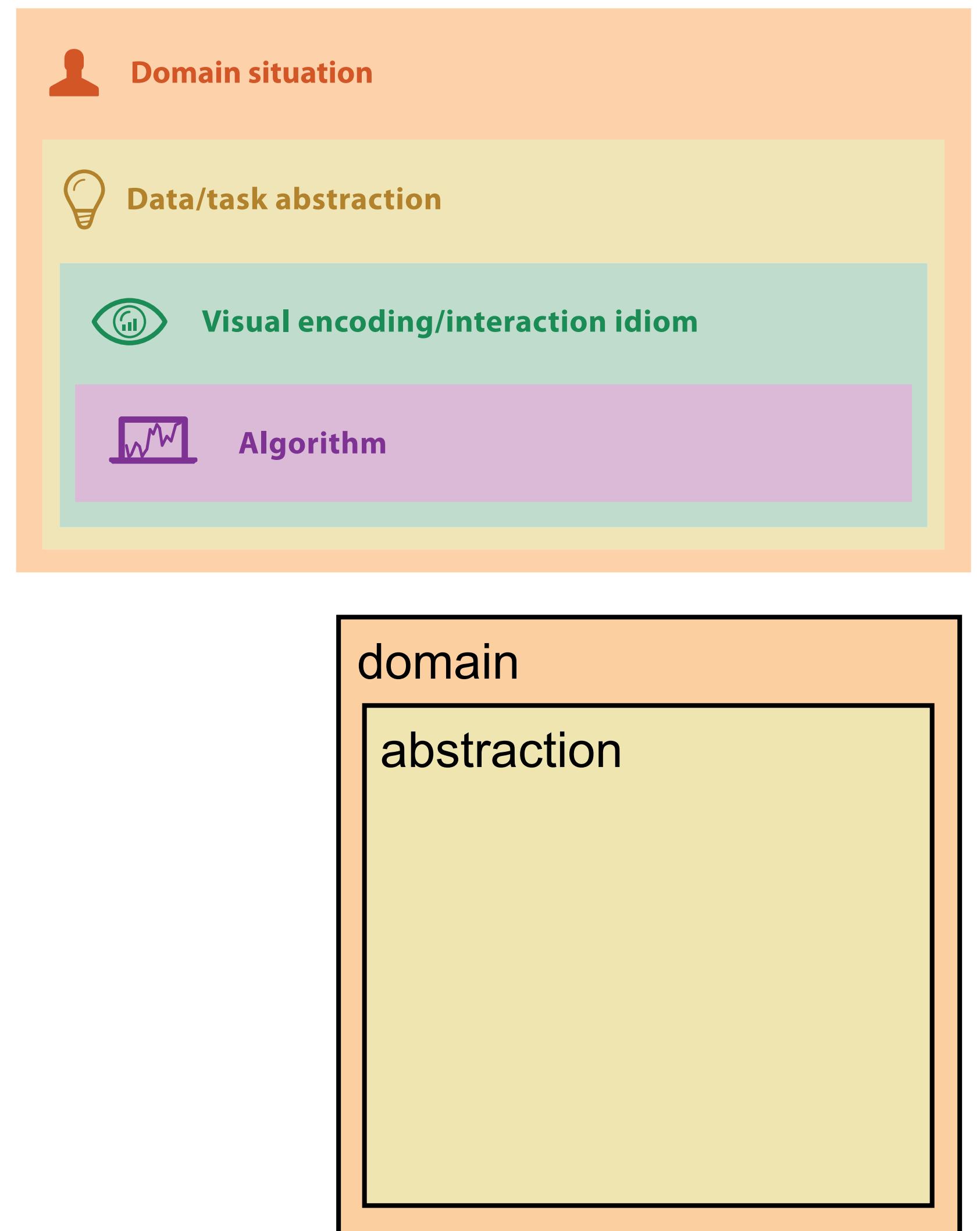
Reading Material

[dv3] Chapter 2 - What: Data Abstraction

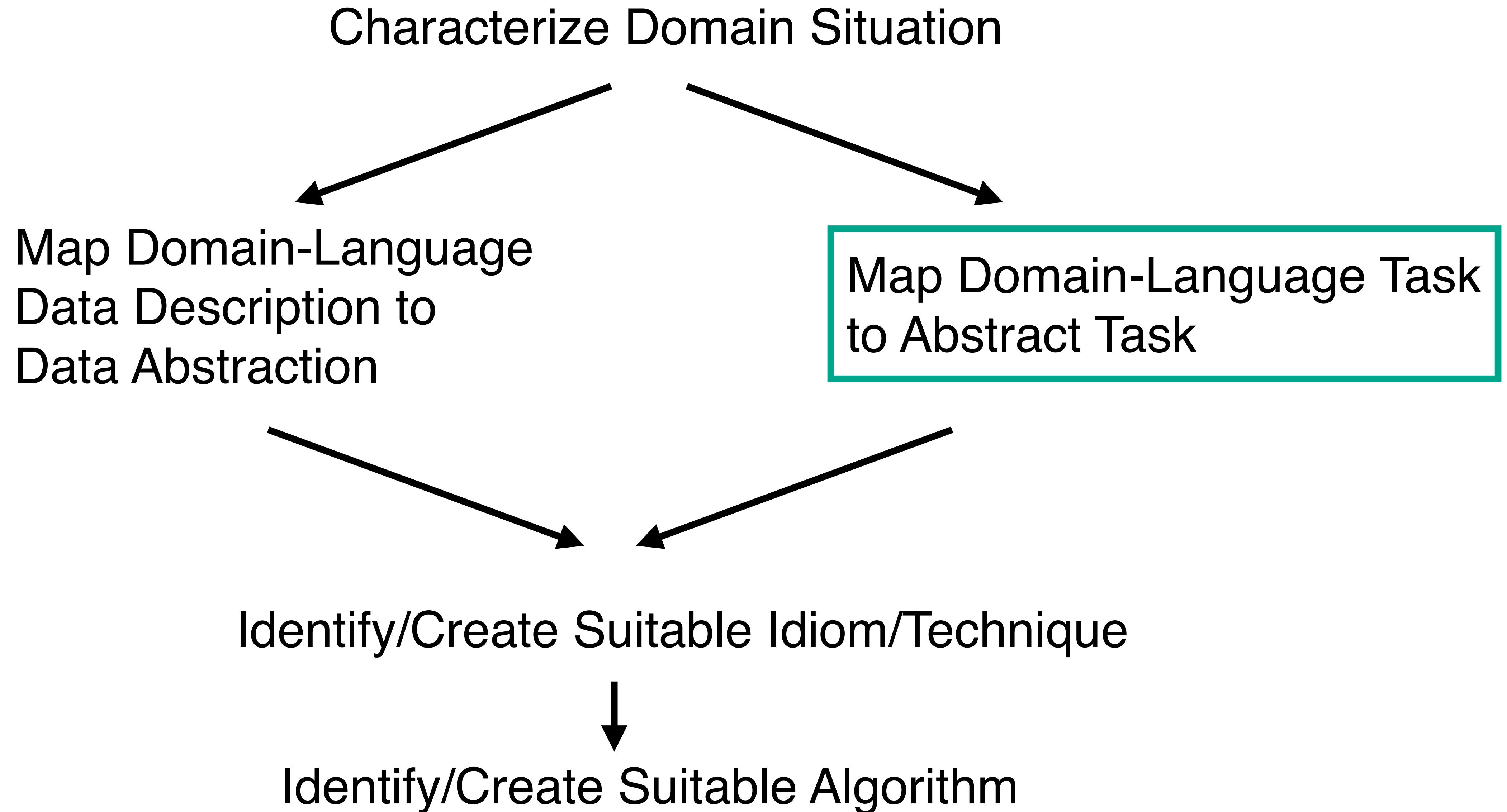
Task Abstraction

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



Task abstraction

Actions and targets

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

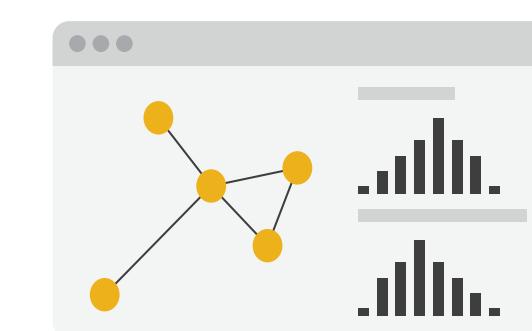
→ Analyze

→ Consume

→ Discover



→ Present



→ Enjoy

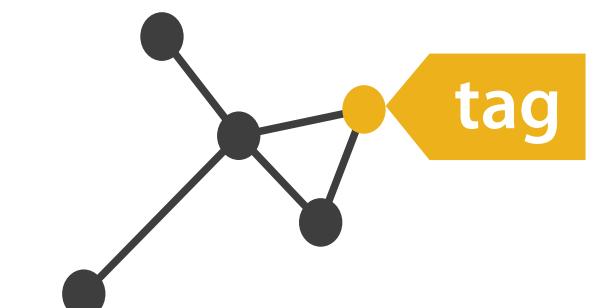


- **produce**

- annotate, record
- derive
 - crucial design choice

→ Produce

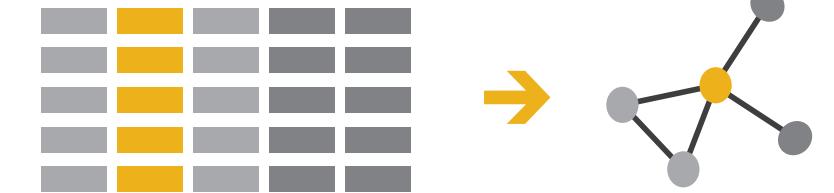
→ Annotate



→ Record



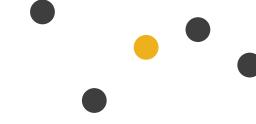
→ Derive



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**

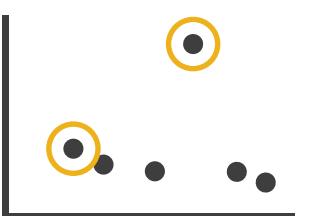
	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>

Actions: Query

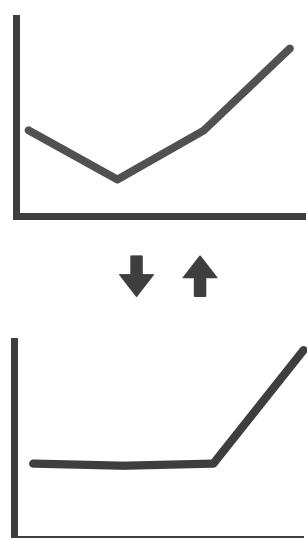
- **how much of the data matters?**
 - one: identify
 - some: compare
 - all: summarize

➔ **Query**

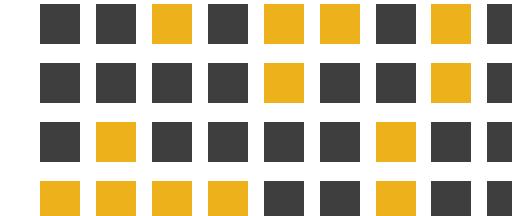
➔ Identify



➔ Compare



➔ Summarize



Actions

Actions

- **independent choices for each of these three levels**
 - analyze, search, query
 - mix and match

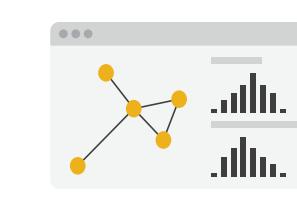
→ Analyze

→ Consume

→ Discover



→ Present

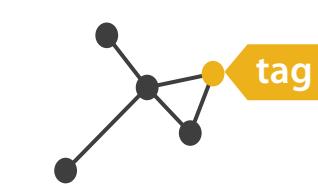


→ Enjoy

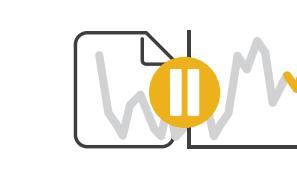


→ Produce

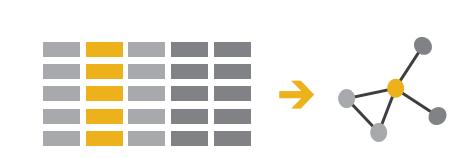
→ Annotate



→ Record



→ Derive

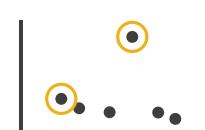


→ Search

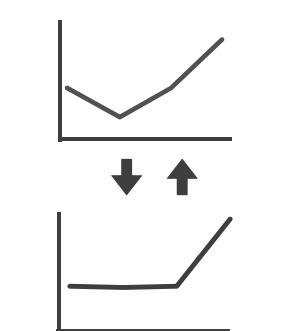
	Target known	Target unknown
Location known	•..•• <i>Lookup</i>	•..•• <i>Browse</i>
Location unknown	◁•○•▷ <i>Locate</i>	◁•○•▷ <i>Explore</i>

→ Query

→ Identify



→ Compare



→ Summarize

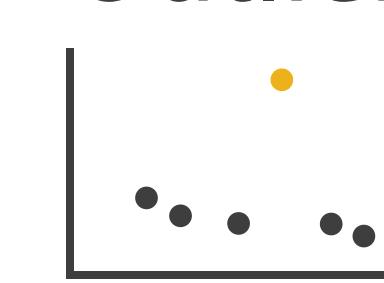
Task abstraction: Targets

→ All Data

→ Trends



→ Outliers



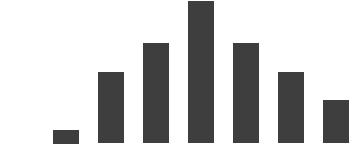
→ Features



→ Attributes

→ One

→ Distribution

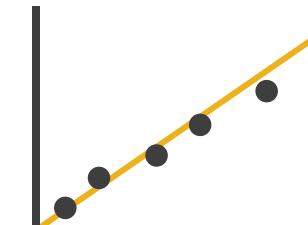


→ Many

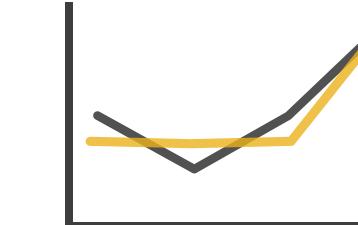
→ Dependency



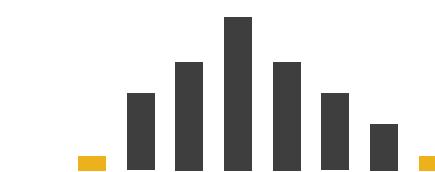
→ Correlation



→ Similarity

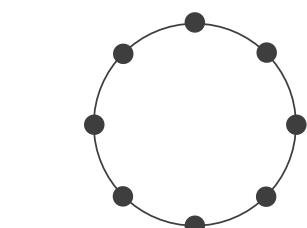
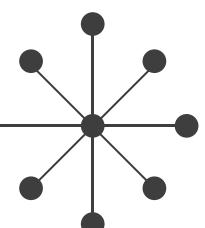
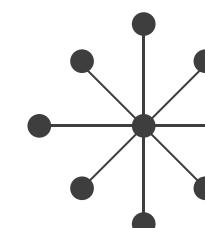
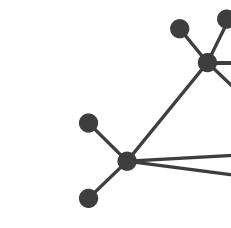


→ Extremes

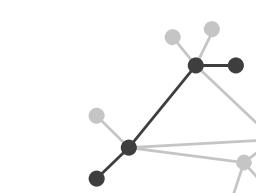


→ Network Data

→ Topology

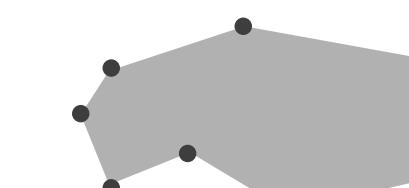


→ Paths



→ Spatial Data

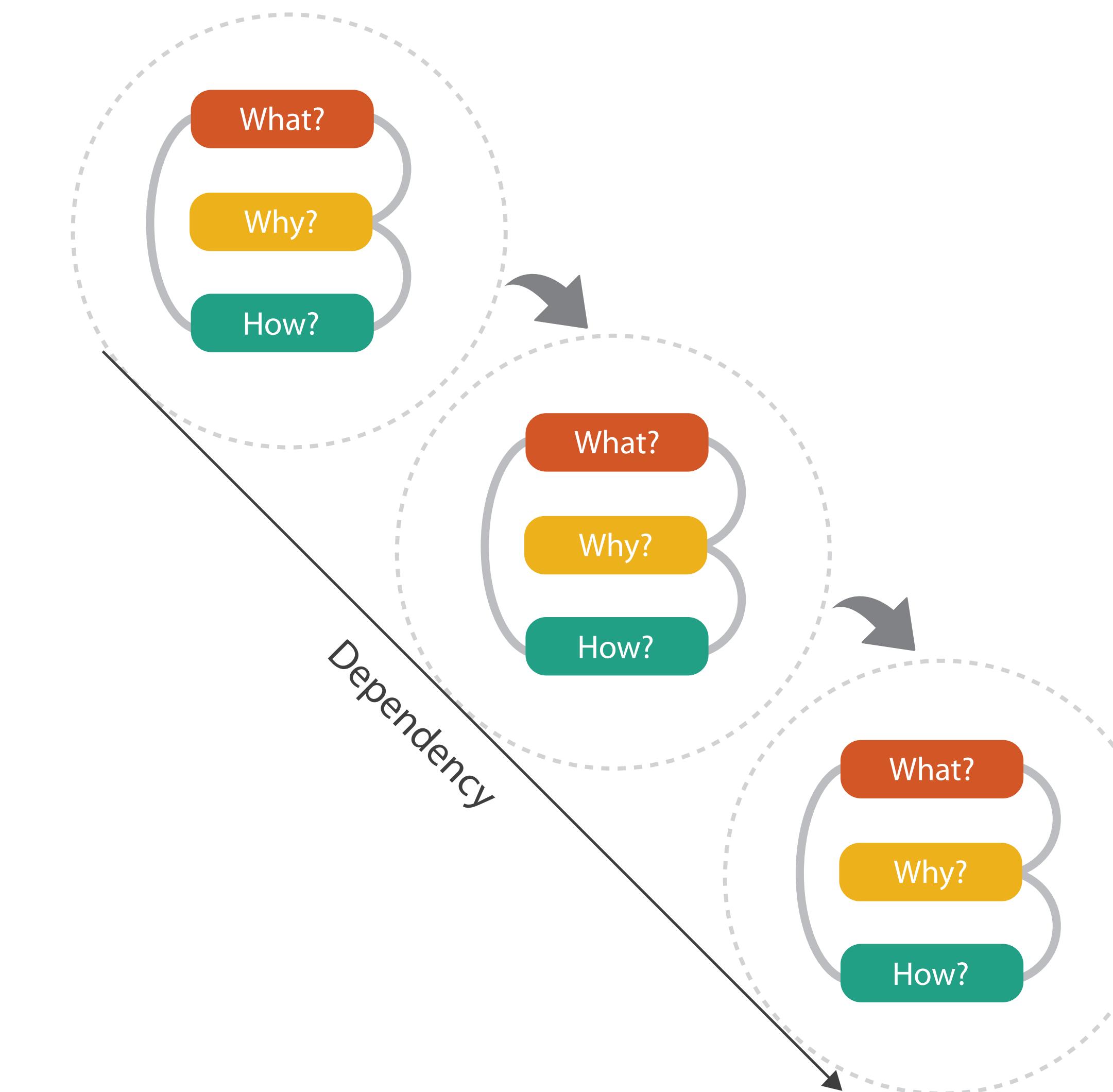
→ Shape

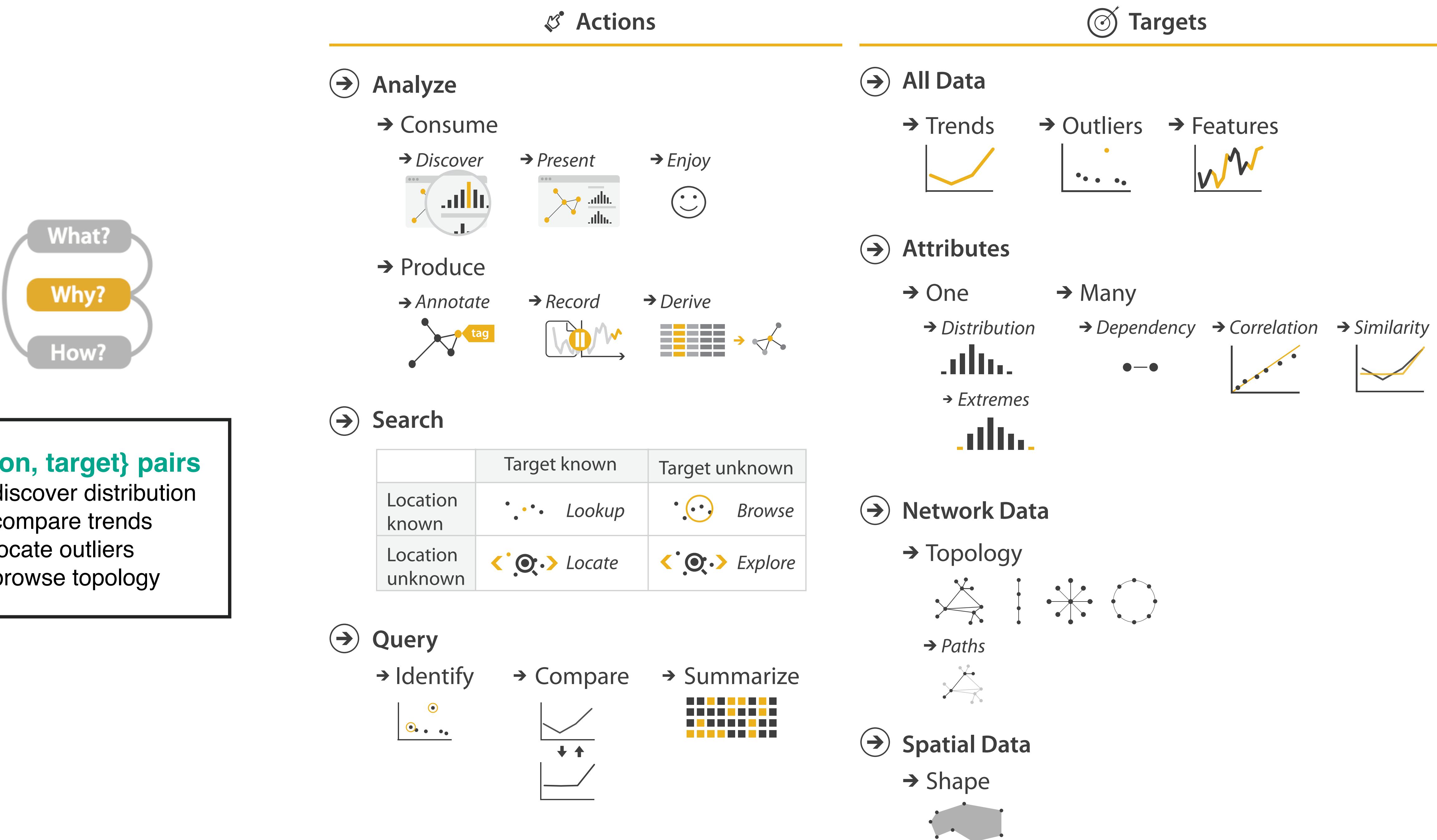


Abstraction

- 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





Reading Material

[dv3] Chapter 3 - Why: Task Abstraction

Questions?

:::::::



@rschifan



rossano.schifanella@unito.it



<http://www.di.unito.it/~schifane>