UNIVERSITÀ DI TORINO

Analisi e Visualizzazione di Reti Complesse

DV02 Nested Model
Data Abstraction
Task Abstraction

Prof. Rossano Schifanella

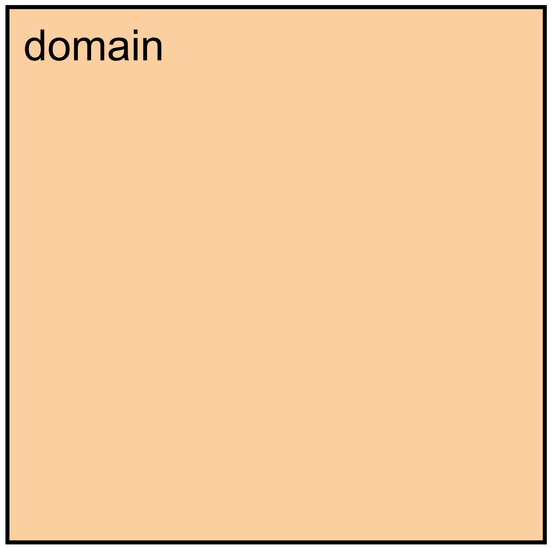
Nested Model

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

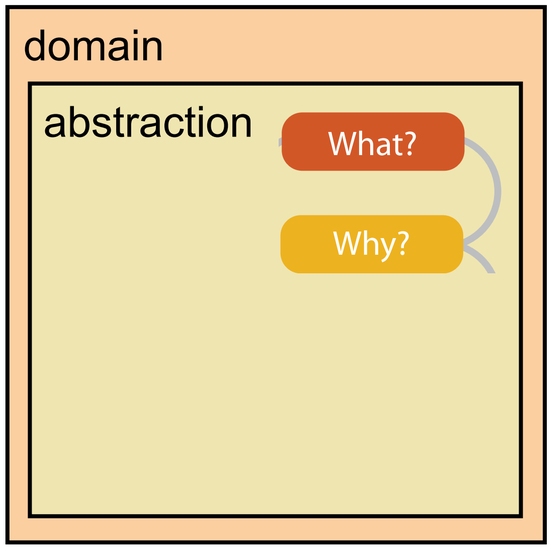
Analysis framework: Four levels, three questions

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



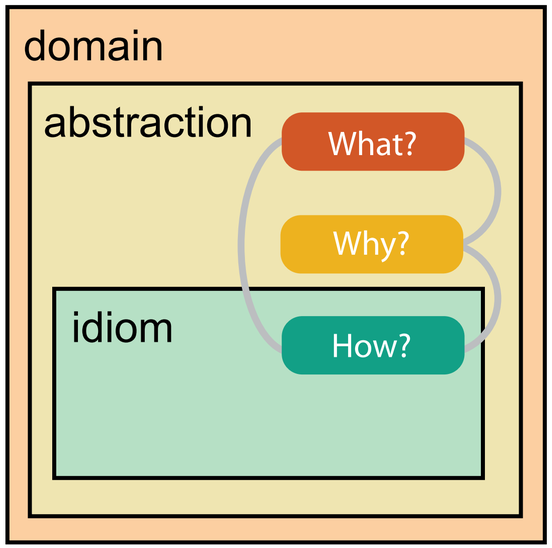
Analysis framework: Four levels, three questions

- **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



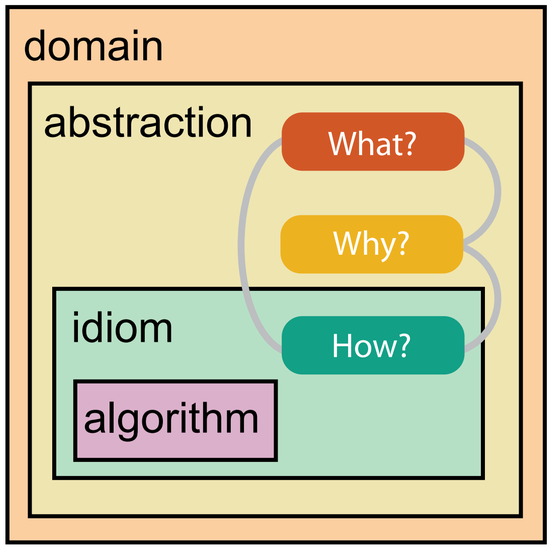
Analysis framework: Four levels, three questions

- **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



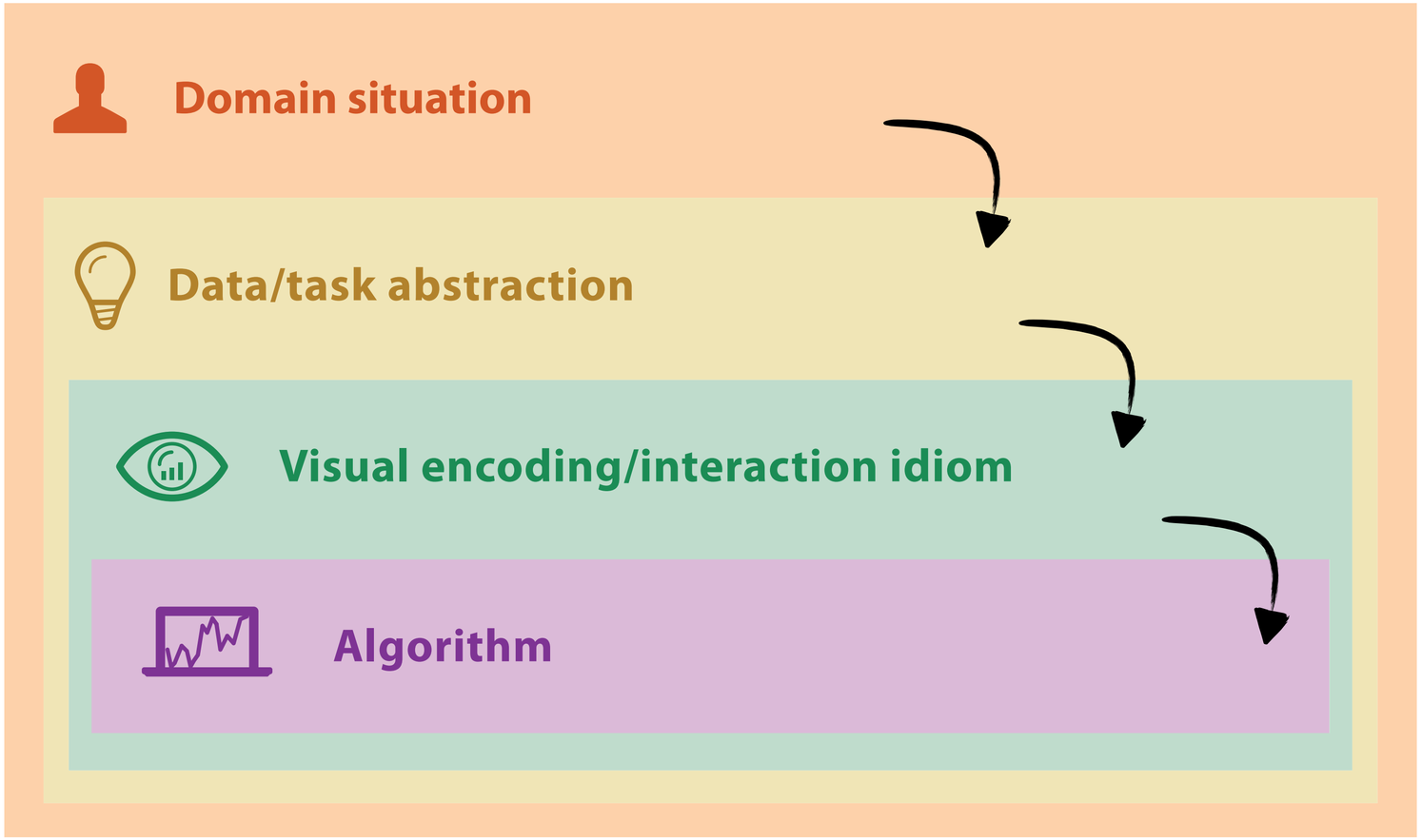
Analysis framework: Four levels, three questions

- **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

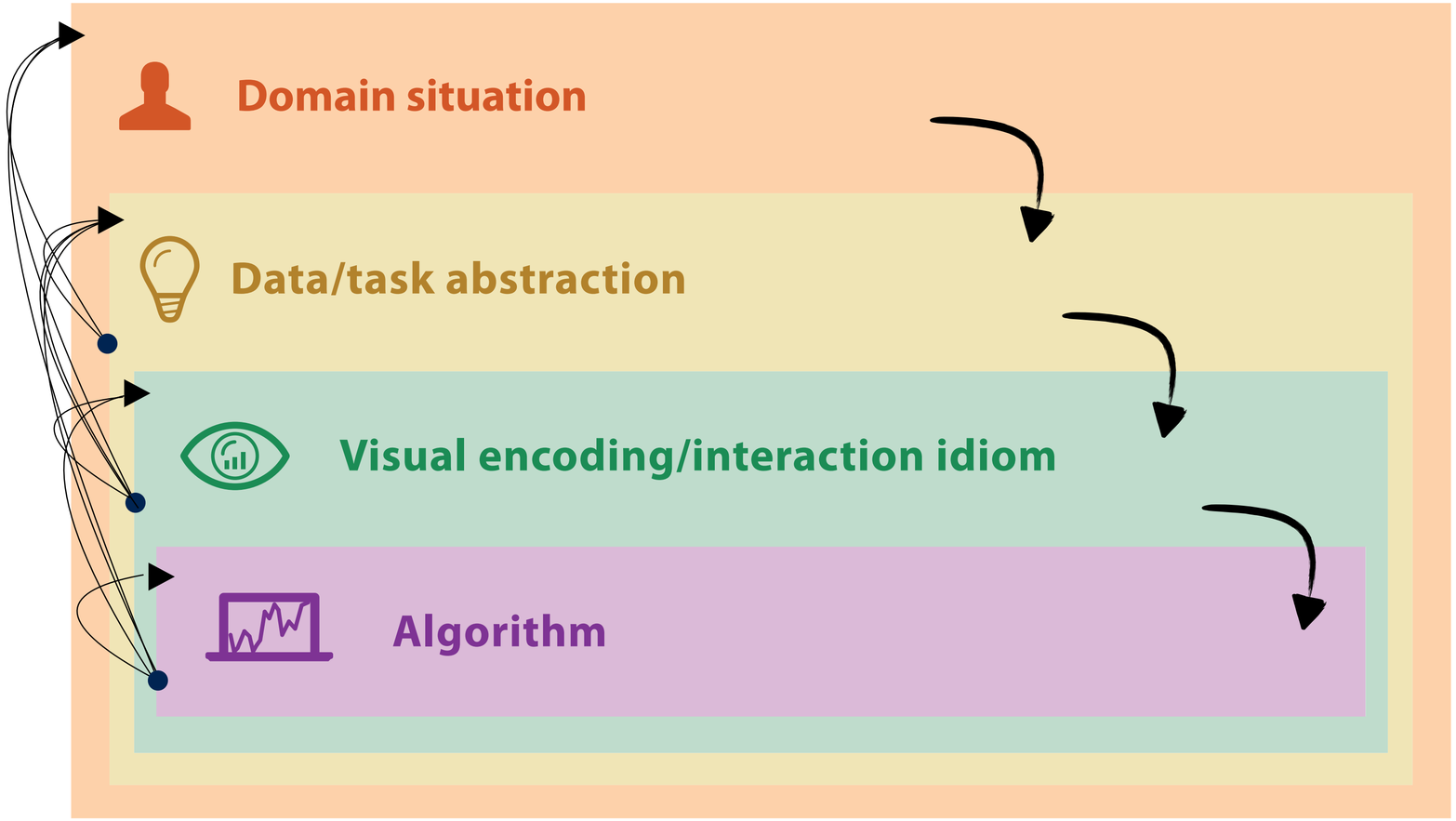
downstream: cascading effects



Nested model

downstream: cascading effects

upstream: iterative refinement



Why is validation difficult?

- different ways to get it wrong at each level

Why is validation difficult?

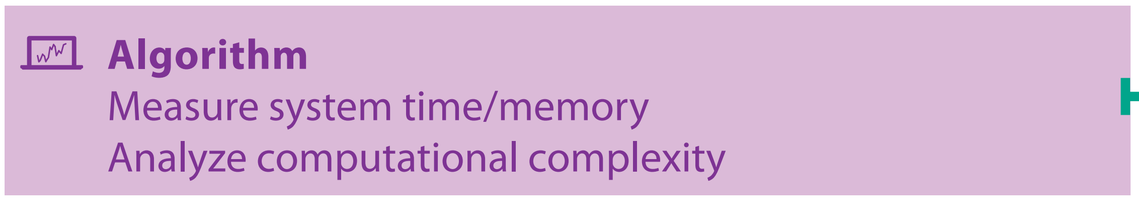
- solution: use methods from different fields at each level

Algorithm
Measure system time/memory
Analyze computational complexity

Why is validation difficult?

- solution: use methods from different fields at each level

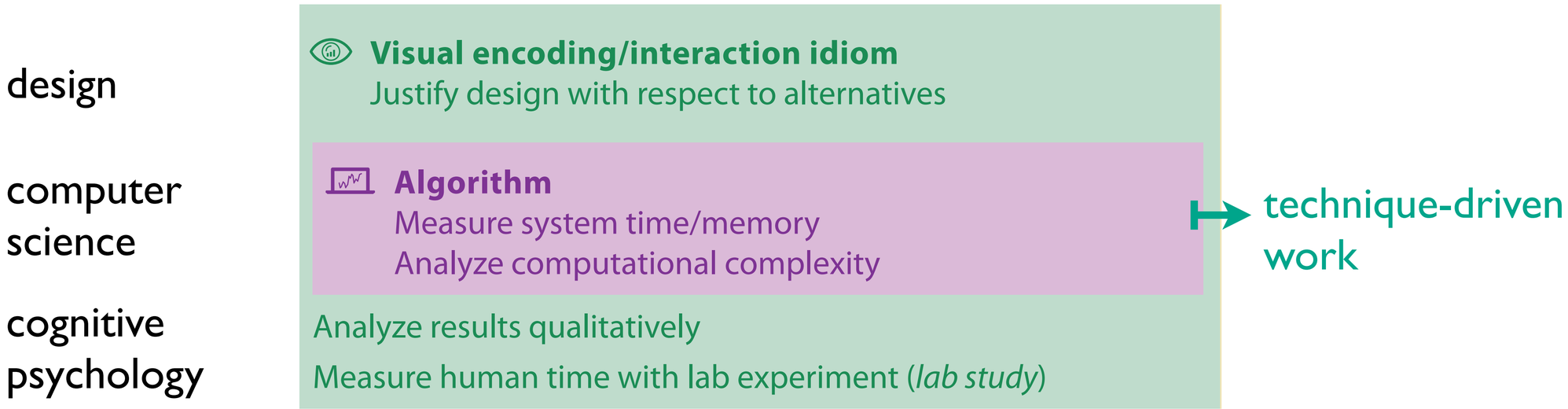
computer science



→ technique-driven work

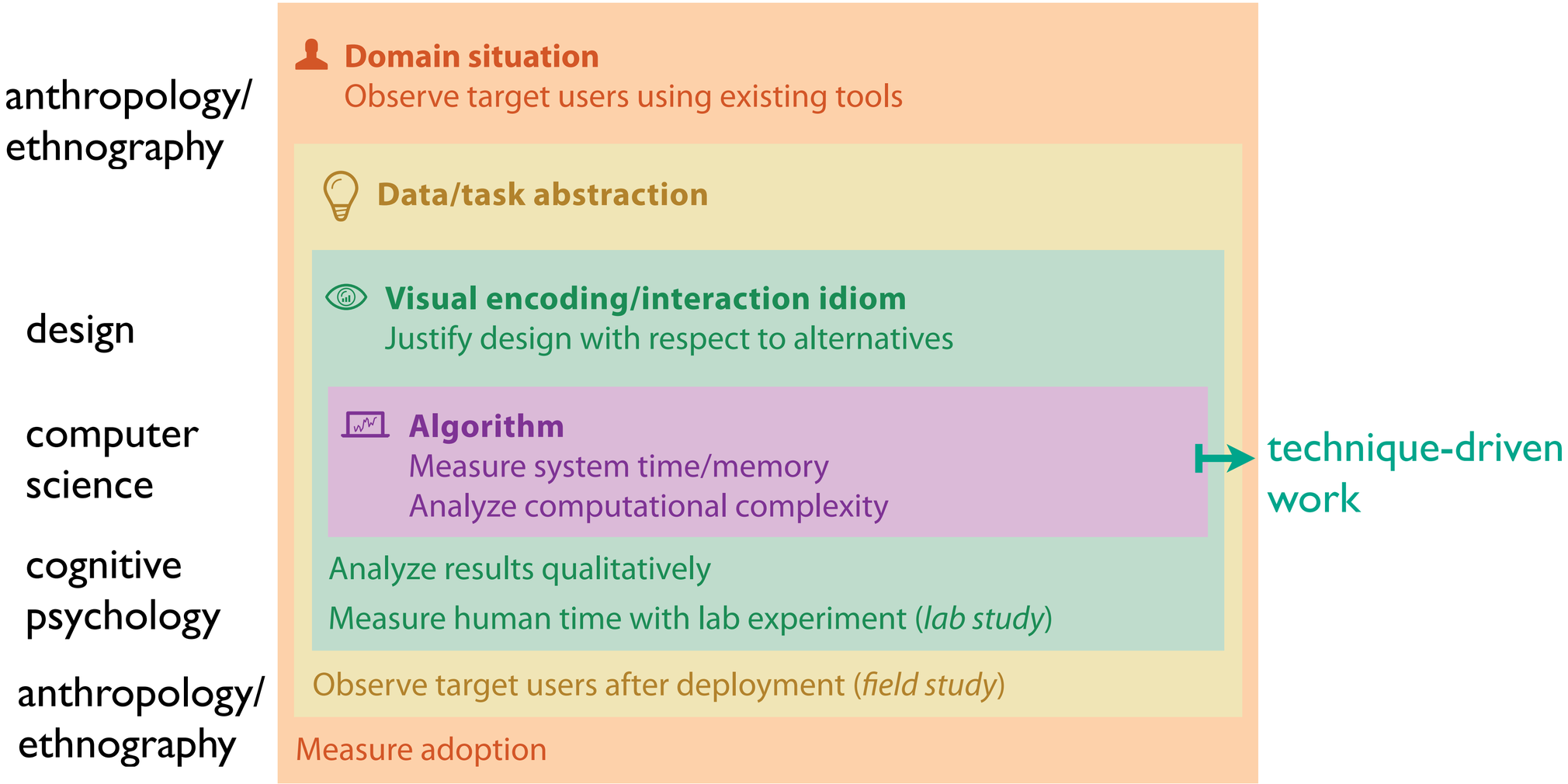
Why is validation difficult?

- solution: use methods from different fields at each level



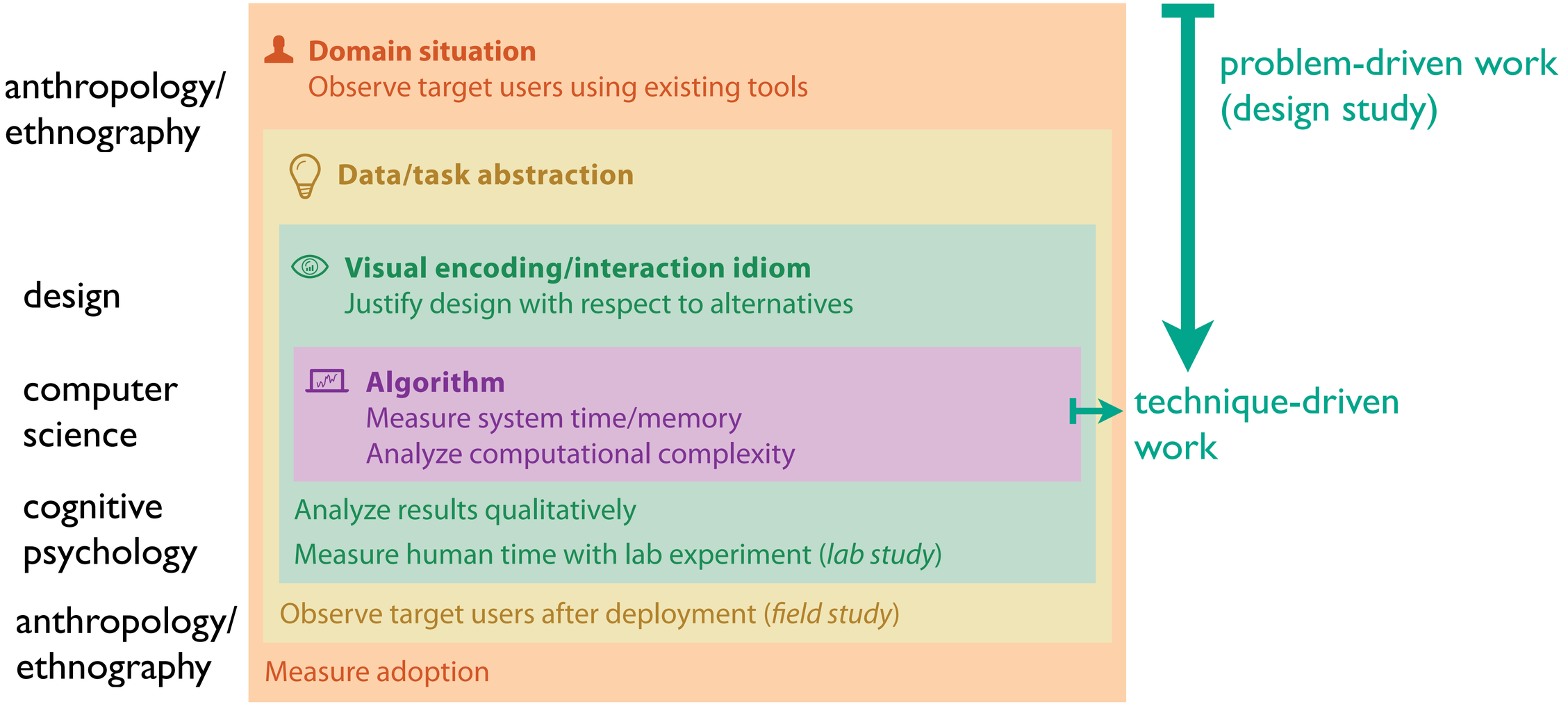
Why is validation difficult?

- solution: use methods from different fields at each level

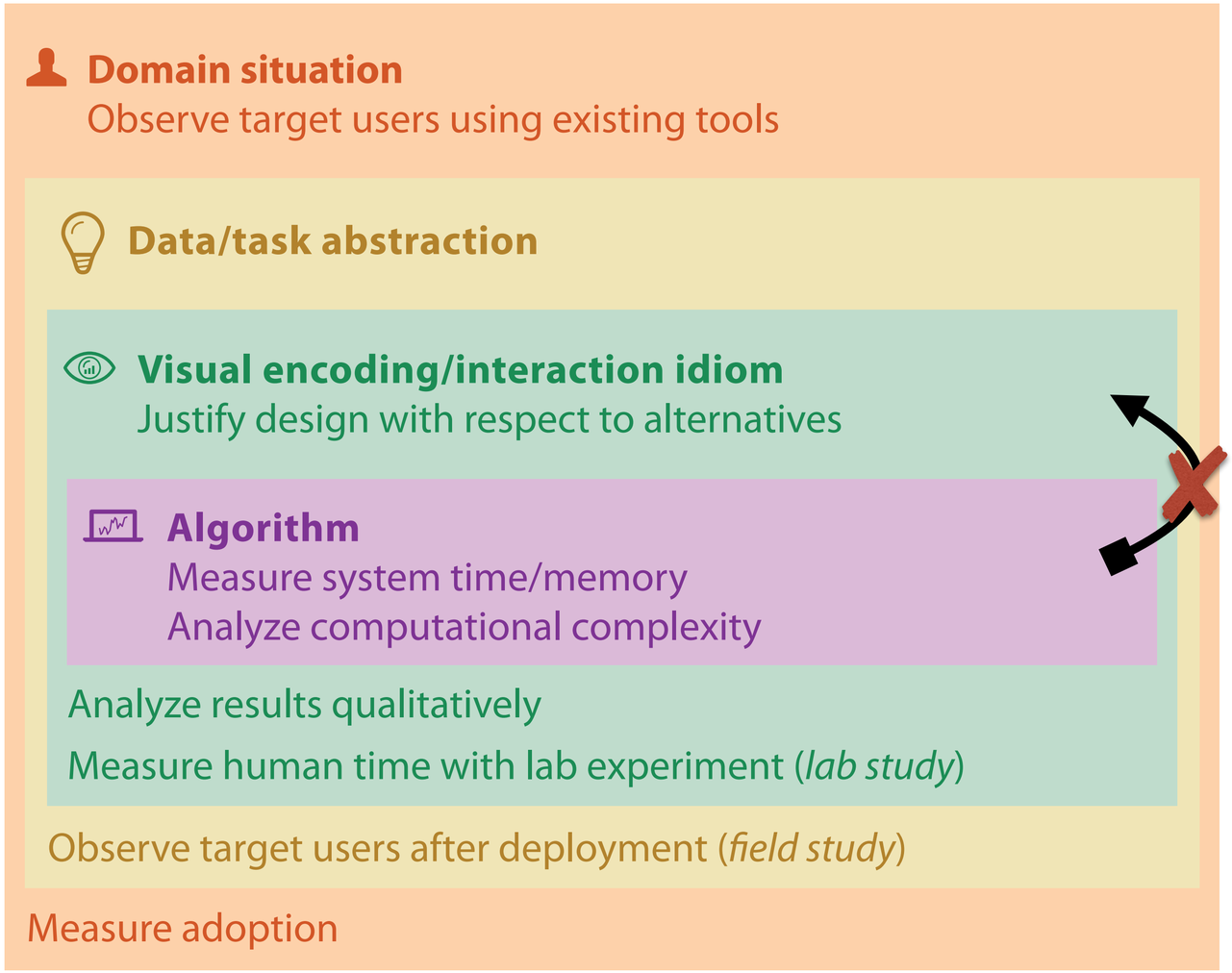


Why is validation difficult?

- solution: use methods from different fields at each level

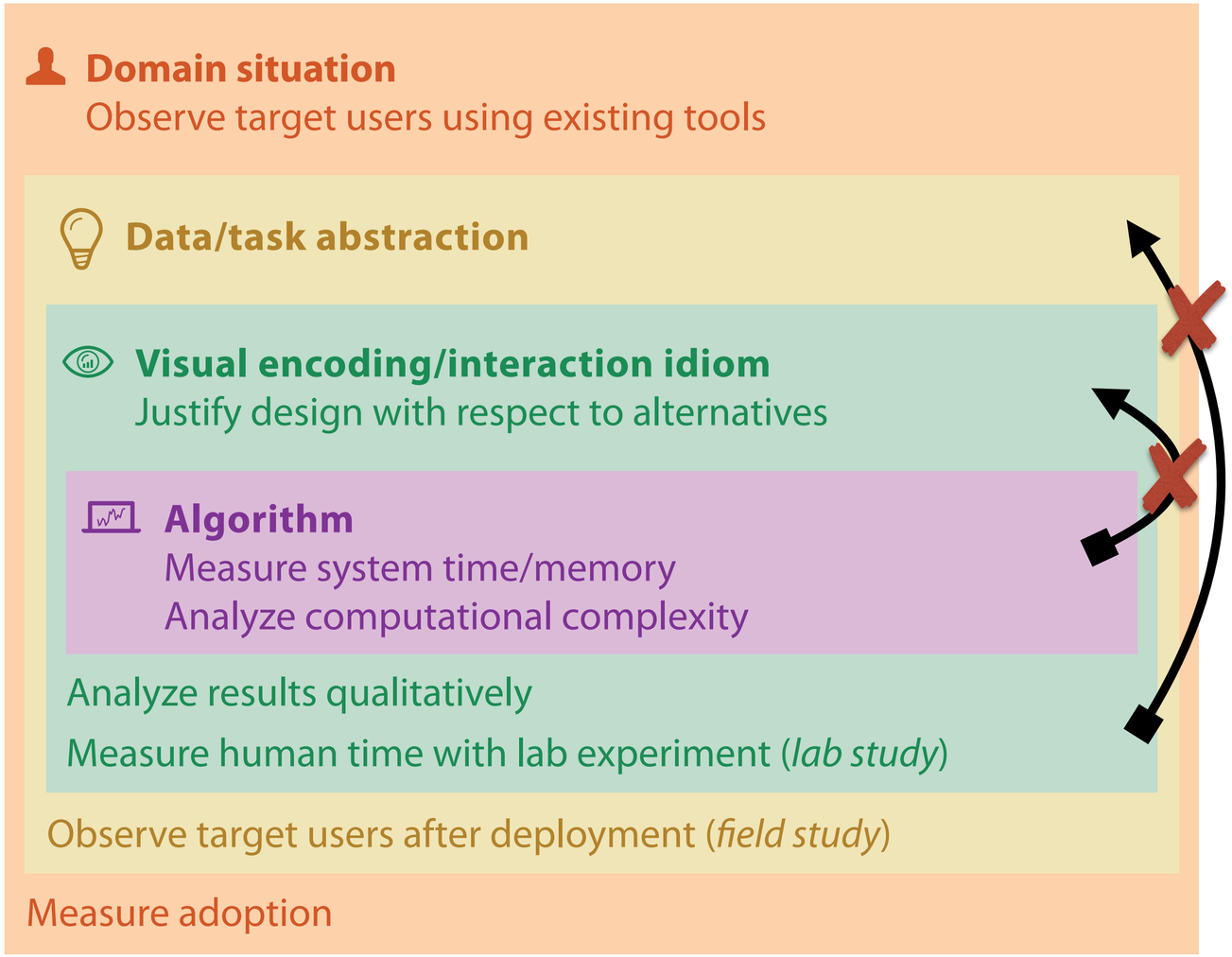


Avoid mismatches



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

flat table

–one item per row

–each column is attribute

–cell holds value for item-attribute pair

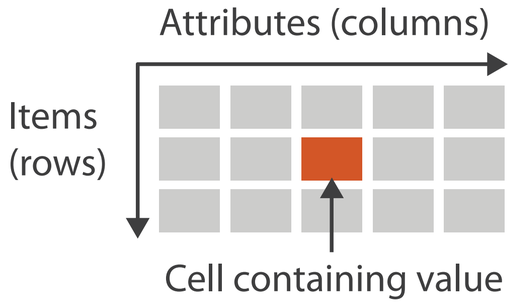
–unique key (could be implicit)

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

➔ Tables



Dataset types

Tables

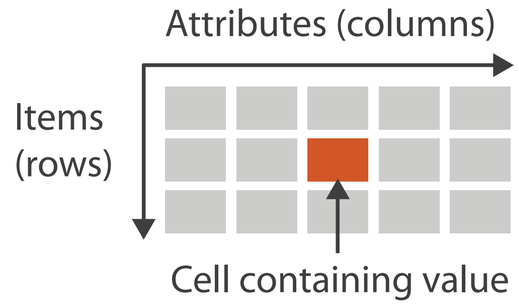
Items

Attributes

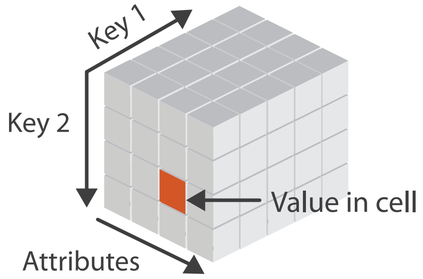
multidimensional tables

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

➔ Tables



➔ *Multidimensional Table*



	A	B	C	D	E
1	#1.2				
2	1500	529			
3	GeneName	DESCRIPTION	TCGA-02-0001-01C-01R-0177-01	TCGA-02-0003-01A-01R-0177-01	TCGA-02-0004-01A-01R-0298-01
4	LTF	LTF	-1.265728057	2.377012066	4.123979585
5	POSTN	POSTN	2.662411805	3.932400324	5.031585377
6	TMSL8	TMSL8	-3.082217838	-2.243148513	-0.02313681
7	HLA-DQA1	HLA-DQA1	-1.739664398	4.577962344	3.127744964
8	RP11-35N6.1	RP11-35N6.1	-3.346352968	-2.895400157	-3.473035067
9	STMN2	STMN2	-2.578511106	-3.051605144	-1.729892888
10	DCX	DCX	-2.26078976	-2.529795801	-2.844966278
11	AGXT2L1	AGXT2L1	-2.639493611	-3.113204863	-0.403975027
12	IL13RA2	IL13RA2	-2.93596915	-1.873600916	2.976256911
13	SLN	SLN	-2.466718221	-2.208406749	1.025827904
14	MEOX2	MEOX2	-2.395054066	-1.062676046	1.783235317
15	COL11A1	COL11A1	1.211934832	-0.399392588	4.733608974
16	NNMT	NNMT	0.703745164	0.664082419	3.069030715
17	F13A1	F13A1	-0.224094042	2.222197544	1.171354775
18	CXCL14	CXCL14	-3.1309694	-1.395056071	2.569540659
19	MBP	MBP	-1.906390566	-2.037626447	-2.935744906
20	TF	TF	-4.334123292	-4.680680246	-2.975788866
21	KCND2	KCND2	-1.777692395	-2.100362021	-1.996306032

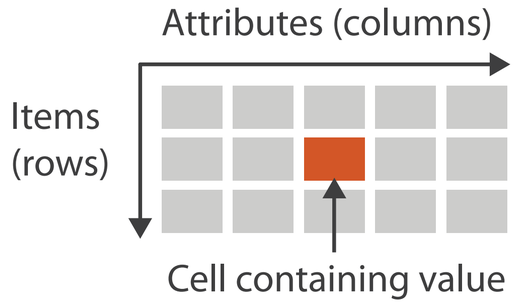
Dataset types

Tables	Networks & Trees
Items	Items (nodes)
Attributes	Links
	Attributes

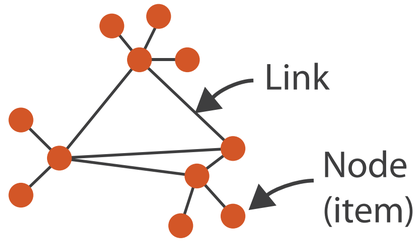
network/graph

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

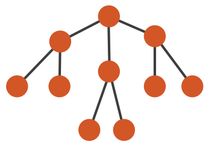
➔ Tables



➔ Networks



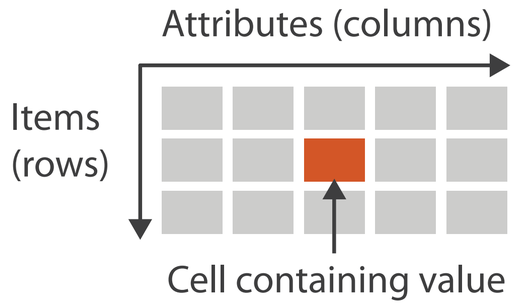
➔ *Trees*



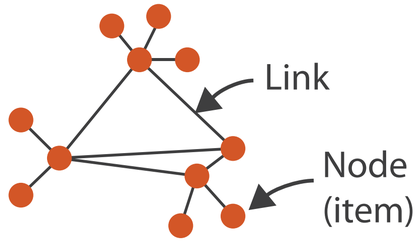
Dataset types

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

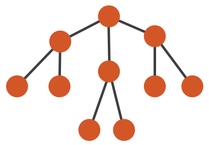
➔ Tables



➔ Networks

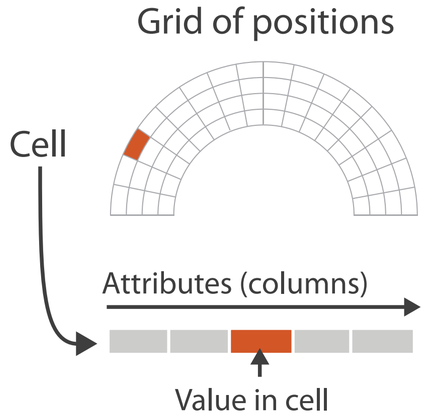


➔ *Trees*



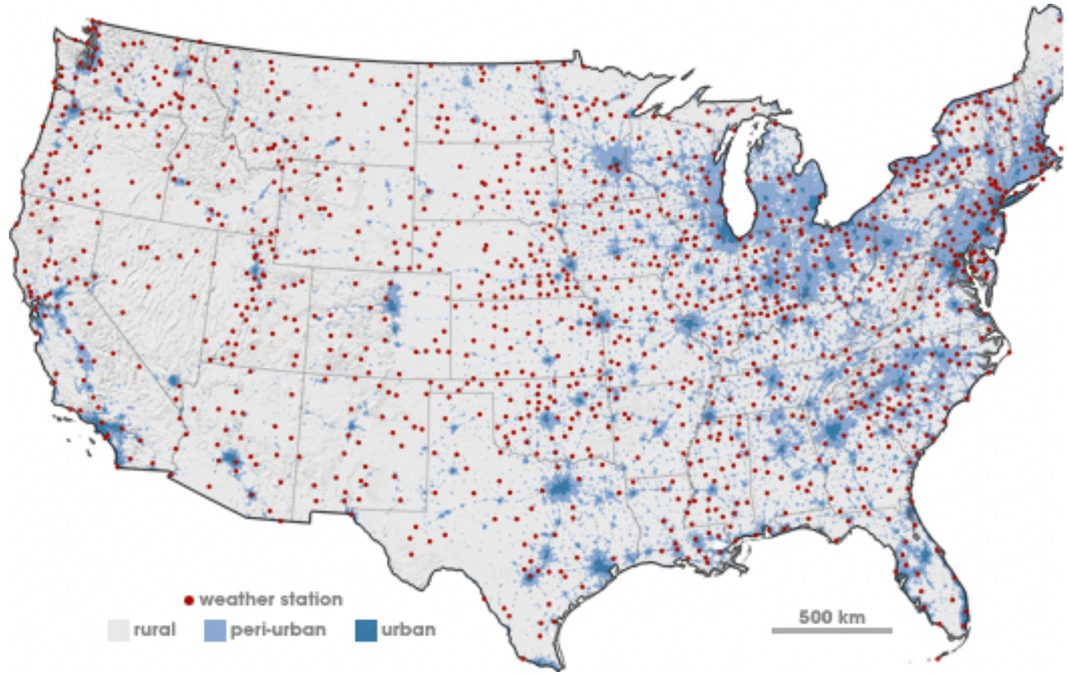
➔ Spatial

➔ Fields (Continuous)



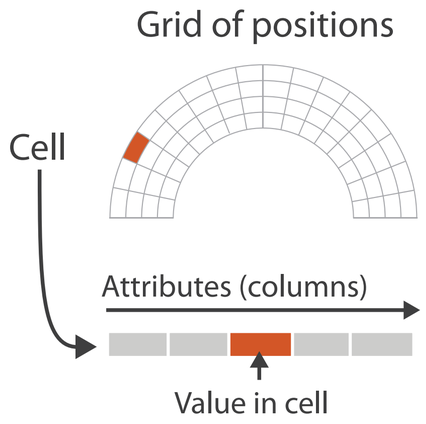
Spatial fields

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



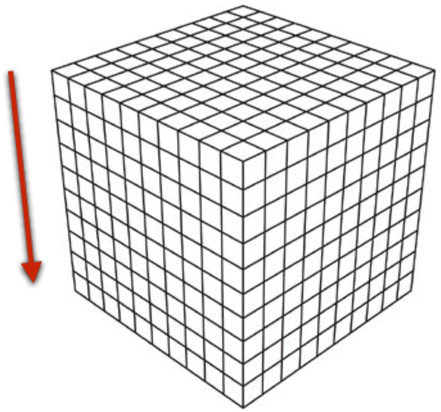
➔ Spatial

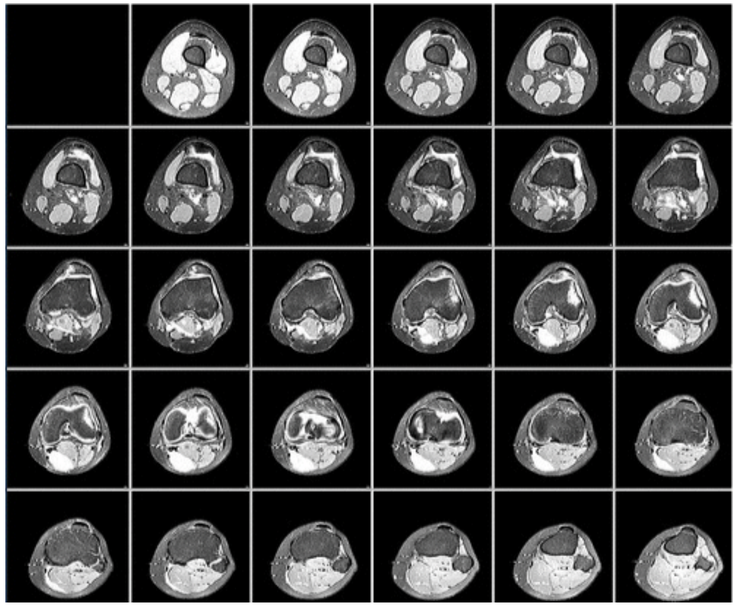
➔ Fields (Continuous)



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

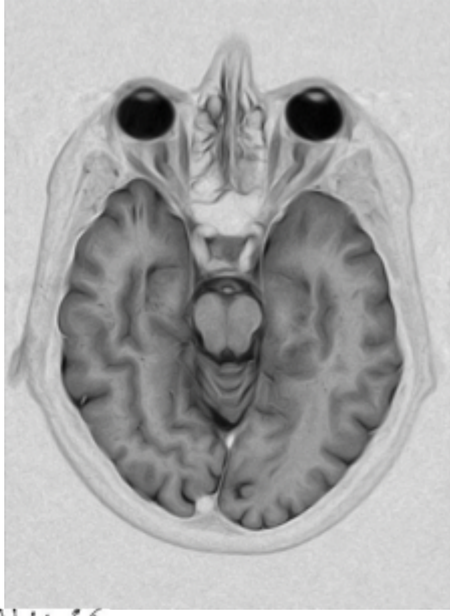




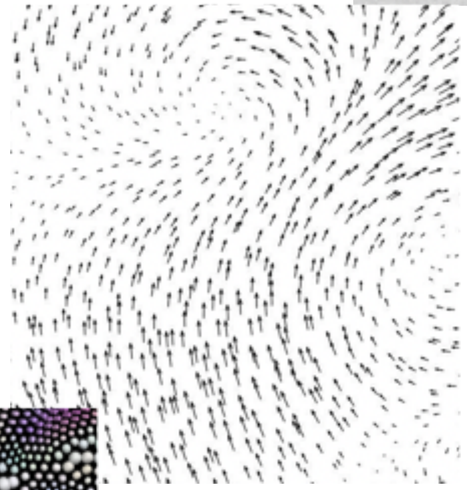
Spatial fields

- 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)

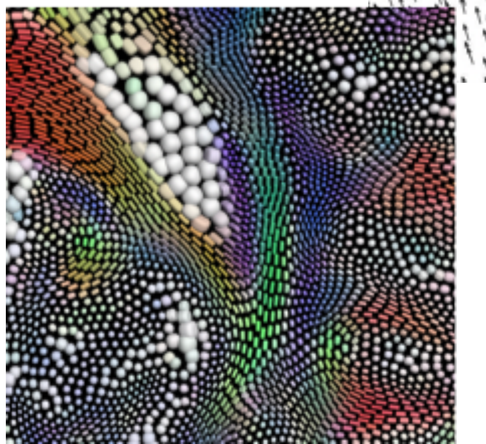
scalar



vector



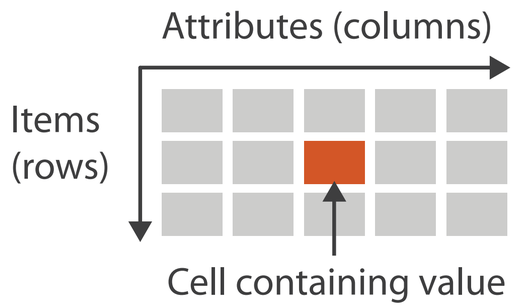
tensor



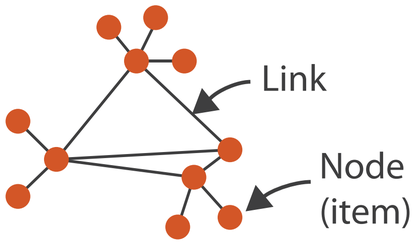
Dataset types

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

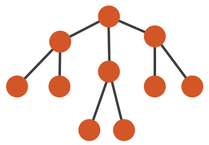
➔ Tables



➔ Networks

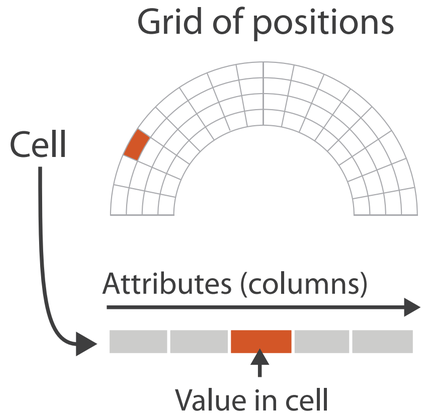


➔ *Trees*

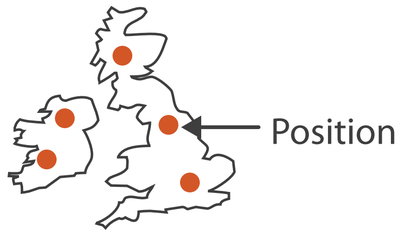


➔ Spatial

➔ Fields (Continuous)

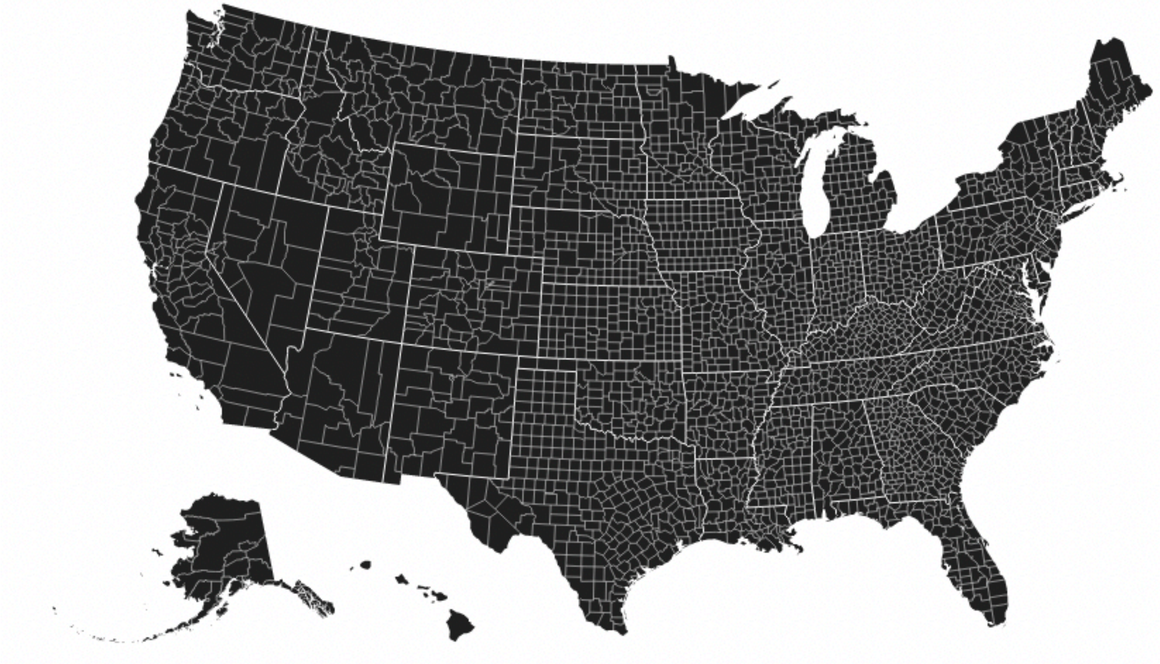


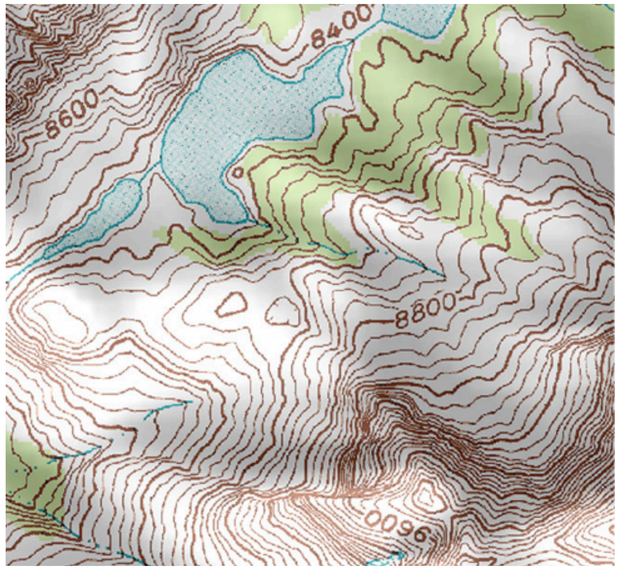
➔ Geometry (Spatial)



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

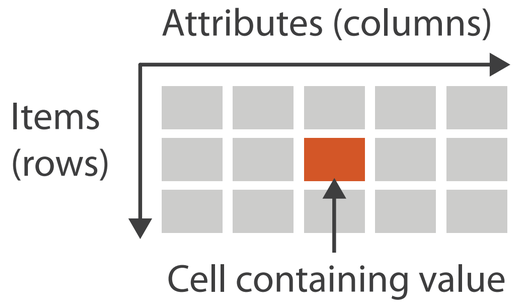




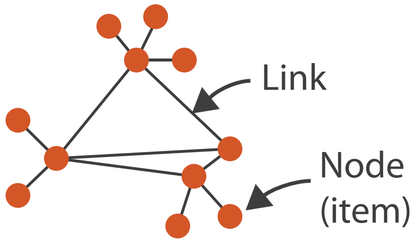
Dataset types

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

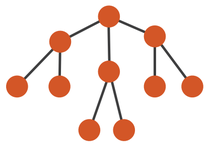
➔ Tables



➔ Networks

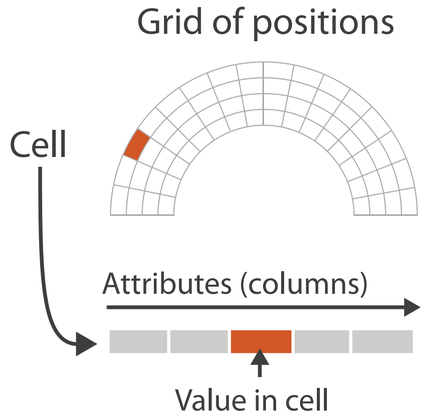


➔ *Trees*

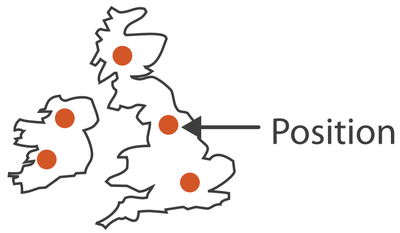


➔ Spatial

➔ Fields (Continuous)

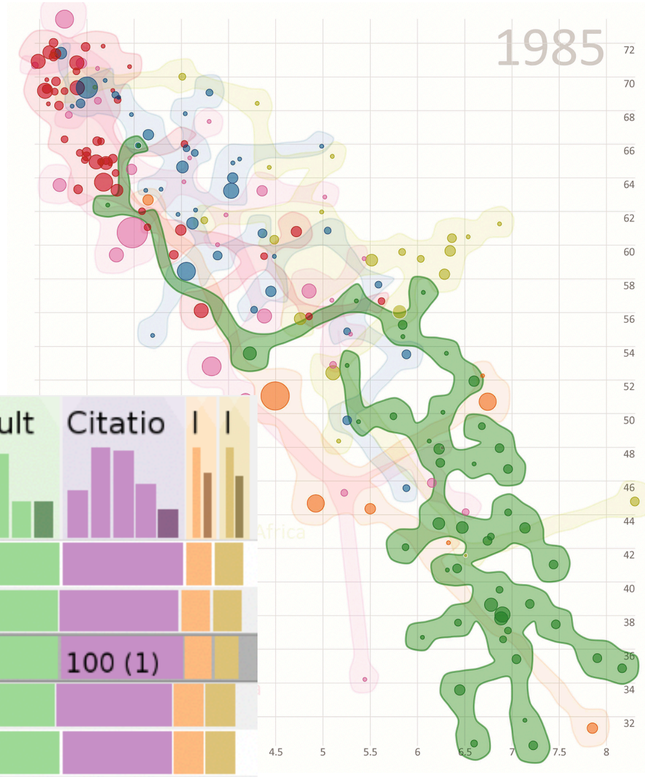


➔ Geometry (Spatial)

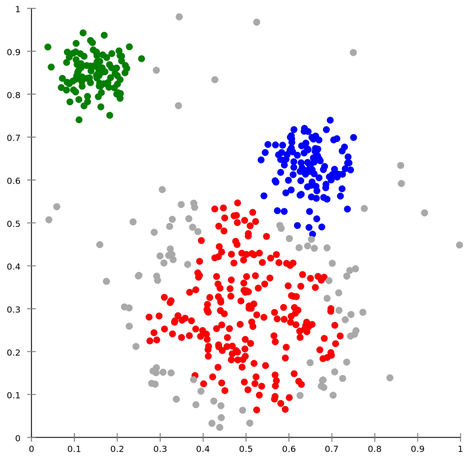


Collections

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



Rank	School Name	Academic repu	E	Facult	Citatio	I	I
	Filter: <None>						
1.	Massachusetts Inst						
2.	University of Camb						
3.	Harvard University	100 (1)			100 (1)		
4.	UCL (University Co						
5.	University of Oxfor						
6.	Imperial College L						
7.	Yale University						
8.	University of Chic						



Data and Dataset Types

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

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	S	T	U
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!]

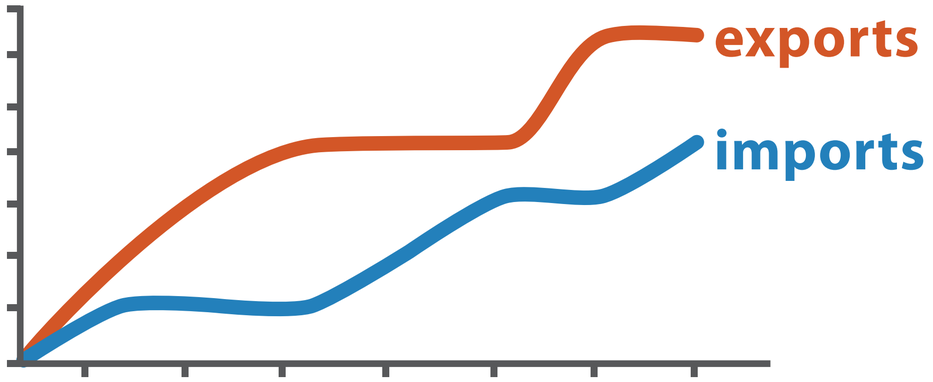
- data abstraction process relies on conceptual model
  - for transforming data if needed

Data vs conceptual model, example

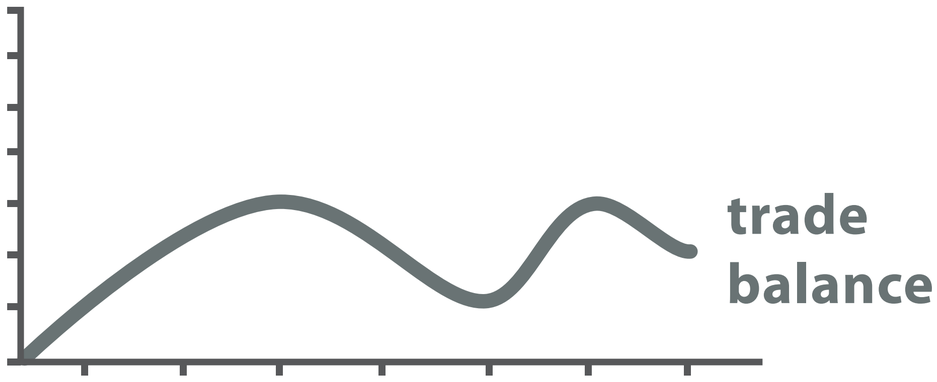
- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready
  - above freezing,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

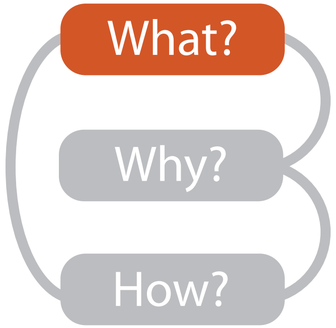


Original Data



trade balance = *exports* – *imports*

Derived Data



What?

Datasets

Data Types

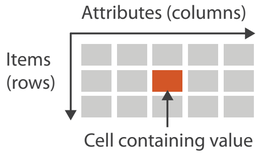
→ Items → Attributes → Links → Positions → Grids

Data and Dataset Types

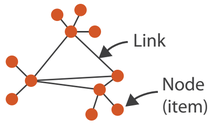
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Types

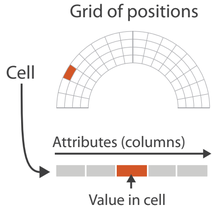
→ Tables



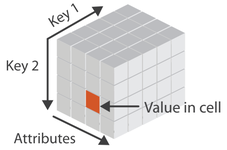
→ Networks



→ Fields (Continuous)



→ *Multidimensional Table*



→ *Trees*

→ Geometry (Spatial)

Dataset Availability

→ Static

→ Dynamic

Attributes

Attribute Types

→ Categorical

→ Ordered

→ *Ordinal*

→ *Quantitative*

Ordering Direction

→ Sequential

→ Diverging

→ Cyclic

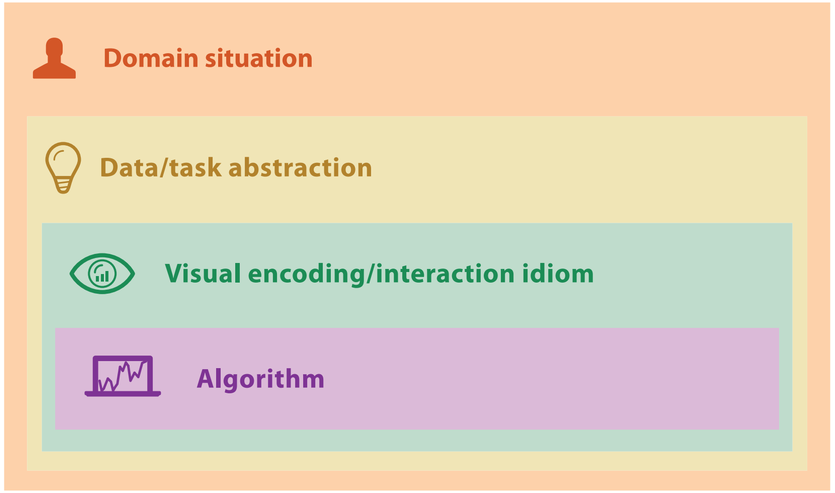
Reading Material

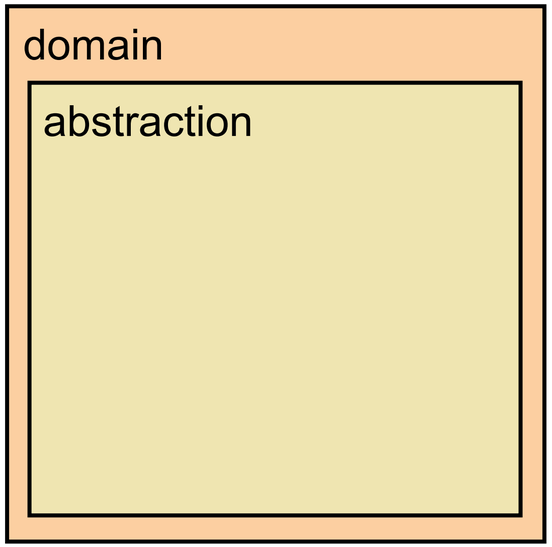
[dv3] **Chapter 2 - What: Data Abstraction**

Task Abstraction

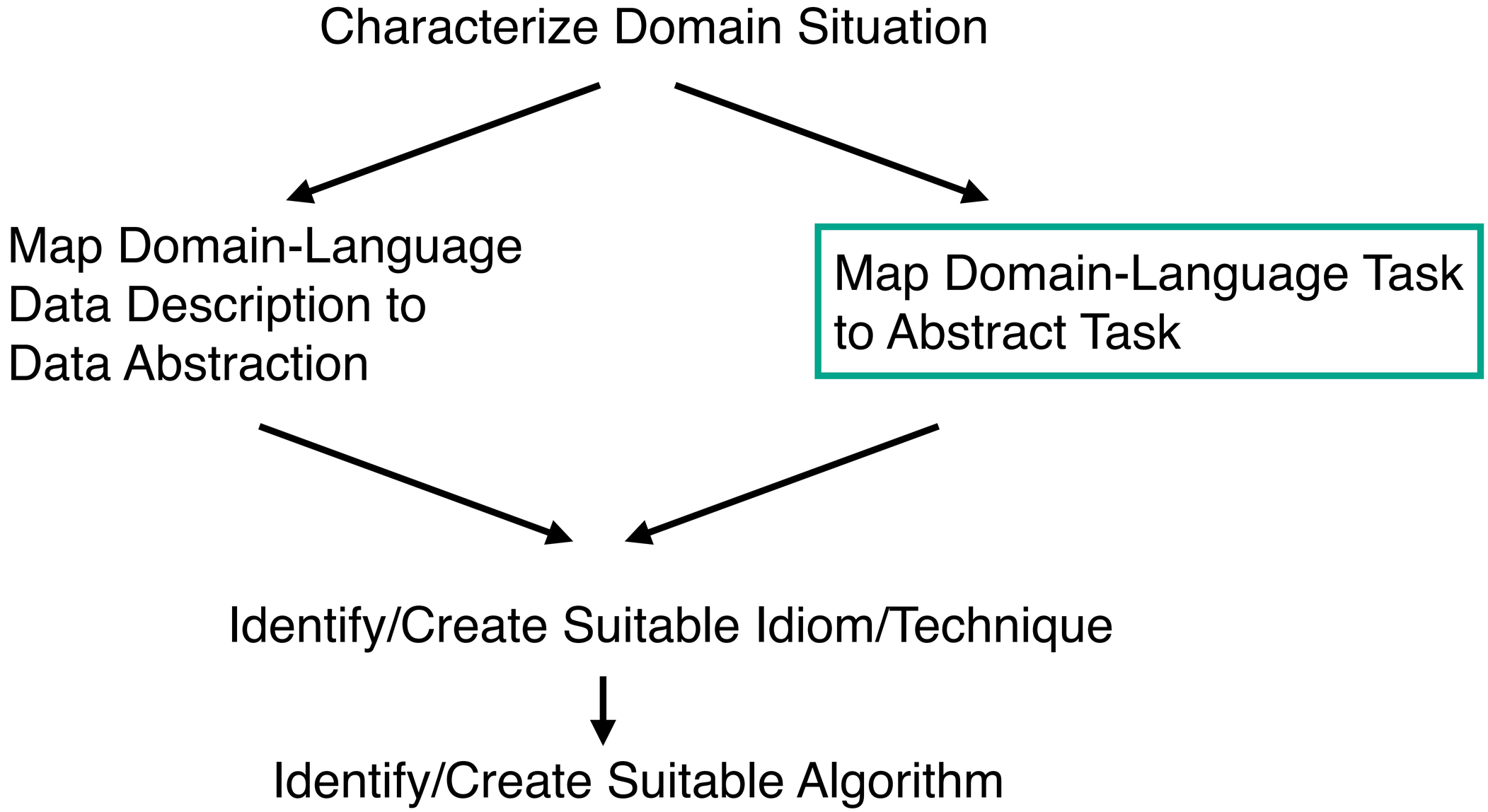
From domain to abstraction

- **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

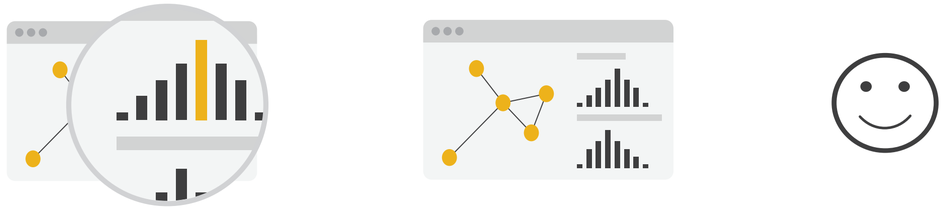
Actions: Analyze

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

➔ **Analyze**

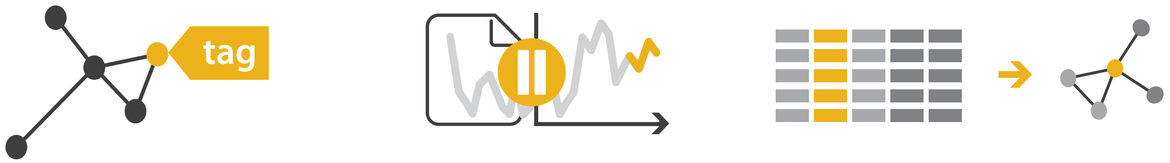
➔ Consume

➔ *Discover* ➔ *Present* ➔ *Enjoy*



➔ Produce

➔ *Annotate* ➔ *Record* ➔ *Derive*



Actions: Search

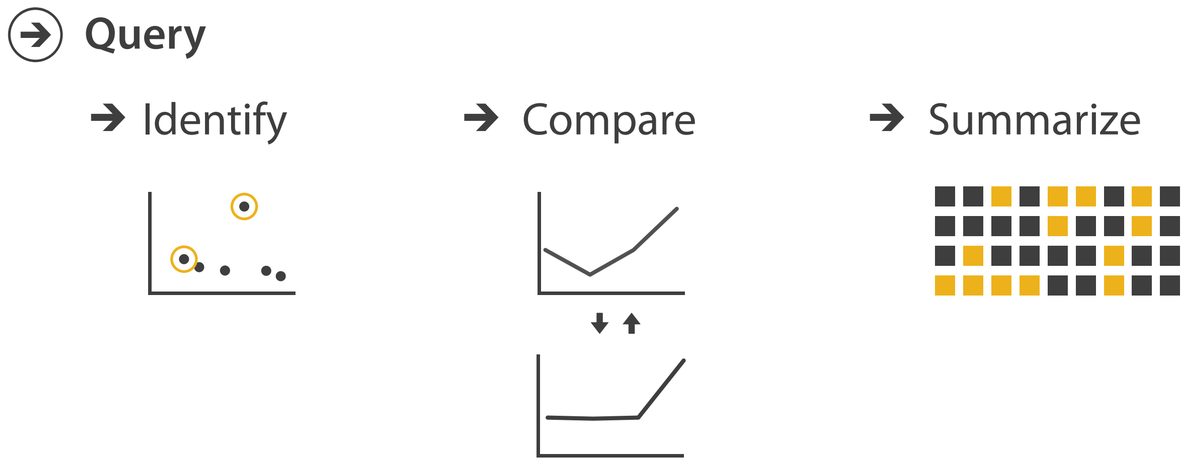
- what does user know?
  - target, location
- lookup
  - ex: word in dictionary
    - alphabetical order
- locate
  - ex: keys in your house
  - ex: node in network
- browse
  - ex: books in bookstore
- explore
  - ex: find cool neighborhood in new city

Search

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

Actions: Query

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



Actions

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

Actions

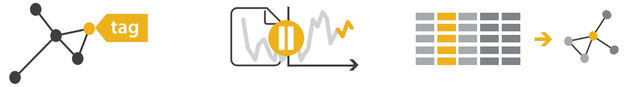
Analyze

→ Consume

→ *Discover* → *Present* → *Enjoy*

→ Produce

→ *Annotate* → *Record* → *Derive*

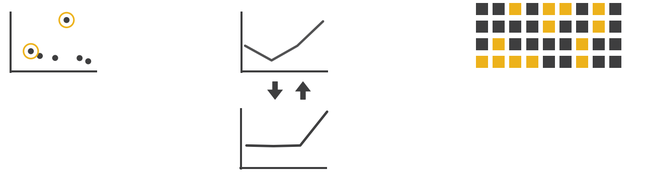


Search

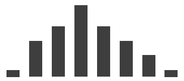
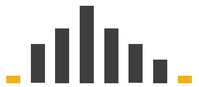
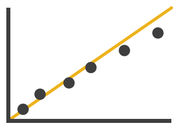
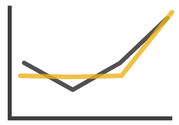
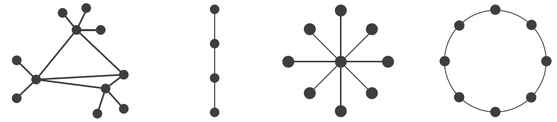
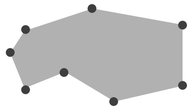
	Target known	Target unknown
Location known	*Lookup*	*Browse*
Location unknown	*Locate*	*Explore*

Query

→ Identify → Compare → Summarize



Task abstraction: Targets

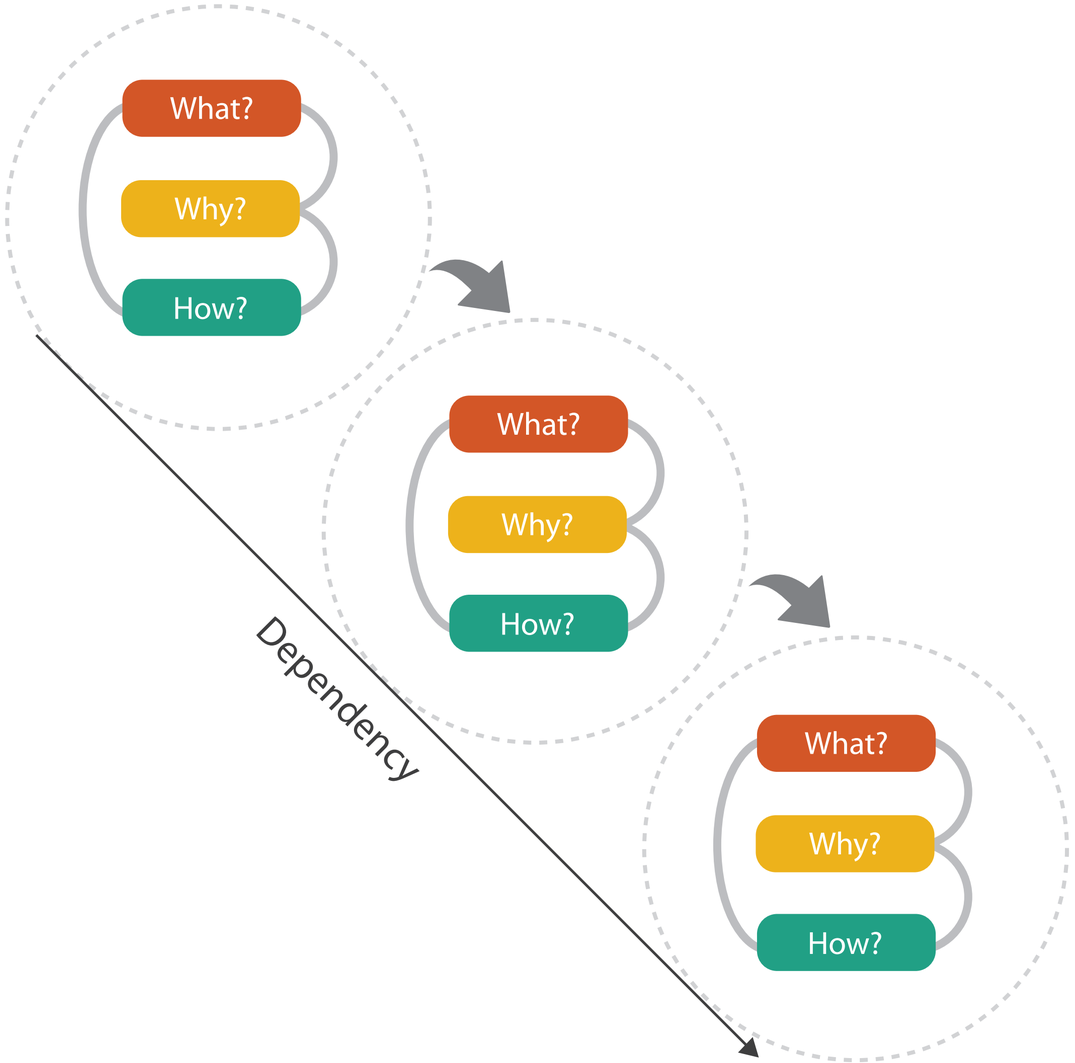
- **All Data**
  - Trends 
  - Outliers 
  - Features 
- **Attributes**
  - One
    - *Distribution* 
    - *Extremes* 
  - Many
    - *Dependency*
    - *Correlation* 
    - *Similarity* 
- **Network Data**
  - Topology 
  - *Paths* 
- **Spatial Data**
  - Shape 

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

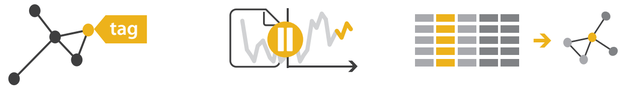
Analyze

→ Consume

→ *Discover* → *Present* → *Enjoy*

→ Produce

→ *Annotate* → *Record* → *Derive*

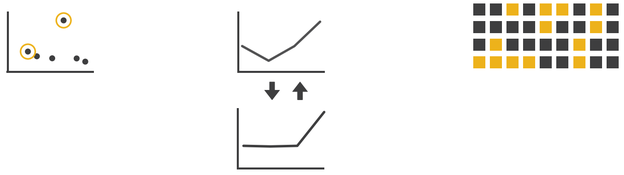


Search

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

Query

→ Identify → Compare → Summarize



Targets

All Data

→ Trends → Outliers → Features

Attributes

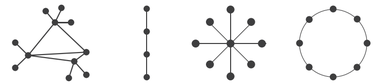
→ One → Many

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

→ *Extremes*

Network Data

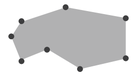
→ Topology



→ *Paths*

Spatial Data

→ Shape



Reading Material

[dv3] **Chapter 3 - Why: Task Abstraction**

Questions?


@rschifan


rossano.schifanella@unito.it


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