

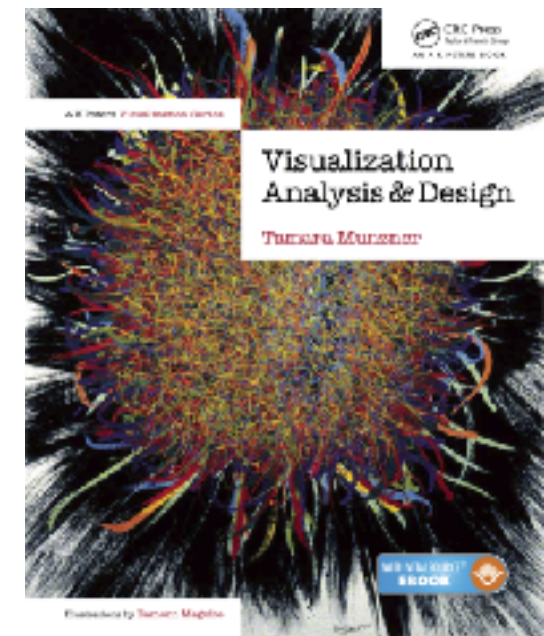
# Visualization Analysis & Design

## *What's Vis, and Why Do It? (Ch 1)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# Defining visualization (vis)

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

# Defining visualization (vis)

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

Why?...

# Why have a human in the loop?

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

# Why have a human in the loop?

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

**Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.**

# Why have a human in the loop?

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

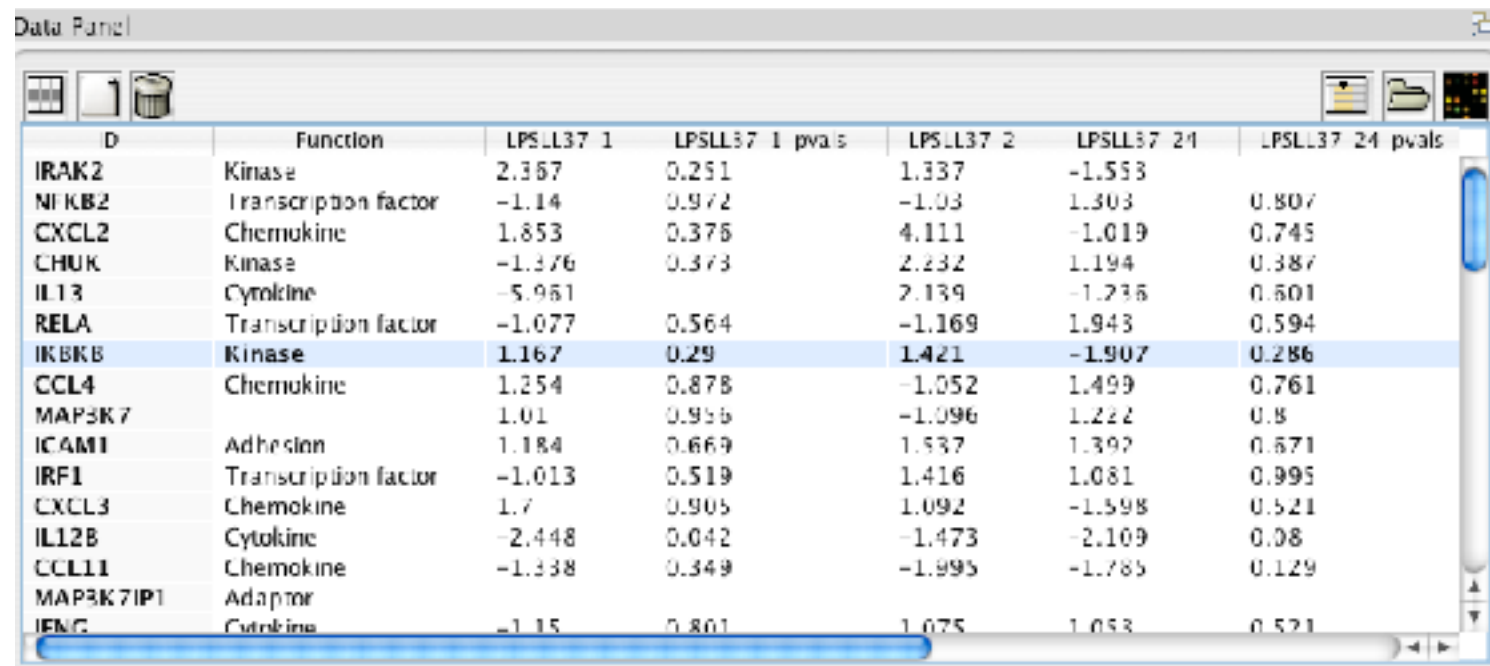
**Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.**

- don't need vis when fully automatic solution exists and is trusted
- many analysis problems ill-specified
  - don't know exactly what questions to ask in advance
- possibilities
  - long-term use for end users (ex: exploratory analysis of scientific data)
  - presentation of known results (ex: New York Times Upshot)
  - stepping stone to assess requirements before developing models
  - help automatic solution developers refine & determine parameters
  - help end users of automatic solutions verify, build trust

# Why use an external representation?

Computer-based visualization systems provide **visual representations** of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception



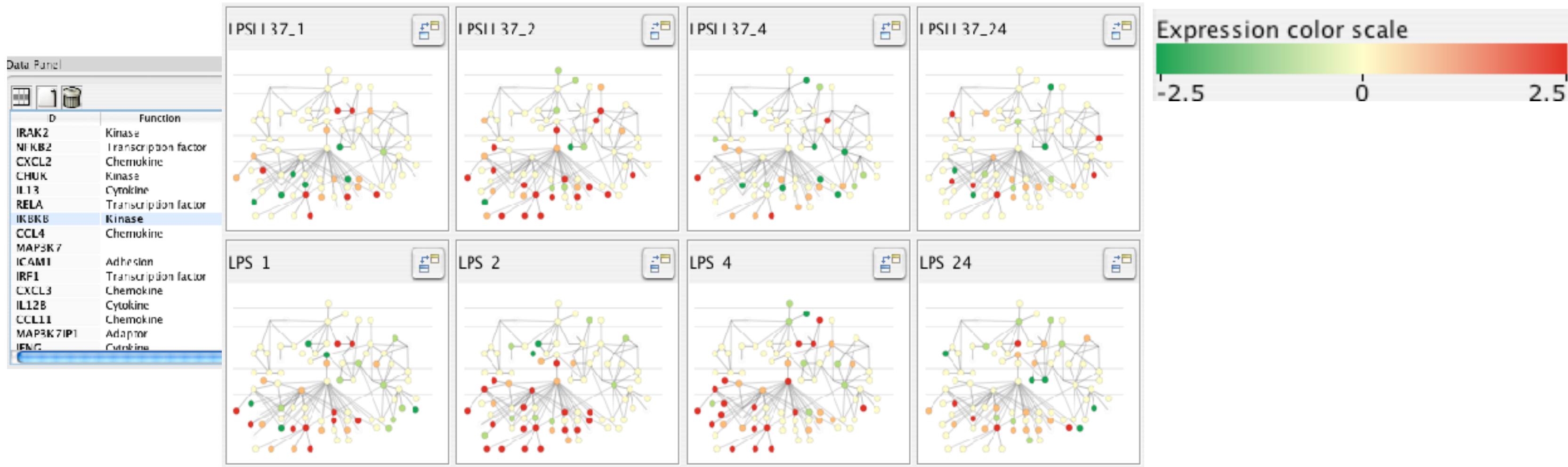
D	Function	LPSLL37 1	LPSLL37 1 pvals	LPSLL37 2	LPSLL37 24	LPSLL37 24 pvals
IRAK2	Kinase	2.357	0.251	1.337	-1.553	
NIK2	Transcription factor	-1.14	0.972	-1.03	1.303	0.807
CXCL2	Chemokine	1.853	0.376	4.111	-1.019	0.745
CHUK	Kinase	-1.376	0.373	2.232	1.194	0.387
IL13	Cytokine	-5.961		2.139	-1.236	0.601
RELA	Transcription factor	-1.077	0.564	-1.169	1.943	0.594
IKKKB	Kinase	1.167	0.29	1.421	-1.907	0.286
CCL4	Chemokine	1.254	0.878	-1.052	1.499	0.761
MAP3K7		1.01	0.956	-1.096	1.222	0.8
ICAM1	Adhesion	1.184	0.669	1.537	1.392	0.671
IRF1	Transcription factor	-1.013	0.519	1.416	1.081	0.995
CXCL3	Chemokine	1.7	0.905	1.092	-1.598	0.521
IL12B	Cytokine	-2.448	0.042	-1.473	-2.109	0.08
CCL11	Chemokine	-1.338	0.349	-1.995	-1.785	0.129
MAP3K7IP1	Adaptor					
IFNG	Cytokine	-1.15	0.801	1.075	1.053	0.521

[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE TVCG (Proc. InfoVis) 14(6):1253-1260, 2008.]

# Why use an external representation?

Computer-based visualization systems provide **visual representations** of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception



[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE TVCG (Proc. InfoVis) 14(6):1253-1260, 2008.]



# Why depend on vision?

**Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.**

- human visual system is high-bandwidth channel to brain
  - overview possible due to background processing
    - subjective experience of seeing everything simultaneously
    - significant processing occurs in parallel and pre-attentively
- sound: lower bandwidth and different semantics
  - overview not supported
    - subjective experience of sequential stream
- touch/haptics: impoverished record/replay capacity
  - only very low-bandwidth communication thus far
- taste, smell: no viable record/replay devices

# Why represent all the data?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
  - confirm expected and find unexpected patterns
  - assess validity of statistical model

## Anscombe's Quartet

Identical statistics	
x mean	9
x variance	10
y mean	7.5
y variance	3.75
x/y correlation	0.816

# Why represent all the data?

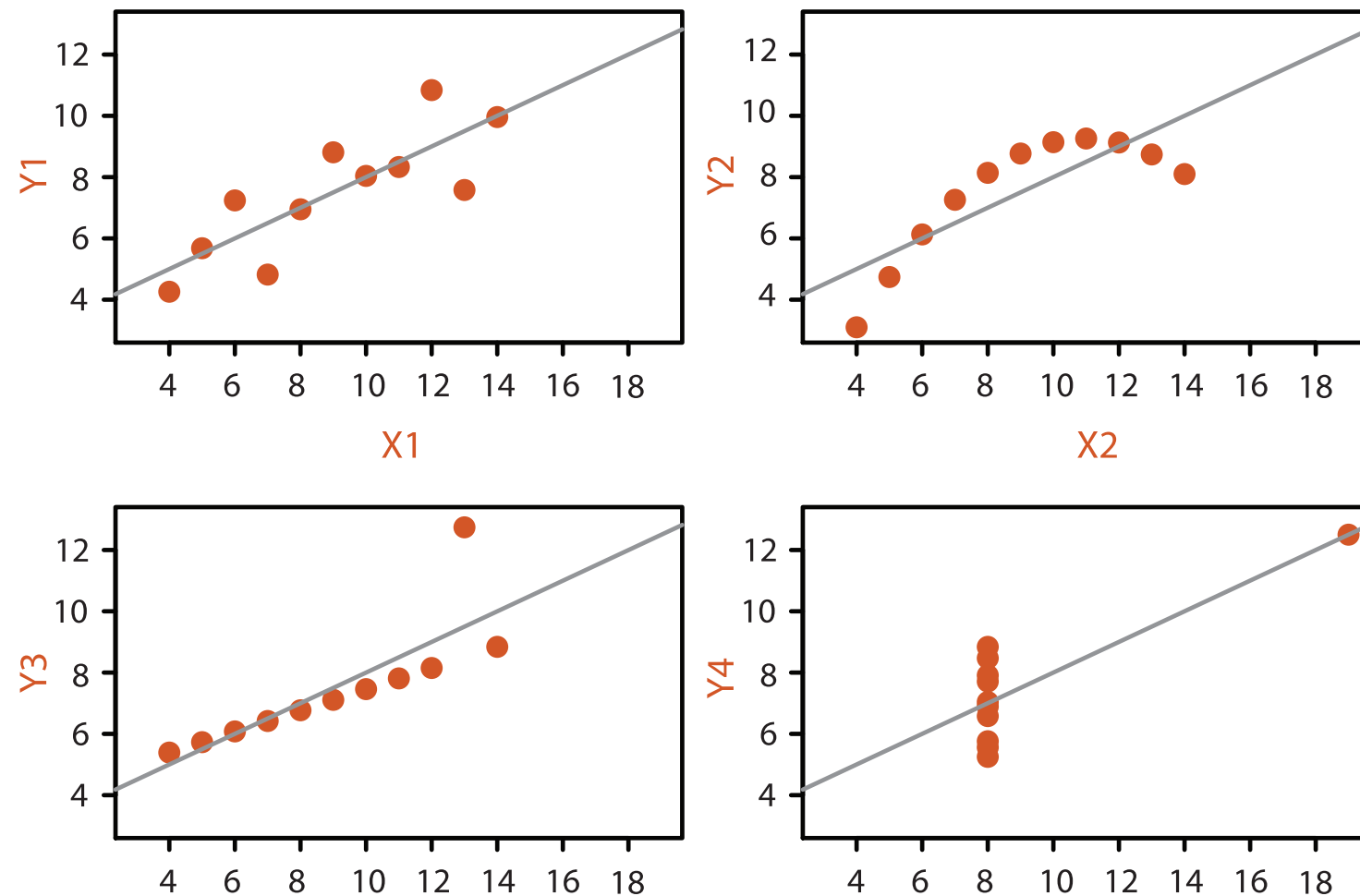
Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
  - confirm expected and find unexpected patterns
  - assess validity of statistical model

## Anscombe's Quartet

### Identical statistics

x mean	9
x variance	10
y mean	7.5
y variance	3.75
x/y correlation	0.816



# What resource limitations are we faced with?

**Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.**

- computational limits
  - computation time, system memory
- display limits
  - pixels are precious & most constrained resource
  - **information density**: ratio of space used to encode info vs unused whitespace
    - tradeoff between clutter and wasting space
    - find sweet spot between dense and sparse
- human limits
  - human time, human memory, human attention

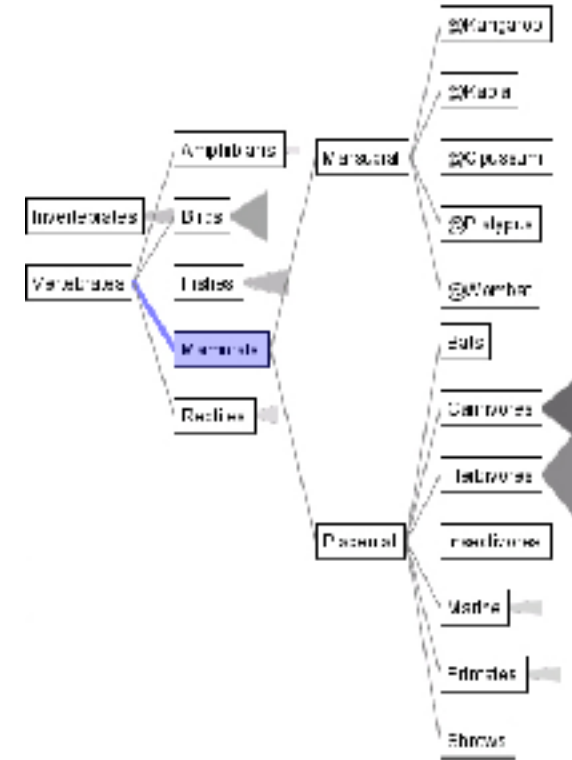
# Why analyze?

- imposes structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new
  - most possibilities ineffective for particular task/data combination

# Why analyze?

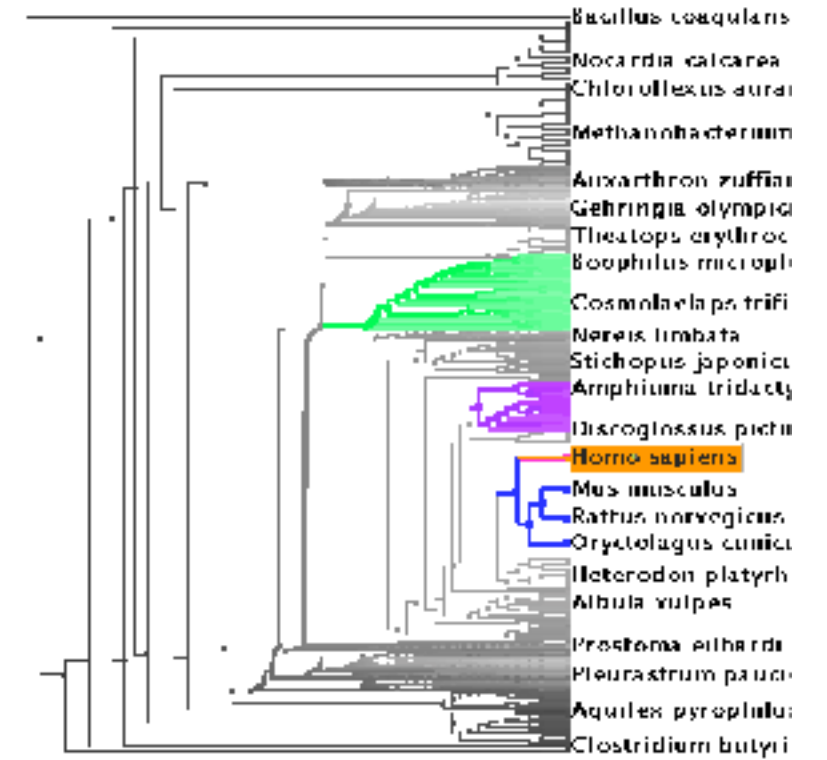
- imposes structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new
  - most possibilities ineffective for particular task/data combination

## SpaceTree



[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

## TreeJuxtaposer

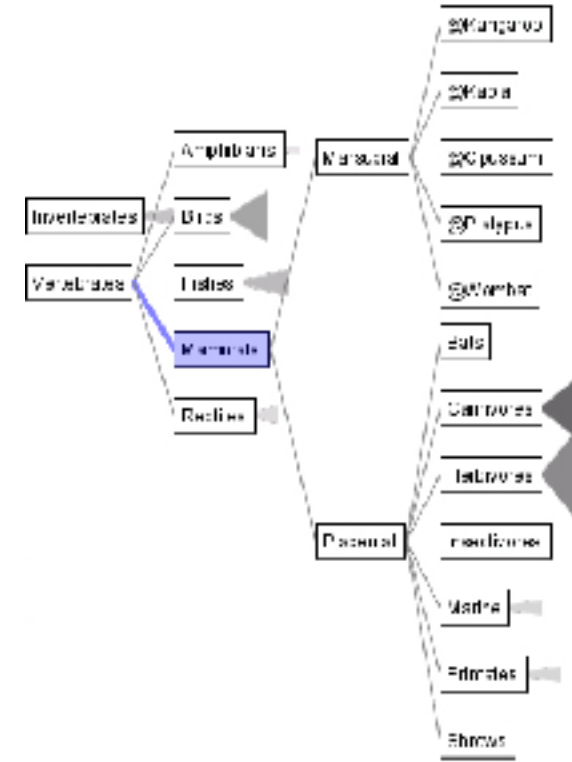


[TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]

# Why analyze?

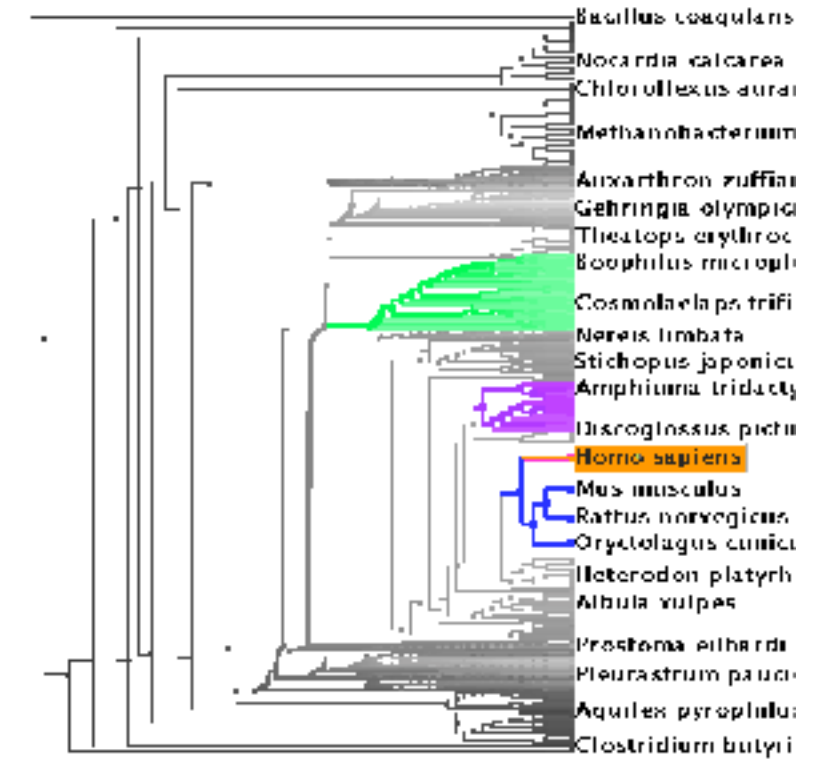
- imposes structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new
  - most possibilities ineffective for particular task/data combination

## SpaceTree



[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

## TreeJuxtaposer



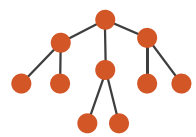
[TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]

### What?

### Why?

### How?

#### → Tree



#### → Actions

→ Present → Locate → Identify



#### → Targets

→ Path between two nodes



#### → SpaceTree

→ Encode → Navigate → Select → Filter → Aggregate



#### → TreeJuxtaposer

→ Encode → Navigate → Select → Arrange



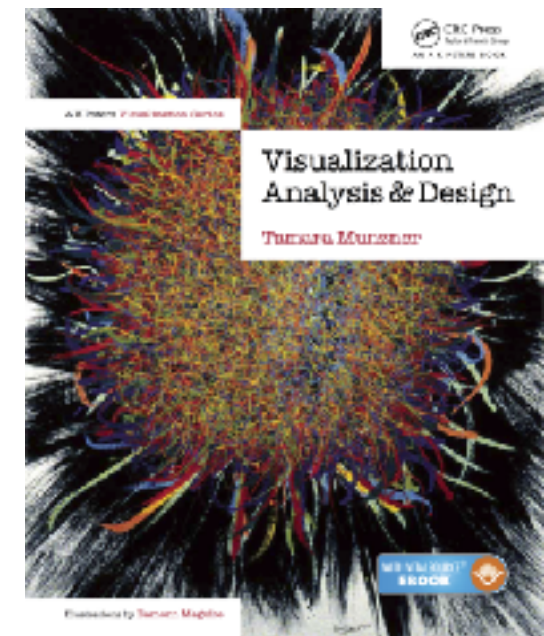
# Visualization Analysis & Design

## *Analysis: Nested Model (Ch 4)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

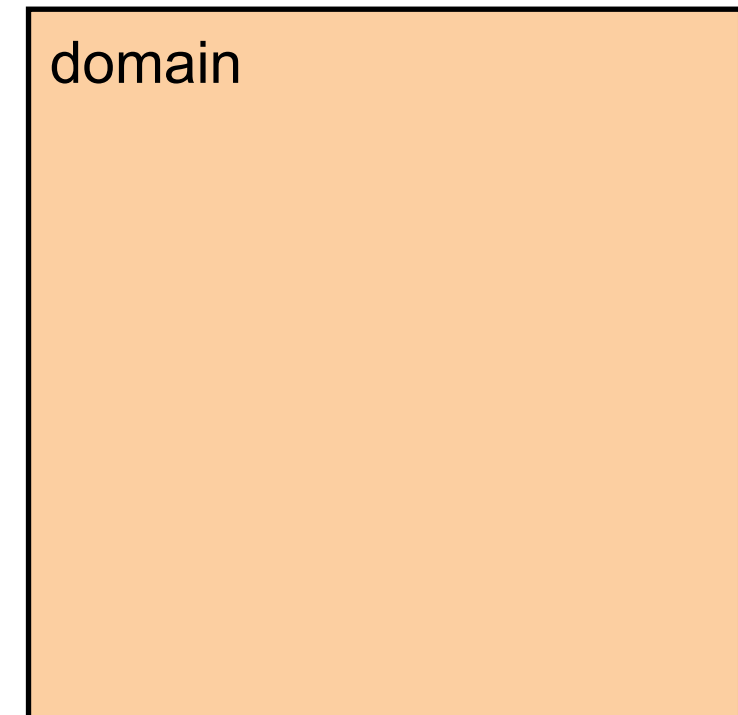
[@tamaramunzner](#)





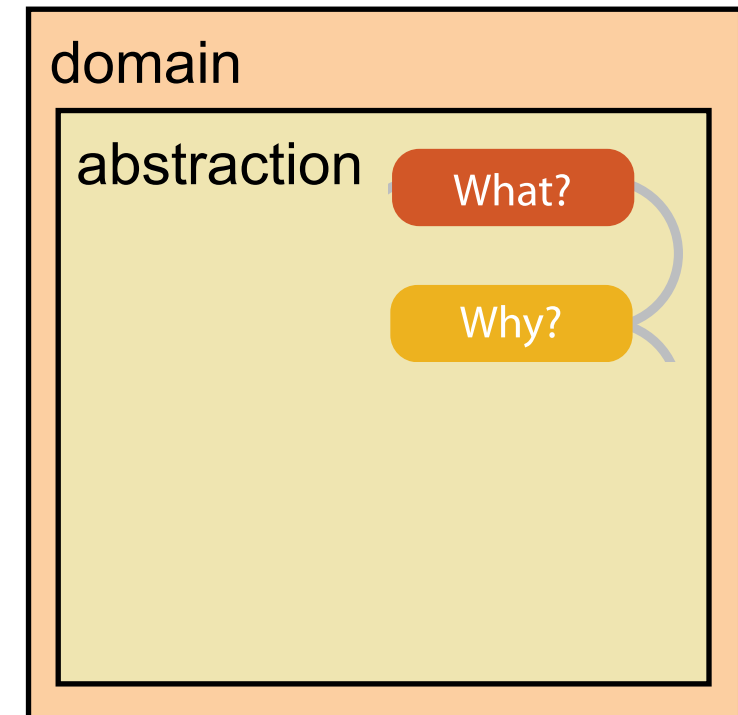
# Analysis framework: Four levels, three questions

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



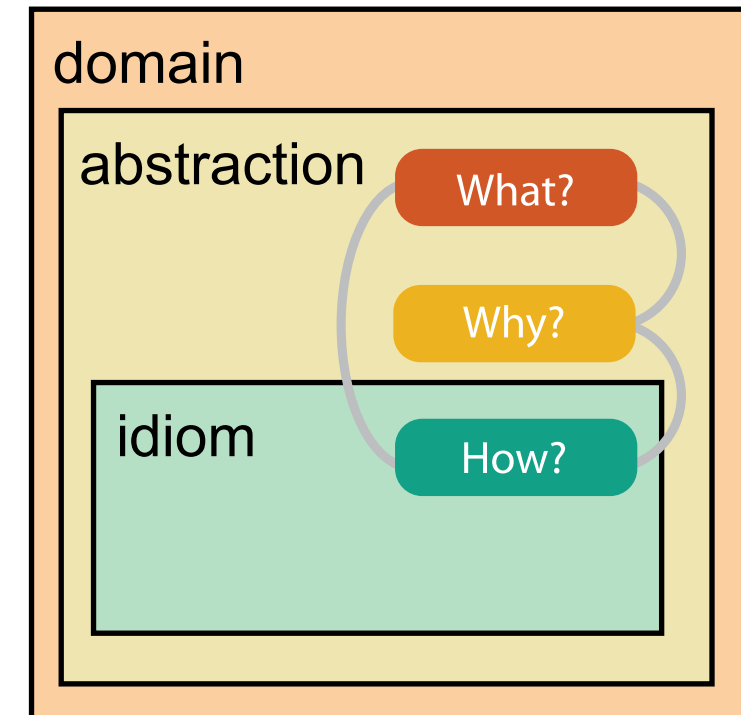
# Analysis framework: Four levels, three questions

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



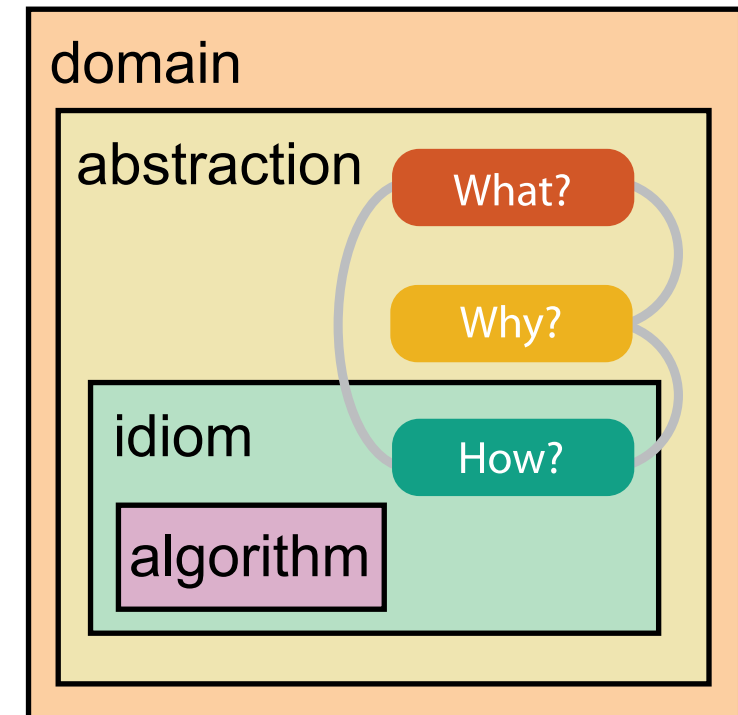
# Analysis framework: Four levels, three questions

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



# 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

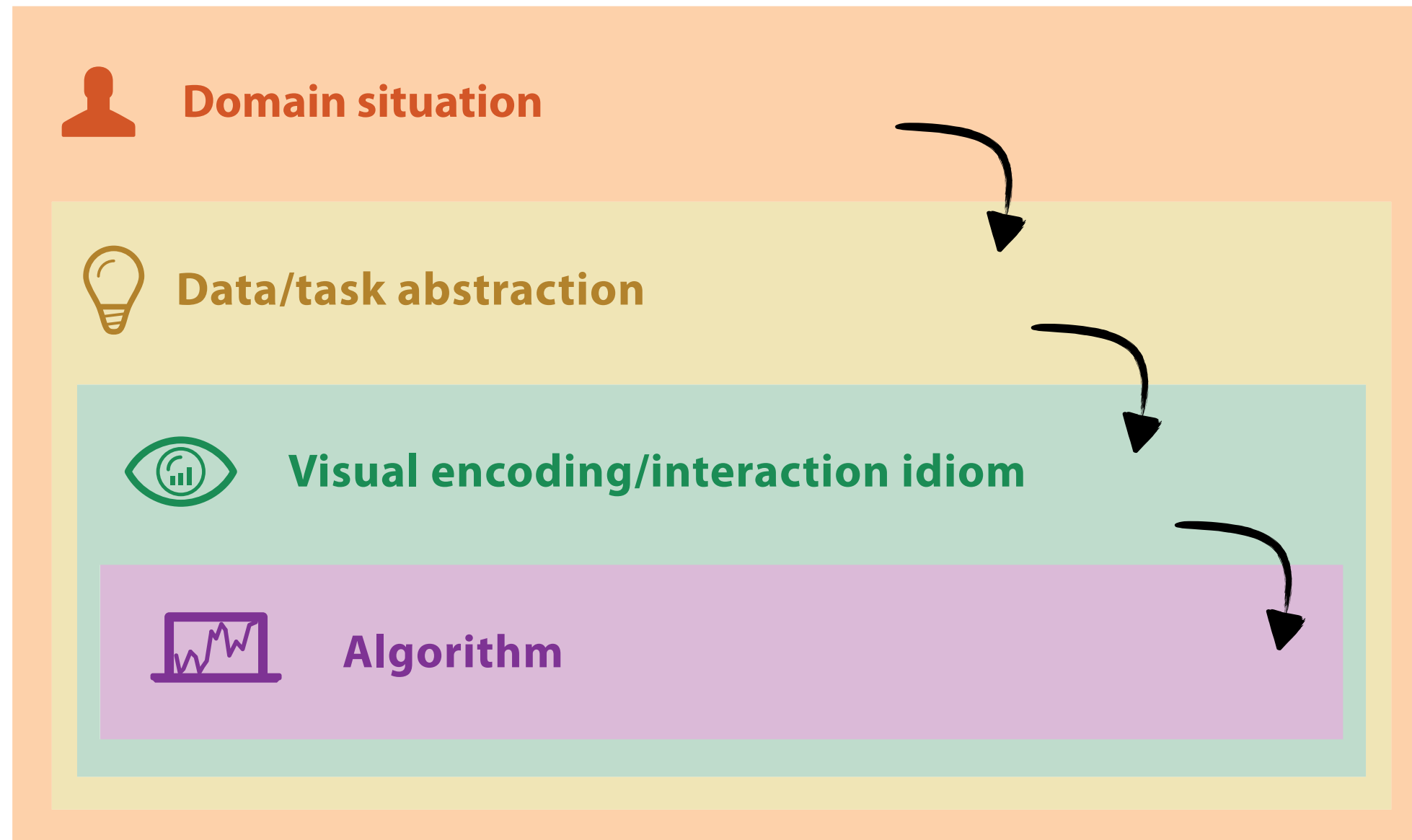


[A Multi-Level Typology of Abstract Visualization Tasks. Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

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

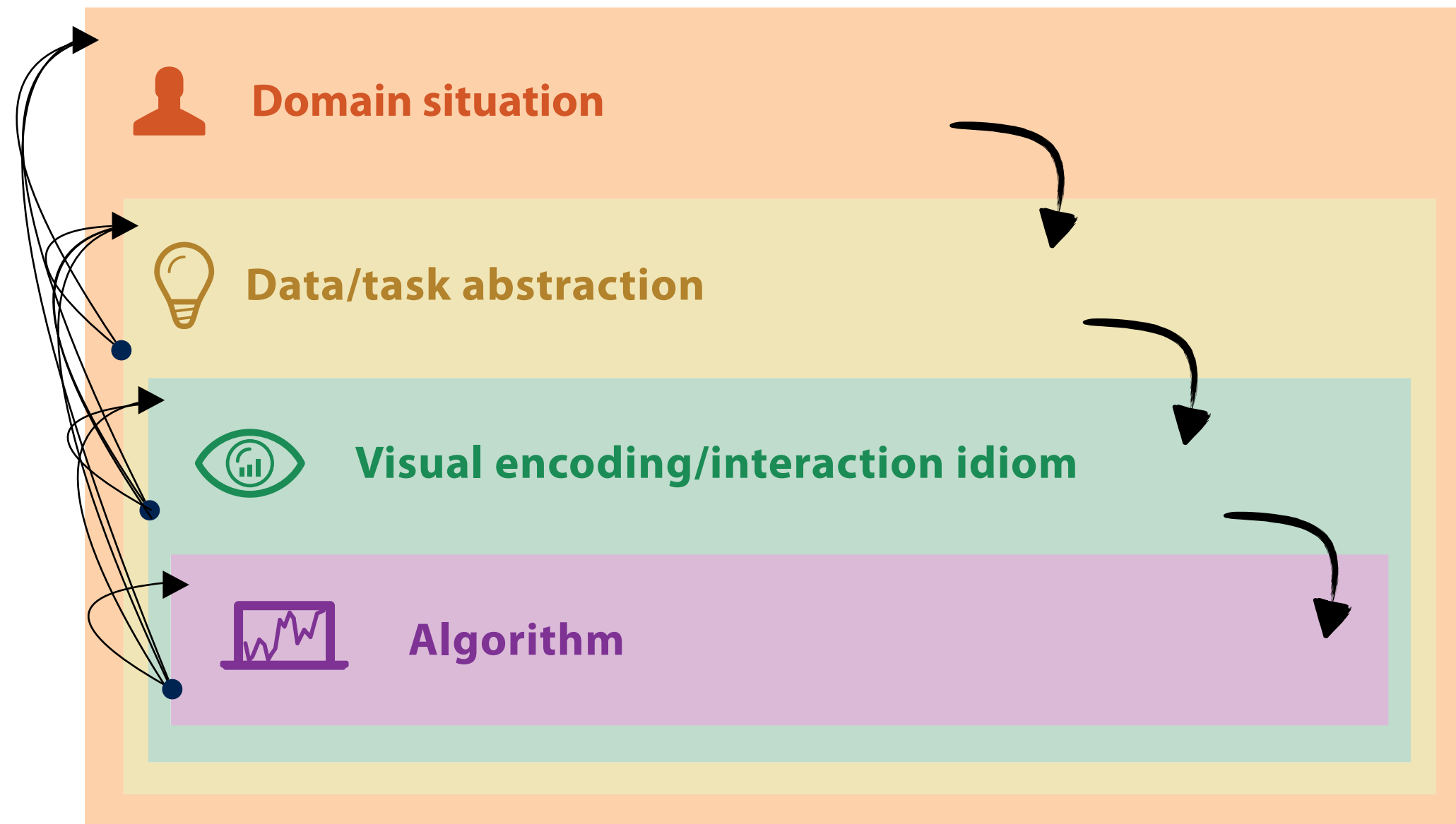
# Nested model

- downstream: cascading effects



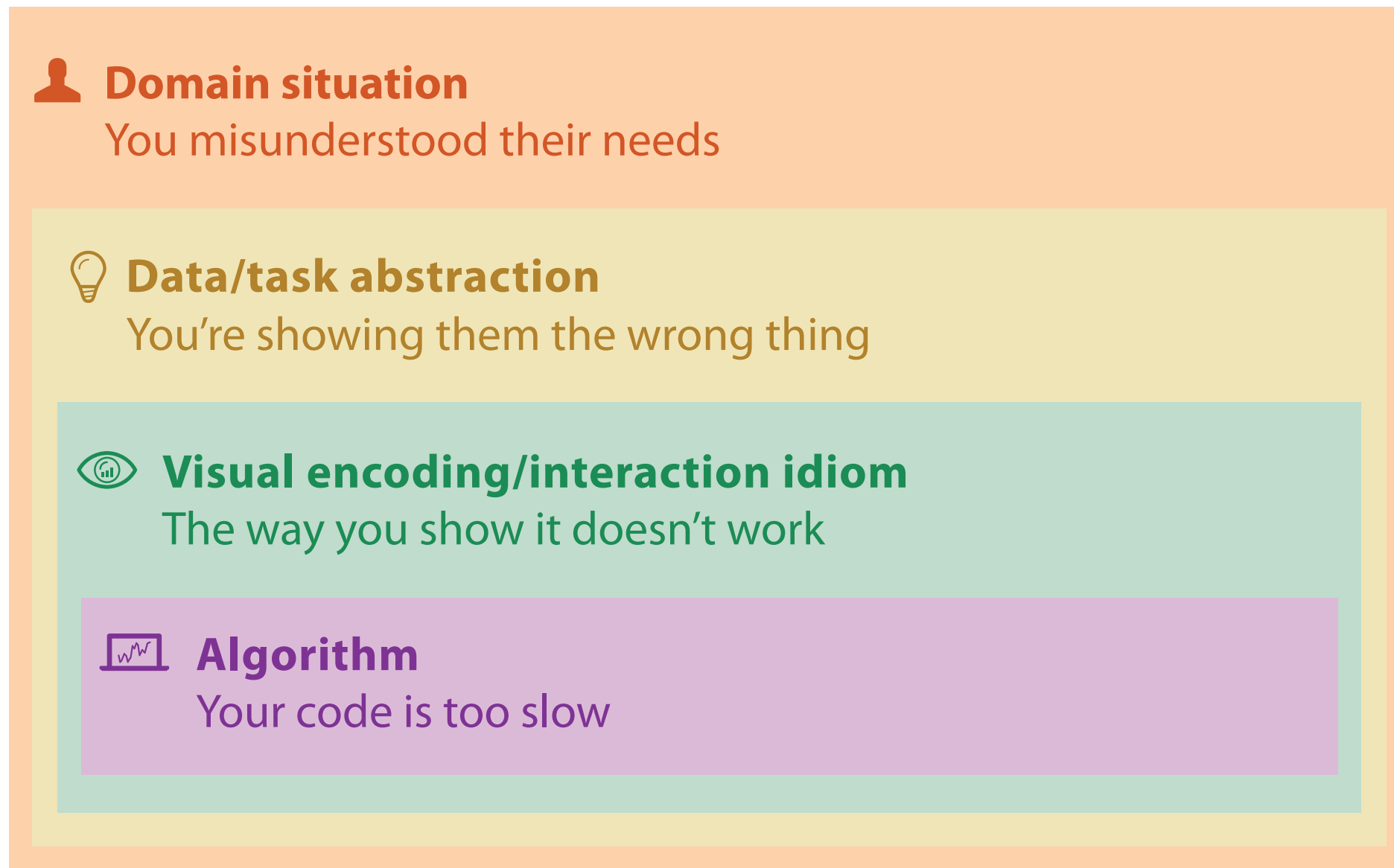
# Nested model

- 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



## Algorithm

Measure system time/memory

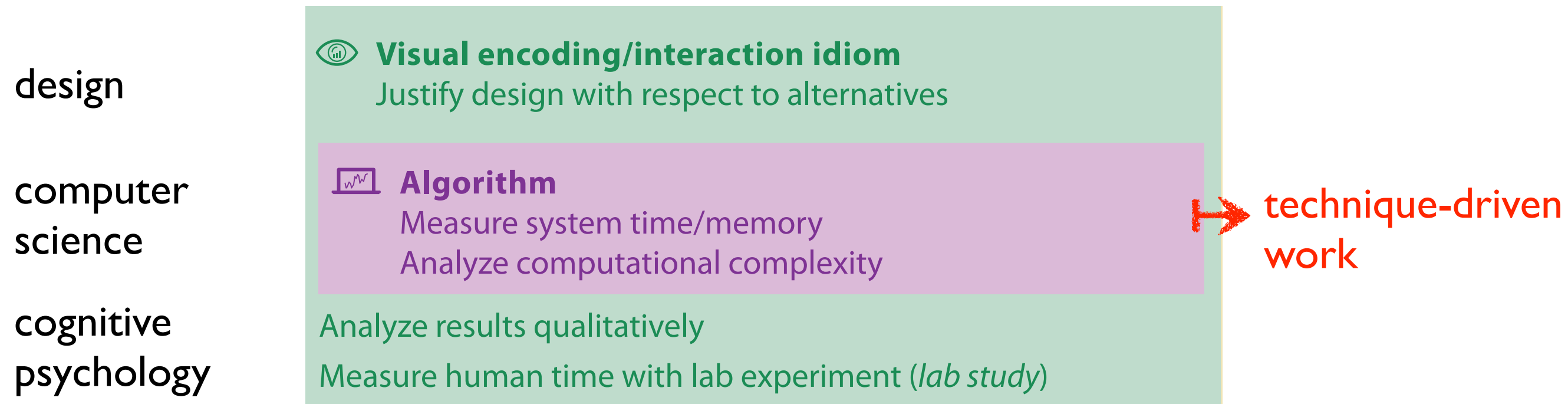
Analyze computational complexity



technique-driven  
work

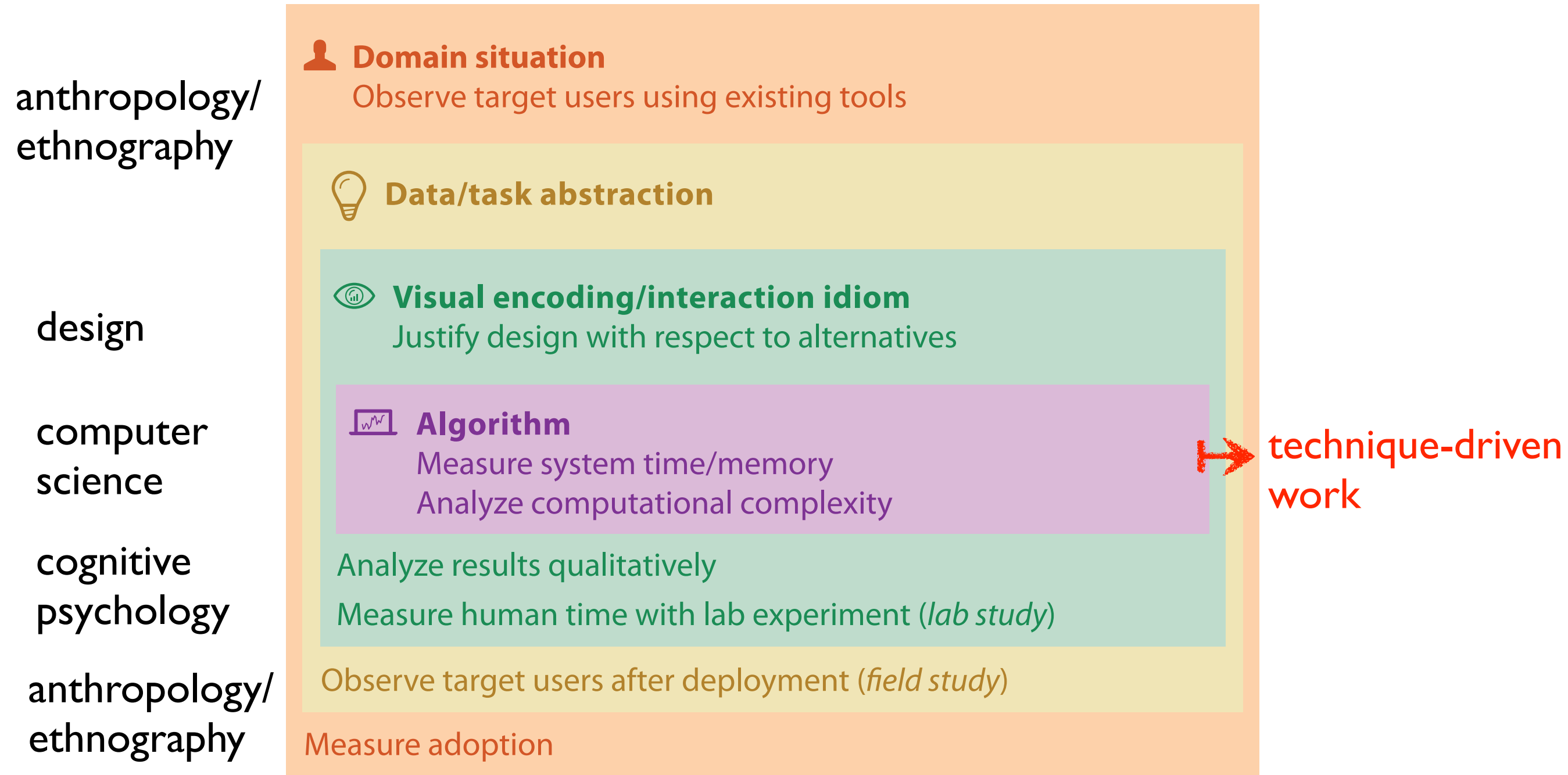
# Why is validation difficult?

- solution: use methods from different fields at each level



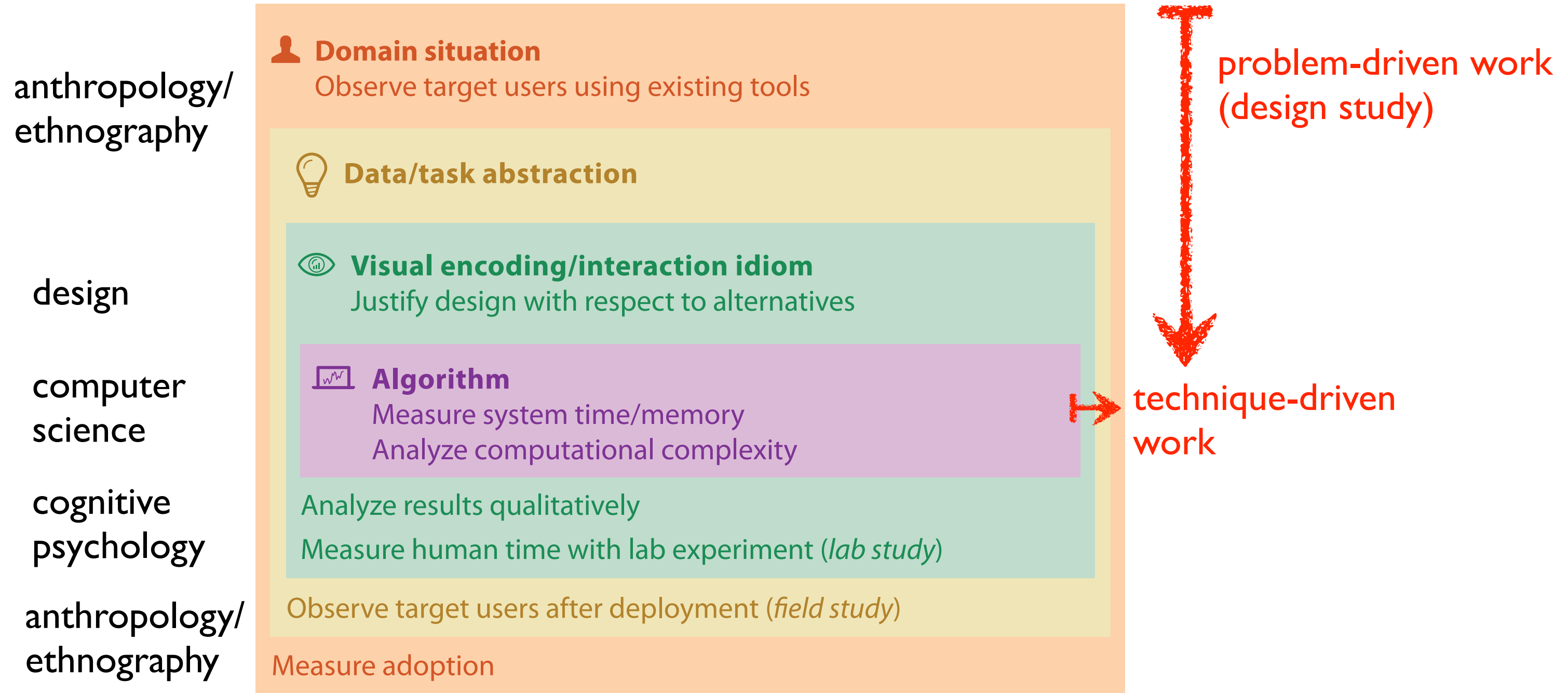
# Why is validation difficult?

- solution: use methods from different fields at each level

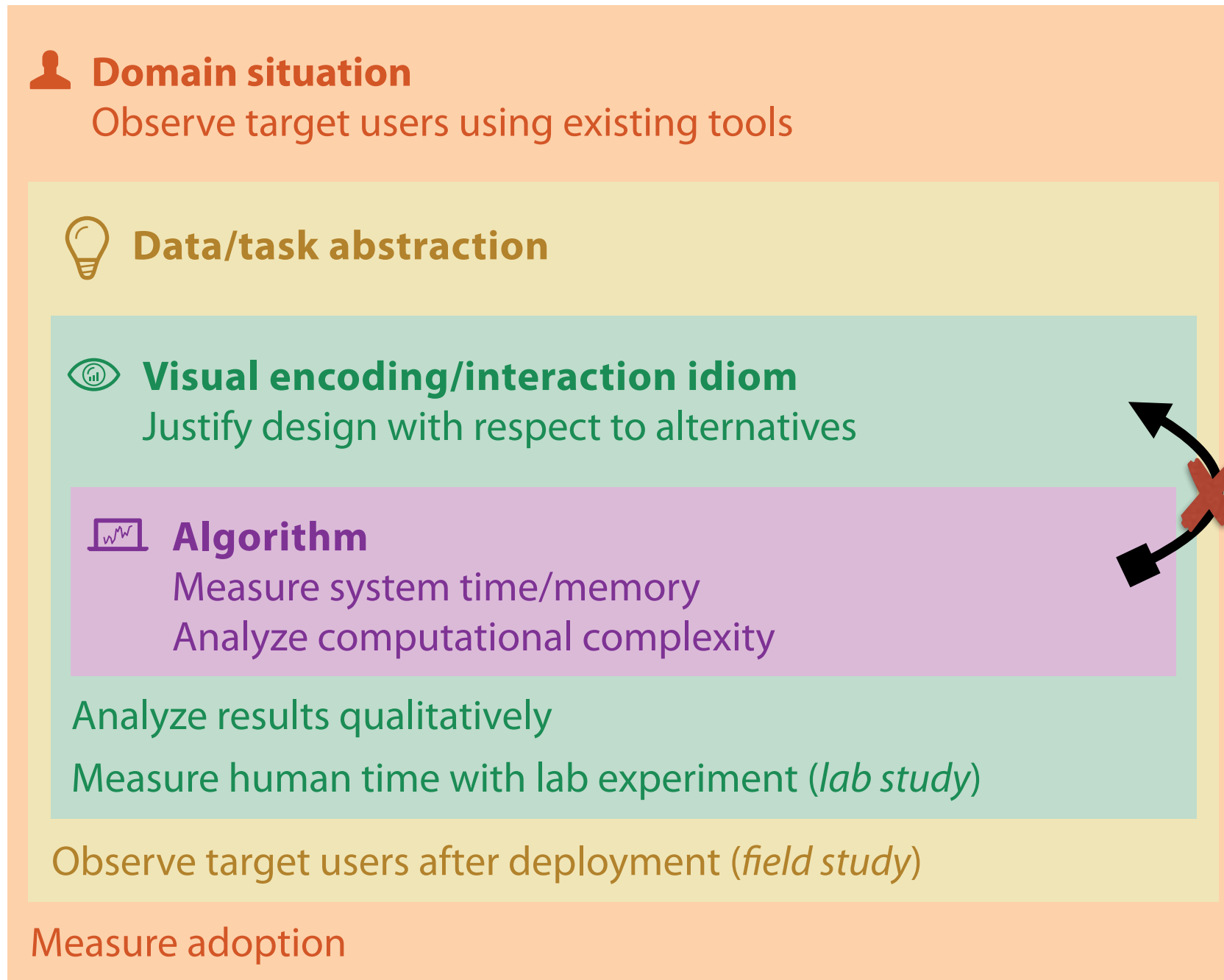


# Why is validation difficult?

- solution: use methods from different fields at each level

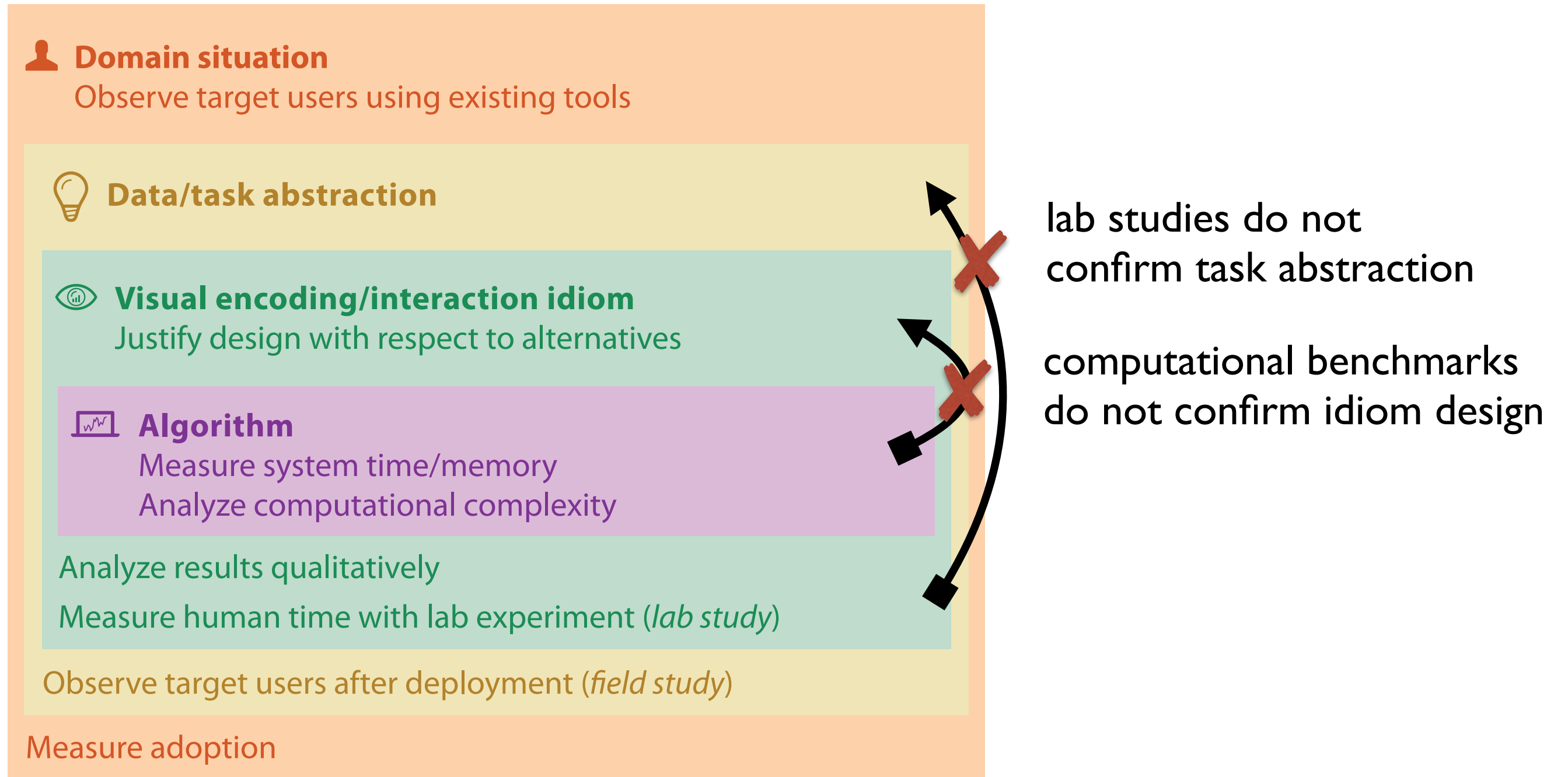


# Avoid mismatches



computational benchmarks  
do not confirm idiom design

# Avoid mismatches



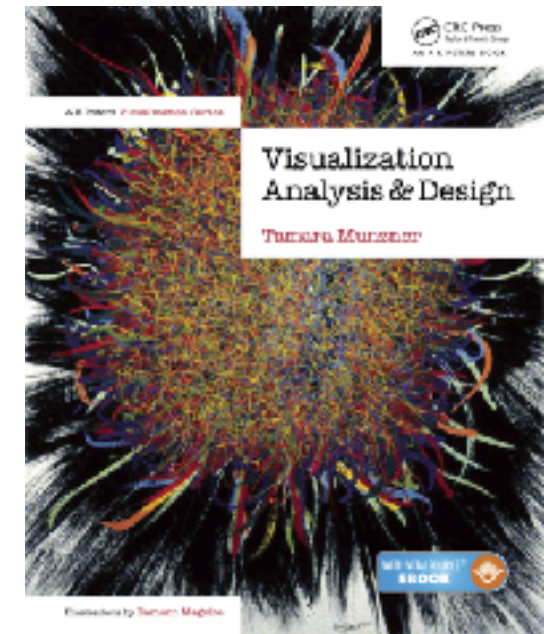
# Visualization Analysis & Design

## *Data Abstraction (Ch 2)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# What does data mean?



# What does data mean?

14, 2.6, 30, 30, 15, 100001

- What does this sequence of six numbers mean?

# 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?

# 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?

# 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??

# 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 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?

# 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 7th day of month

# 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 7th day of month
  - Basil Point neighborhood of city had 7 inches of snow cleared by the Pear Creek Limited snow removal service



# 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 7th day of month
  - Basil Point neighborhood of city had 7 inches of snow cleared by the Pear Creek Limited snow removal service
  - lab rat Basil made 7 attempts to find way through south section of maze, these trials used pear as reward food

# Now what?

- semantics: real-world meaning

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

# Now what?

- semantics: real-world meaning

<b>Name</b>	<b>Age</b>	<b>Shirt Size</b>	<b>Favorite Fruit</b>
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

# 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"

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"

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

# 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 patient
  - eg horsepower, make for car
  - "dependent variable"

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

# 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 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 relationship between two items
  - eg 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

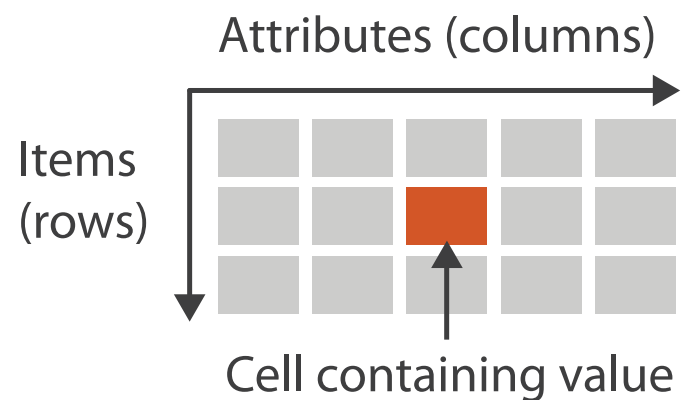
- flat table
  - one item per row
  - each column is attribute
  - cell holds value for item-attribute pair

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

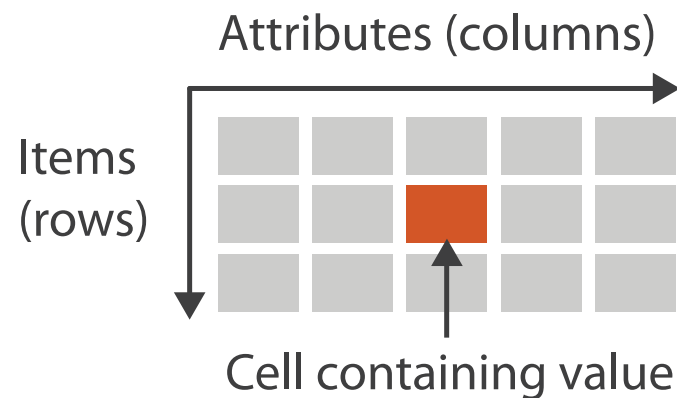
- flat table
  - one item per row
  - each column is attribute
  - cell holds value for item-attribute pair
  - unique key (could be implicit)

attributes: name, age, shirt size, fave fruit

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

item: person

→ Tables



# Table

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
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

# Table

item

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
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

# Table

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
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

item

attribute

# Table

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
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

item

cell

attribute

# Dataset types

Tables

Items

Attributes

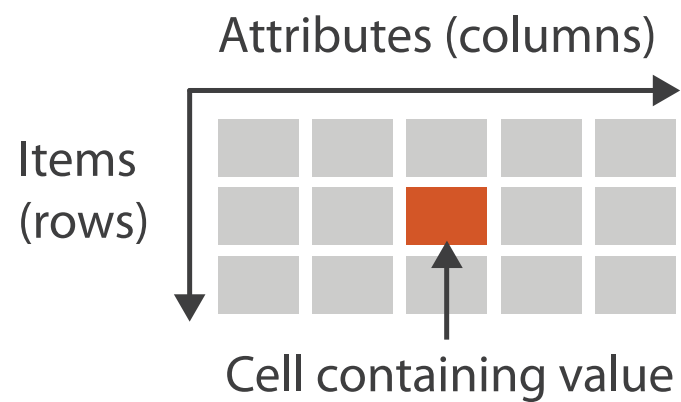
- multidimensional tables
  - indexing based on multiple keys
  - eg genes, patients

	A	B	C	D	E
1					
2	1	1	1	1	1
3	2	1	1	1	1
4	3	2	1	1	1
5	4	3	2	1	1
6	5	4	3	2	1
7	6	5	4	3	2
8	7	6	5	4	3
9	8	7	6	5	4
10	9	8	7	6	5
11	10	9	8	7	6
12	11	10	9	8	7
13	12	11	10	9	8
14	13	12	11	10	9
15	14	13	12	11	10
16	15	14	13	12	11
17	16	15	14	13	12
18	17	16	15	14	13
19	18	17	16	15	14
20	19	18	17	16	15
21	20	19	18	17	16
22	21	20	19	18	17

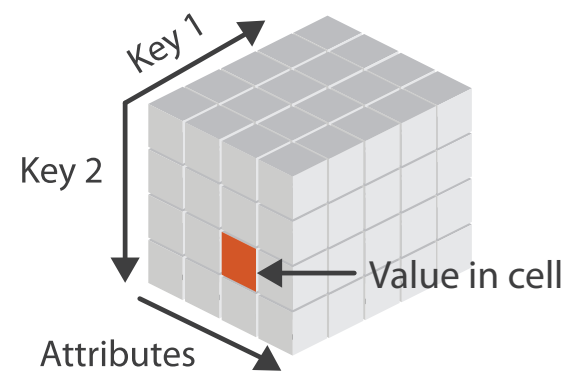
  

	A	B	C	D	E
1	#1	#2			
2	GeneName	DESCRIPTION	TCGA-02-0001-01C-01R-0177-01	TCGA-02-0003-01A-01R-0177-01	TCGA-02-0004-01A-01R-0298-01
3	LTF	LTF	-1.265728057	2.377012066	4.123979585
4	POSTN	POSTN	2.662411806	3.932400324	5.031585377
5	TMSL8	TMSL8	-3.082217838	-2.243148513	-0.02313581
6	HLA-DQA1	HLA-DQA1	-1.739664398	4.577962344	3.127744964
7	RP11-35N5.1	RP11-35N6.1	-3.345352958	-2.895400157	-3.473035067
8	STMN2	STMN2	-2.573511106	-3.051605144	-1.729892888
9	DCX	DCX	-2.25078975	-2.529795801	-2.844966278
10	AGXT2L1	AGXT2L1	-2.633493611	-3.113204863	-0.403975027
11	IL13RA2	IL13RA2	-2.93596915	-1.873600916	2.976256911
12	SLN	SLN	-2.465718221	-2.208406749	1.025827904
13	MEOX2	MEOX2	-2.395054056	1.062676046	1.783235317
14	COL11A1	COL11A1	1.211934832	-0.399392588	4.733608974
15	NNMT	NNMT	0.703745154	0.664082419	3.069030715
16	F13A1	F13A1	-0.229094012	2.222197544	1.171354775
17	CXCL14	CXCL14	-3.1309694	-1.395056071	2.569540659
18	MBP	MBP	-1.905390566	-2.037626447	-2.935744906
19	TF	TF	-4.334123292	-4.680680246	-2.975788866
20	KCND2	KCND2	-1.777692395	-2.100362021	-1.996306032

→ Tables



→ Multidimensional Table





# Dataset types

Tables

Items

Attributes

Networks &  
Trees

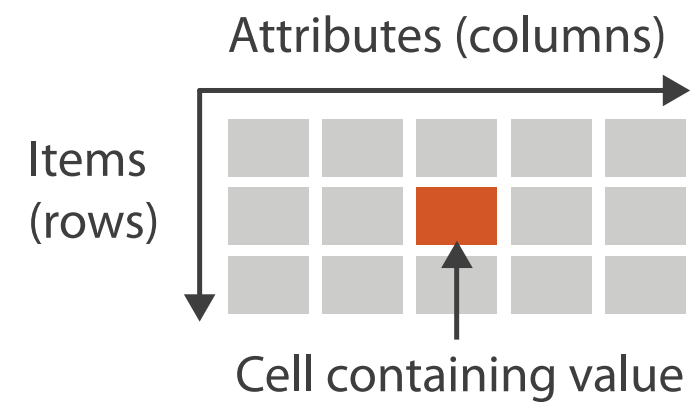
Items (nodes)

Links

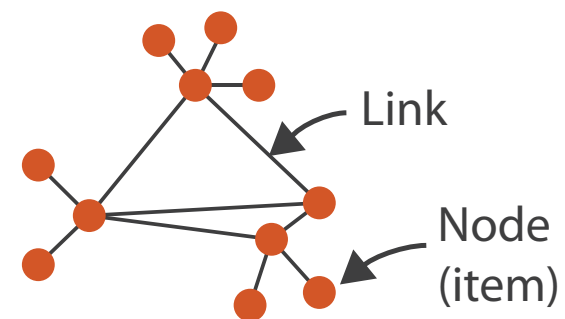
Attributes

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

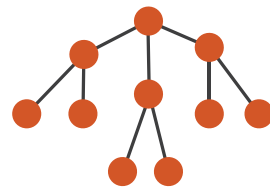
→ Tables



→ Networks



→ Trees



# Dataset types

Tables

Items

Attributes

Networks &  
Trees

Items (nodes)

Links

Attributes

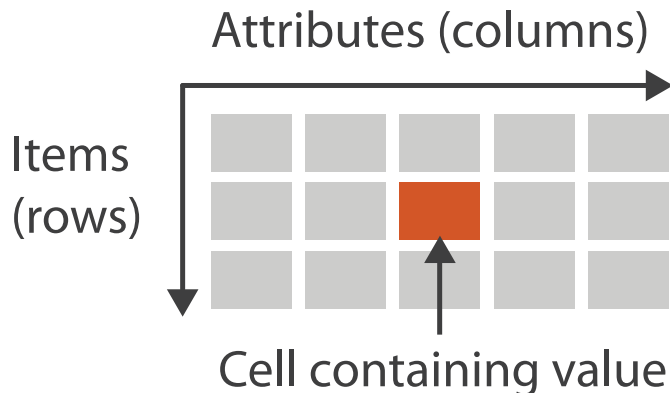
Fields

Grids

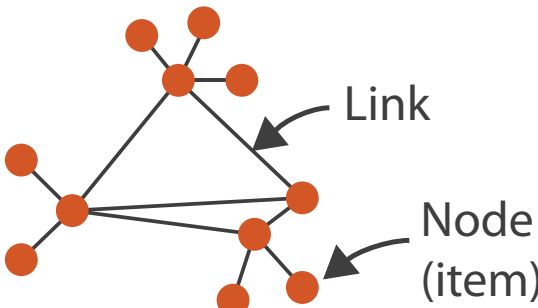
Positions

Attributes

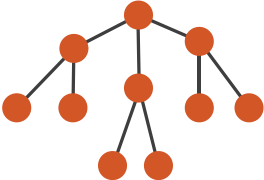
→ Tables



→ Networks

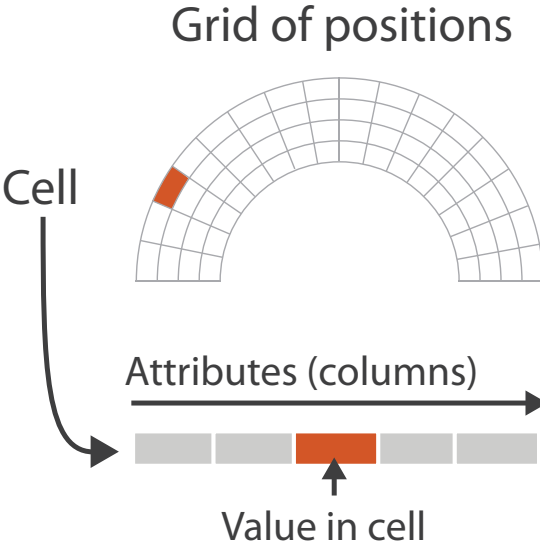


→ Trees



→ Spatial

→ Fields (Continuous)

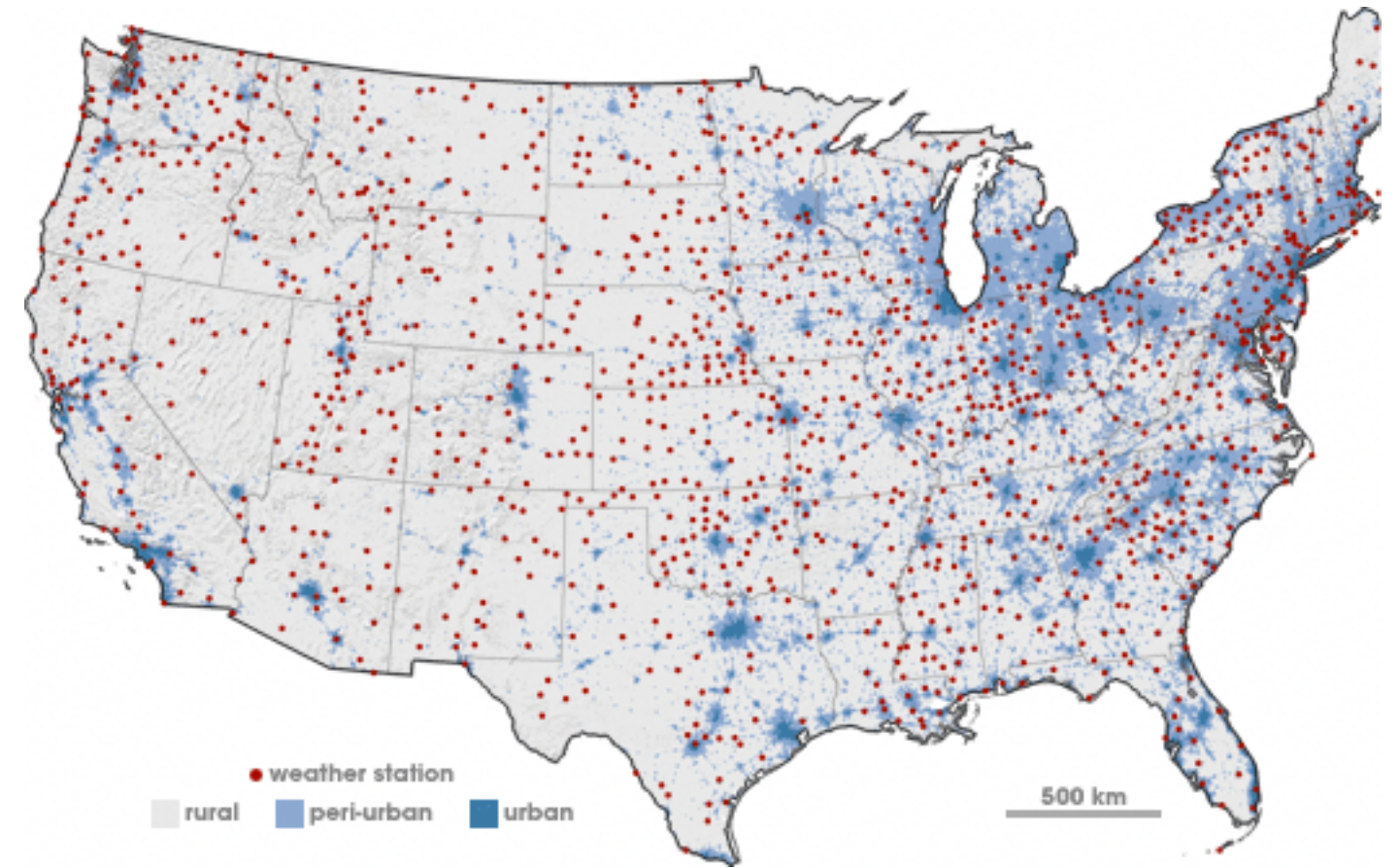
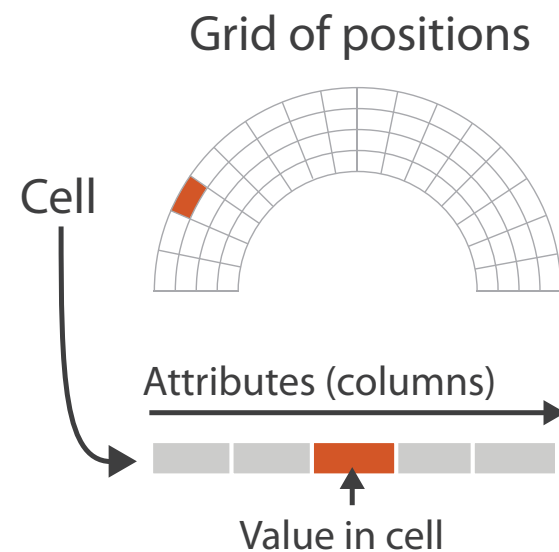


# Spatial fields

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

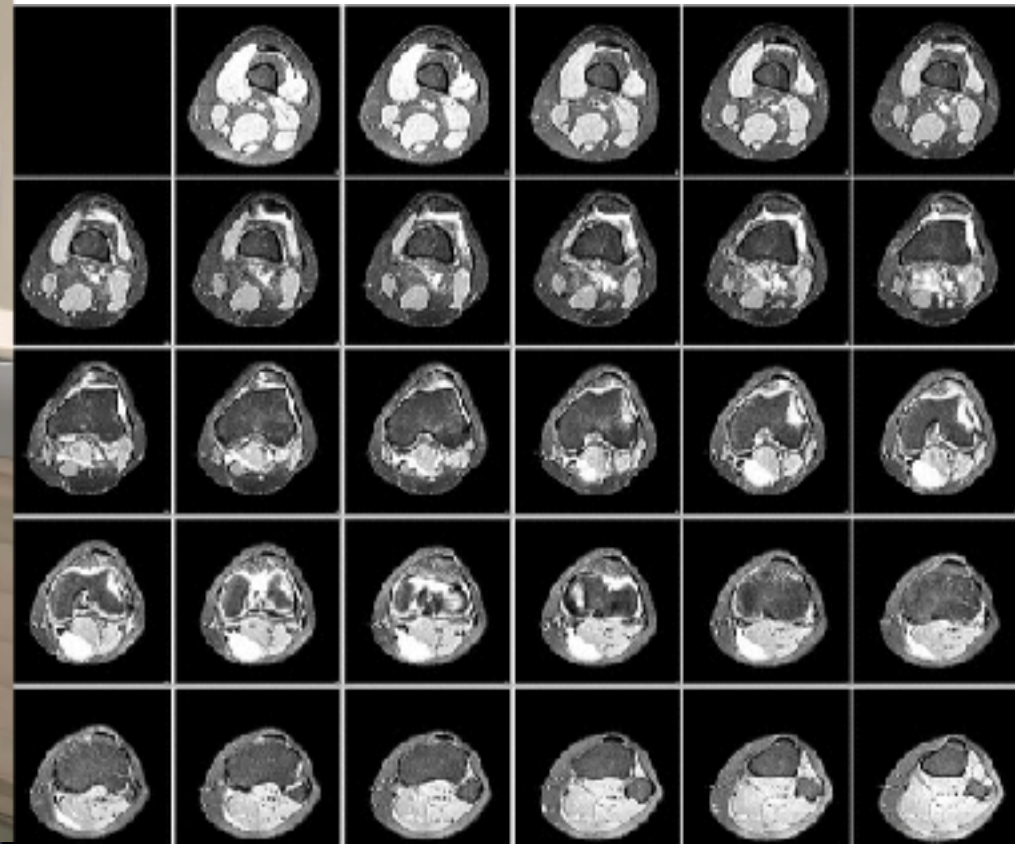
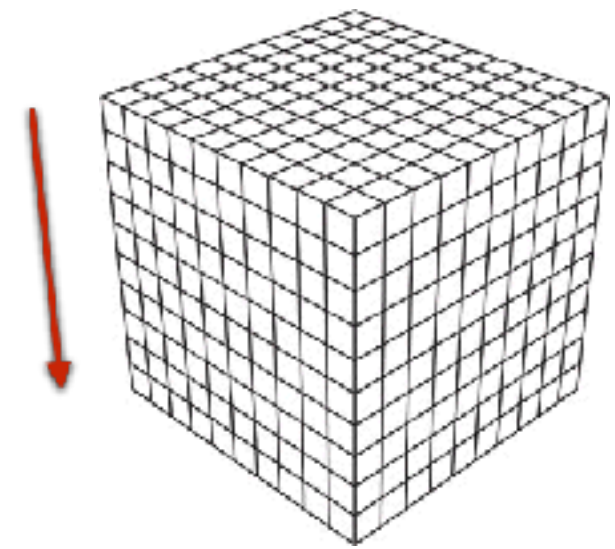
→ Spatial

→ Fields (Continuous)



# Spatial fields

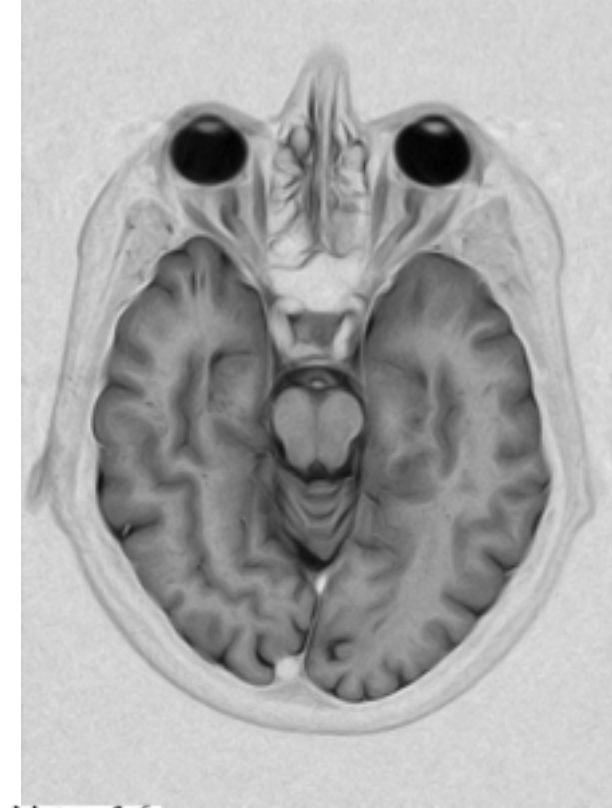
- attribute values associated w/ cells
- cell contains value from continuous domain
  - eg temperature, pressure, wind velocity
- measured or simulated
- major concerns
  - sampling:  
where attributes are measured
  - interpolation:  
how to model attributes elsewhere
  - grid types



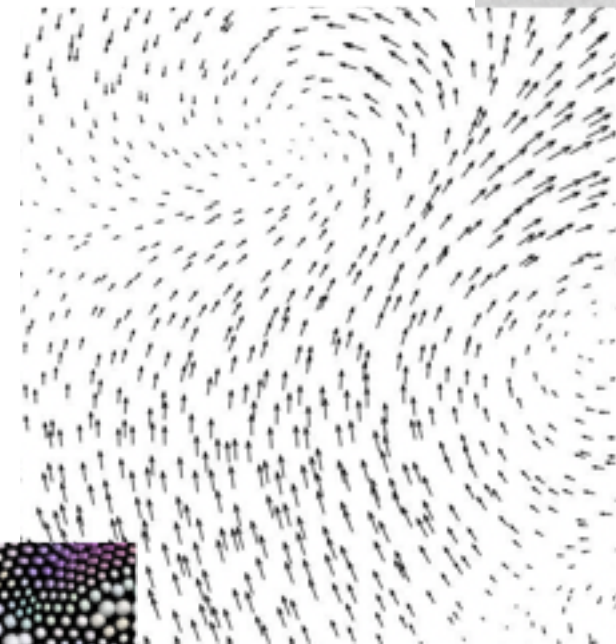
# Spatial fields

- 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)

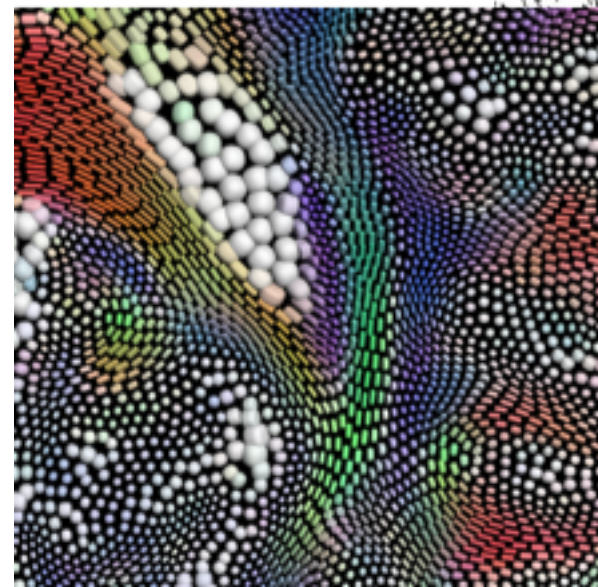
scalar



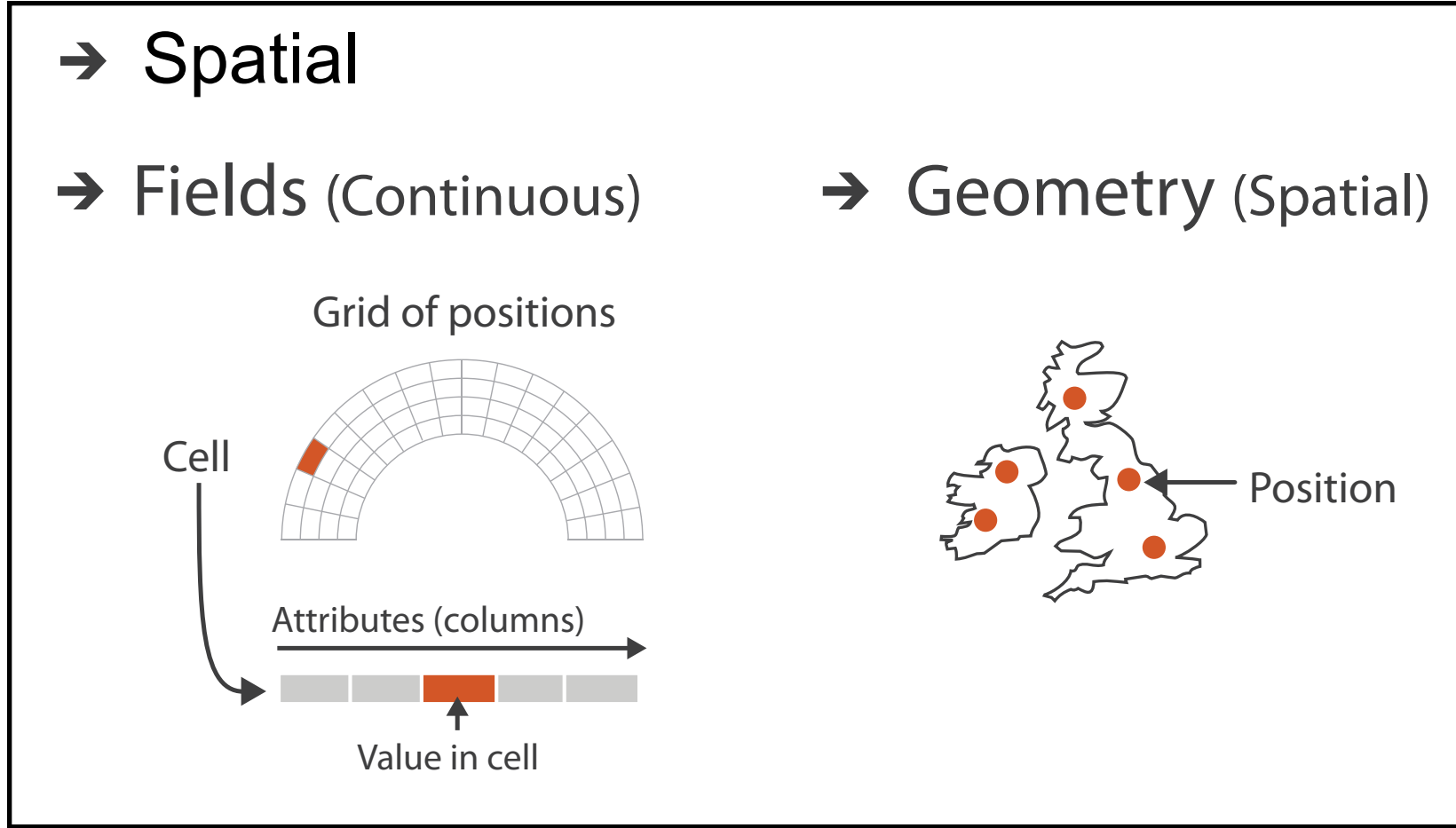
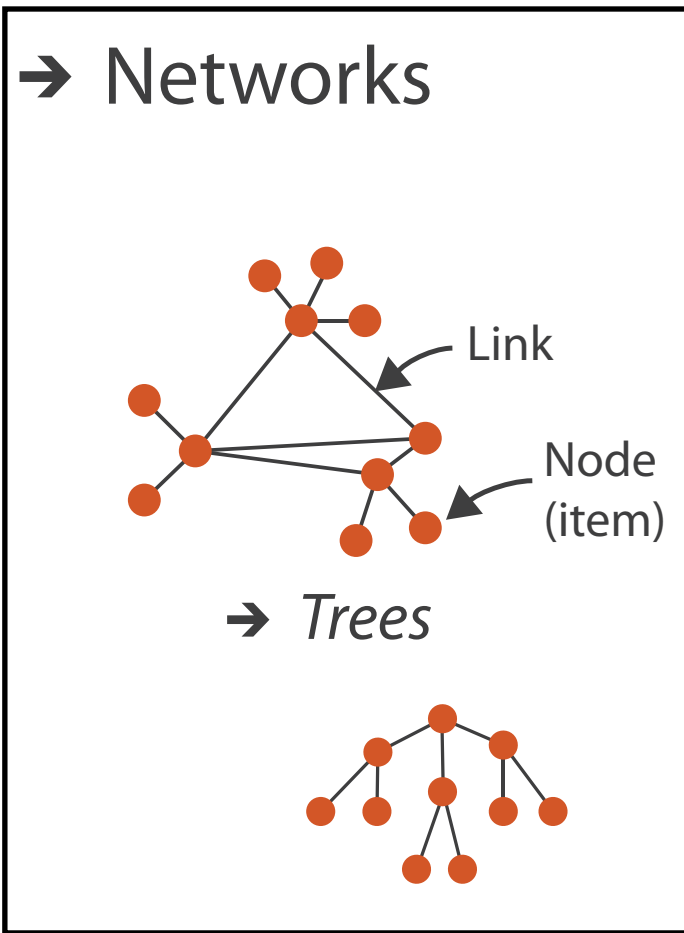
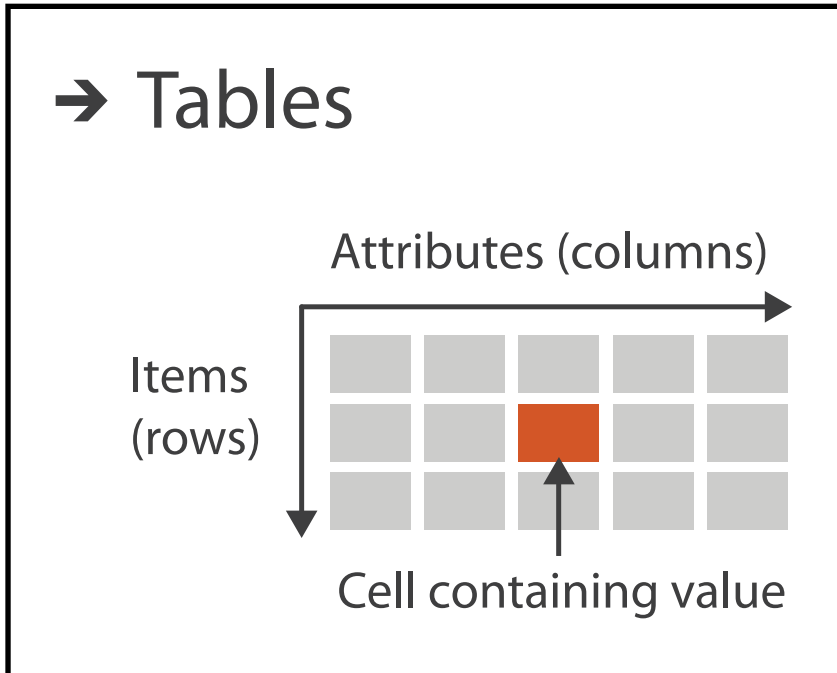
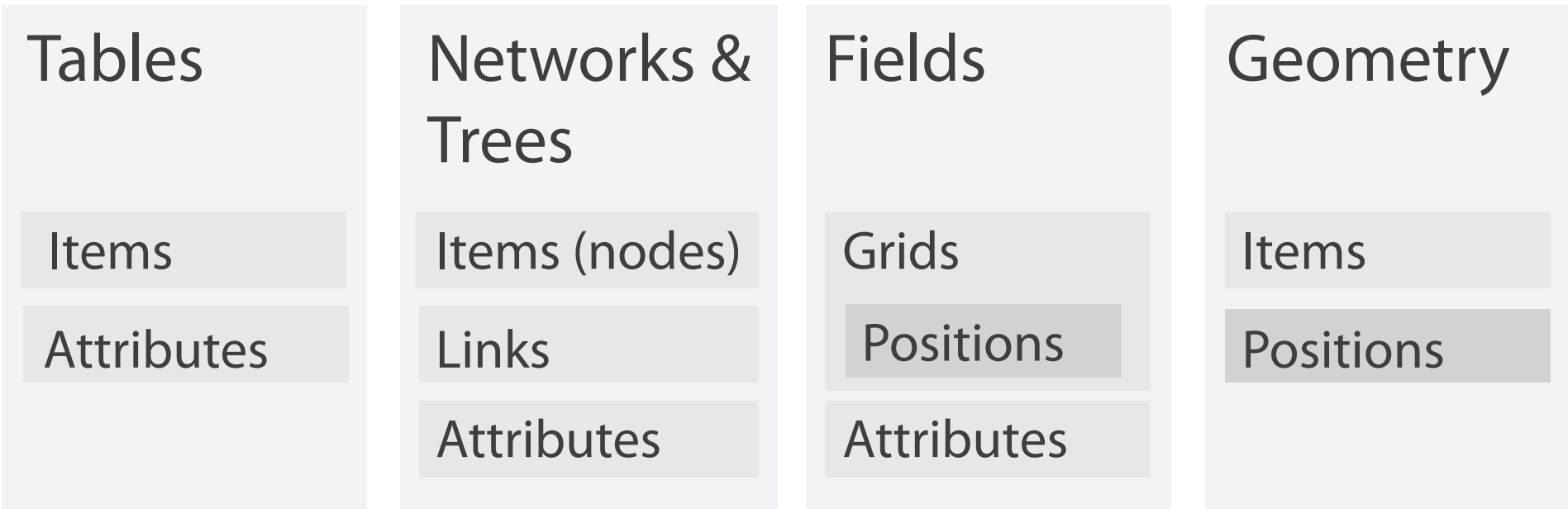
vector



tensor

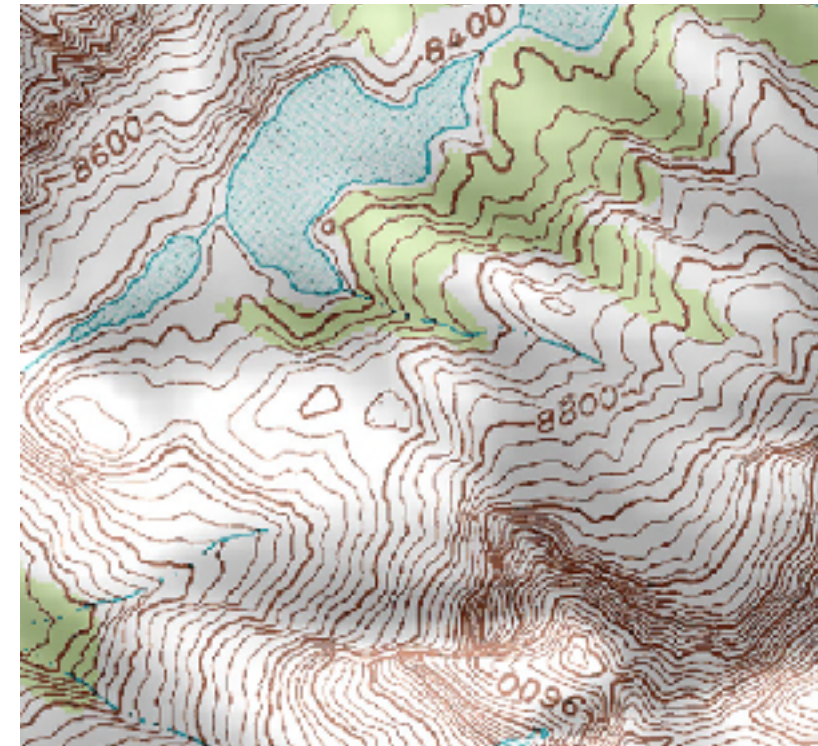


# Dataset types



# 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

Items

Attributes

Networks &  
Trees

Items (nodes)

Links

Attributes

Fields

Grids

Positions

Attributes

Geometry

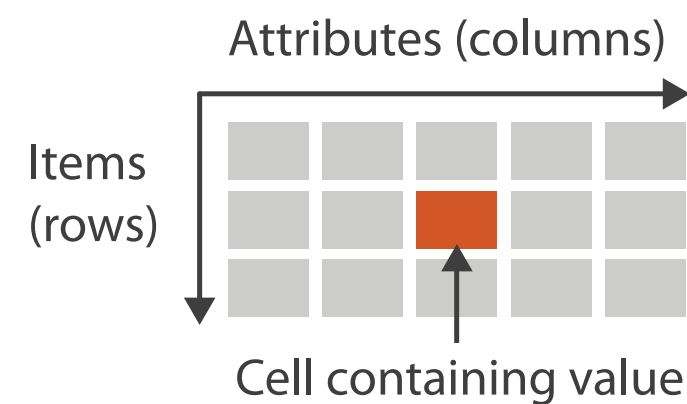
Items

Positions

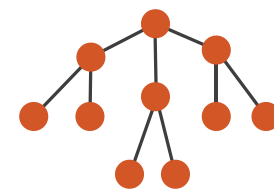
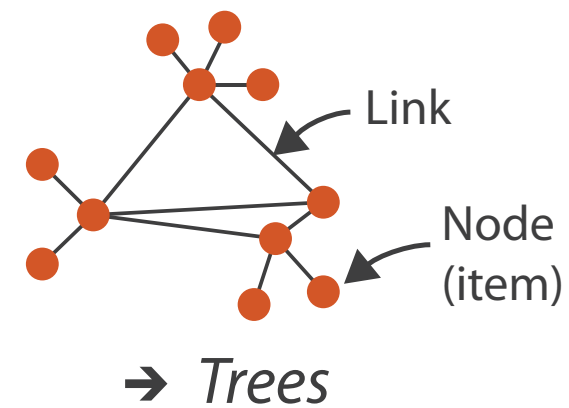
Clusters,  
Sets, Lists

Items

→ Tables

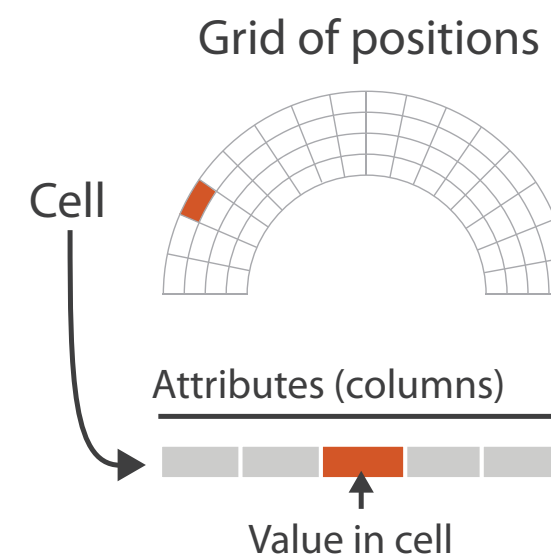


→ Networks



→ Spatial

→ Fields (Continuous)



→ Geometry (Spatial)



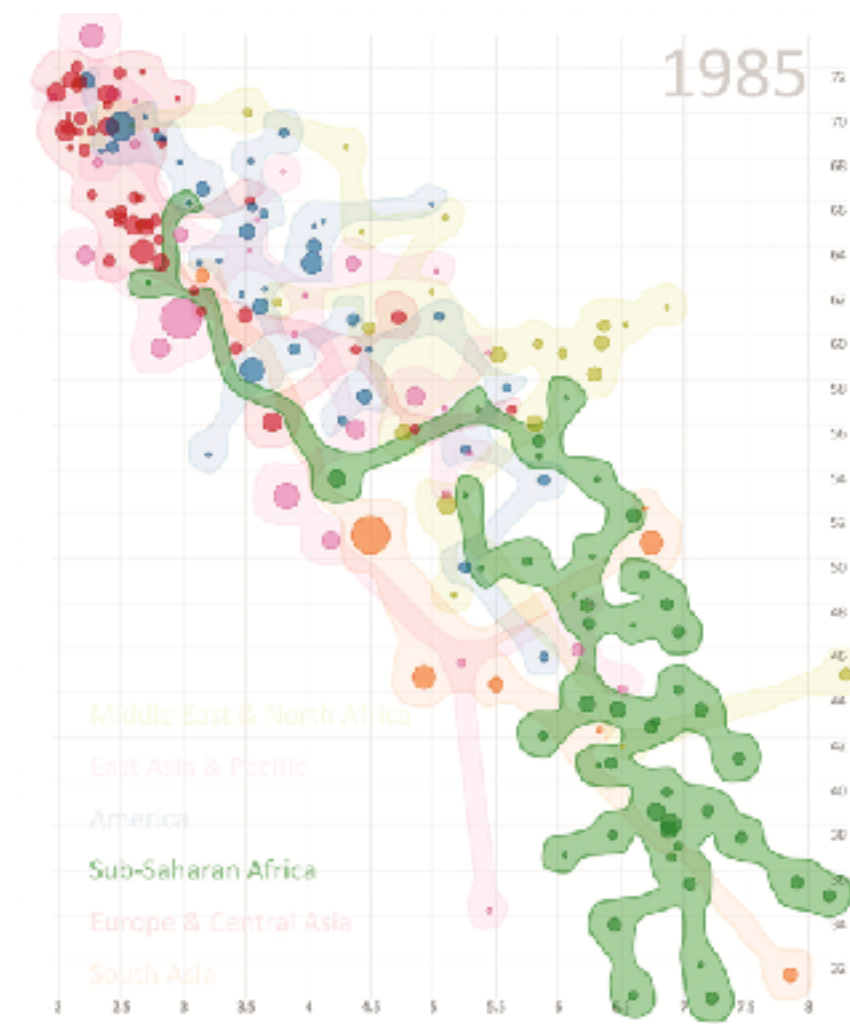


# Collections

- how we group items

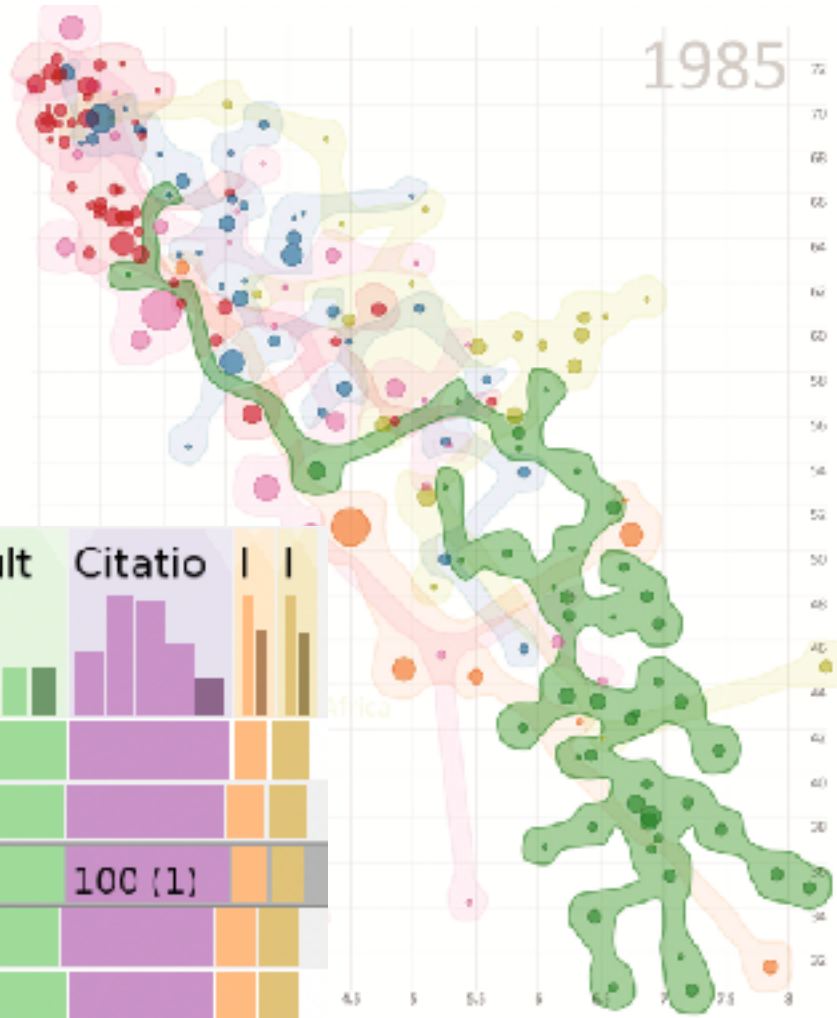
# Collections

- how we group items
- sets
  - unique items, unordered



# Collections

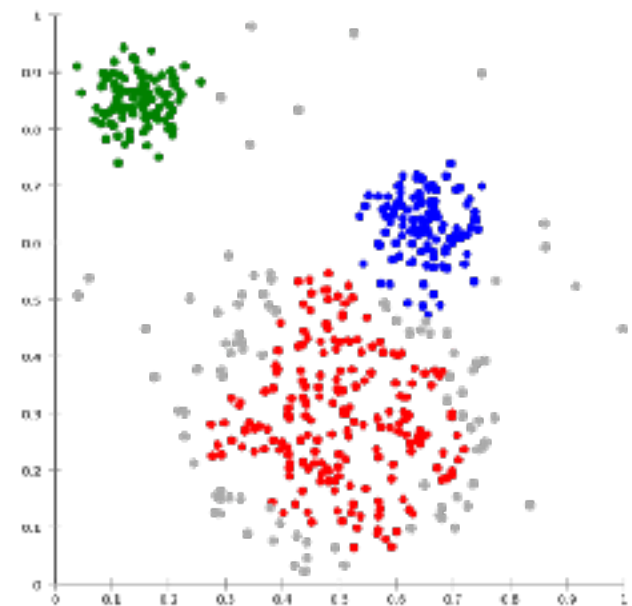
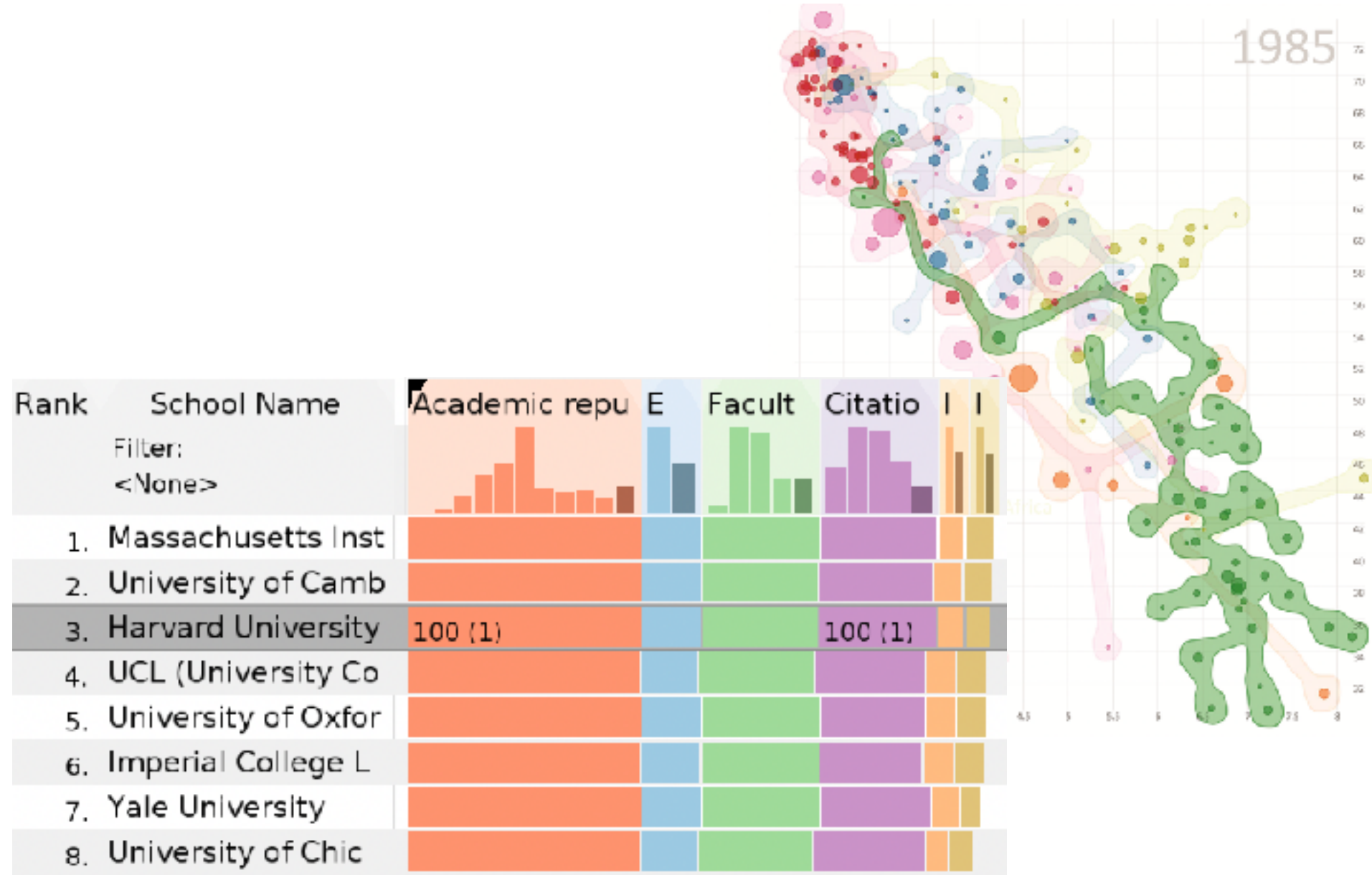
- how we group items
- sets
  - unique items, unordered
- lists
  - ordered, duplicates possible



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						

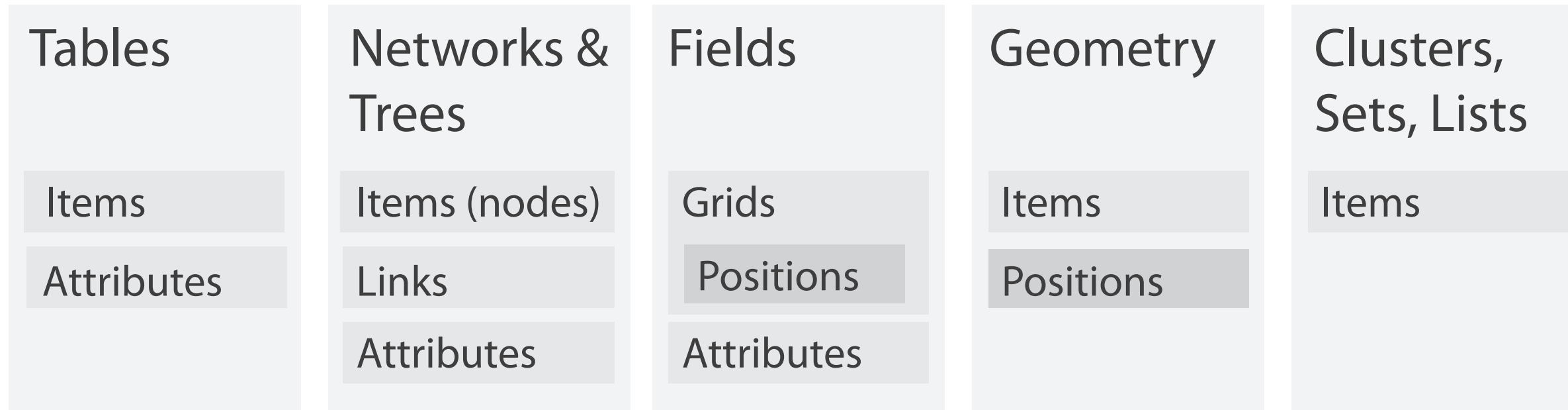
# Collections

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



# Dataset and data types

## → Data and Dataset Types



## → 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

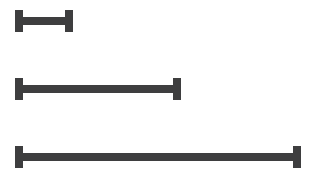


➔ Ordered

➔ *Ordinal*



➔ *Quantitative*



# Table

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
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

**categorical**  
**ordinal**  
**quantitative**

A	B	C	S	I	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

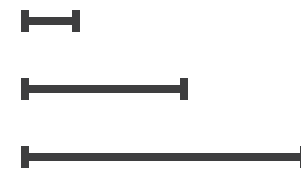


➔ Ordered

➔ Ordinal



➔ Quantitative



## ➔ Ordering Direction

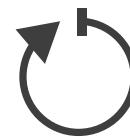
➔ Sequential



➔ 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 vs conceptual model, example

- data model: floats
  - 32.52, 54.06, -14.35, ...

# Data vs conceptual model, example

- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature

# Data vs conceptual model, example

- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions

# 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



# Data vs conceptual model, example

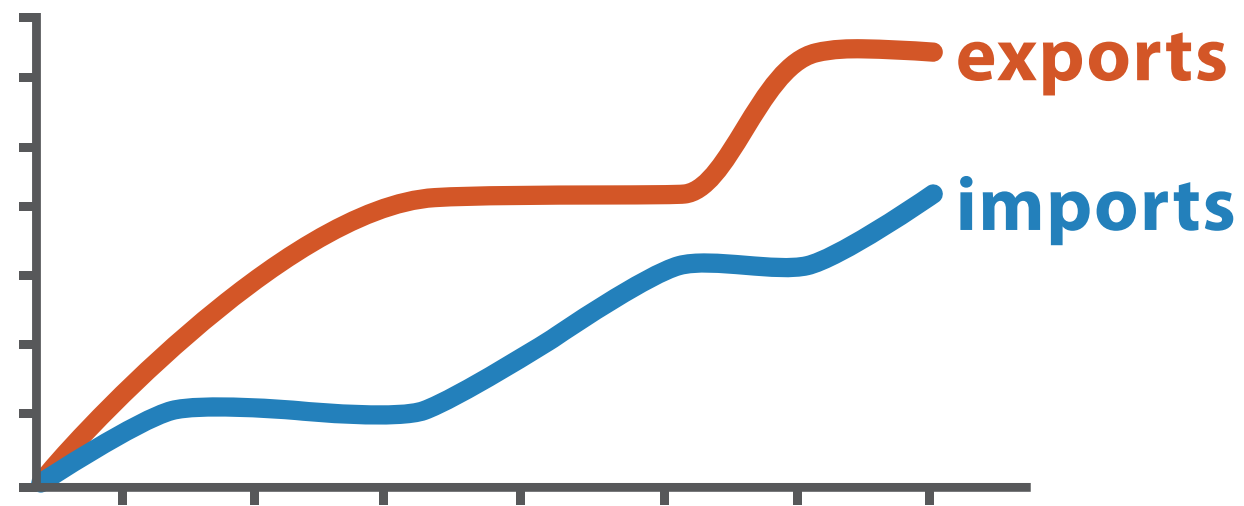
- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready

# Data vs conceptual model, example

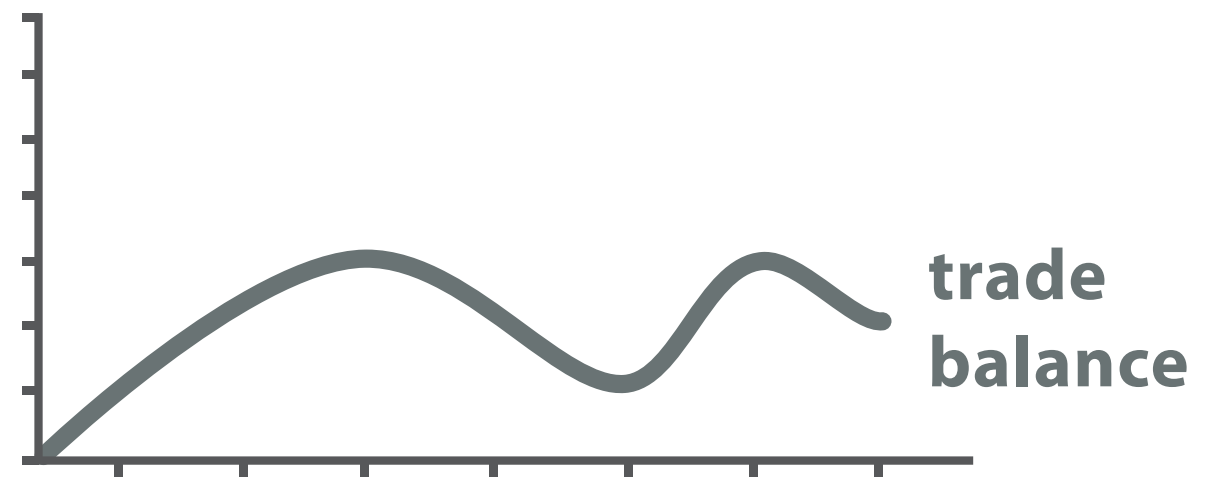
- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready
  - above freezing,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



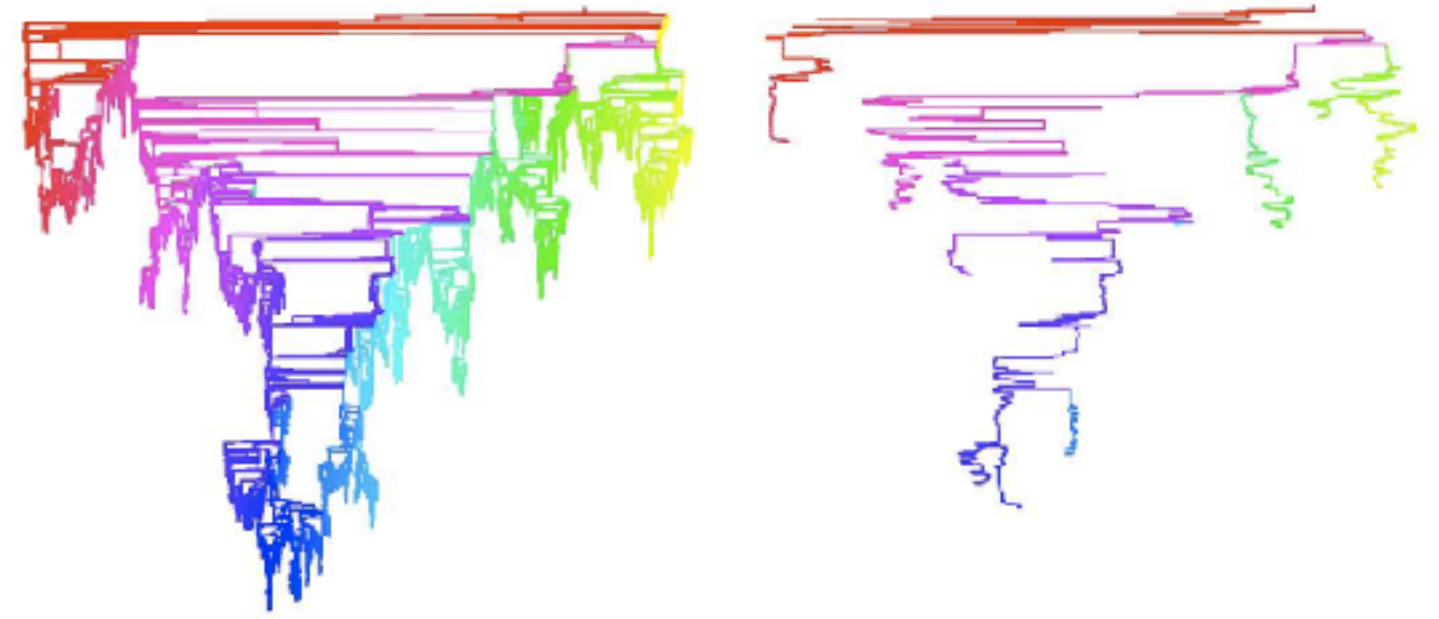
$$\text{trade balance} = \text{exports} - \text{imports}$$

Derived Data

# Analysis example: Derive one attribute

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

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



**Task 1**

**In Tree** → **Out Quantitative attribute on nodes**

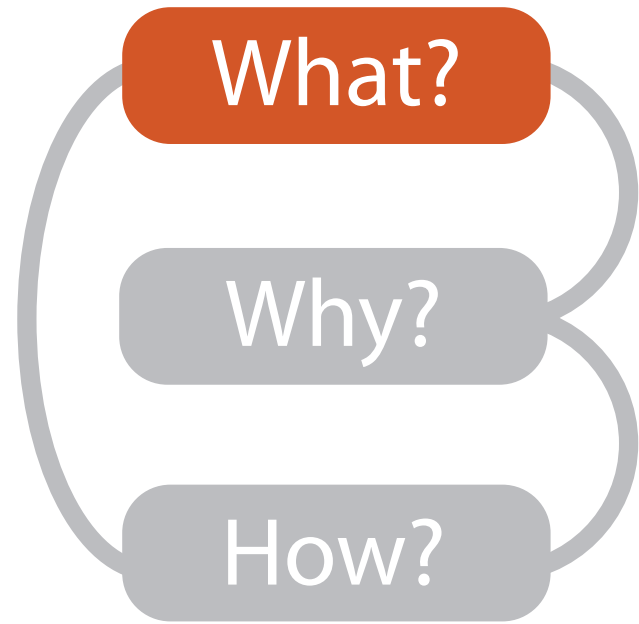
What?	Why?
→ In Tree	→ Derive
→ Out Quantitative attribute on nodes	

**Task 2**

**In Tree** + **In Quantitative attribute on nodes** → **Out Filtered Tree**

Removed unimportant parts

What?	Why?	How?
→ In Tree	→ Summarize	→ Reduce
→ In Quantitative attribute on nodes	→ Topology	→ Filter
→ Out Filtered Tree		



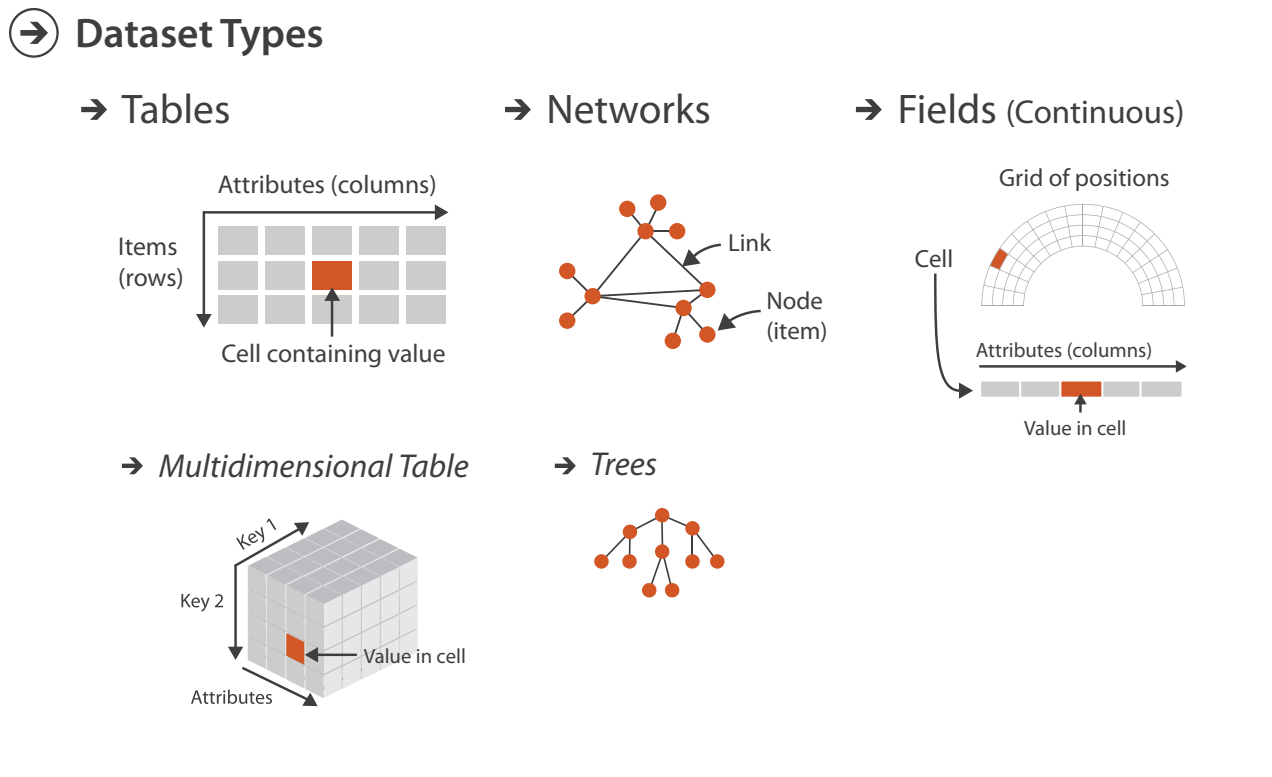
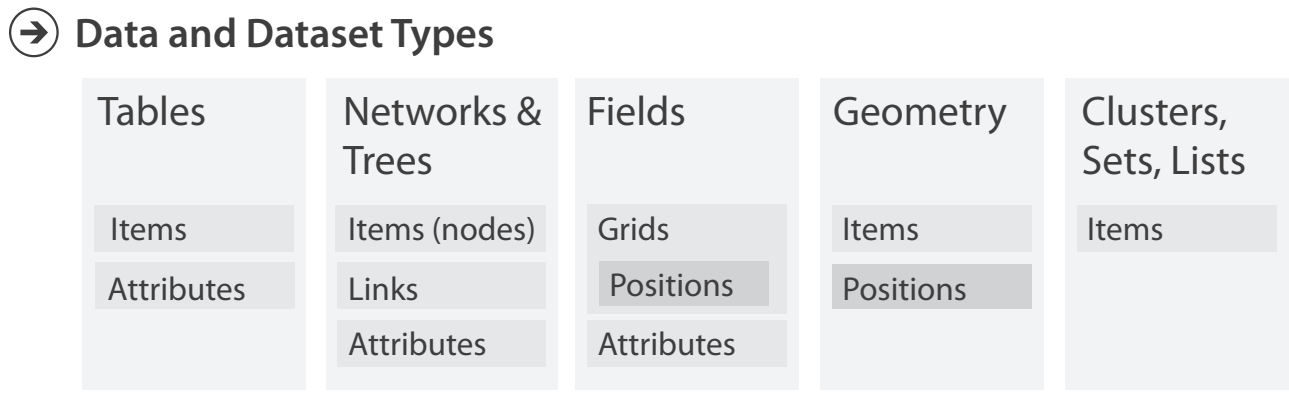
# What?

## Datasets

## Attributes

- Data Types
  - Items
  - Attributes
  - Links
  - Positions
  - Grids

- Attribute Types
  - Categorical
    - + ● ■ ▲
  - Ordered
    - Ordinal
      - 👕 👕 👕
    - Quantitative
      - ┆ ┆ ┆



- Ordering Direction
  - Sequential
    -
  - Diverging
    - ←→
  - Cyclic
    - ↻



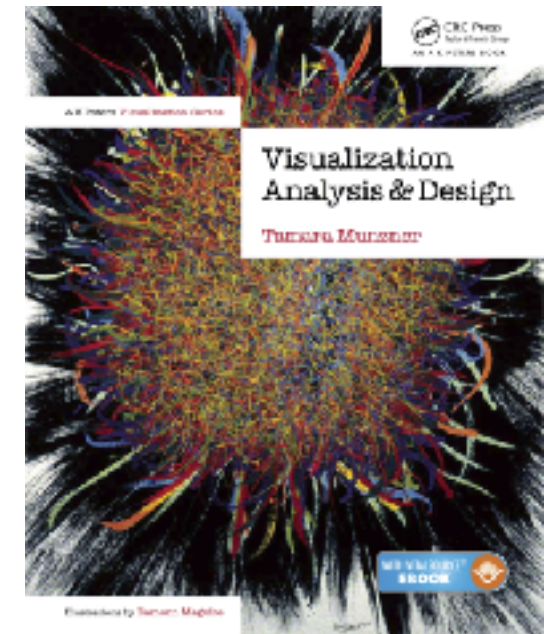
# Visualization Analysis & Design

## *Task Abstraction (Ch 3)*

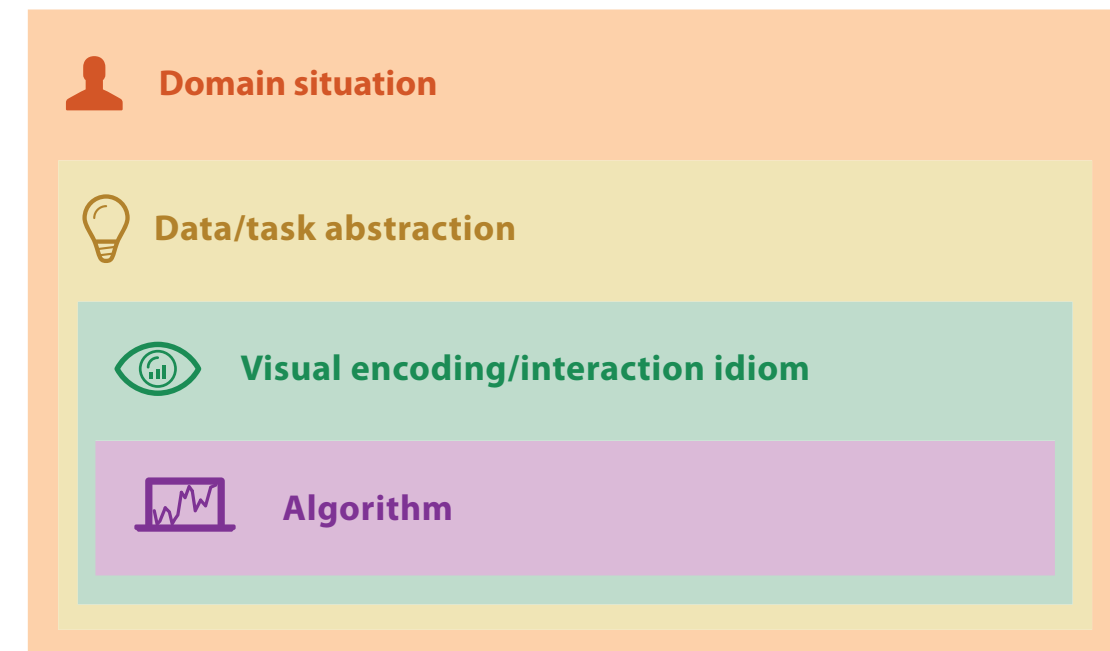
**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)

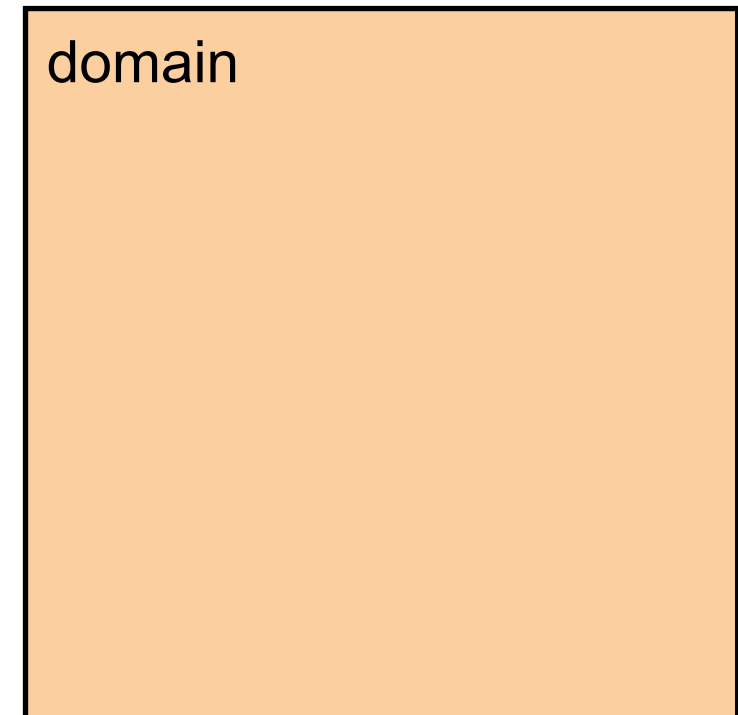
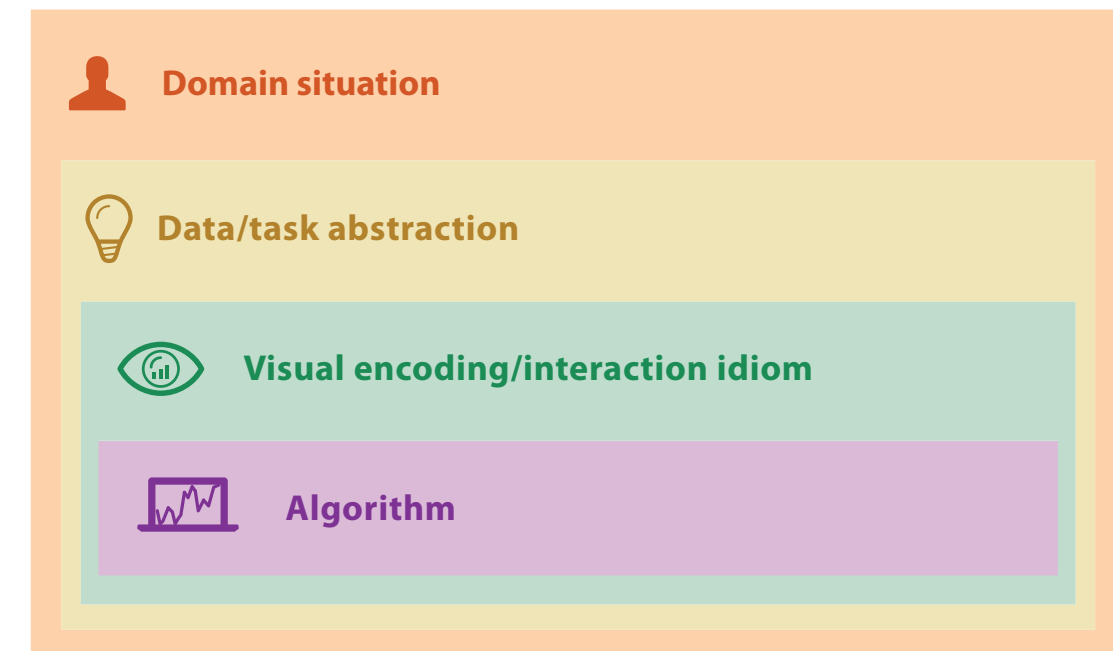


# From domain to abstraction



# From domain to abstraction

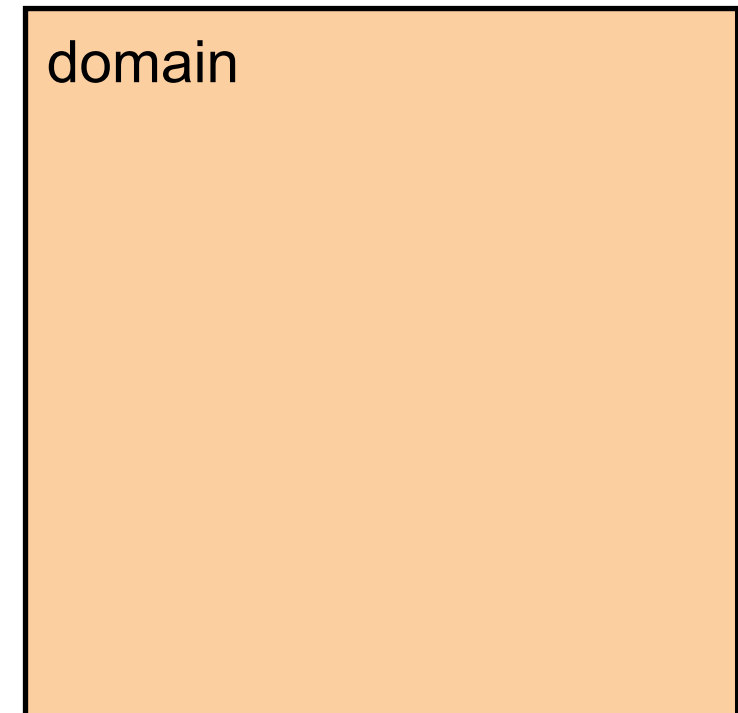
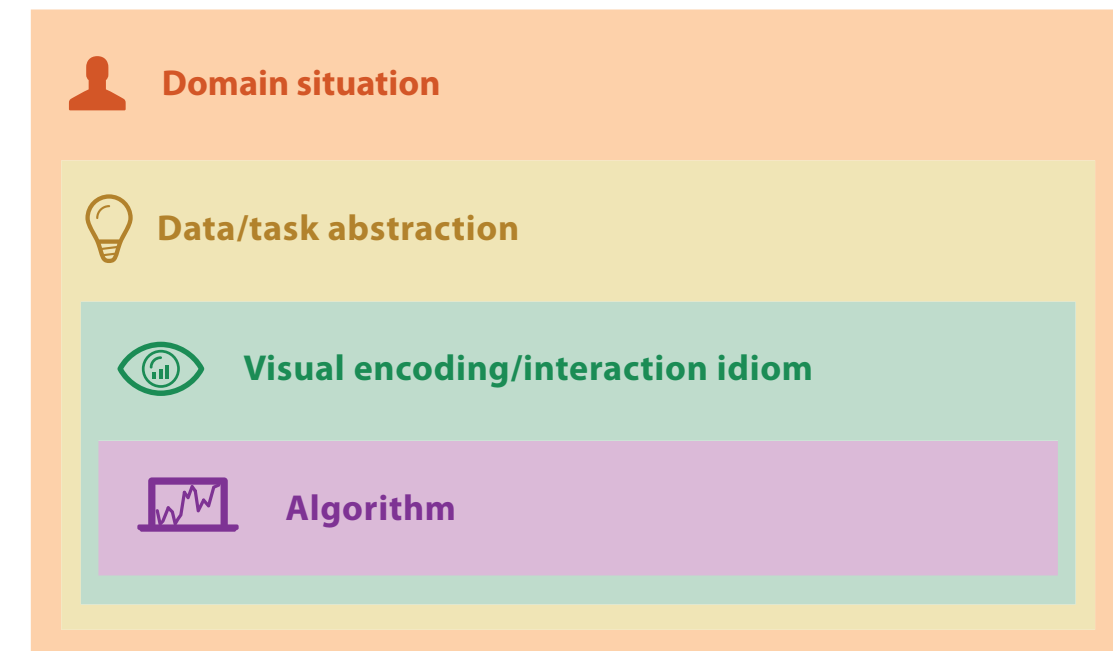
- domain characterization:  
details of application domain





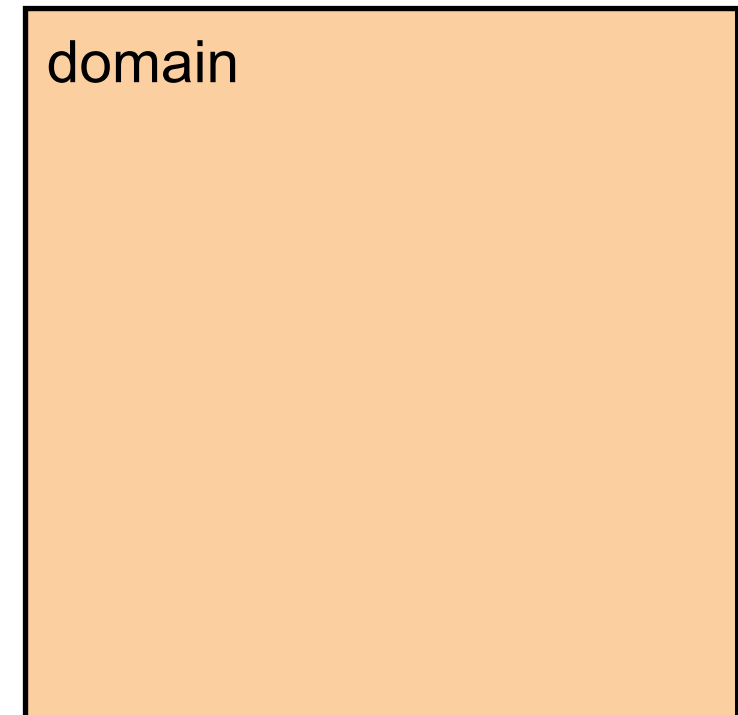
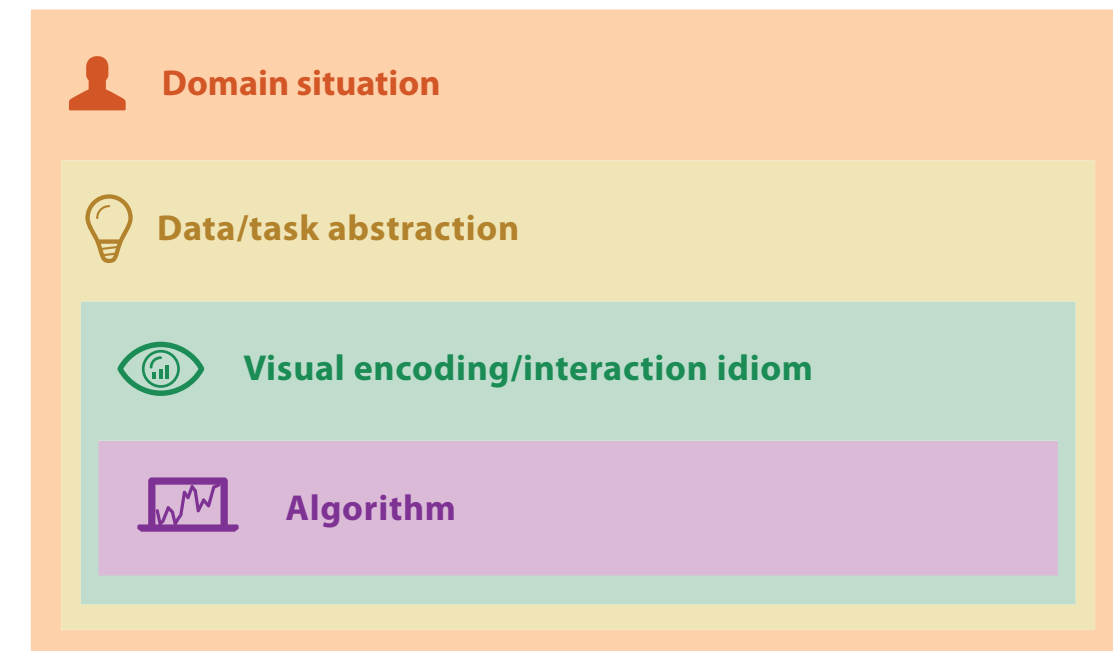
# From domain to abstraction

- 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



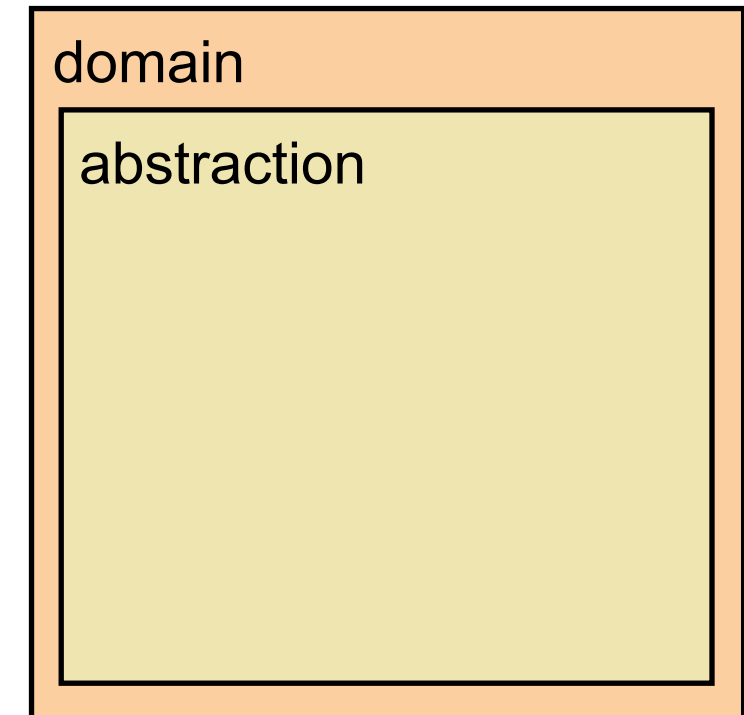
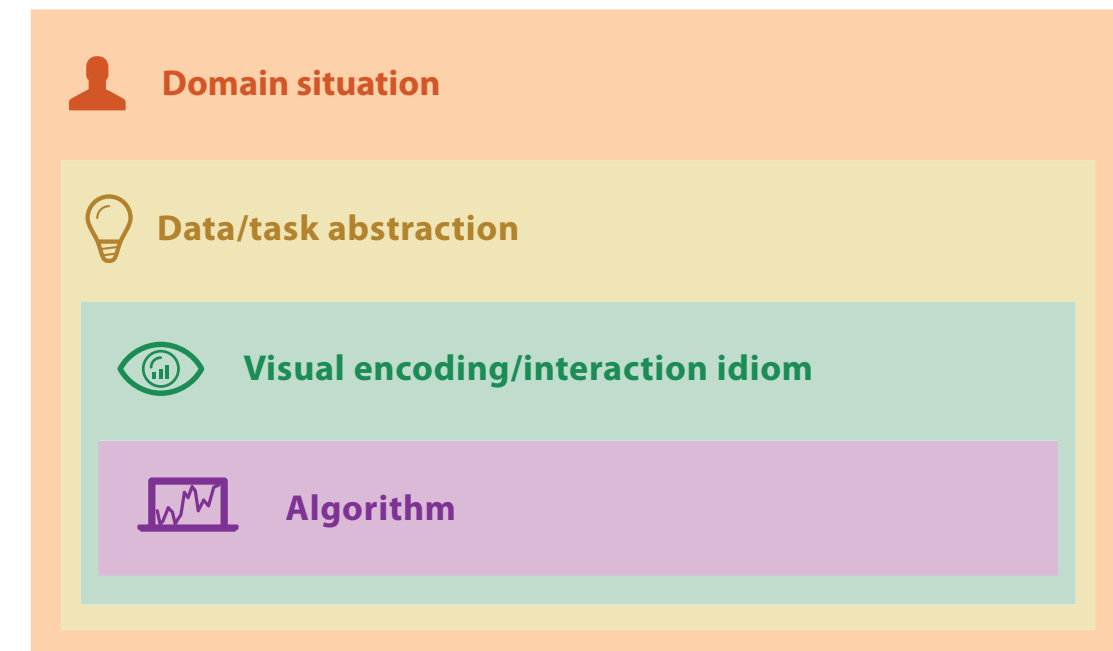
# From domain to abstraction

- 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



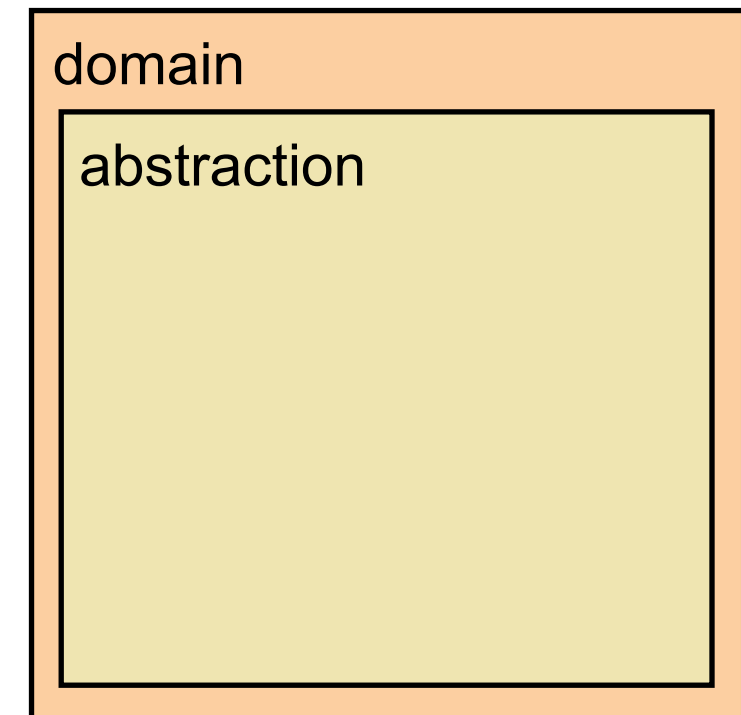
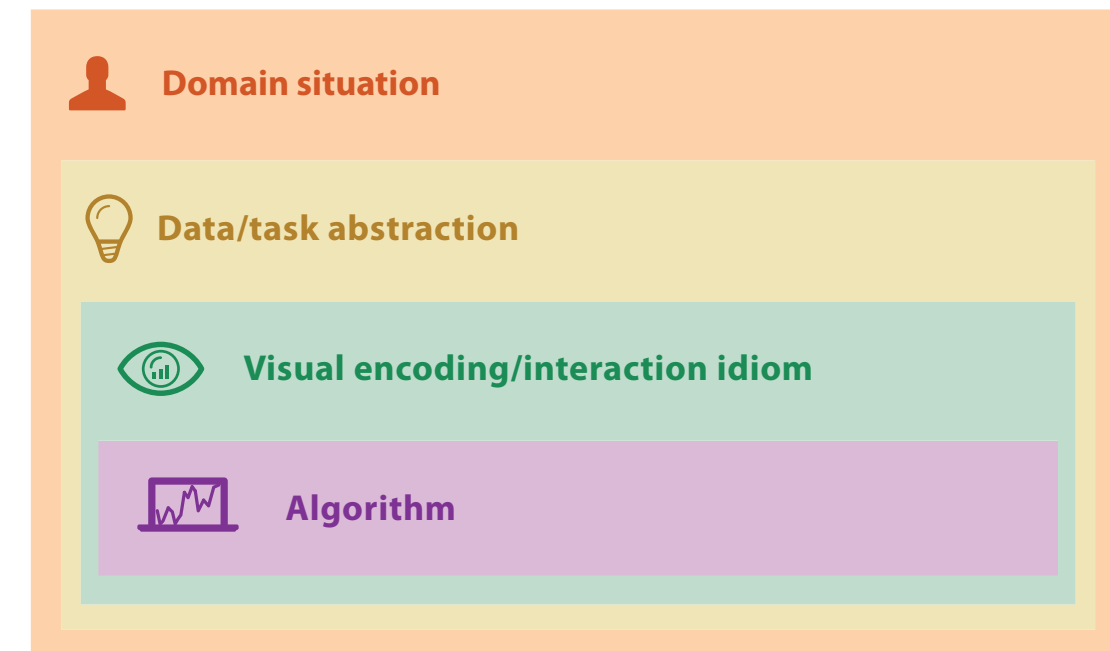
# From domain to abstraction

- 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

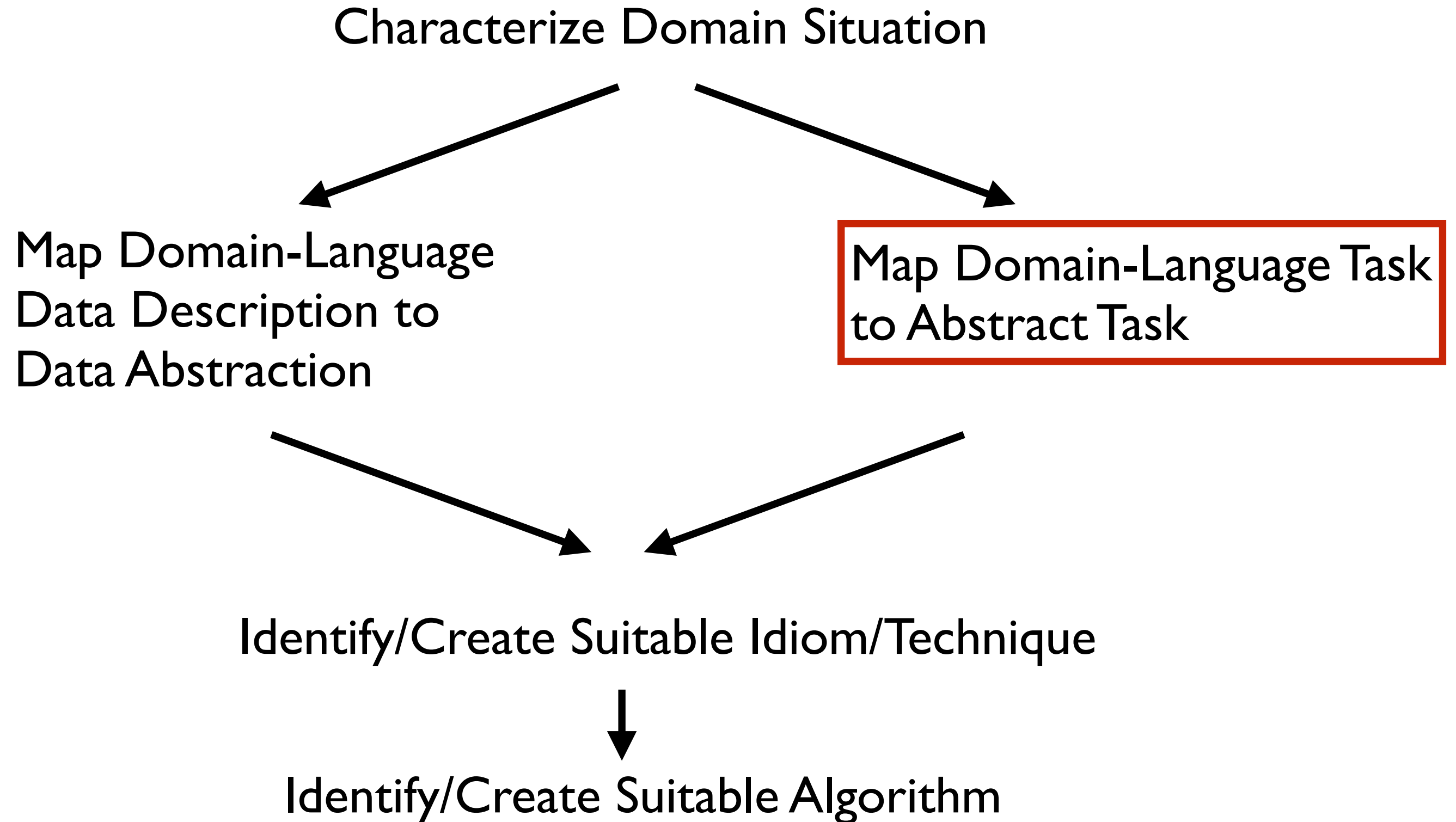


# From domain to abstraction

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

# Task abstraction: Actions and targets

- very high-level pattern
- actions
  - analyze
    - high-level choices
  - search
    - find a known/unknown item
  - query
    - find out about characteristics of item
- {action, target} pairs
  - *discover distribution*
  - *compare trends*
  - *locate outliers*
  - *browse topology*

# 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
- produce
  - newcomer
  - aka casual, social
- produce
  - annotate, record
  - derive
    - crucial design choice

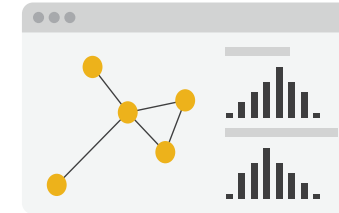
## ➔ Analyze

### ➔ Consume

➔ *Discover*



➔ *Present*

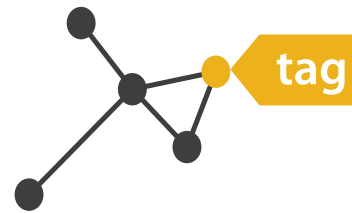


➔ *Enjoy*



### ➔ Produce

➔ *Annotate*



➔ *Record*



➔ *Derive*







# Actions: Search

# Actions: Search

- what does user know?
  - target, location





## ➔ Search

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

# Actions: Search

- what does user know?
  - target, location
- lookup
  - ex: word in dictionary
    - alphabetical order





## ➔ Search

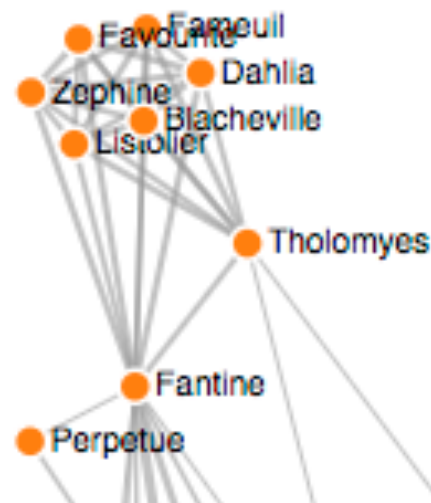
	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>

# 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

## ➔ Search

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







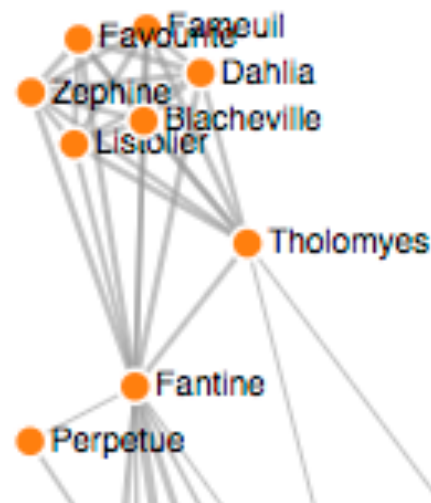
<https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8>

# 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

## ➔ Search

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







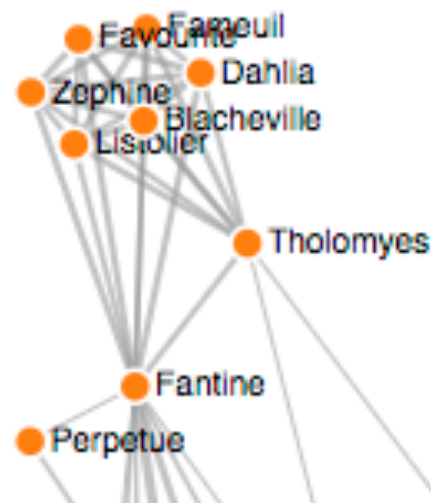
<https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8>

# 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>



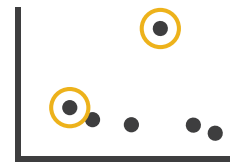
<https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8>

# Actions: Query

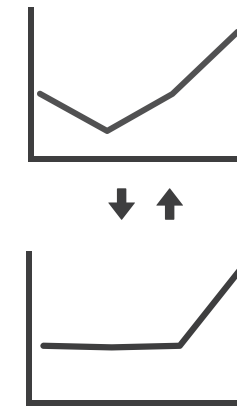
- how much of the data matters?
  - one: identify
  - some: compare
  - all: summarize

## → Query

→ Identify



→ Compare



→ Summarize





# Actions

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

## 👉 Actions

### ➔ Analyze

➔ Consume

➔ Discover



➔ Present



➔ Enjoy



➔ Produce

➔ Annotate



➔ Record



➔ Derive

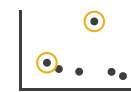


### ➔ Search

	Target known	Target unknown
Location known	•••• Lookup	•••• Browse
Location unknown	<••••> Locate	<••••> Explore

### ➔ Query

➔ Identify



➔ Compare



➔ Summarize



# Task abstraction: Targets

# Task abstraction: Targets

## → All Data

→ Trends



→ Outliers



→ Features



# Task abstraction: Targets

## → All Data

→ Trends



→ Outliers



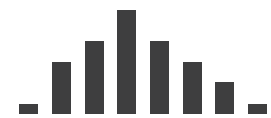
→ Features



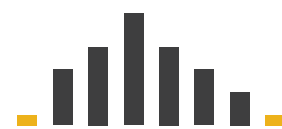
## → Attributes

→ One

→ *Distribution*



→ *Extremes*

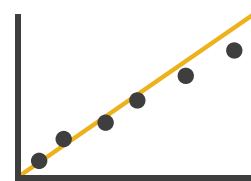


→ Many

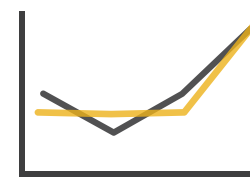
→ *Dependency*



→ *Correlation*



→ *Similarity*



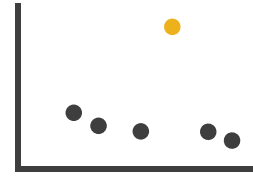
# Task abstraction: Targets

## → All Data

→ Trends



→ Outliers



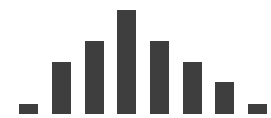
→ Features



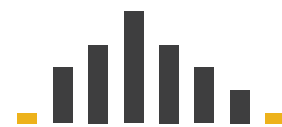
## → Attributes

→ One

→ *Distribution*

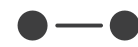


→ *Extremes*

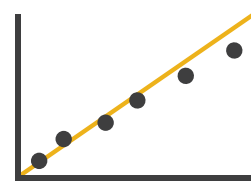


→ Many

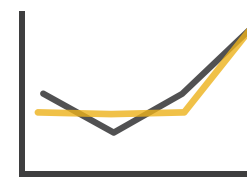
→ *Dependency*



→ *Correlation*

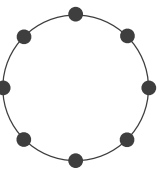
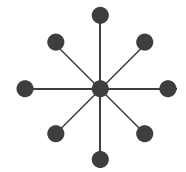
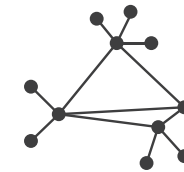


→ *Similarity*

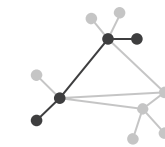


## → Network Data

→ Topology



→ *Paths*



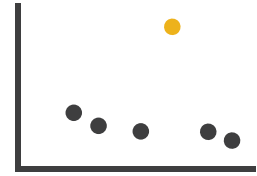
# Task abstraction: Targets

## → All Data

→ Trends



→ Outliers



→ Features



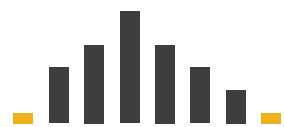
## → Attributes

→ One

→ *Distribution*



→ *Extremes*

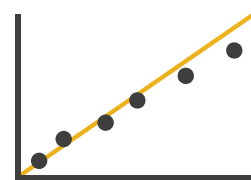


→ Many

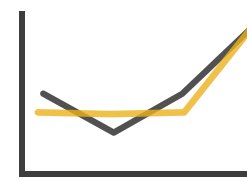
→ *Dependency*



→ *Correlation*

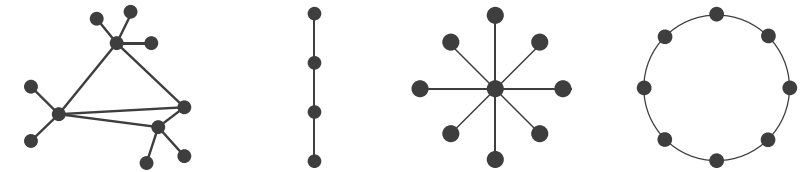


→ *Similarity*



## → Network Data

→ Topology

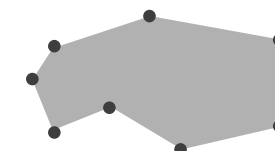


→ *Paths*



## → Spatial Data

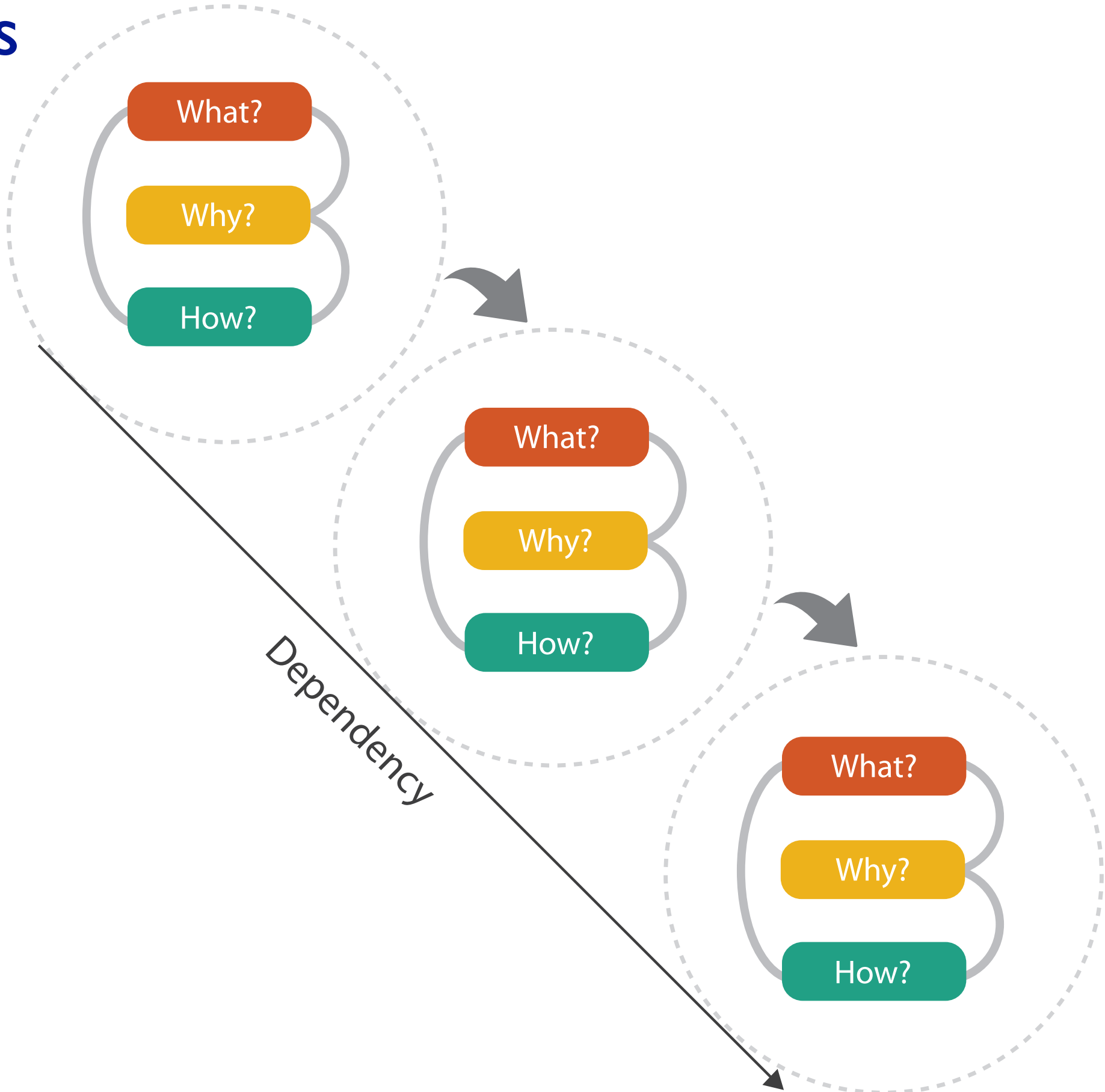
→ Shape



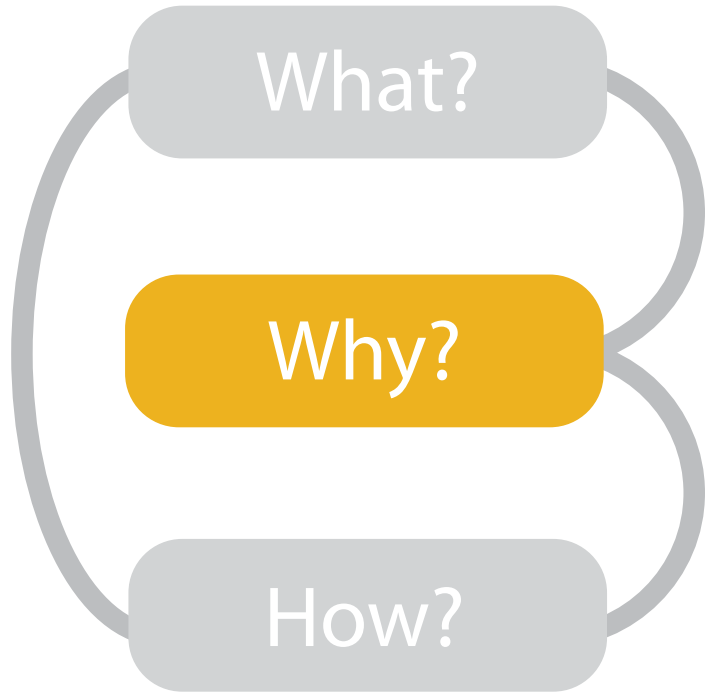
# Abstraction

- 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







## 👉 Actions

## 🎯 Targets

➔ **Analyze**

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

➔ **All Data**

- ➔ Trends
- ➔ Outliers
- ➔ Features

➔ **Attributes**

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

➔ **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

➔ **Network Data**

- ➔ Topology
- ➔ Paths

➔ **Spatial Data**

- ➔ Shape

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

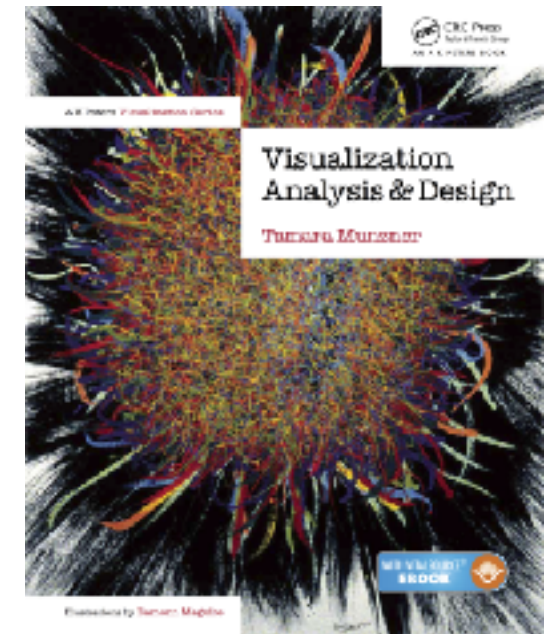
# Visualization Analysis & Design

## *Marks & Channels (Ch 5) I*

**Tamara Munzner**

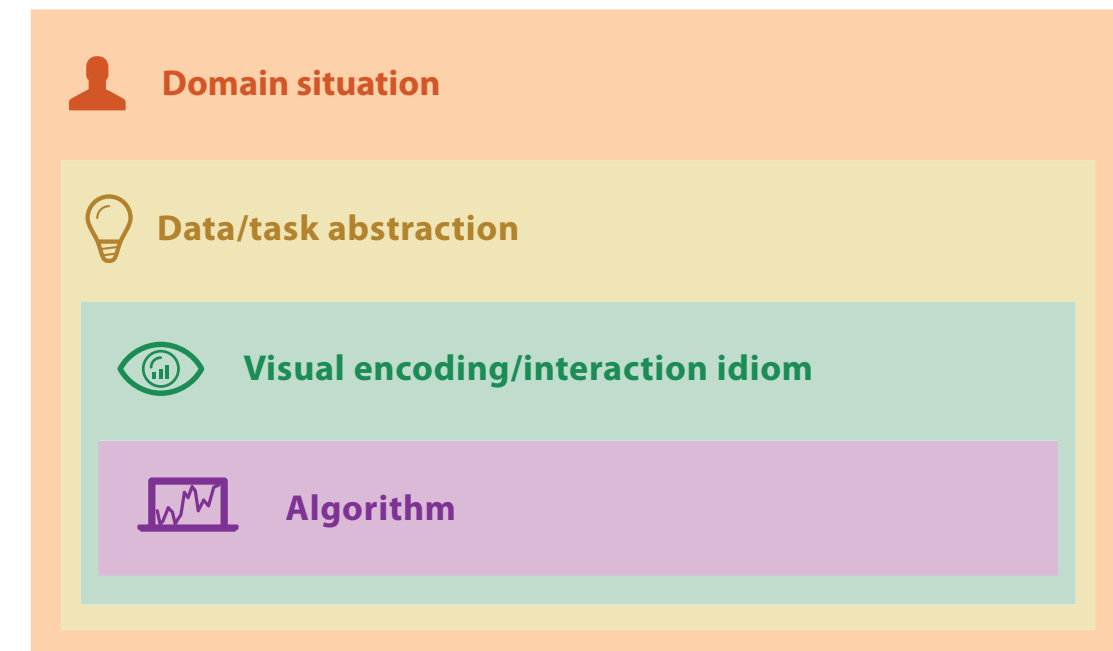
Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



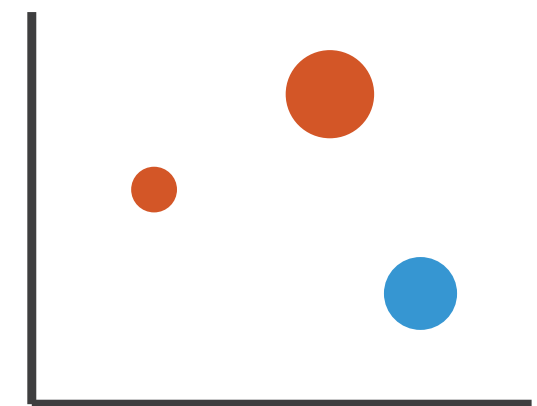
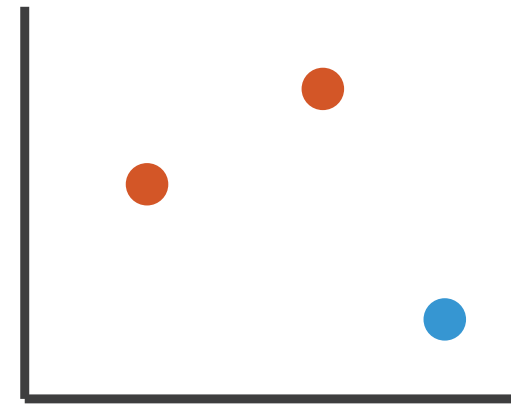
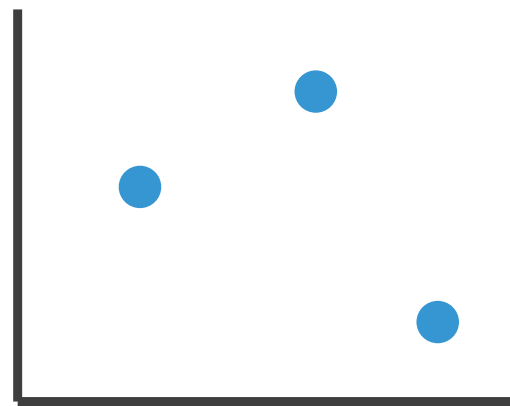
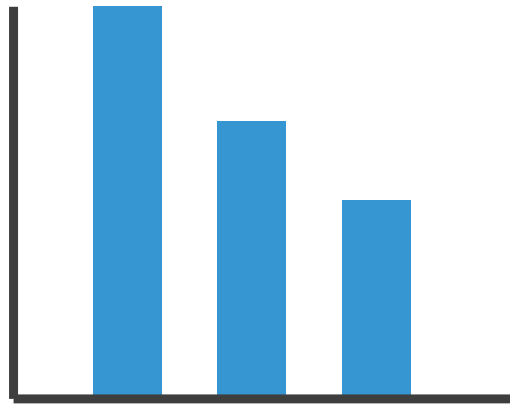
# Visual encoding

- how to systematically analyze idiom structure?



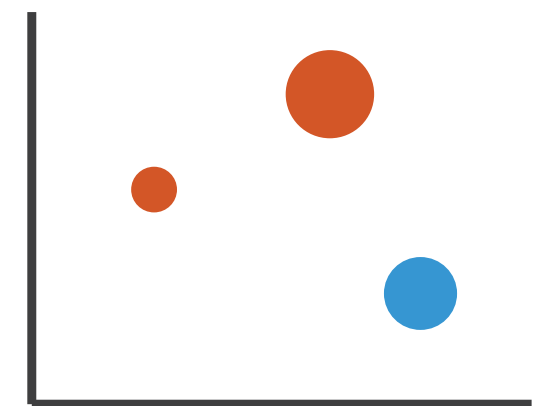
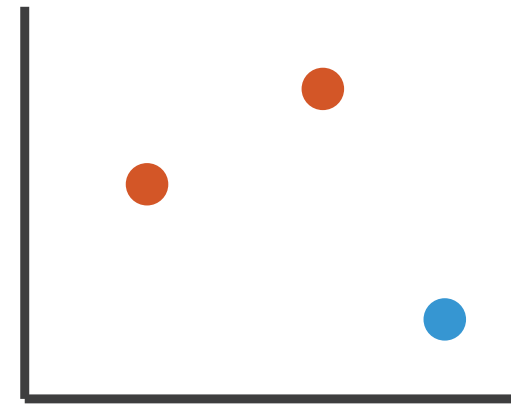
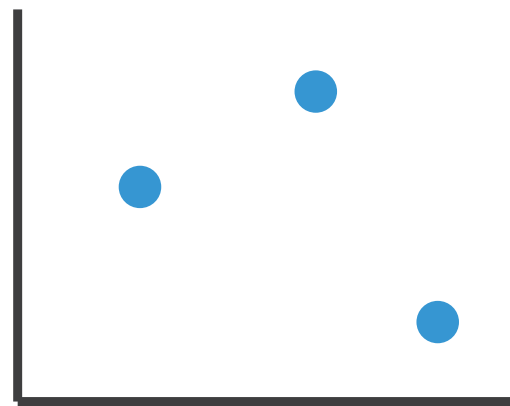
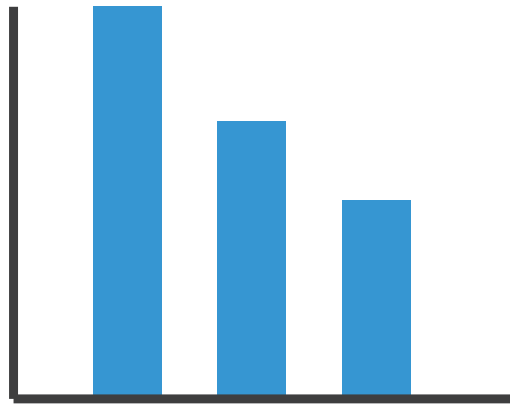
# Visual encoding

- how to systematically analyze idiom structure?



# Visual encoding

- how to systematically analyze idiom structure?



- marks & channels
  - marks: represent items or links
  - channels: change appearance of marks based on attributes

# Marks for items

- basic geometric elements

➔ Points



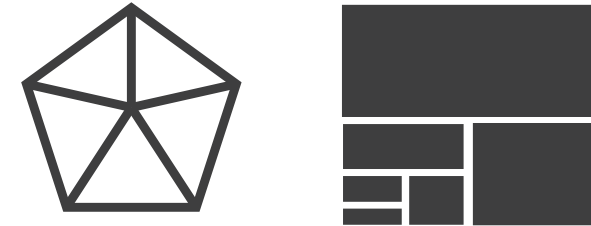
0D

➔ Lines



1D

➔ Interlocking Areas

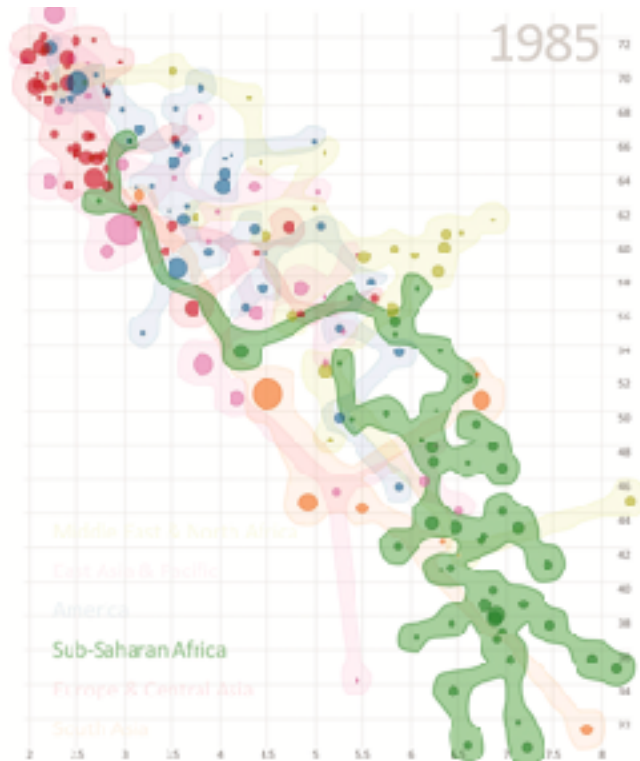
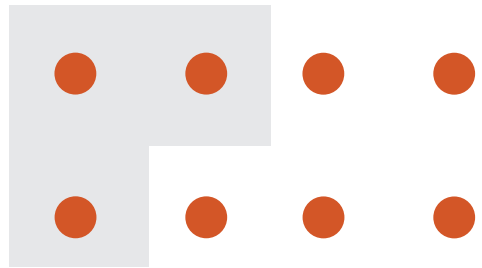


2D

- 3D mark: volume, rarely used

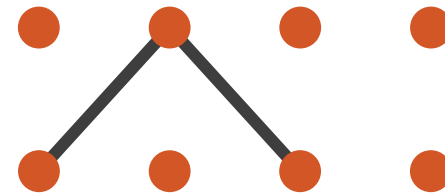
# Marks for links

## ➔ Containment



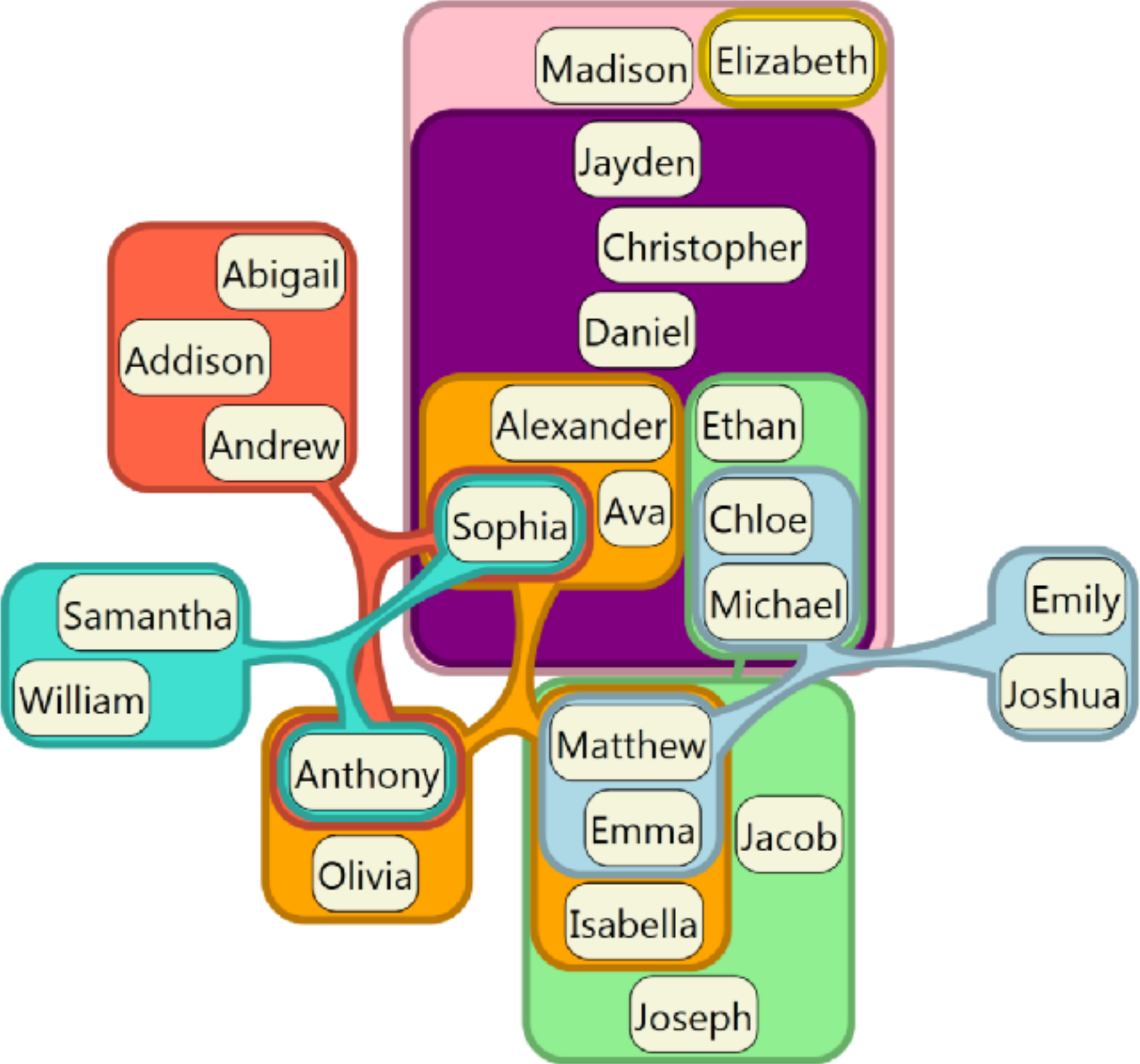
[vialab.science.uoit.ca/portfolio/bubblesets](http://vialab.science.uoit.ca/portfolio/bubblesets)

## ➔ Connection



<https://observablehq.com/@d3/force-directed-graph>

# Containment can be nested



[[Untangling Euler Diagrams, Riche and Dwyer, 2010](#)]



# Channels

- control appearance of marks
  - proportional to or based on attributes
- many names
  - **visual channels**
  - visual variables
  - retinal channels
  - visual dimensions
  - ...

## → Position

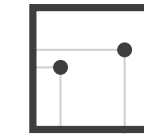
→ Horizontal



→ Vertical



→ Both

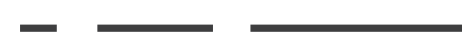


## → Shape



## → Size

→ Length



→ Area



## → Color



## → Tilt



→ Volume



# Definitions: Marks and channels

- marks
  - geometric primitives

→ Points



→ Lines



→ Areas



# Definitions: Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks

→ Points



→ Lines



→ Interlocking Areas



→ Position

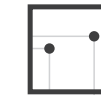
→ Horizontal



→ Vertical



→ Both



→ Color



→ Shape



→ Tilt



→ Size

→ Length



→ Area



→ Volume



# Definitions: Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks
- channel properties differ
  - type & amount of information that can be conveyed to human perceptual system

→ Points



→ Lines



→ Interlocking Areas



→ Position

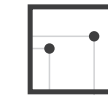
→ Horizontal



→ Vertical



→ Both



→ Color



→ Shape



→ Tilt



→ Size

→ Length



→ Area

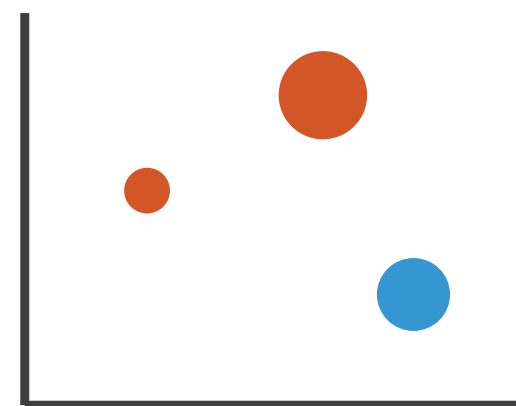
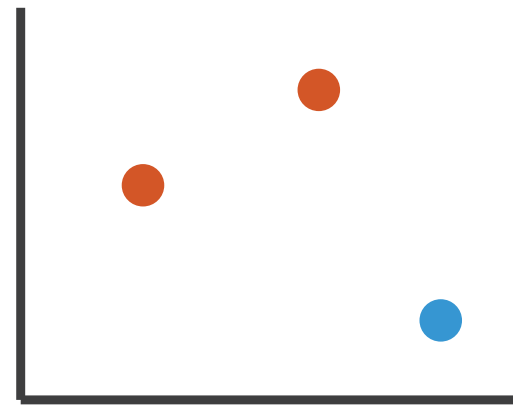
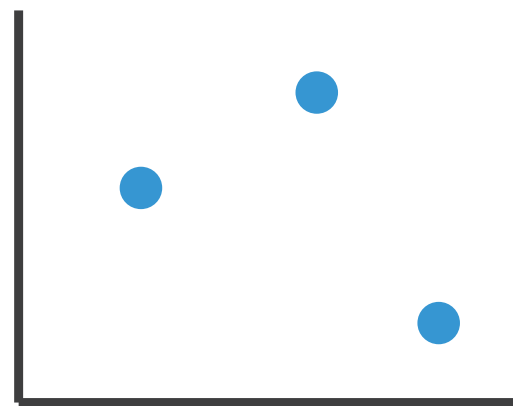
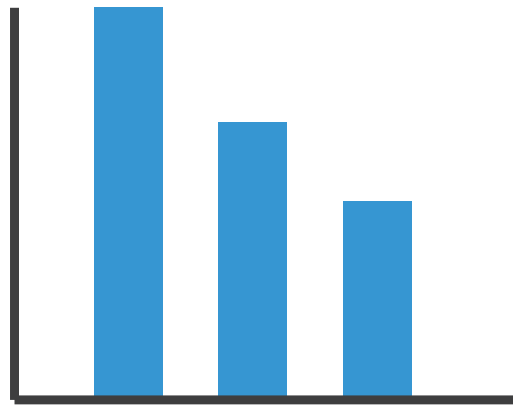


→ Volume



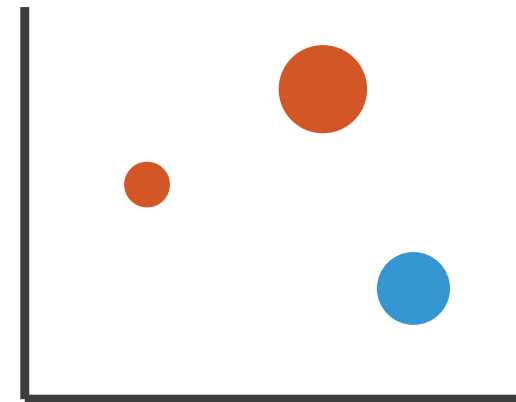
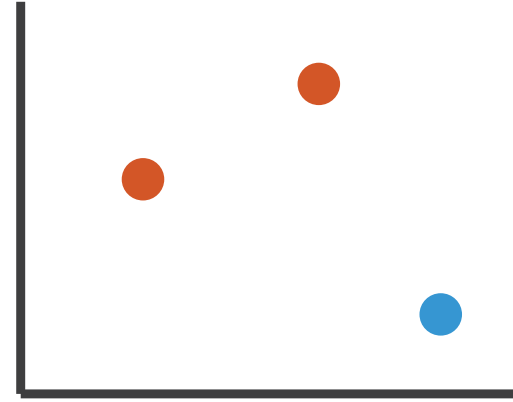
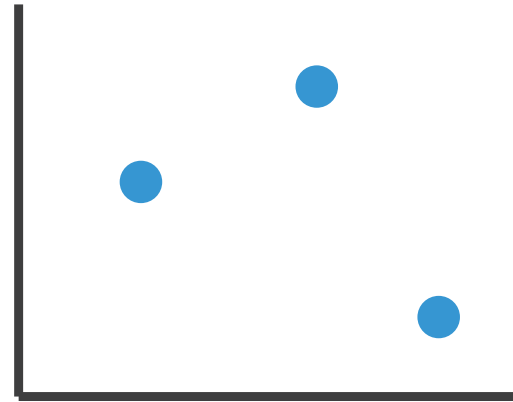
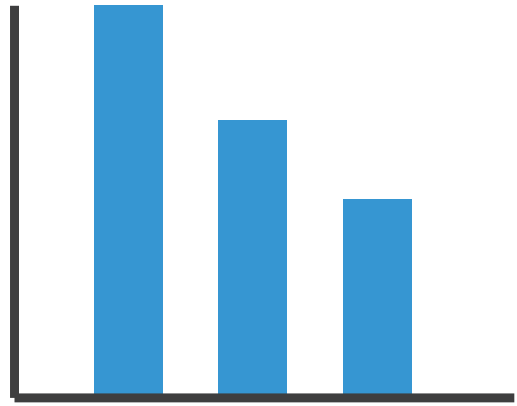
# Visual encoding

- analyze idiom structure as combination of marks and channels



# Visual encoding

- analyze idiom structure as combination of marks and channels

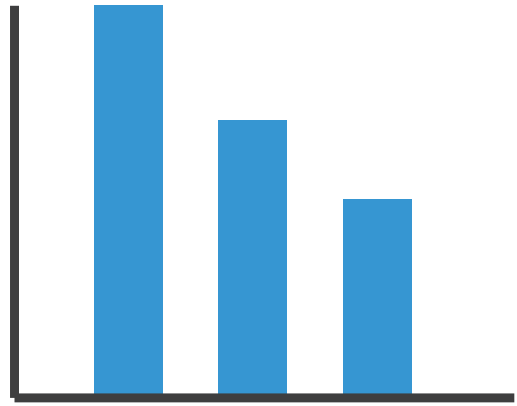


1:  
vertical position

mark: line

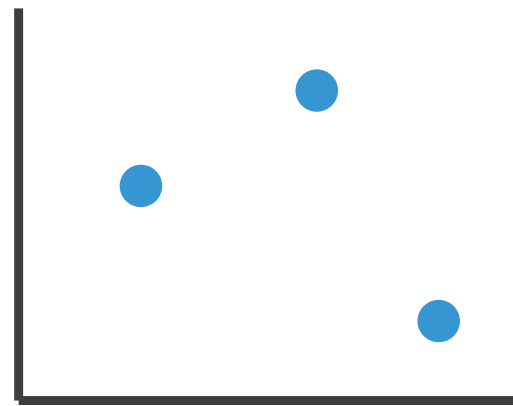
# Visual encoding

- analyze idiom structure as combination of marks and channels



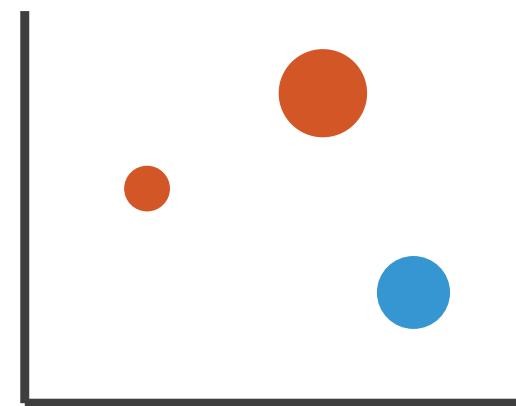
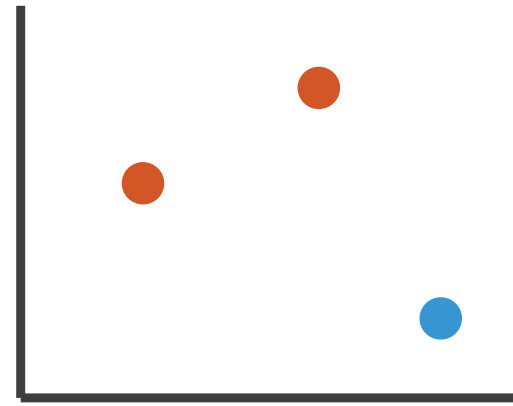
1:  
vertical position

mark: line



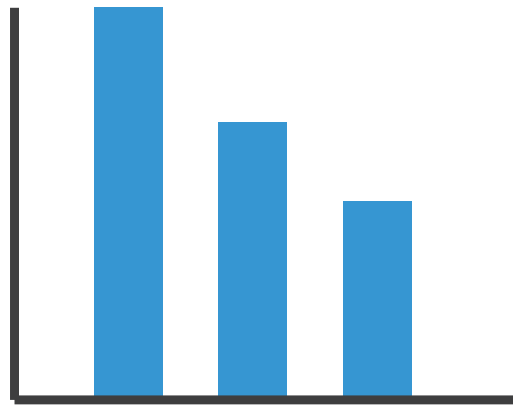
2:  
vertical position  
horizontal position

mark: point



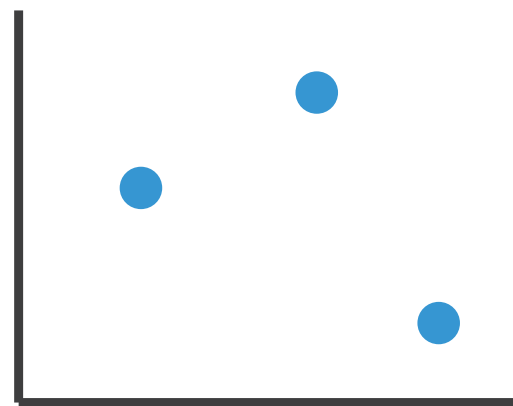
# Visual encoding

- analyze idiom structure as combination of marks and channels



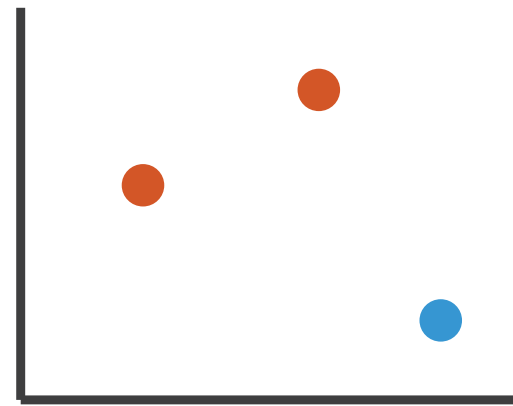
1:  
vertical position

mark: line



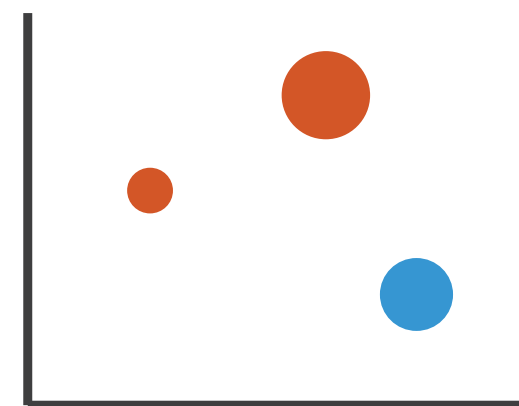
2:  
vertical position  
horizontal position

mark: point



3:  
vertical position  
horizontal position  
color hue

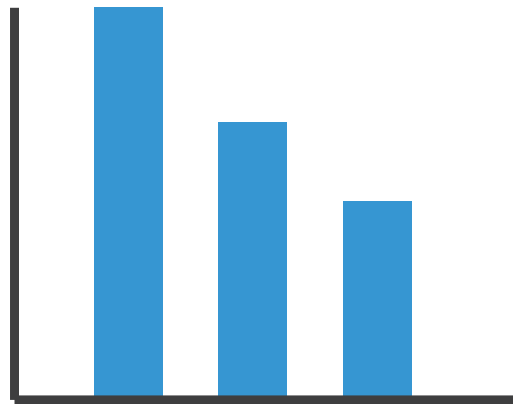
mark: point





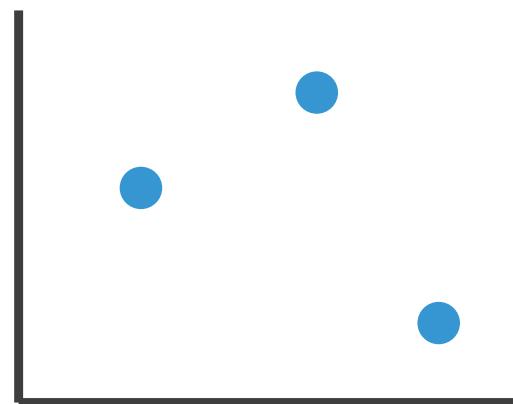
# Visual encoding

- analyze idiom structure as combination of marks and channels



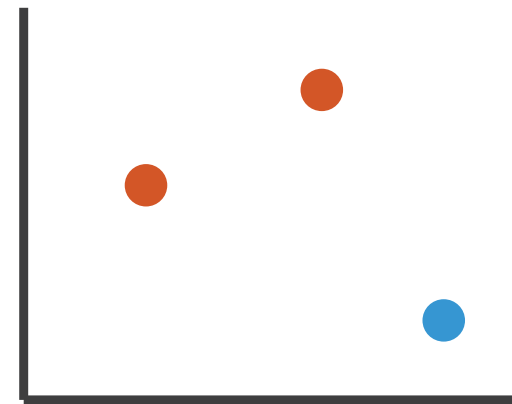
1:  
vertical position

mark: line



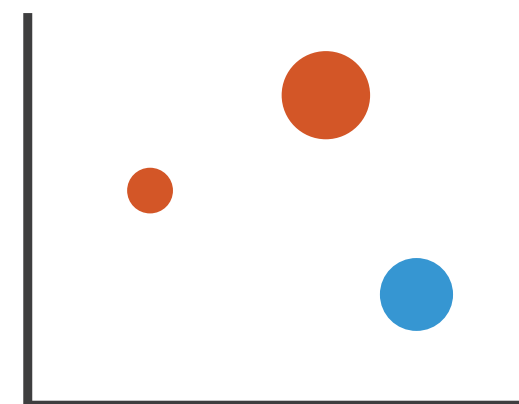
2:  
vertical position  
horizontal position

mark: point



3:  
vertical position  
horizontal position  
color hue

mark: point

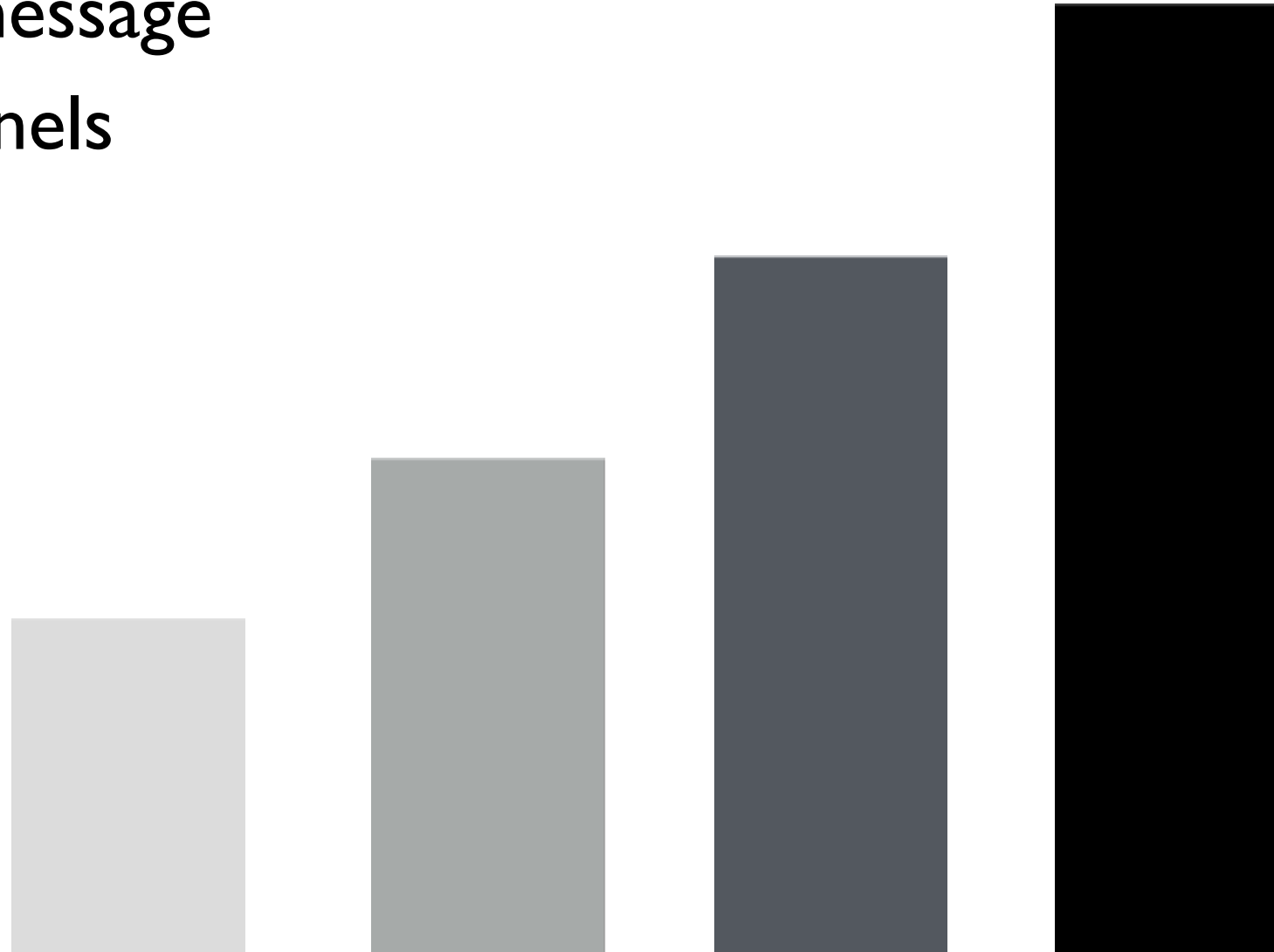


4:  
vertical position  
horizontal position  
color hue  
size (area)

mark: point

# Redundant encoding

- multiple channels
  - sends stronger message
  - but uses up channels



Length and Luminance

# Marks as constraints

- math view: geometric primitives have dimensions

→ Points

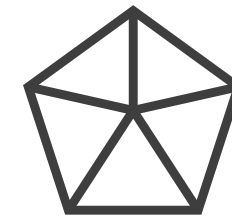
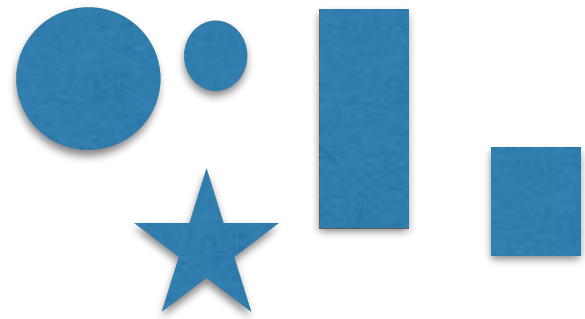
0D

→ Lines

1D

→ Interlocking Areas

2D



# Marks as constraints

- math view: geometric primitives have dimensions

→ Points

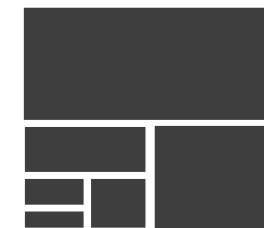
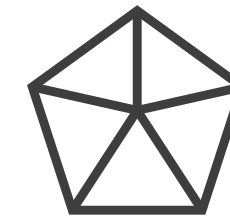
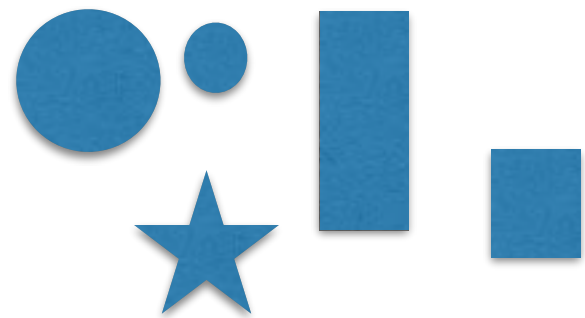
0D

→ Lines

1D

→ Interlocking Areas

2D



- constraint view: mark type constrains what else can be encoded
  - points: 0 constraints on size, can encode more attributes w/ size & shape
  - lines: 1 constraint on size (length), can still size code other way (width)
  - interlocking areas: 2 constraints on size (length/width), cannot size or shape code
    - interlocking: size, shape, position

# Marks as constraints

- math view: geometric primitives have dimensions

➔ Points

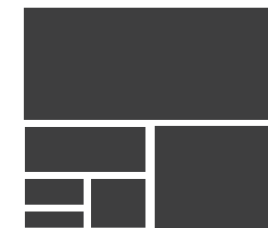
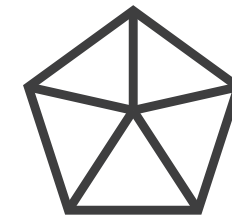
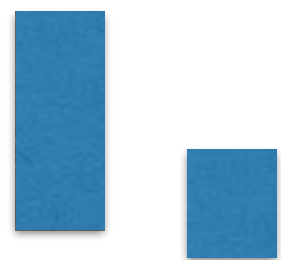
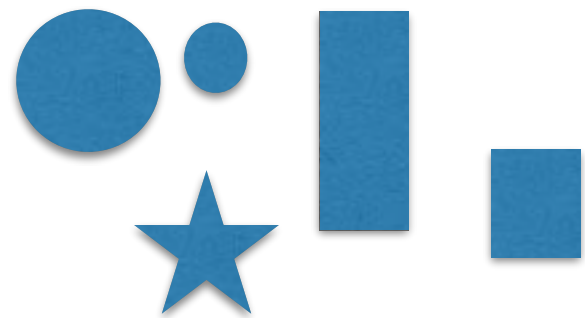
0D

➔ Lines

1D

➔ Interlocking Areas

2D



- constraint view: mark type constrains what else can be encoded
  - points: 0 constraints on size, can encode more attributes w/ size & shape
  - lines: 1 constraint on size (length), can still size code other way (width)
  - interlocking areas: 2 constraints on size (length/width), cannot size or shape code
    - interlocking: size, shape, position
- quick check: can you size-code another attribute
  - or is size/shape in use?

# Scope of analysis

- simplifying assumptions: one mark per item, single view
- later on
  - multiple views
  - multiple marks in a region (glyph)
  - some items not represented by marks (aggregation and filtering)

When to use which channel?

**expressiveness**

match channel type to data type

**effectiveness**

some channels are better than others

# Channels: Rankings

Position on common scale



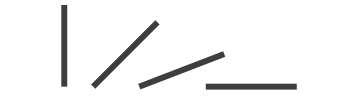
Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



Depth (3D position)



Color luminance



Color saturation



Curvature



Volume (3D size)



Same

Spatial region



Color hue



Motion



Shape





# Channels: Rankings

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale 

Position on unaligned scale 

Length (1D size) 

Tilt/angle 

Area (2D size) 

Depth (3D position) 

Color luminance 

Color saturation 

Curvature 

Volume (3D size) 

Same

## ➔ Identity Channels: Categorical Attributes

Spatial region 

Color hue 

Motion 

Shape 

- **expressiveness**
  - match channel and data characteristics

# Channels: Rankings

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale 

Position on unaligned scale 

Length (1D size) 

Tilt/angle 

Area (2D size) 

Depth (3D position) 

Color luminance 

Color saturation 

Curvature 

Volume (3D size) 


Same  
Same

## ➔ Identity Channels: Categorical Attributes

Spatial region 

Color hue 

Motion 

Shape 

### ➔ Attribute Types

➔ Categorical



➔ Ordered

➔ Ordinal



➔ Quantitative



- **expressiveness**

- match channel and data characteristics

- magnitude for ordered

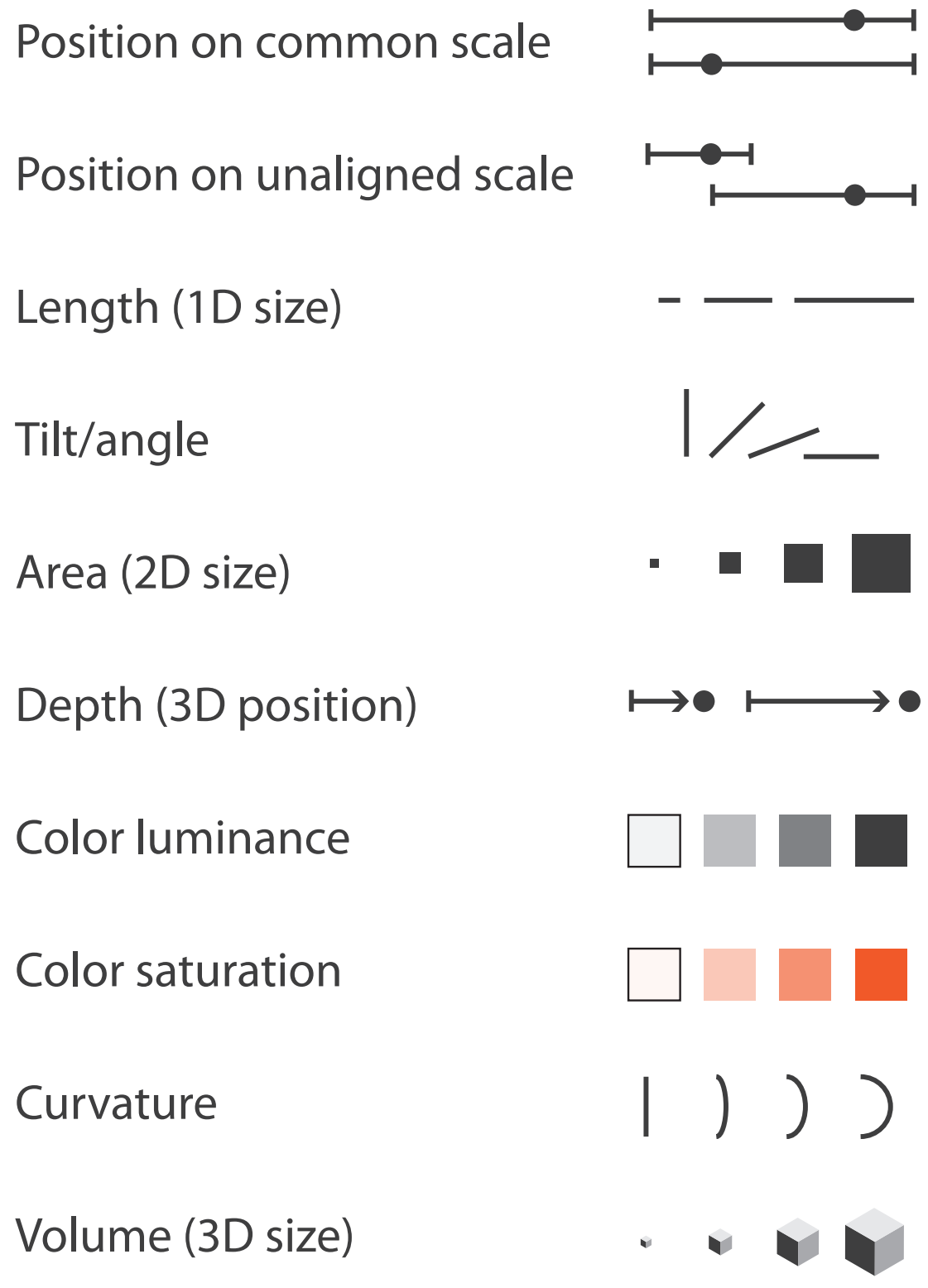
- how much? which rank?

- identity for categorical

- what?

# Channels: Rankings

## ➔ Magnitude Channels: Ordered Attributes



## ➔ Identity Channels: Categorical Attributes



- **expressiveness**
  - match channel and data characteristics
- **effectiveness**
  - channels differ in accuracy of perception

# Channels: Rankings

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale



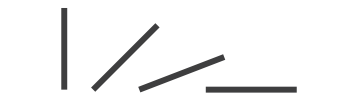
Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



Depth (3D position)



Color luminance



Color saturation



Curvature



Volume (3D size)



Same

Same

## ➔ Identity Channels: Categorical Attributes

Spatial region



Color hue



Motion



Shape



Best

Effectiveness

Least

- **expressiveness**
  - match channel and data characteristics
- **effectiveness**
  - channels differ in accuracy of perception
  - spatial position ranks high for both

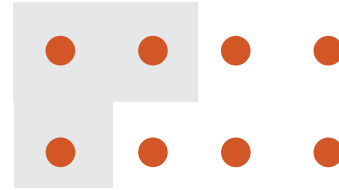
# Grouping

- containment
- connection

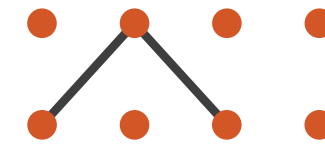
- proximity
  - same spatial region
- similarity
  - same values as other categorical channels

## Marks as Links

### ➔ Containment



### ➔ Connection



### ➔ Identity Channels: Categorical Attributes

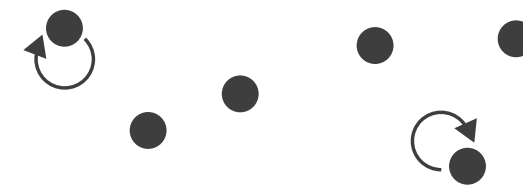
Spatial region



Color hue



Motion



Shape



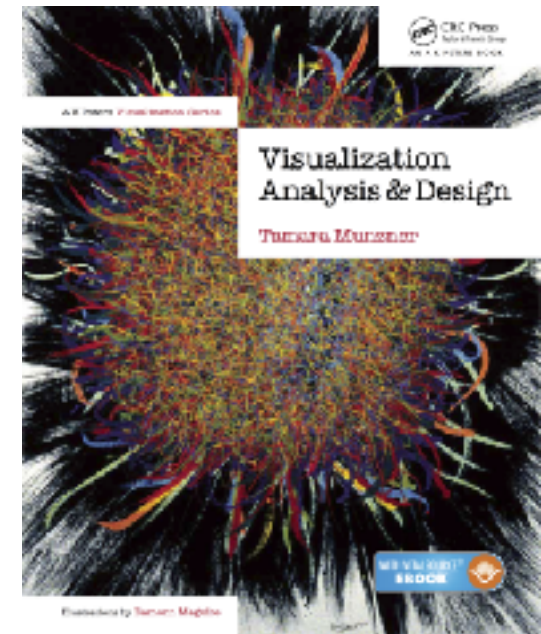
# Visualization Analysis & Design

## *Arrange Tables (Ch 7) I*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

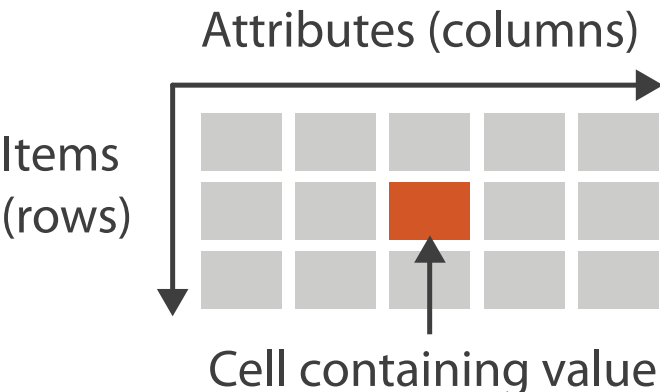
[@tamaramunzner](#)



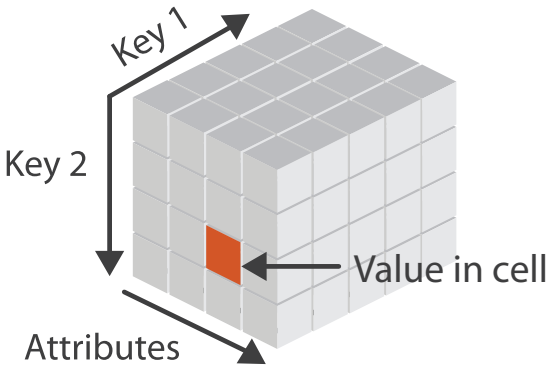
# Focus on Tables

## → Dataset Types

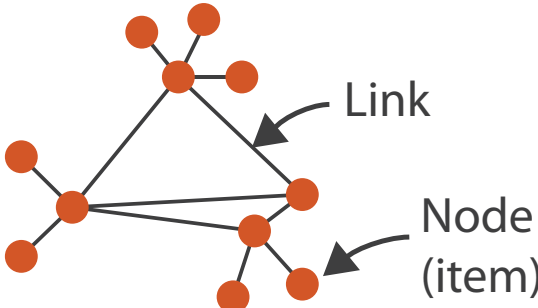
### → Tables



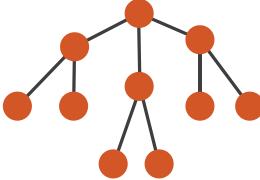
### → Multidimensional Table



### → Networks

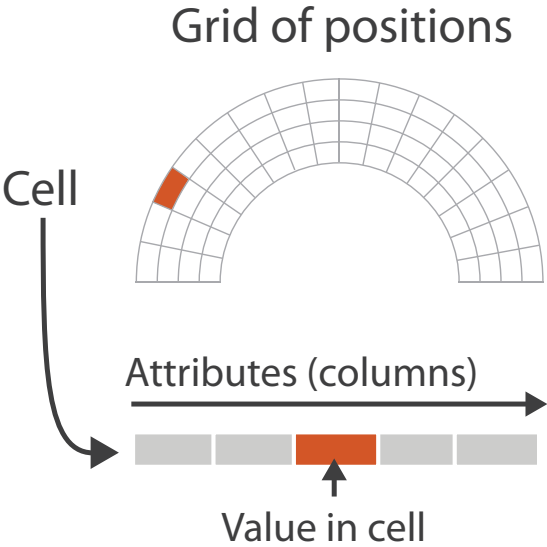


### → Trees



### → Spatial

#### → Fields (Continuous)



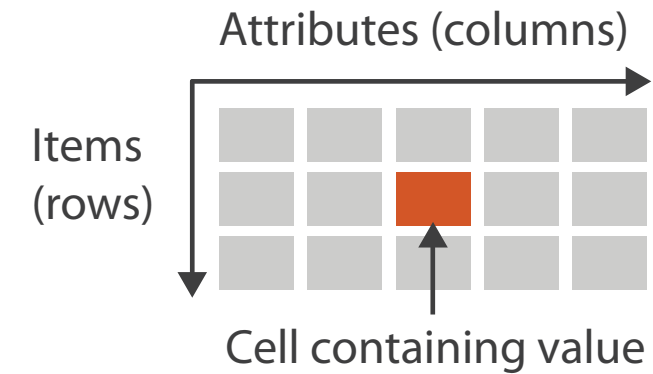
#### → Geometry (Spatial)



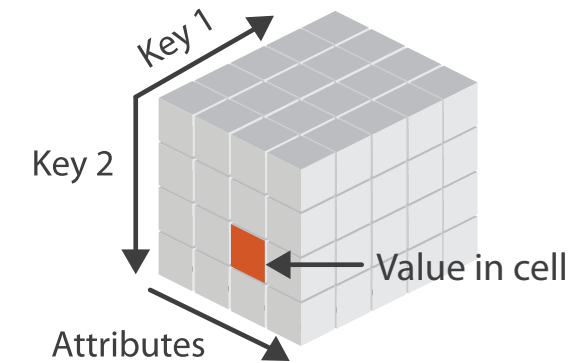
# Keys and values

- **key**
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - multidimensional tables: multiple keys
- **value**
  - dependent attribute, value of cell

→ Tables



→ *Multidimensional Table*





# Keys and values

- **key**
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - multidimensional tables: multiple keys
- **value**
  - dependent attribute, value of cell
- **classify arrangements by keys used**
  - 0, 1, 2, ...

→ 0 Keys

⊕ Express Values



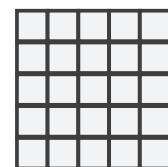
→ 1 Key

List

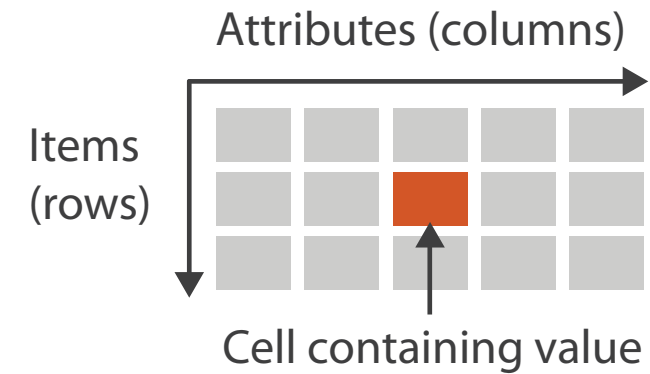


→ 2 Keys

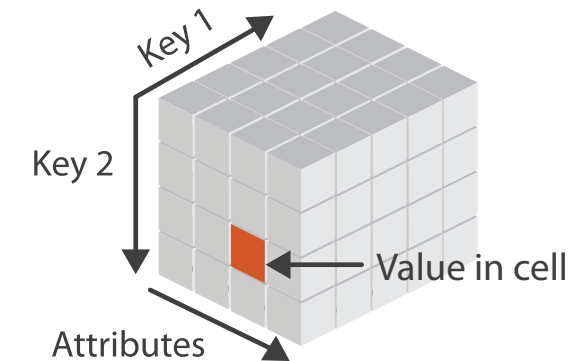
Matrix



→ Tables



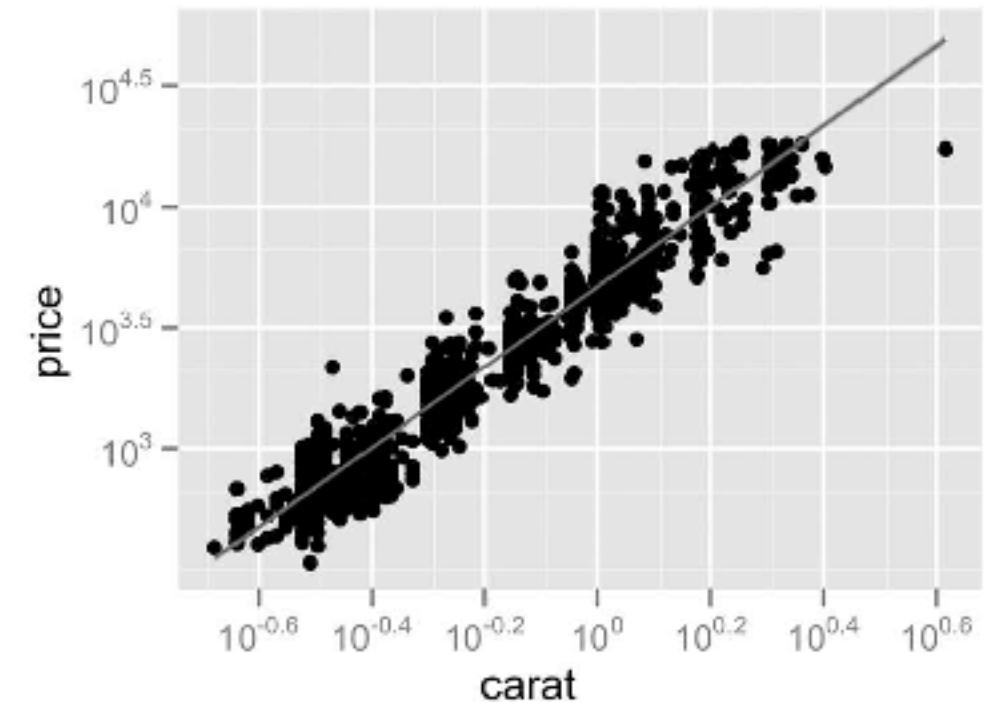
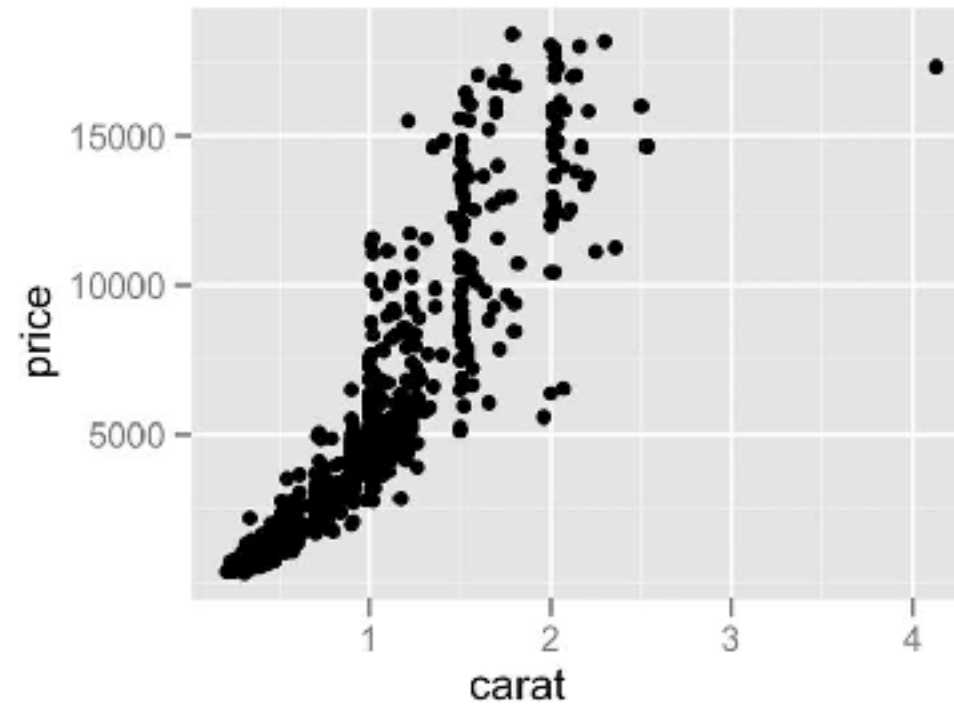
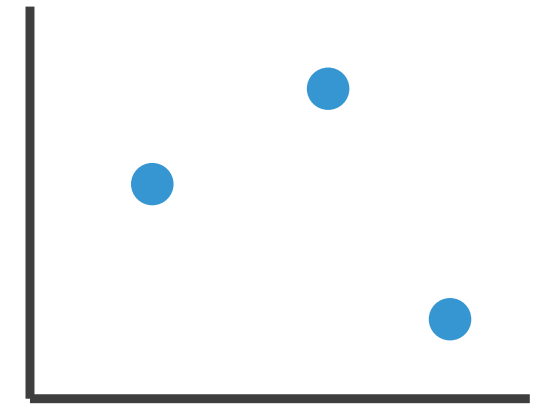
→ Multidimensional Table



# Idiom: scatterplot

- **express** values (magnitudes)
  - quantitative attributes
- no keys, only values

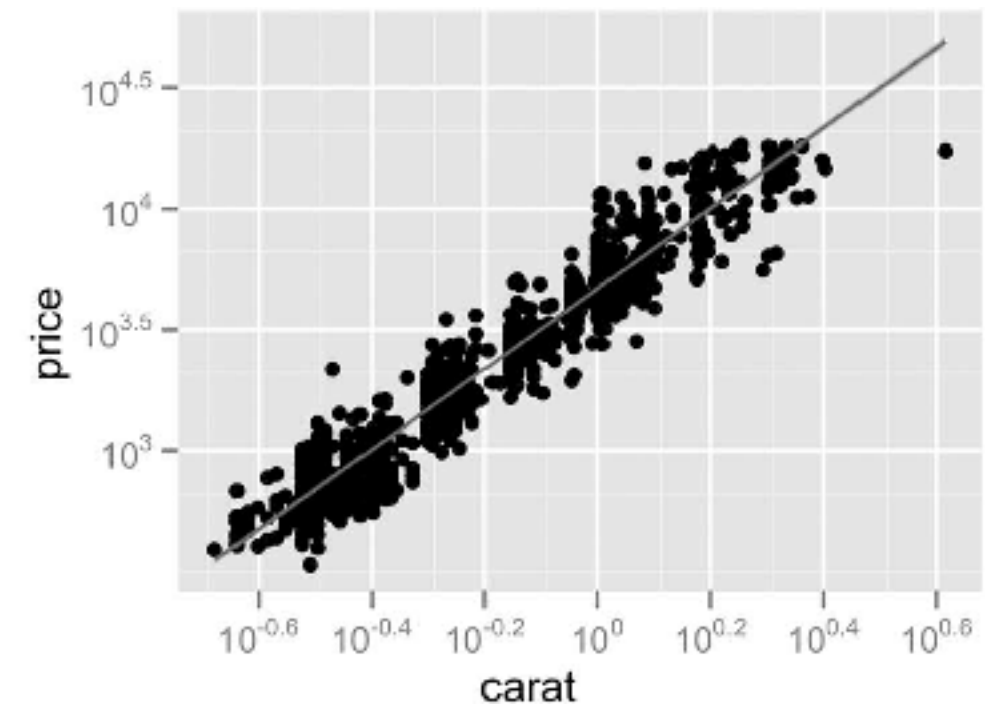
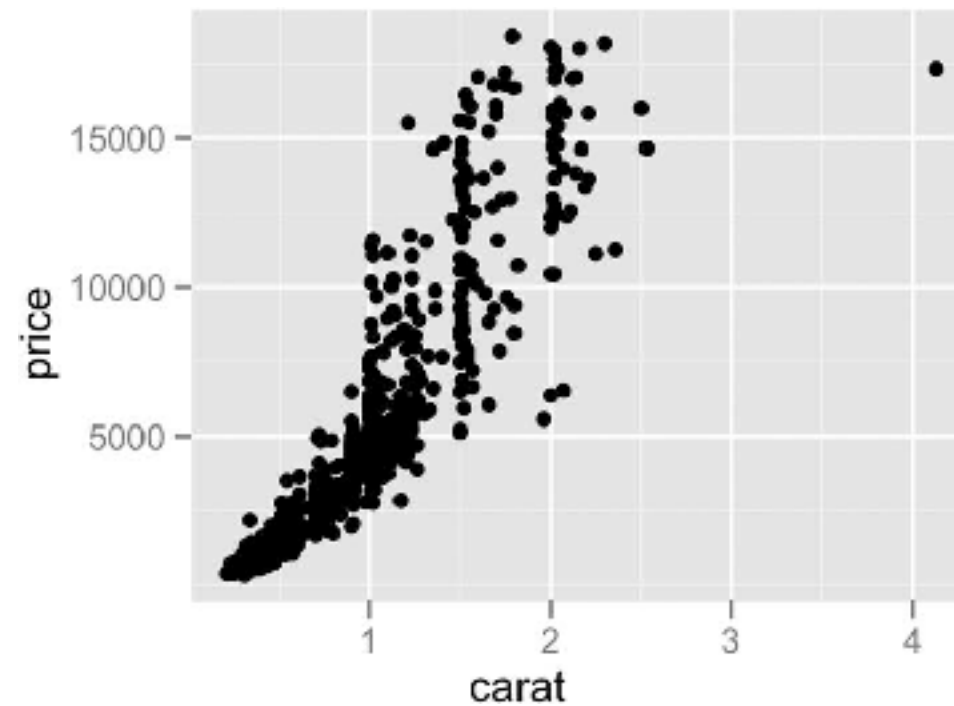
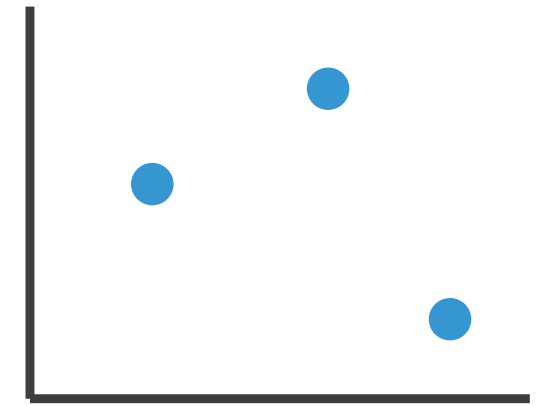
⊙ → Express Values



# Idiom: scatterplot

- **express** values (magnitudes)
  - quantitative attributes
- no keys, only values
  - data
    - 2 quant attribs
  - mark: points
  - channels
    - horiz + vert position

⊙ → Express Values



# Idiom: scatterplot

- **express** values (magnitudes)

- quantitative attributes

- no keys, only values

- data

- 2 quant attribs

- mark: points

- channels

- horiz + vert position

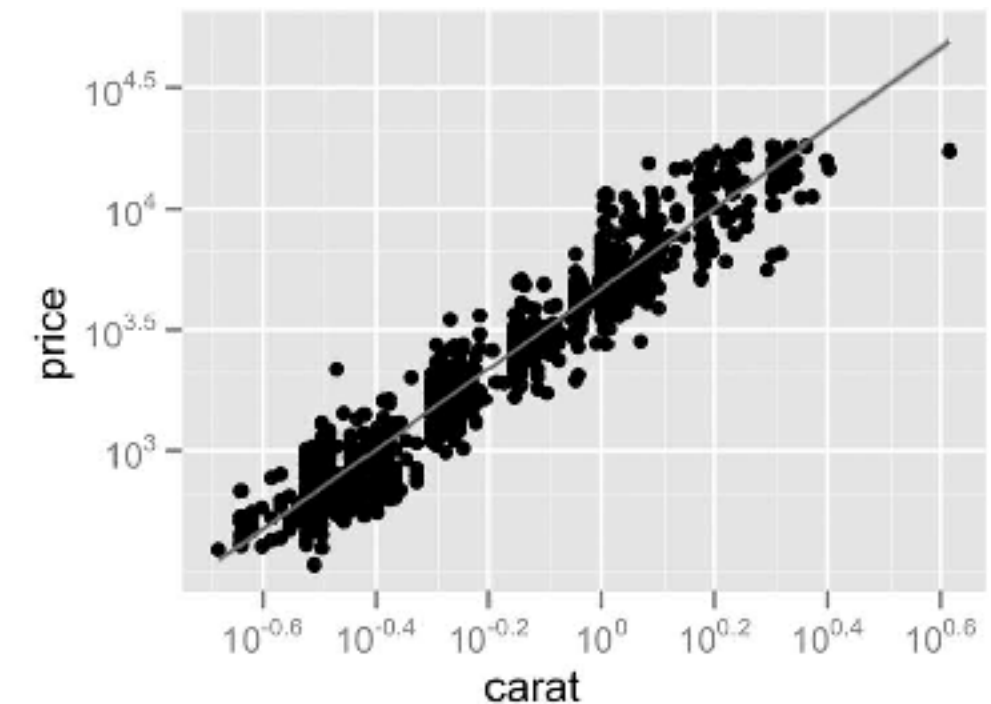
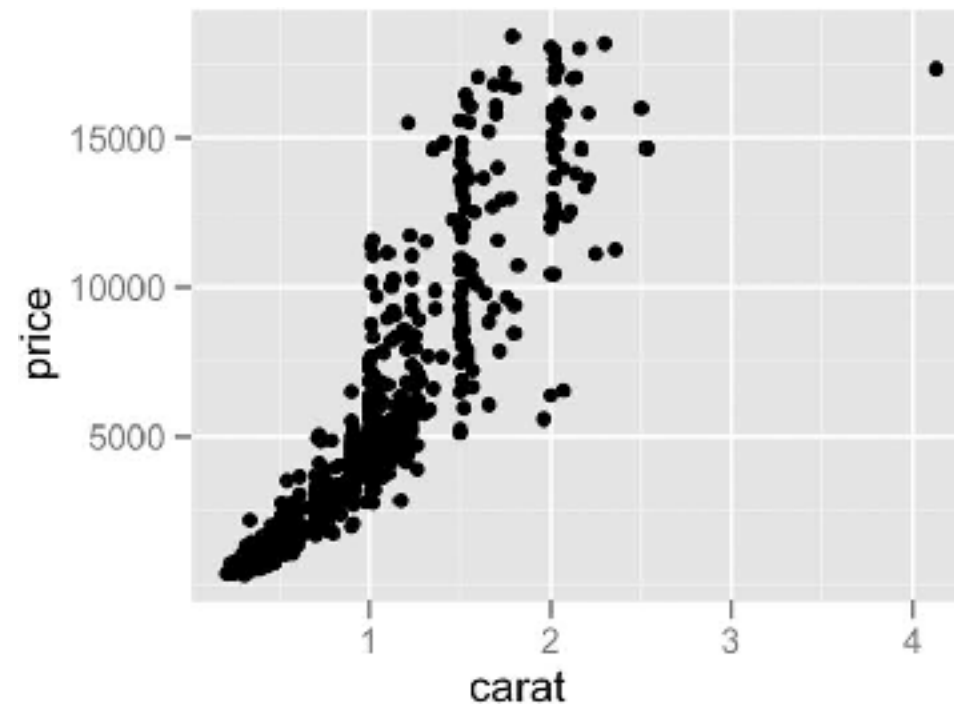
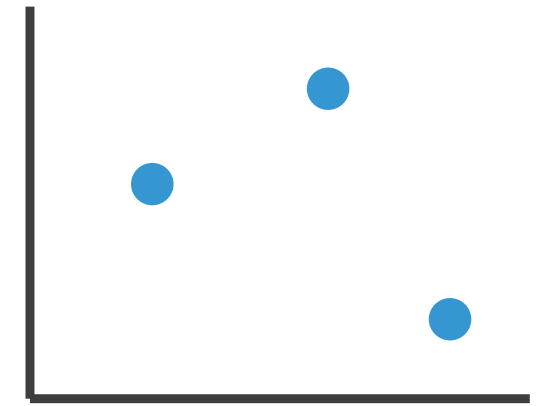
- tasks

- find trends, outliers, distribution, correlation, clusters

- scalability

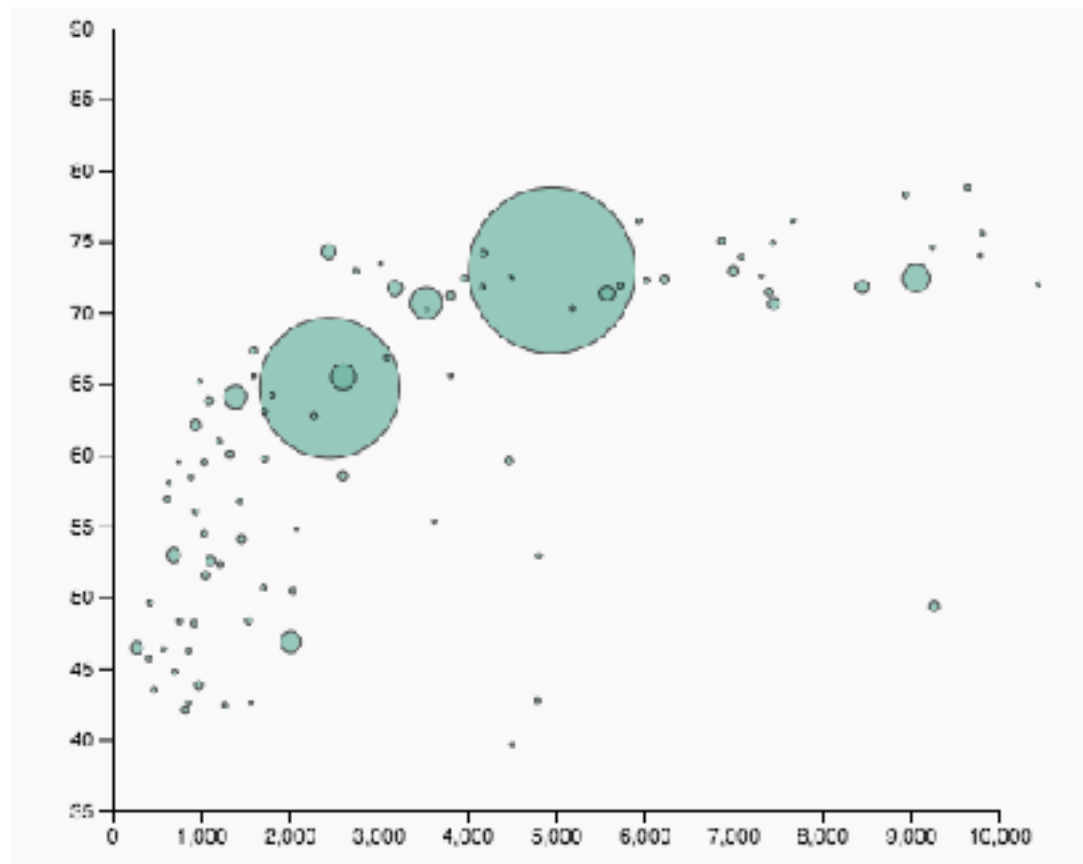
- hundreds of items

⊙ → Express Values

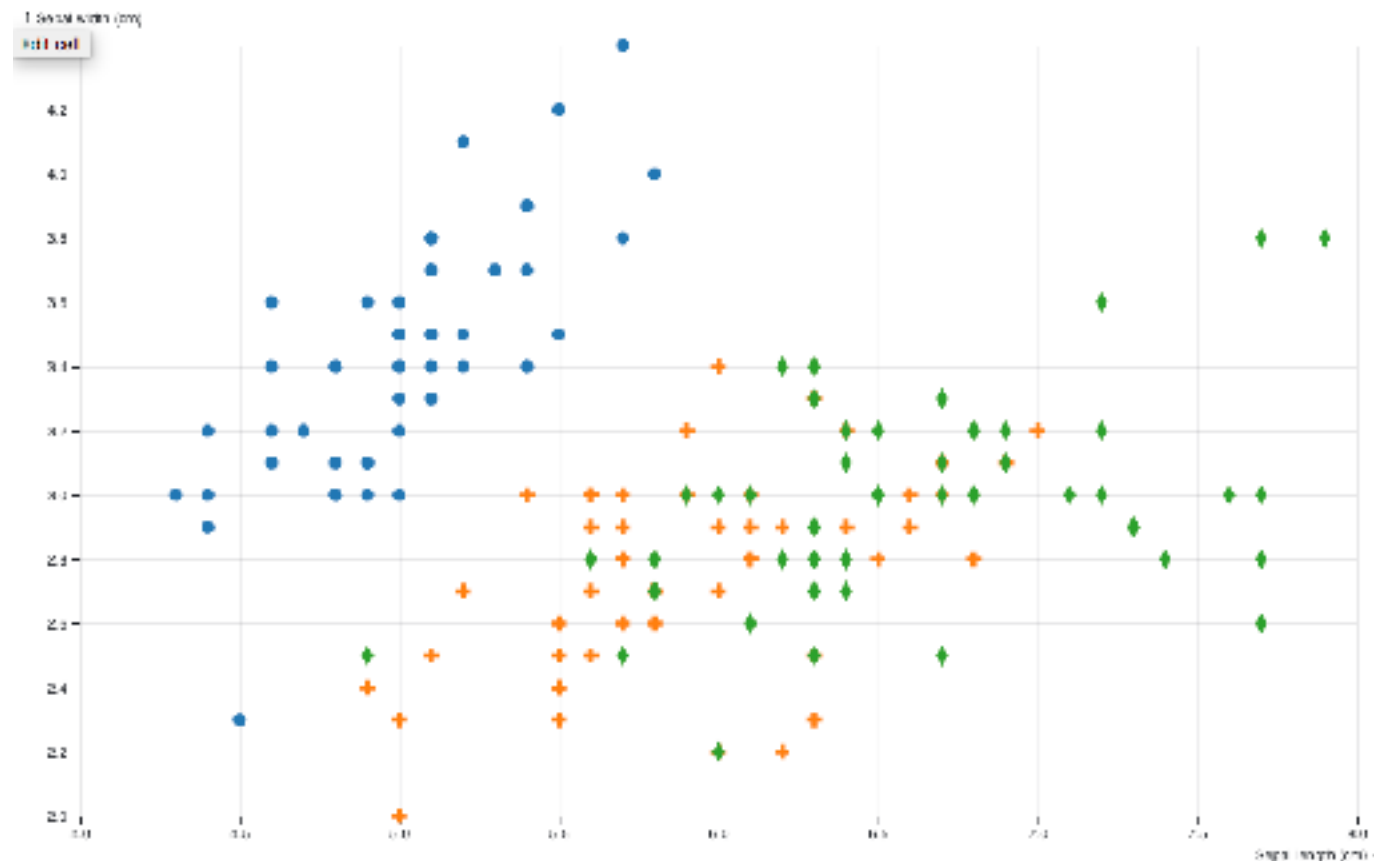


# Scatterplots: Encoding more channels

- additional channels viable since using point marks
  - color
  - size (1 quant attribute, used to control 2D area)
    - note radius would mislead, take square root since area grows quadratically
  - shape



[https://www.d3-graph-gallery.com/graph/bubble\\_basic.html](https://www.d3-graph-gallery.com/graph/bubble_basic.html)

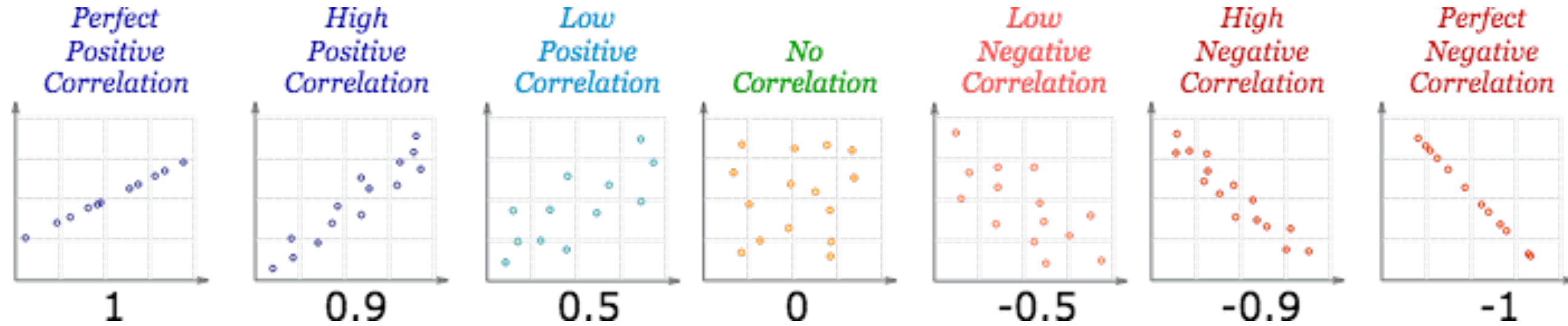


<https://observablehq.com/@d3/scatterplot-with-shapes>

# Scatterplot tasks

# Scatterplot tasks

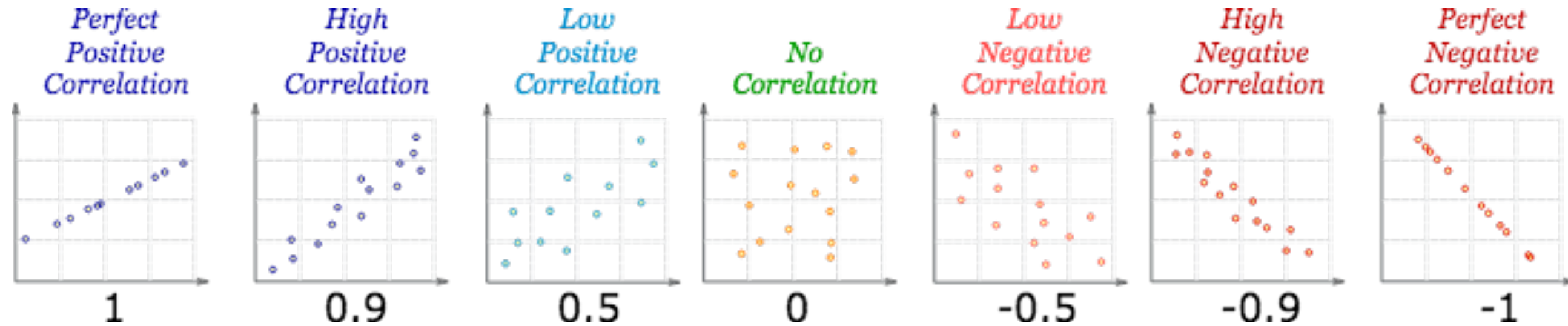
- correlation



<https://www.mathsisfun.com/data/scatter-xy-plots.html>

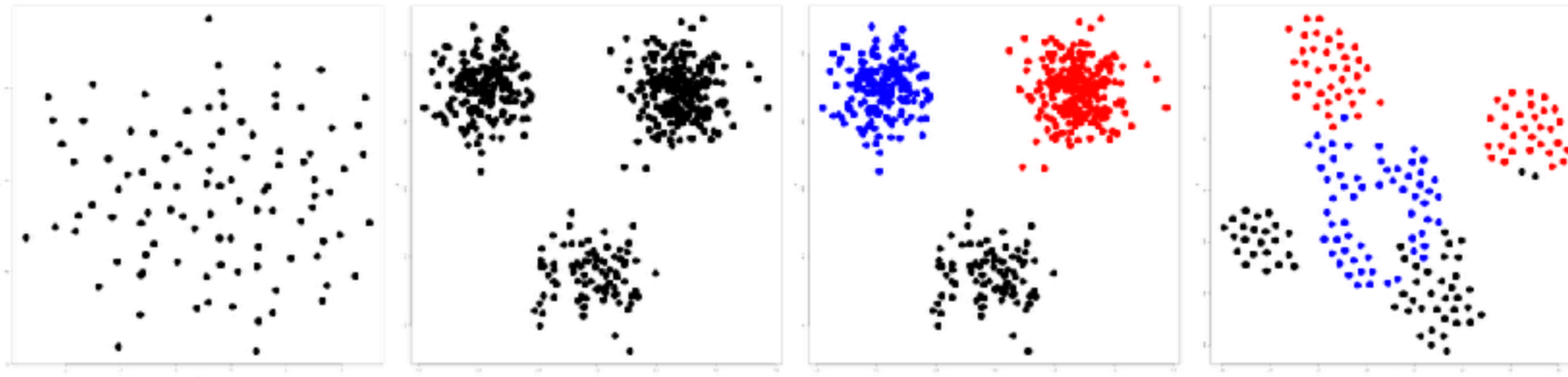
# Scatterplot tasks

- correlation



<https://www.mathsisfun.com/data/scatter-xy-plots.html>

- clusters/groups, and clusters vs classes



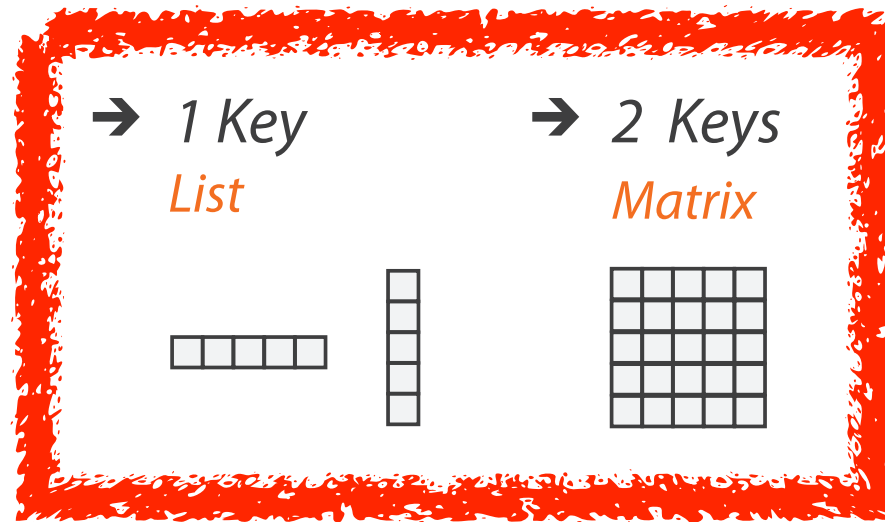
<https://www.cs.ubc.ca/labs/imager/tr/2014/DRVisTasks/>



# Some keys

→ 0 Keys

⊙ Express Values

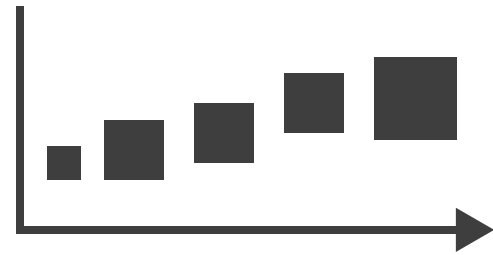


# Some keys: Categorical regions

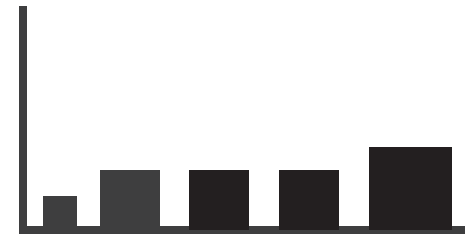
→ Separate



→ Order



→ Align

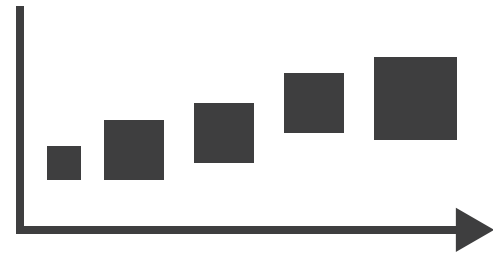


# Regions: Separate, order, align

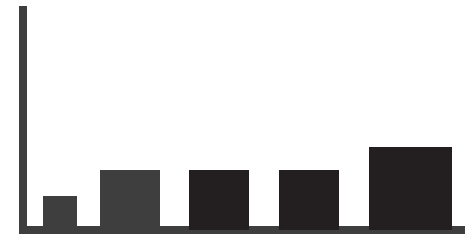
→ Separate



→ Order

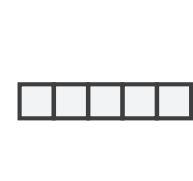


→ Align

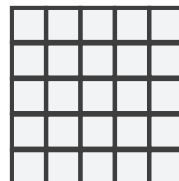


- regions: contiguous bounded areas distinct from each other
  - separate into spatial regions: one mark per region (for now)
- use categorical or ordered attribute to separate into regions
  - no conflict with expressiveness principle for categorical attributes
- use ordered attribute to order and align regions

→ 1 Key  
*List*

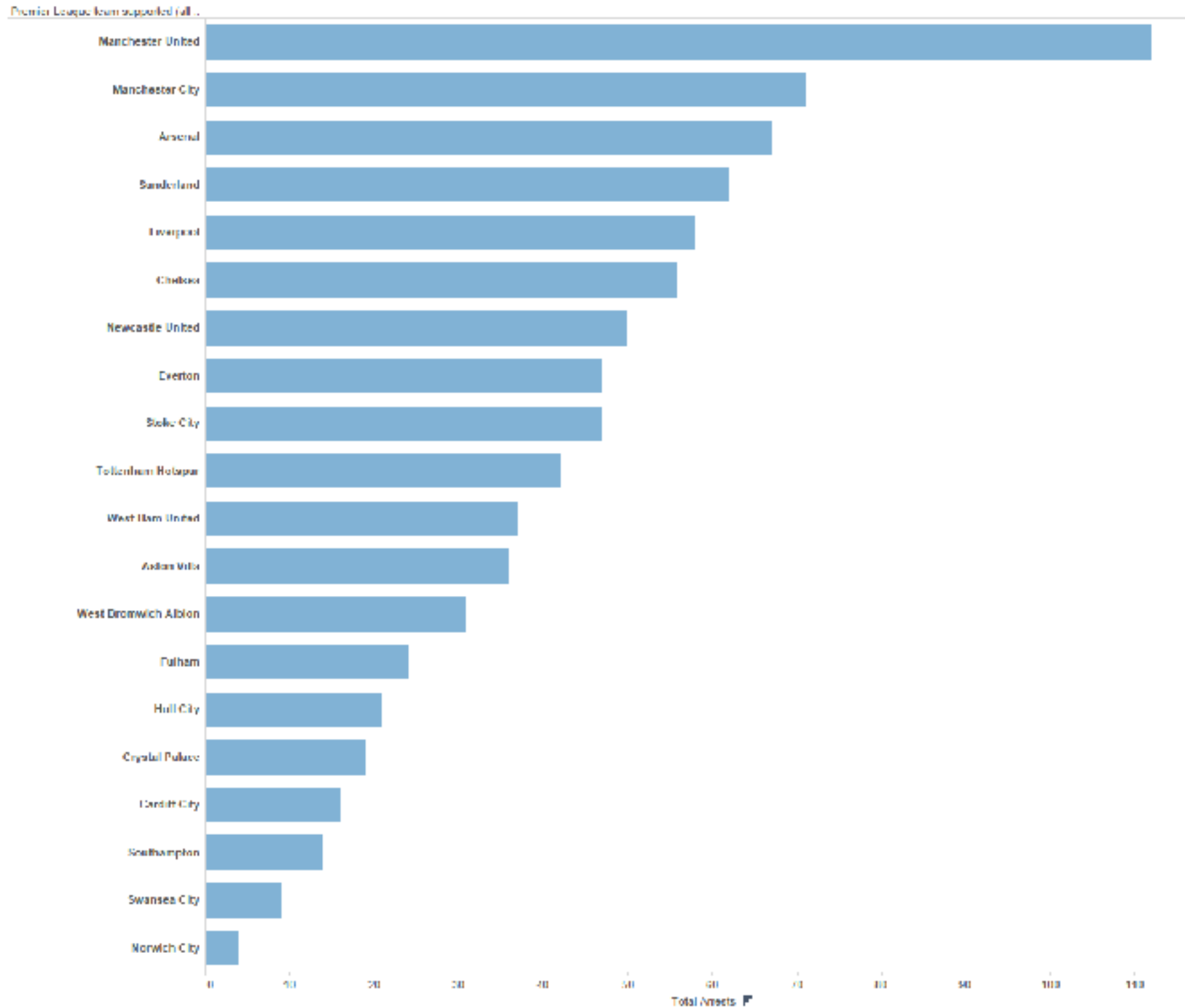


→ 2 Keys  
*Matrix*



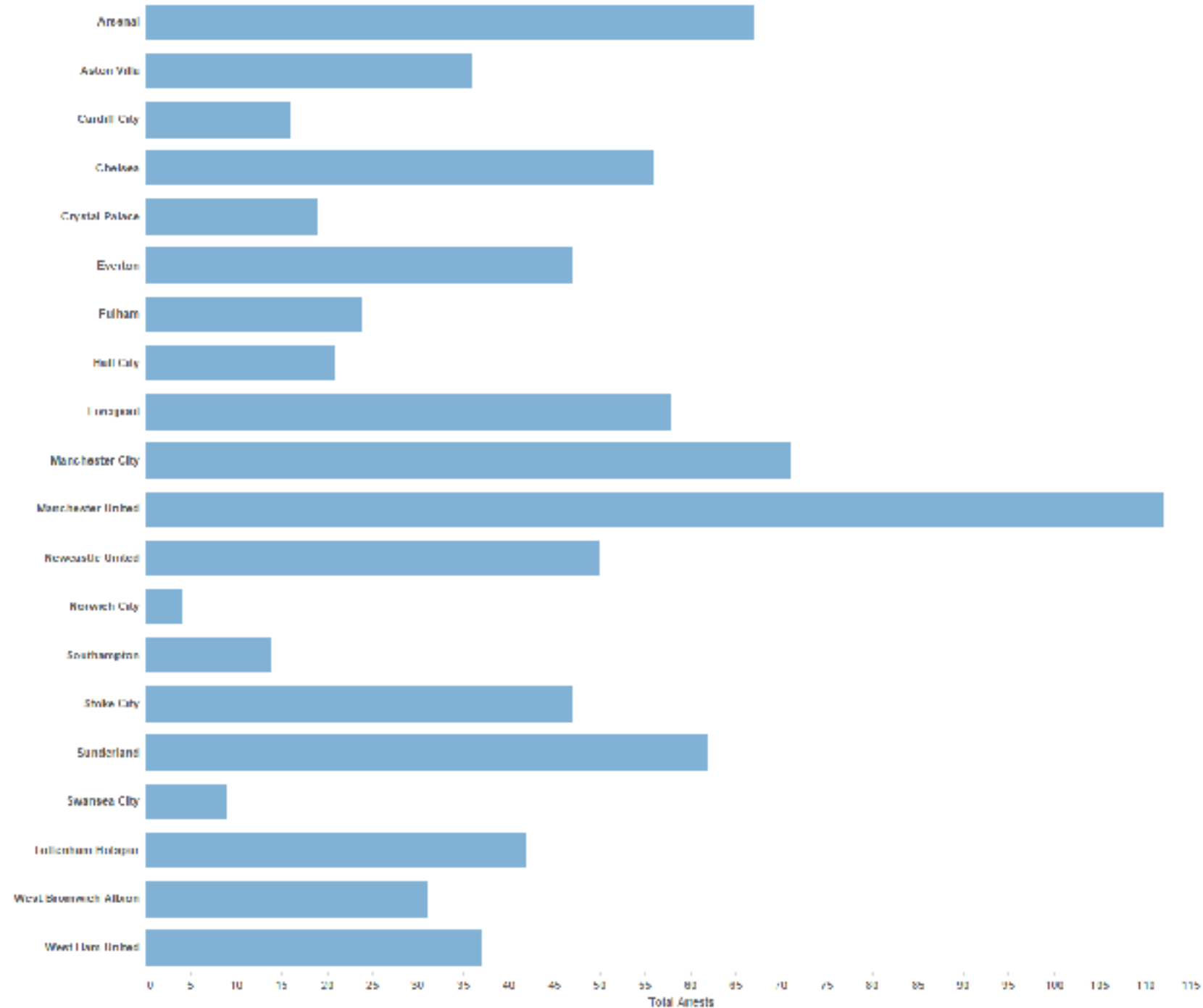
# Separated and aligned and ordered

- best case



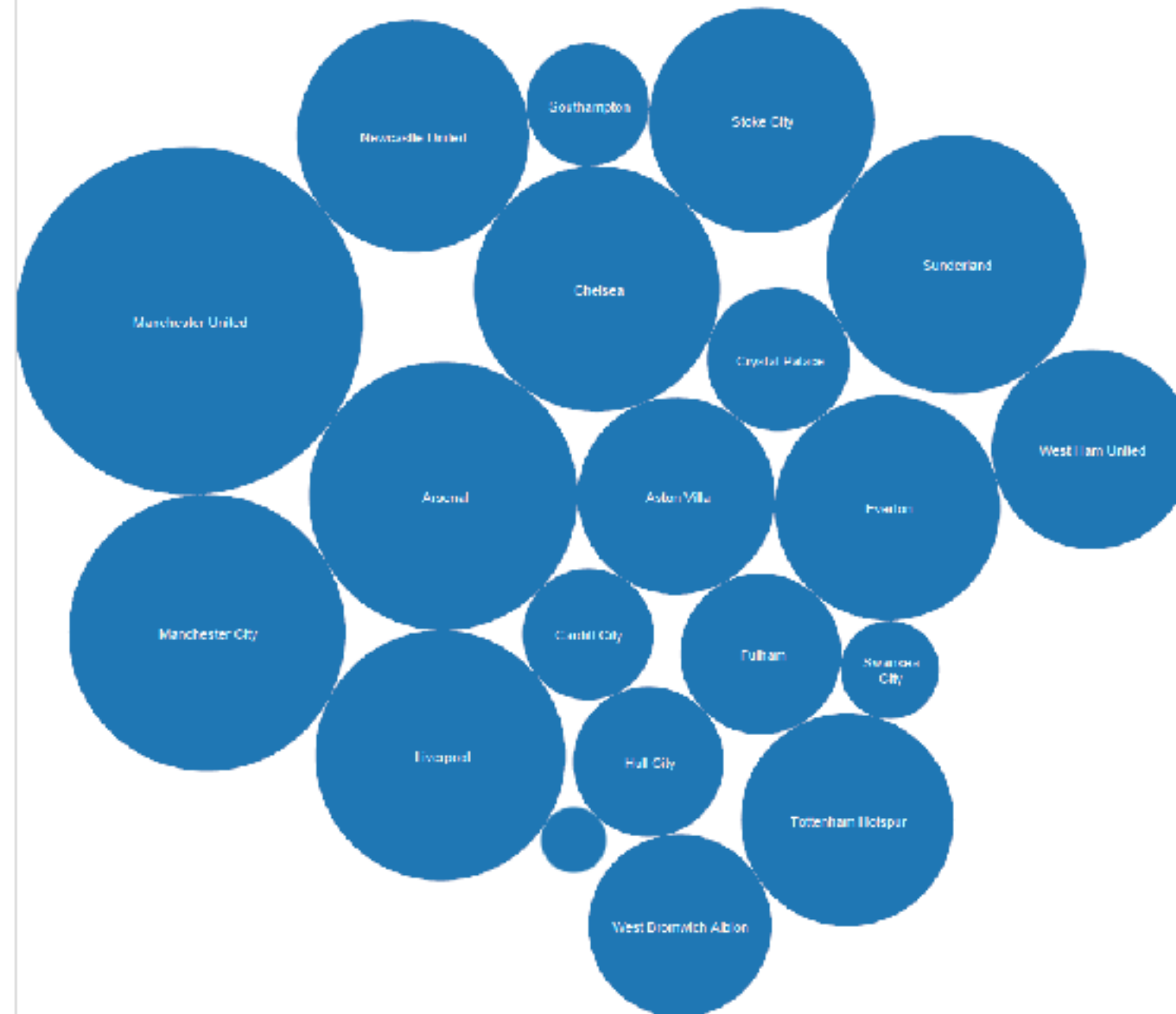
# Separated and aligned but not ordered

- limitation: hard to know rank. what's 4th? what's 7th?



# Separated but not aligned or ordered

- limitation: hard to make comparisons with size (vs aligned position)



# Idiom: bar chart

- one key, one value

- data

- 1 categ attrib, 1 quant attrib

- mark: lines

- channels

- length to express quant value

- spatial regions: one per mark

- separated horizontally, aligned vertically

- ordered by quant attrib

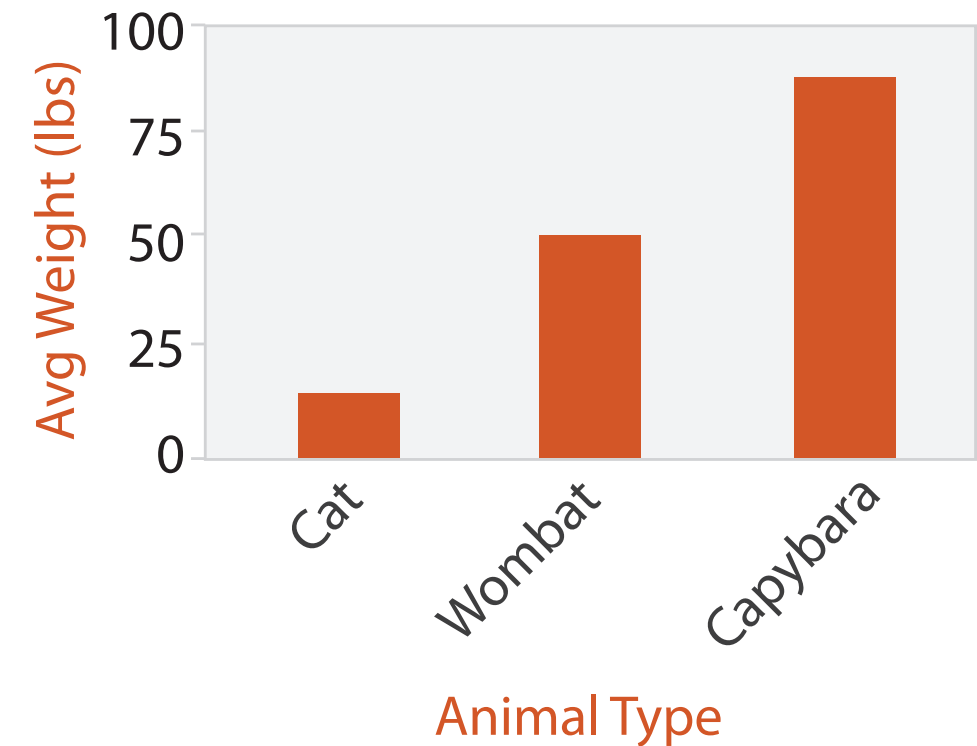
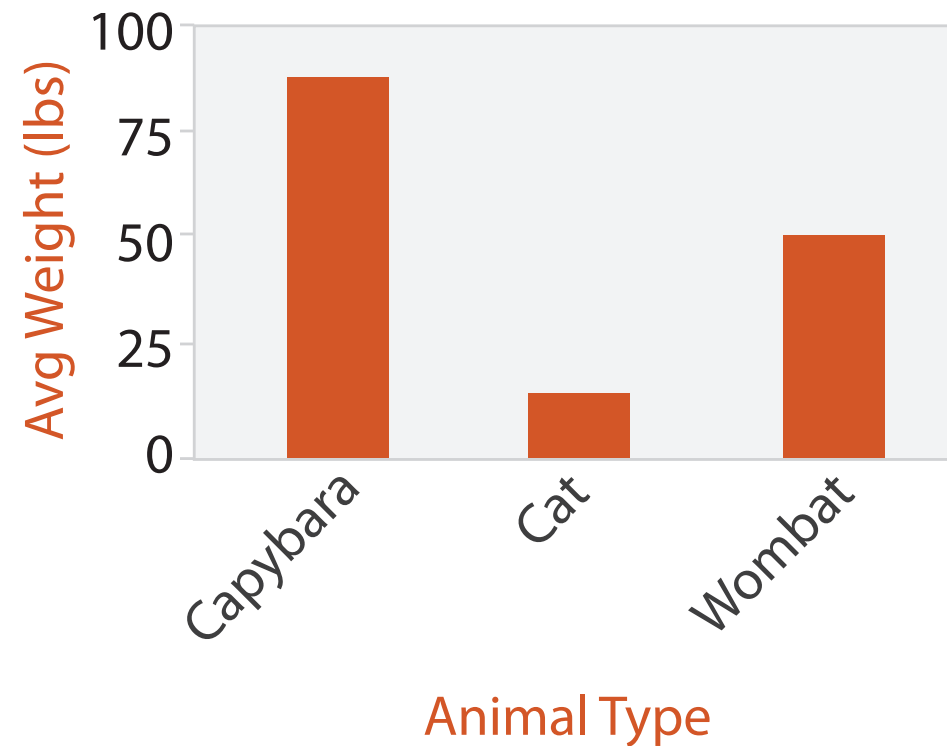
- » by label (alphabetical), by length attrib (data-driven)

- task

- compare, lookup values

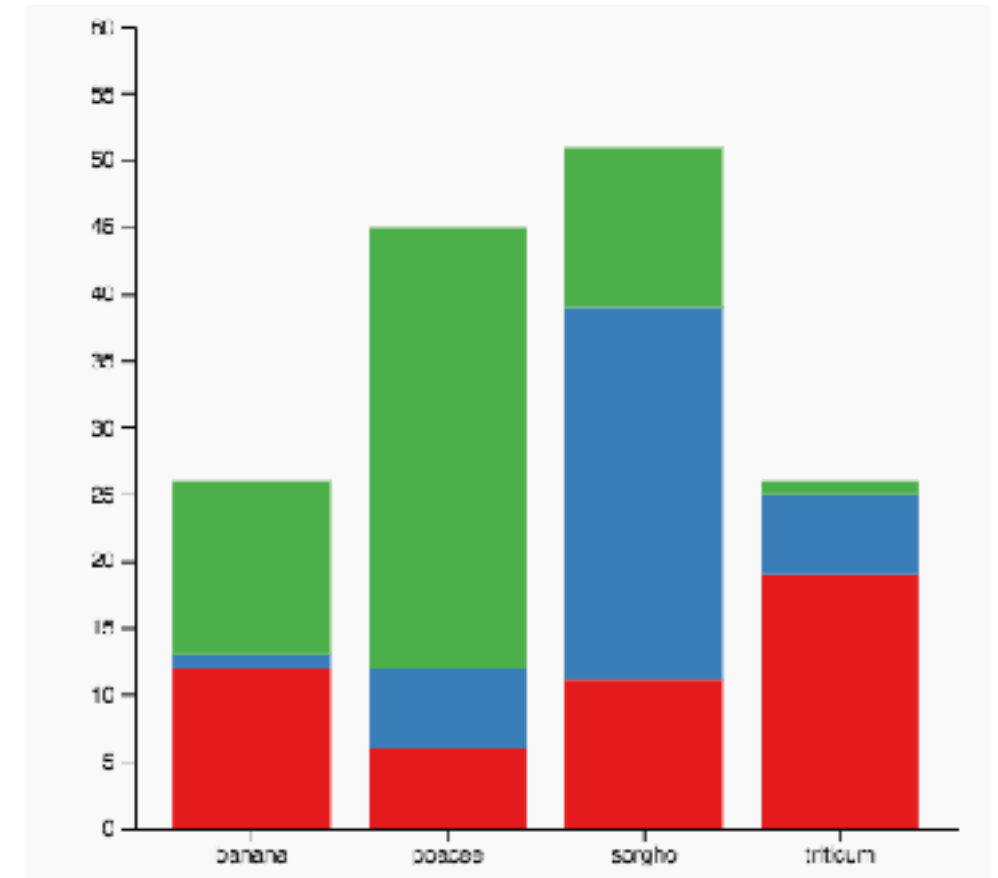
- scalability

- dozens to hundreds of levels for key attrib [bars], hundreds for values



# Idiom: stacked bar chart

- one more key
  - data
    - 2 categ attrib, 1 quant attrib
  - mark: vertical stack of line marks
    - **glyph**: composite object, internal structure from multiple marks
  - channels
    - length and color hue
    - spatial regions: one per glyph
      - aligned: full glyph, lowest bar component
      - unaligned: other bar components
  - task
    - part-to-whole relationship
  - scalability: asymmetric
    - for *stacked* key attrib, 10-12 levels [segments]
    - for *main* key attrib, dozens to hundreds of levels [bars]

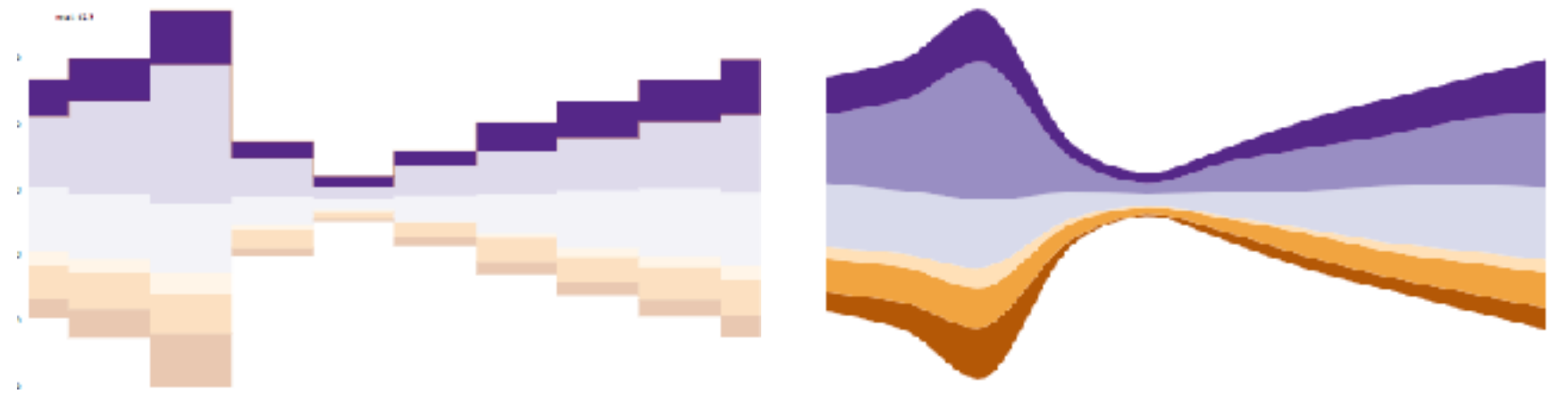


[https://www.d3-graph-gallery.com/graph/barplot\\_stacked\\_basicWide.html](https://www.d3-graph-gallery.com/graph/barplot_stacked_basicWide.html)



# Idiom: streamgraph

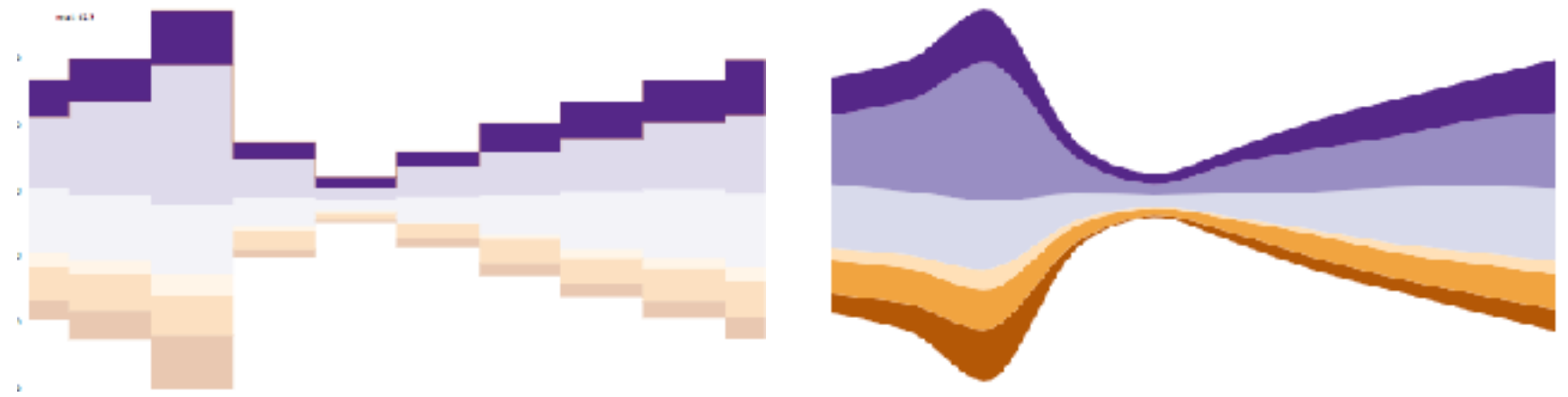
- generalized stacked graph
  - emphasizing horizontal continuity
    - vs vertical items
  - data
    - 1 categ key attrib (movies)
    - 1 ordered key attrib (time)
    - 1 quant value attrib (counts)
  - derived data
    - geometry: layers, where height encodes counts
    - 1 quant attrib (layer ordering)



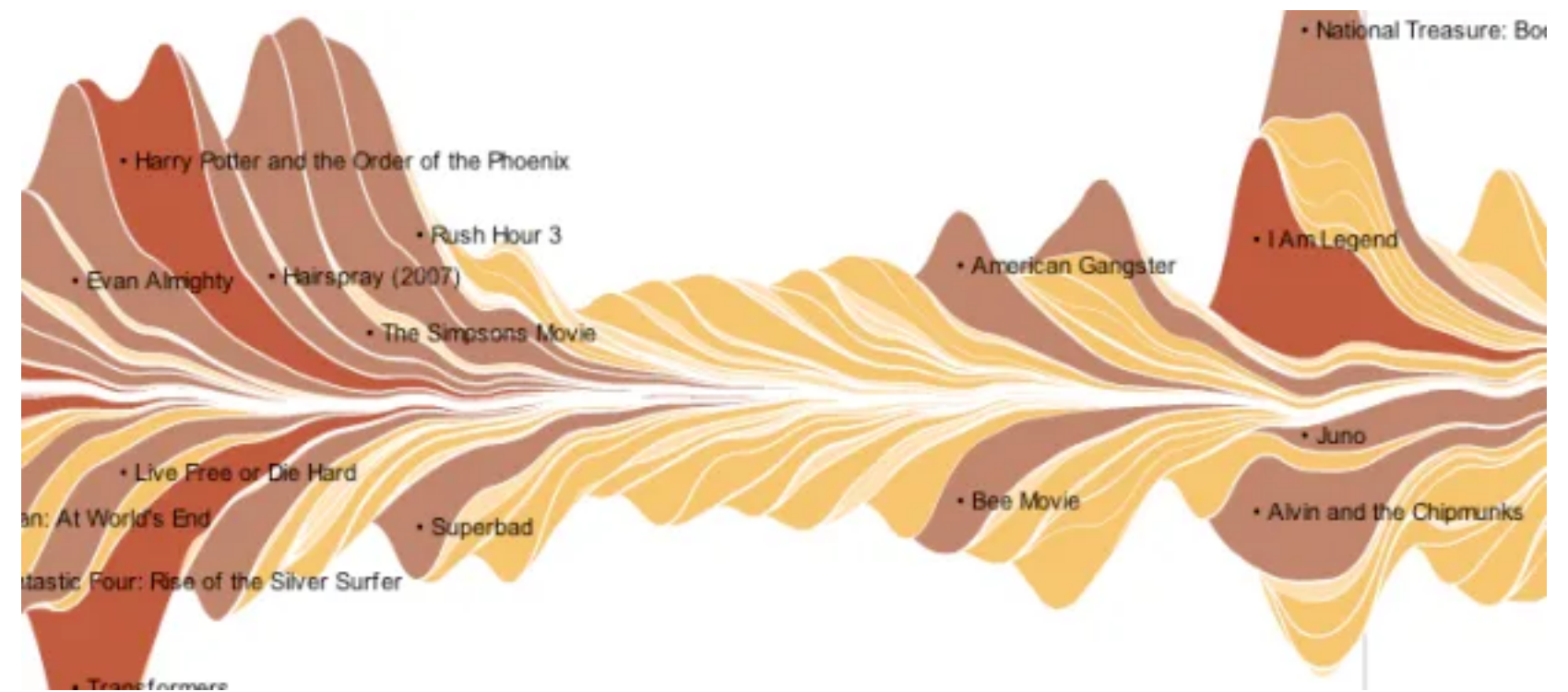
[Stacked Graphs Geometry & Aesthetics. Byron and Wattenberg. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008)* 14(6): 1245–1252, (2008).]

# Idiom: streamgraph

- generalized stacked graph
  - emphasizing horizontal continuity
    - vs vertical items
  - data
    - 1 categ key attrib (movies)
    - 1 ordered key attrib (time)
    - 1 quant value attrib (counts)
  - derived data
    - geometry: layers, where height encodes counts
    - 1 quant attrib (layer ordering)
  - scalability
    - hundreds of time keys
    - dozens to hundreds of movies keys
      - more than stacked bars: most layers don't extend across whole chart



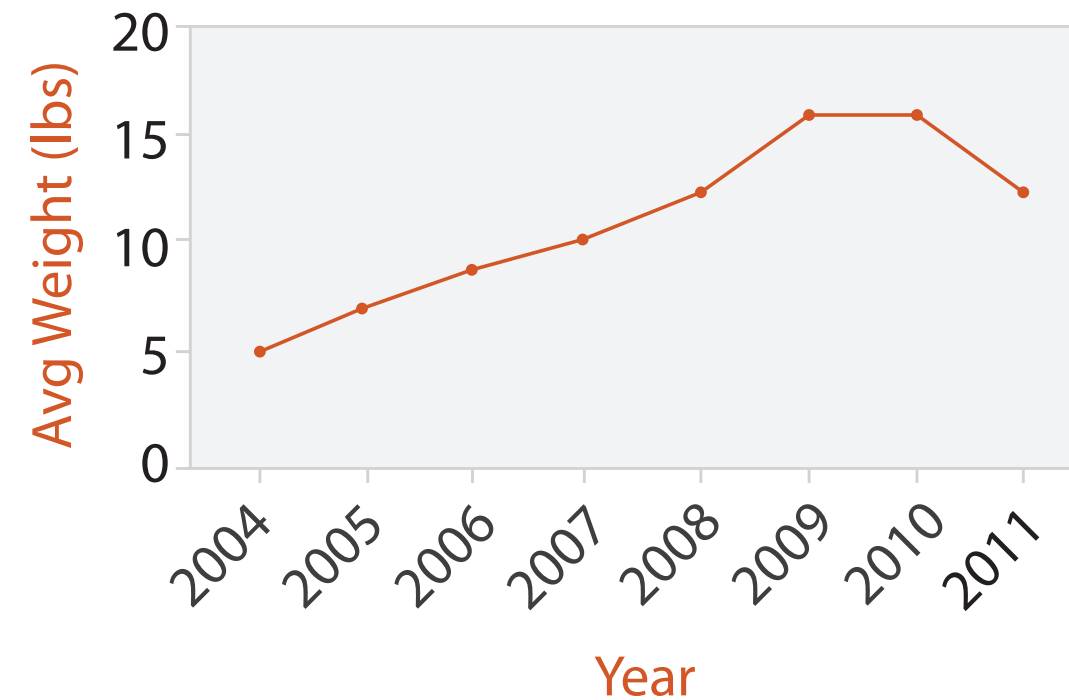
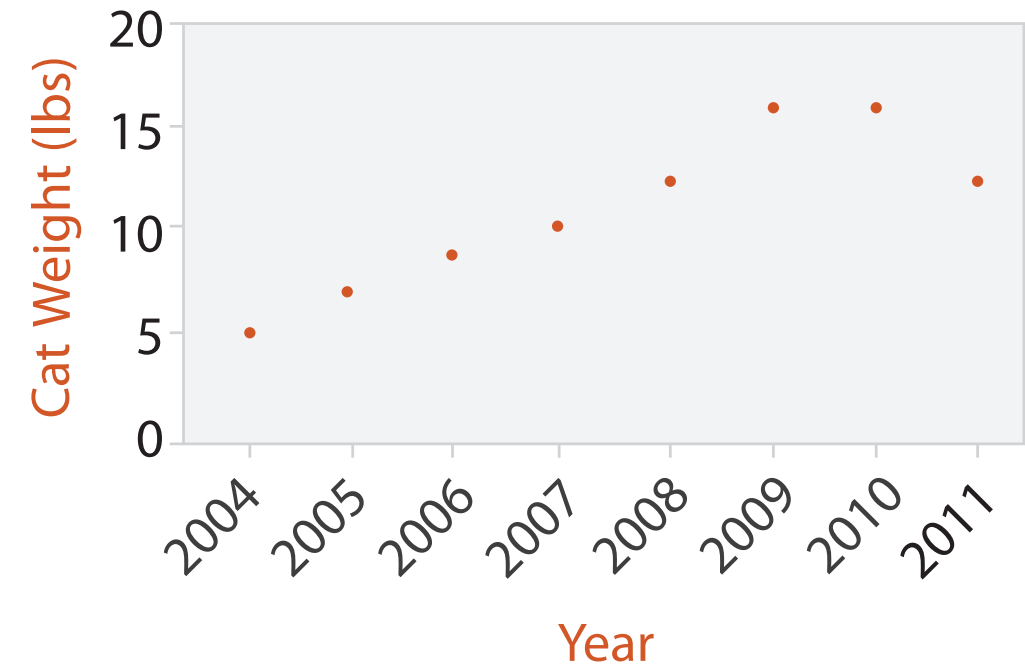
[Stacked Graphs Geometry & Aesthetics. Byron and Wattenberg. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14(6): 1245–1252, (2008).]



<https://flowingdata.com/2008/02/25/ebb-and-flow-of-box-office-receipts-over-past-20-years/>

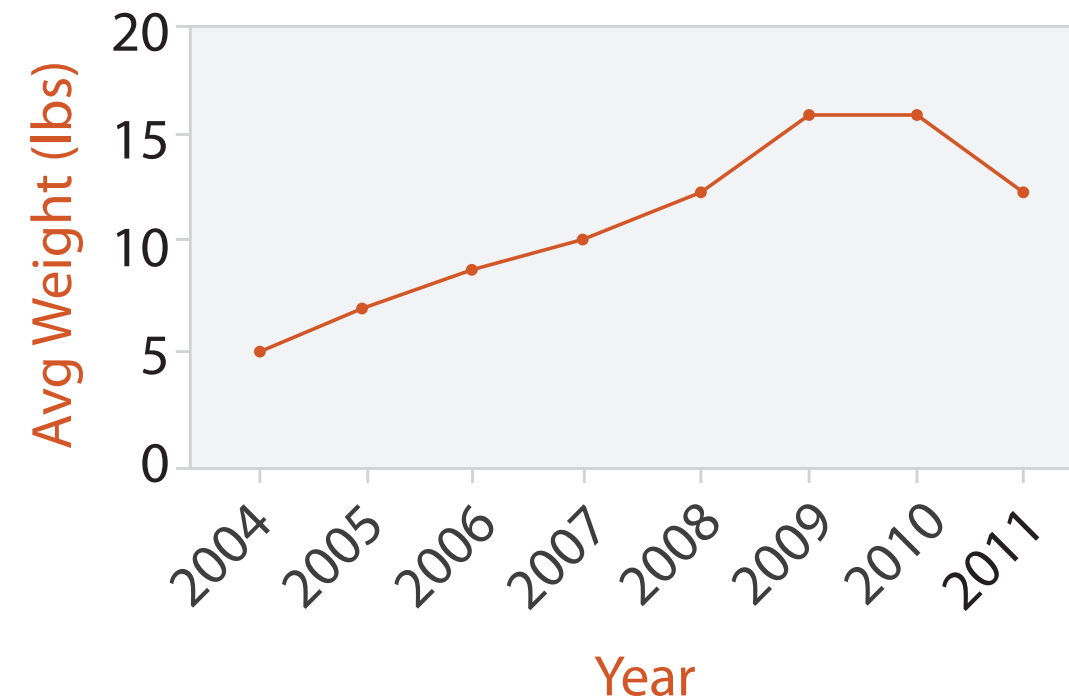
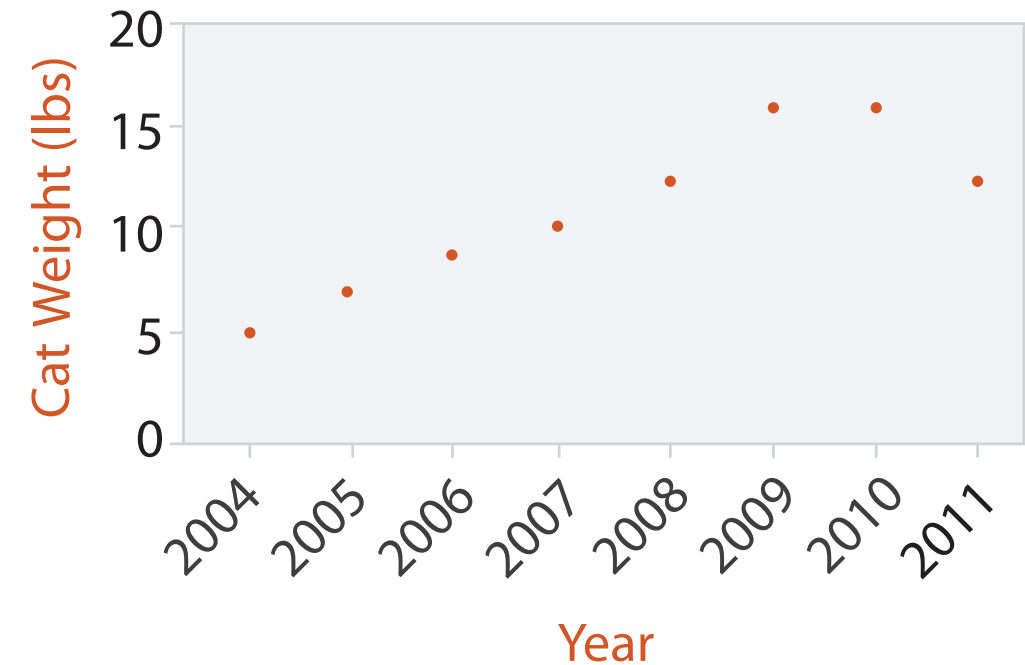
# Idiom: dot / line chart

- one key, one value
  - data
    - 2 quant attribs
  - mark: points
    - AND line connection marks between them
  - channels
    - aligned lengths to express quant value
    - separated and ordered by key attrib into horizontal regions



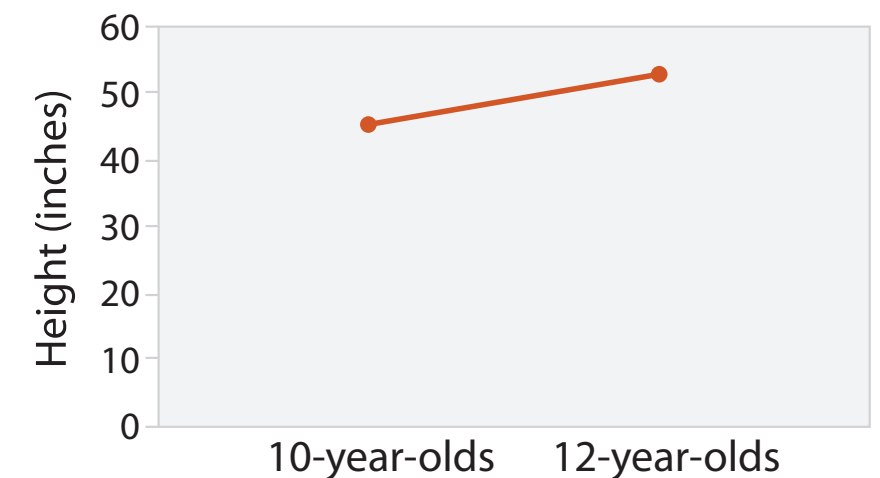
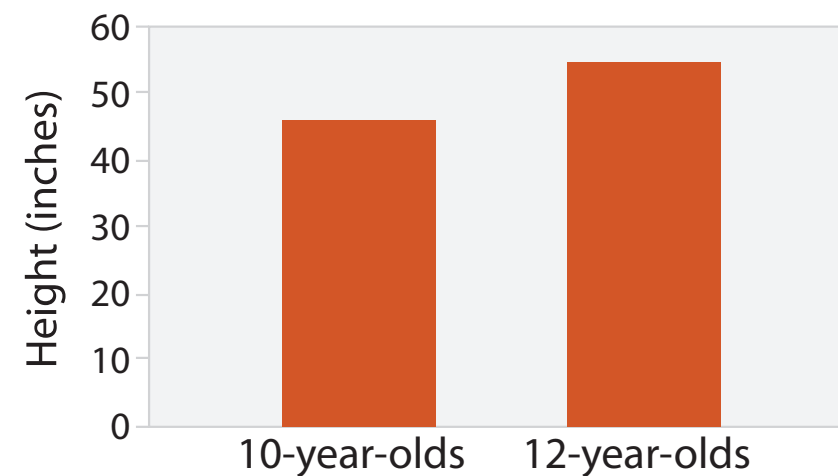
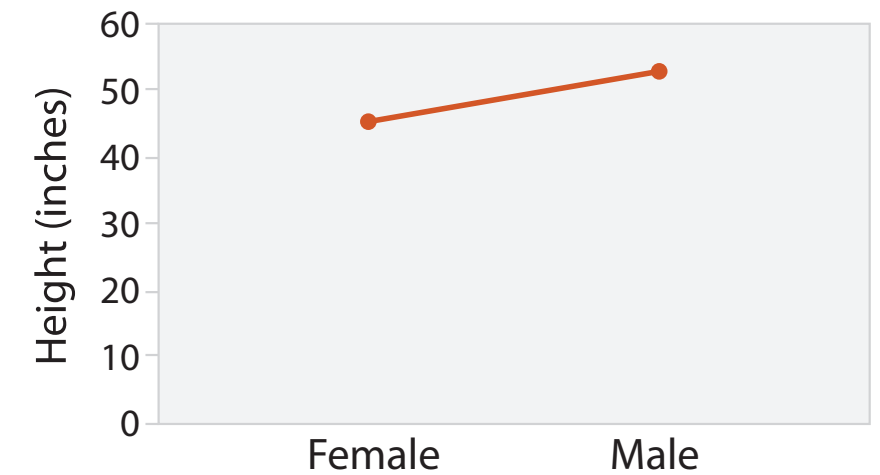
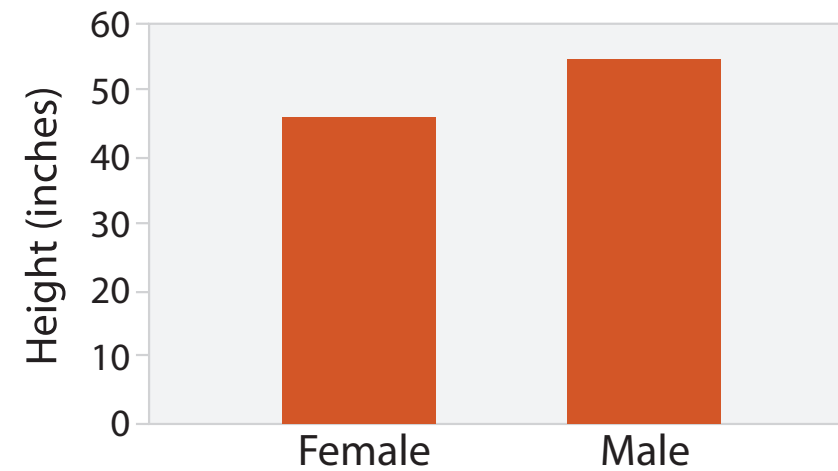
# Idiom: dot / line chart

- one key, one value
  - data
    - 2 quant attribs
  - mark: points
    - AND line connection marks between them
  - channels
    - aligned lengths to express quant value
    - separated and ordered by key attrib into horizontal regions
  - task
    - find trend
      - connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next
  - scalability
    - hundreds of key levels, hundreds of value levels



# Choosing bar vs line charts

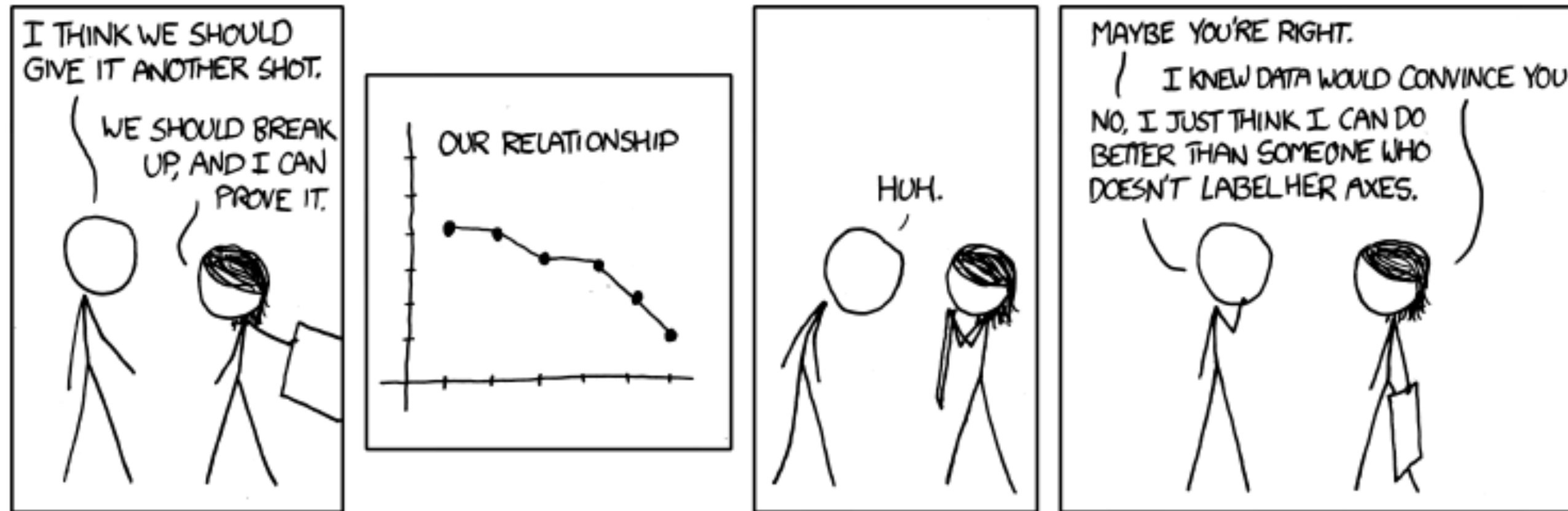
- depends on type of key attrib
  - bar charts if categorical
  - line charts if ordered
- do not use line charts for categorical key attribs
  - violates expressiveness principle
    - implication of trend so strong that it overrides semantics!
      - “The more male a person is, the taller he/she is”



after [Bars and Lines: A Study of Graphic Communication. Zacks and Tversky. *Memory and Cognition* 27:6 (1999), 1073–1079.]

# Chart axes: label them!

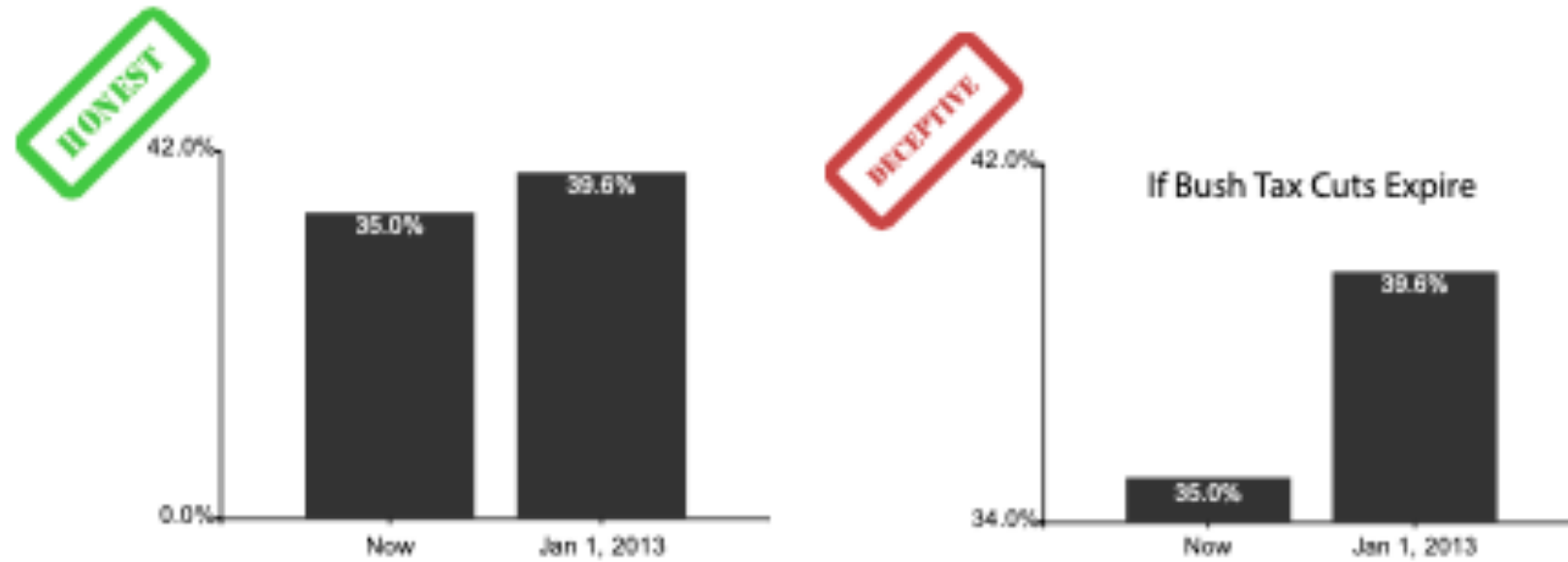
- best practice to label
  - few exceptions: individual small multiple views could share axis label



<https://xkcd.com/833/>

# Chart axes: avoid cropping y axis

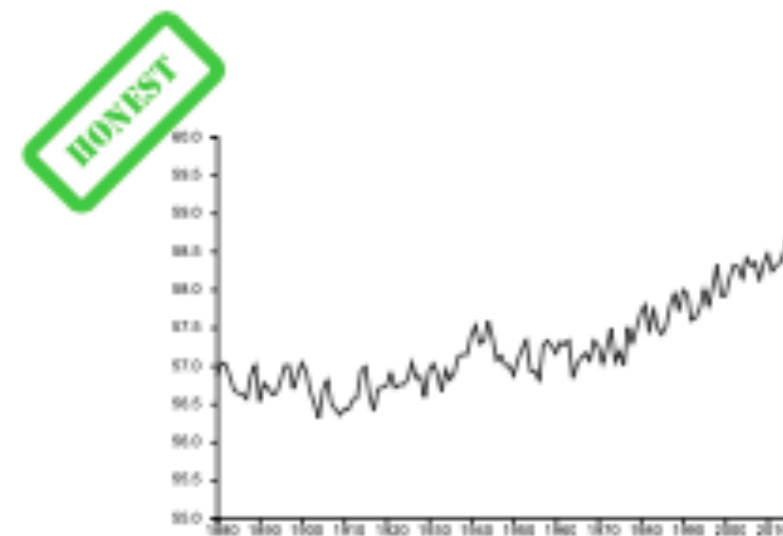
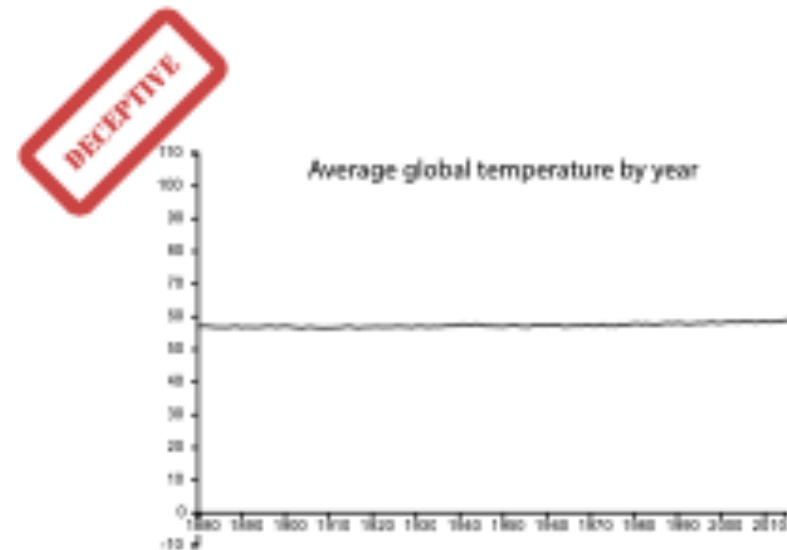
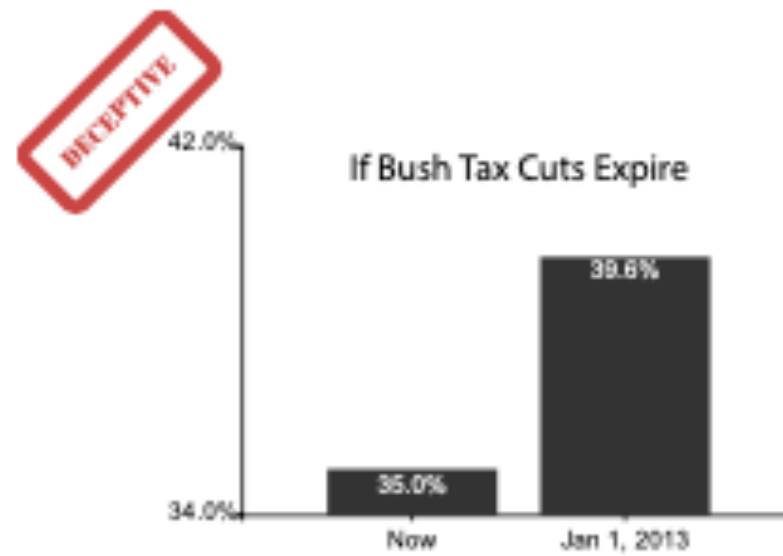
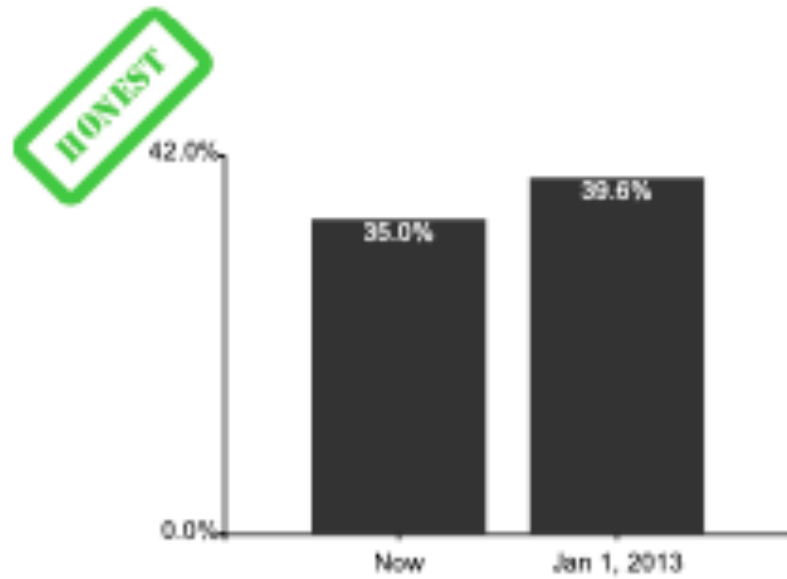
- include 0 at bottom left or slope misleads



*[Truncating the Y-Axis: Threat or Menace?  
Correll, Bertini, & Franconeri, CHI 2020.]*

# Chart axes: avoid cropping y axis

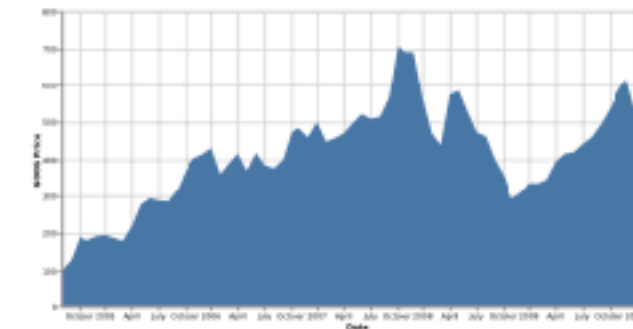
- include 0 at bottom left or slope misleads
  - some exceptions (arbitrary 0, small change matters)



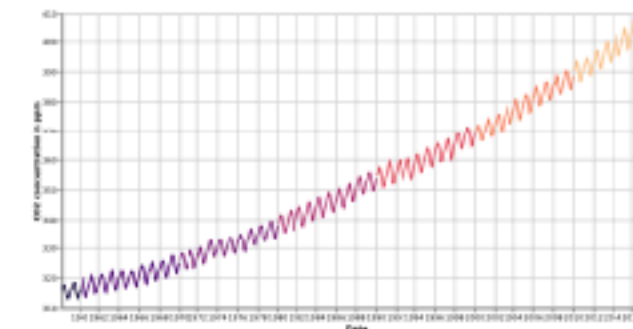
(a) Statistical process charts rely on comparison to an expected value, and so deviations from that value, not from zero, are important



(b) Index charts compare to an indexed value rather than zero.



(c) Stock charts must show small differences in stock value, as these can translate to enormous monetary gains or losses.



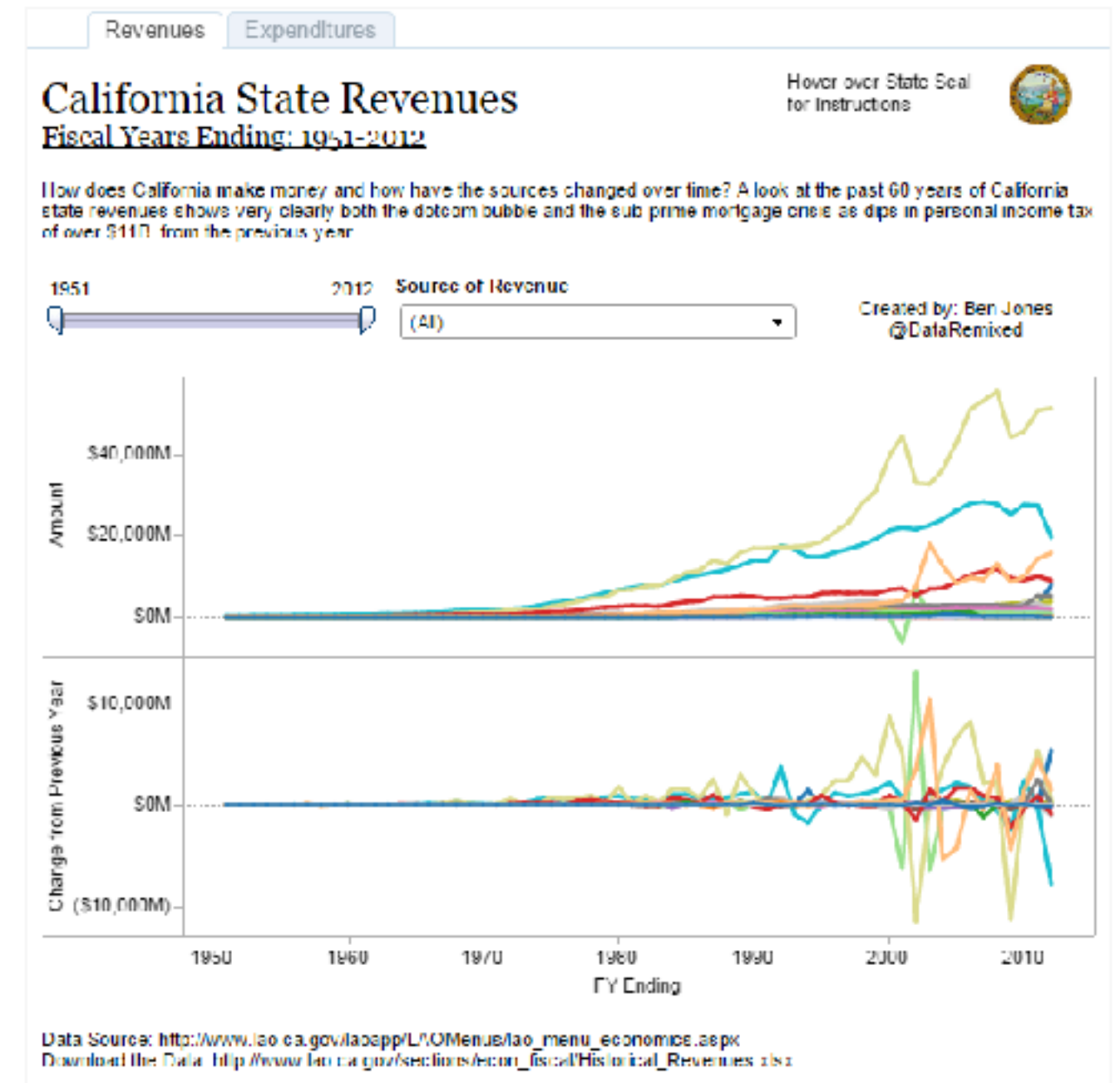
(d) Climate Anomaly charts rely on both highlighting deviation from a non-zero expected value but also emphasize the potentially disastrous impact of even minute changes in climate.

[Truncating the Y-Axis: Threat or Menace?  
Correll, Bertini, & Franconeri, CHI 2020.]



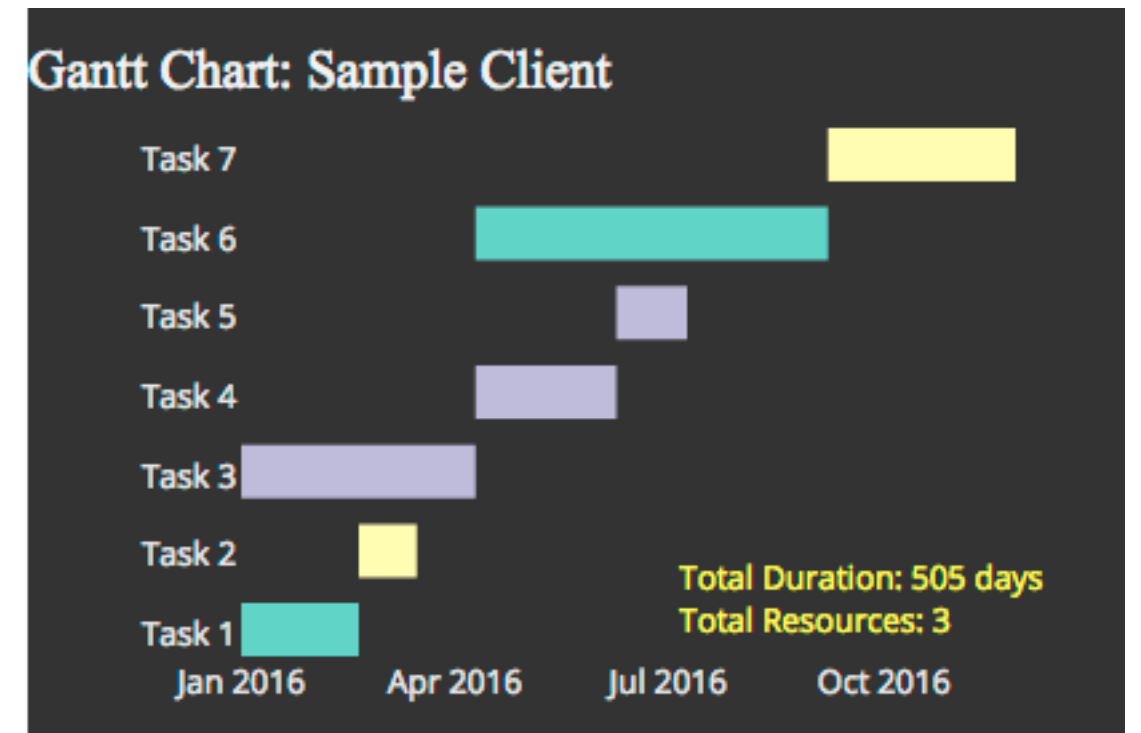
# Idiom: Indexed line charts

- data: 2 quant attribs
  - 1 key + 1 value
- derived data: new quant value attrib
  - index
  - plot instead of original value
- task: show change over time
  - principle: normalized, not absolute
- scalability
  - same as standard line chart



# Idiom: Gantt charts

- one key, two (related) values
  - data
    - 1 categ attrib, 2 quant attribs
  - mark: line
    - length: duration
  - channels
    - horiz position: start time  
(+end from duration)
  - task
    - emphasize temporal overlaps & start/end dependencies between items
  - scalability
    - dozens of key levels [bars]
    - hundreds of value levels [durations]

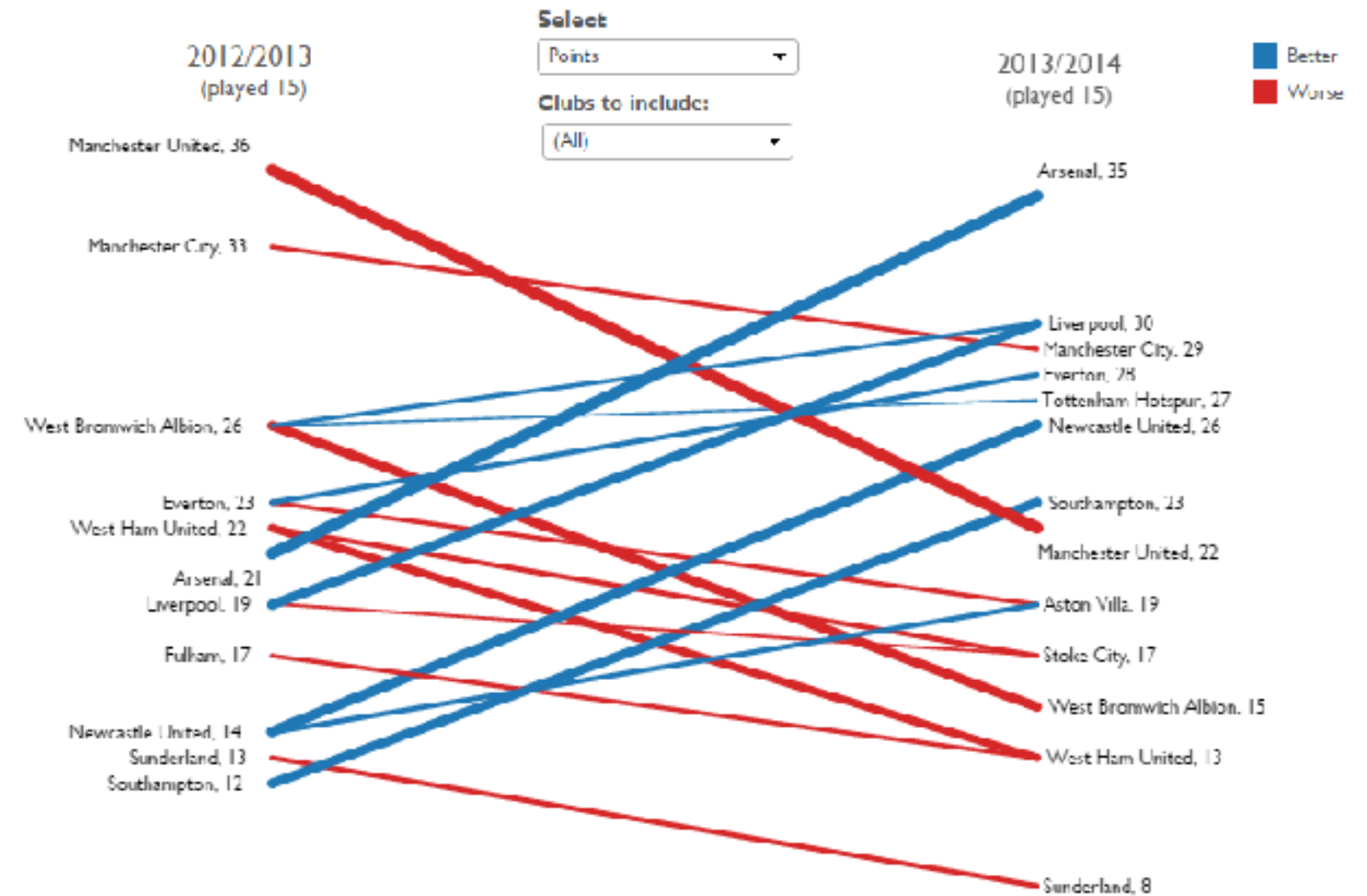


<https://www.r-bloggers.com/gantt-charts-in-r-using-plotly/>

# Idiom: Slopegraphs

- two values
  - data
    - 2 quant value attribs
    - (1 derived attrib: change magnitude)
  - mark: point + line
    - line connecting mark between pts
  - channels
    - 2 vertical pos: express attrib value
    - (linewidth/size, color)
  - task
    - emphasize changes in rank/value
  - scalability
    - hundreds of value levels
    - dozens of items

## Barclay's Premier League Tables: Comparing 2012/2013 Starts to 2013/2014 Starts



<https://public.tableau.com/profile/ben.jones#!/vizhome/Slopegraphs/Slopegraphs>

# 2 Keys

→ 0 Keys

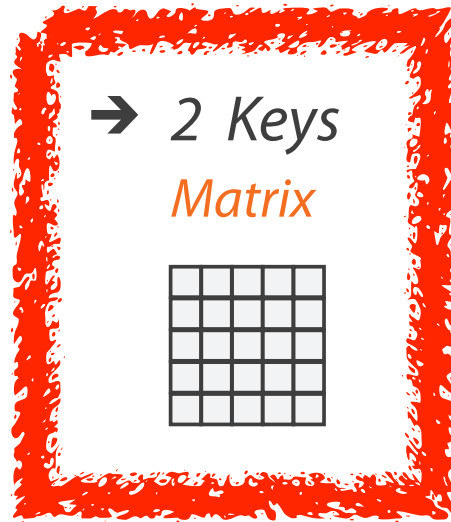
→ Express Values



→ 1 Key  
*List*



→ 2 Keys  
*Matrix*

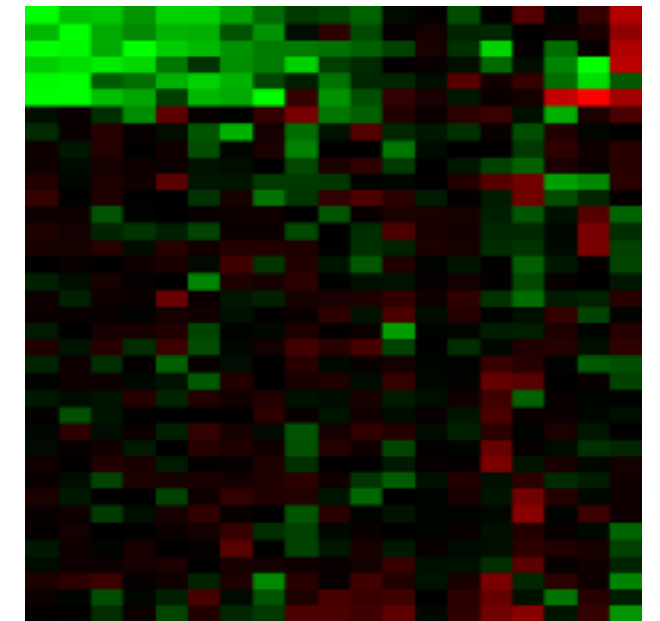
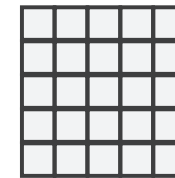


# Idiom: heatmap

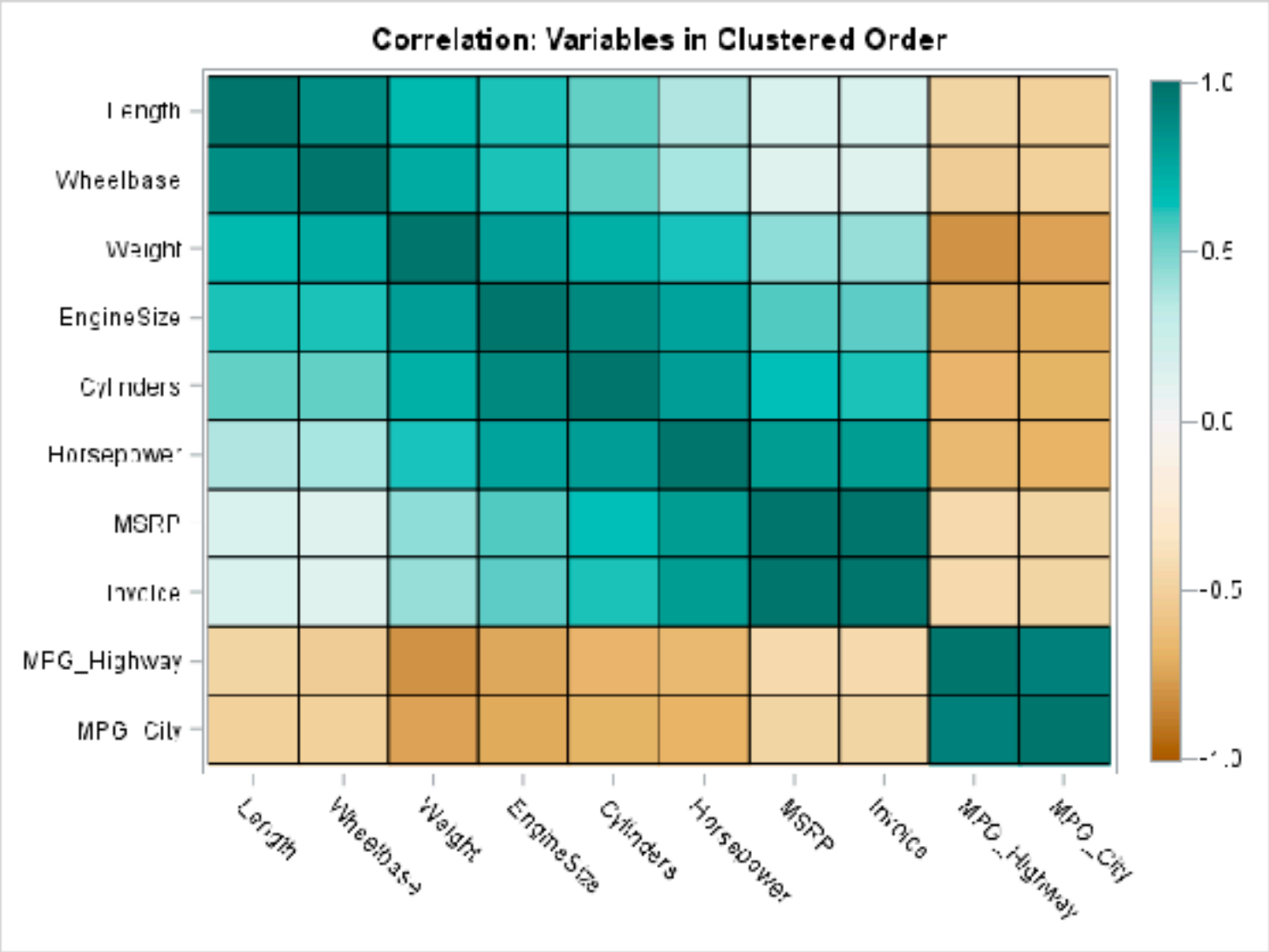
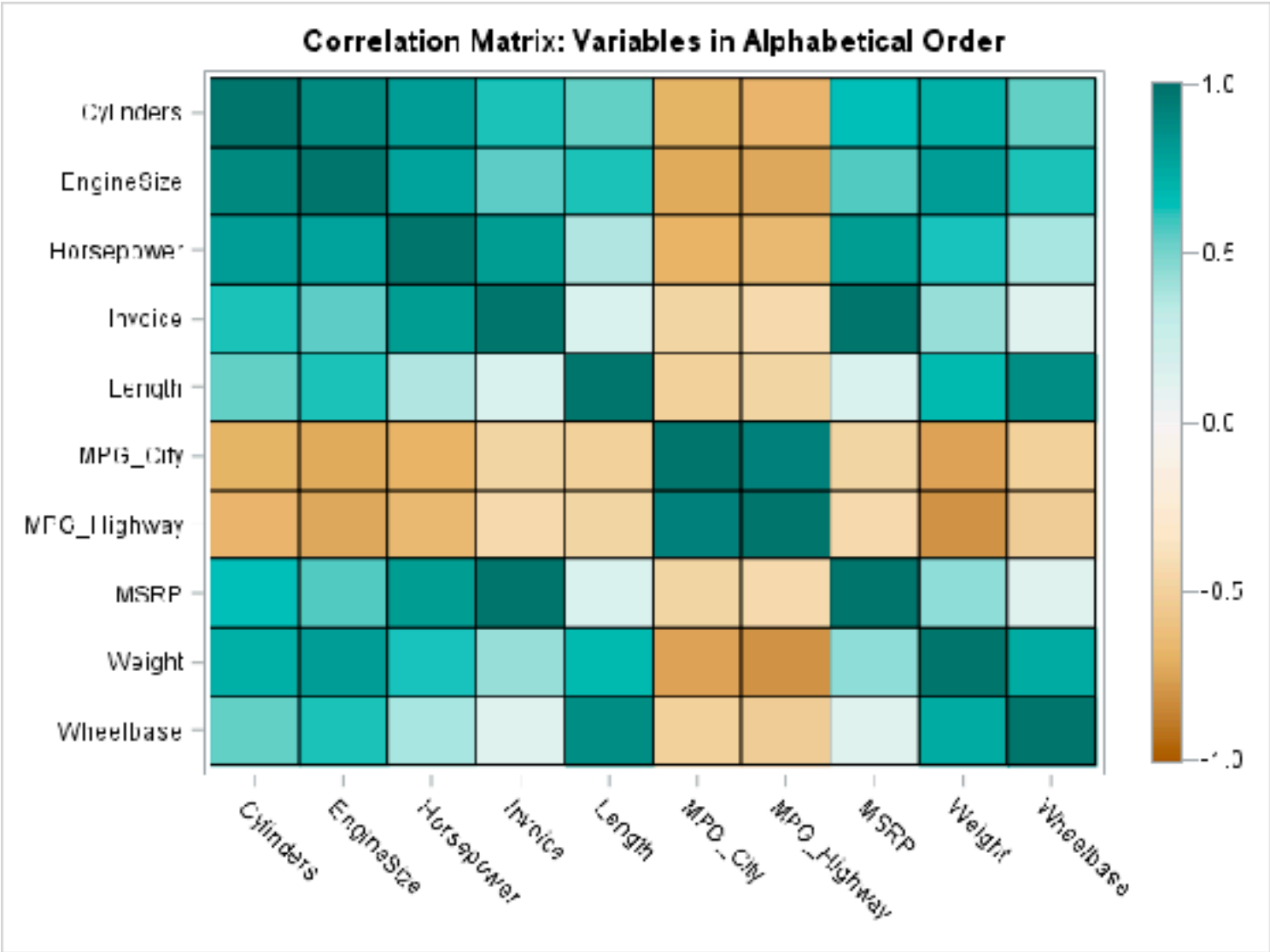
- two keys, one value
  - data
    - 2 categ attribs (gene, experimental condition)
    - 1 quant attrib (expression levels)
  - marks: point
    - separate and align in 2D matrix
      - indexed by 2 categorical attributes
  - channels
    - color by quant attrib
      - (ordered diverging colormap)
  - task
    - find clusters, outliers
  - scalability
    - 1M items, 100s of categ levels, ~10 quant attrib levels

→ 2 Keys

*Matrix*

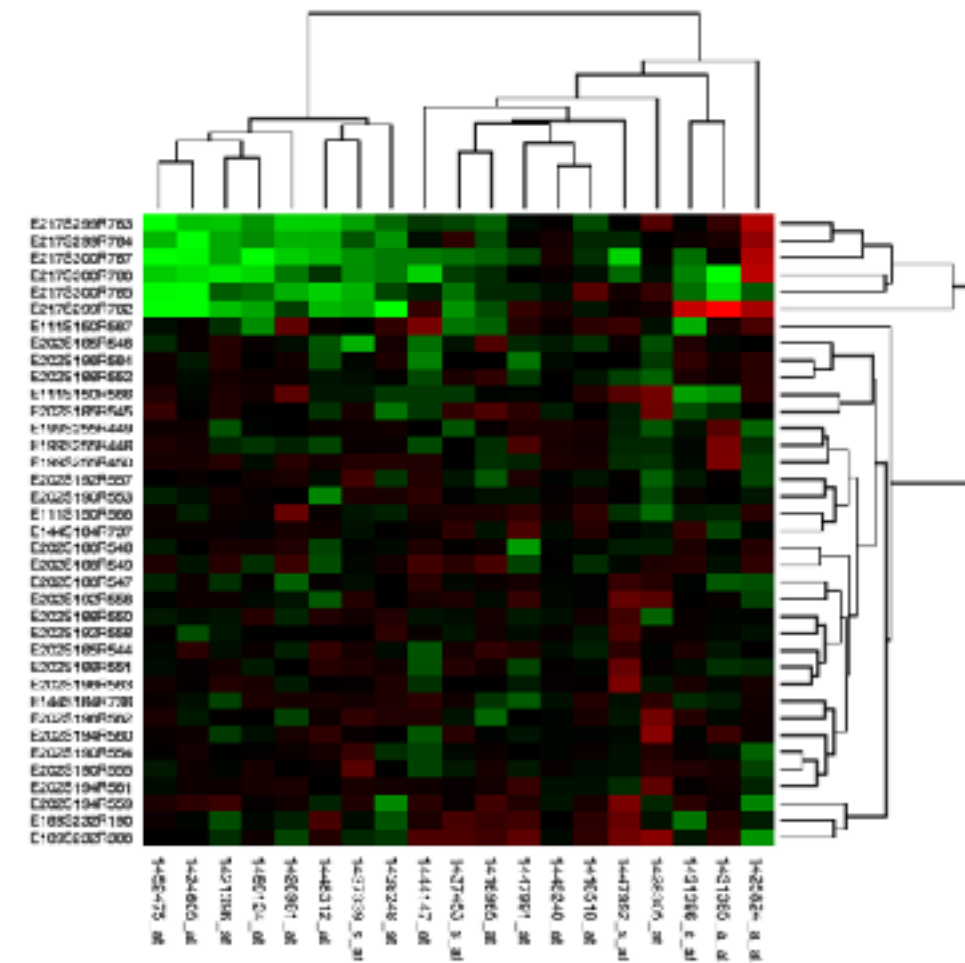


# Heatmap reordering



# Idiom: cluster heatmap

- in addition
  - derived data
    - 2 cluster hierarchies
  - dendrogram
    - parent-child relationships in tree with connection line marks
    - leaves aligned so interior branch heights easy to compare
  - heatmap
    - marks (re-)ordered by cluster hierarchy traversal
    - task: assess quality of clusters found by automatic methods



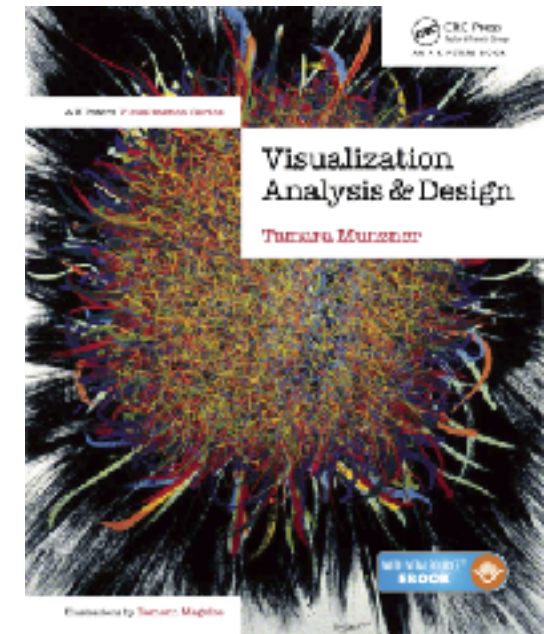
# Visualization Analysis & Design

## *Tables (Ch 7) II*

**Tamara Munzner**

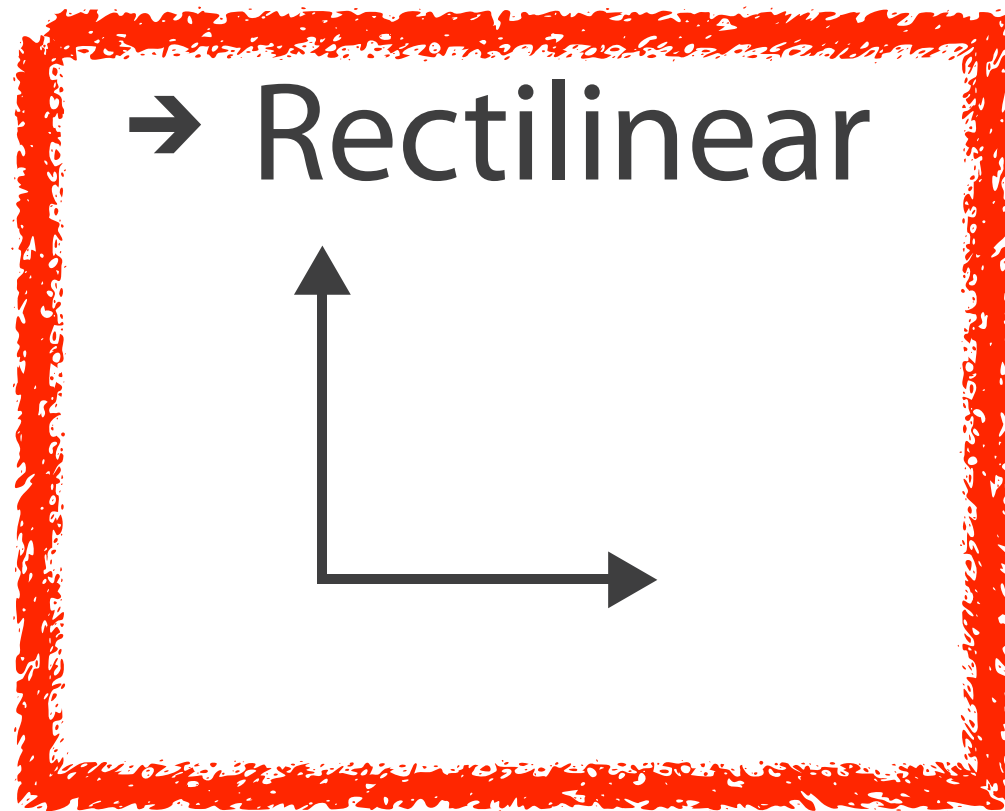
Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)

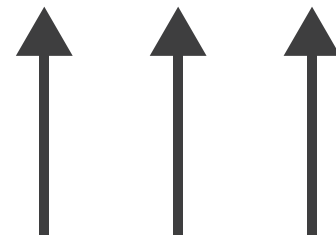




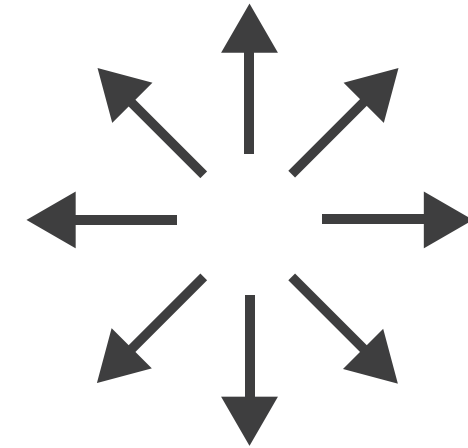
# → Axis Orientation



→ Parallel

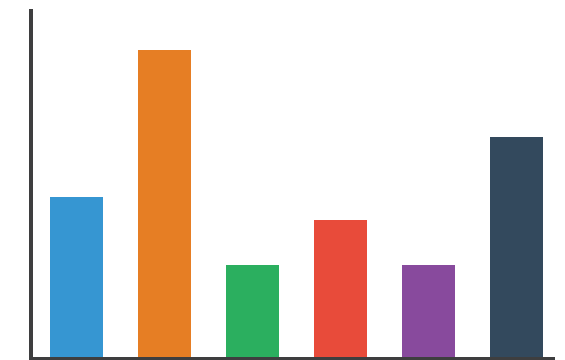
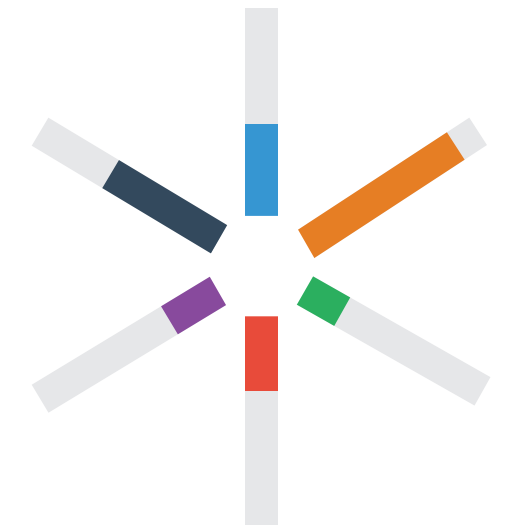


→ Radial



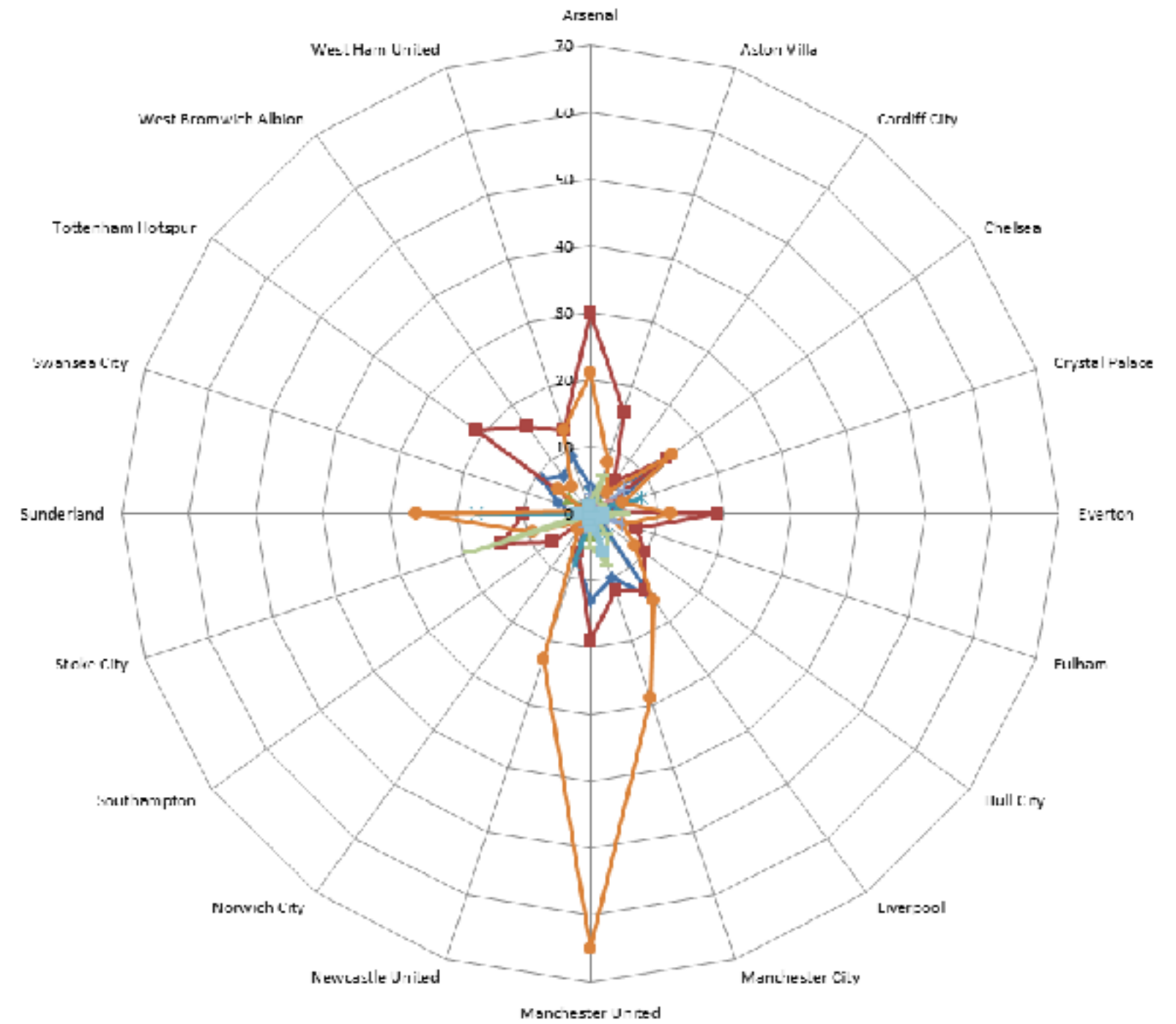
# Idioms: radial bar chart, star plot

- star plot
  - line mark, radial axes meet at central point
- radial bar chart
  - line mark, radial axes meet at central ring
  - channels: length, angle/orientation
- bar chart
  - rectilinear axes, aligned vertically
- accuracy
  - length not aligned with radial layouts
    - less accurately perceived than rectilinear aligned



# Idiom: radar plot

- radial line chart
  - point marks, radial layout
  - connecting line marks
- avoid unless data is cyclic



# “Radar graphs: Avoid them (99.9% of the time)”



## Os sinais da bússola eleitoral

Disputa de 2010 foi parecida com a de 2006

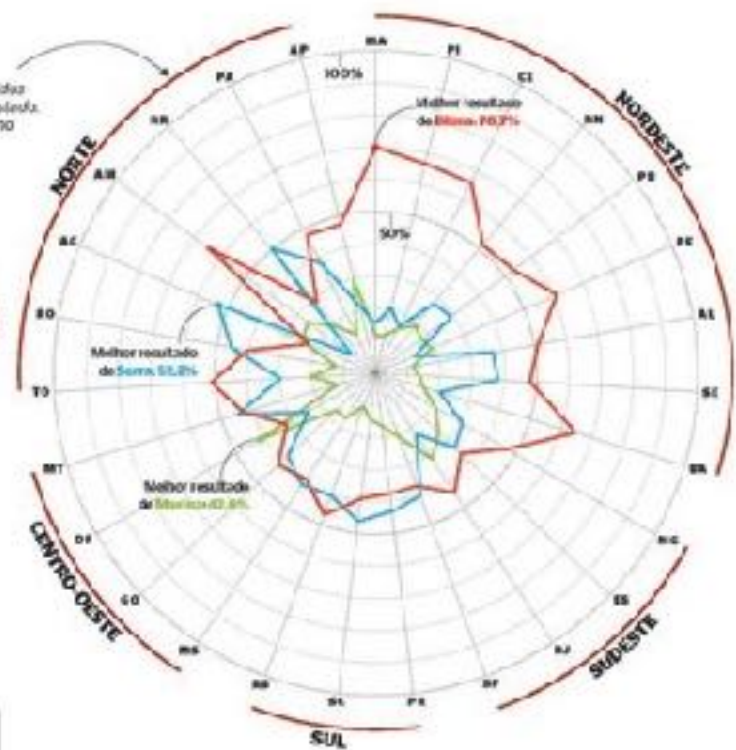
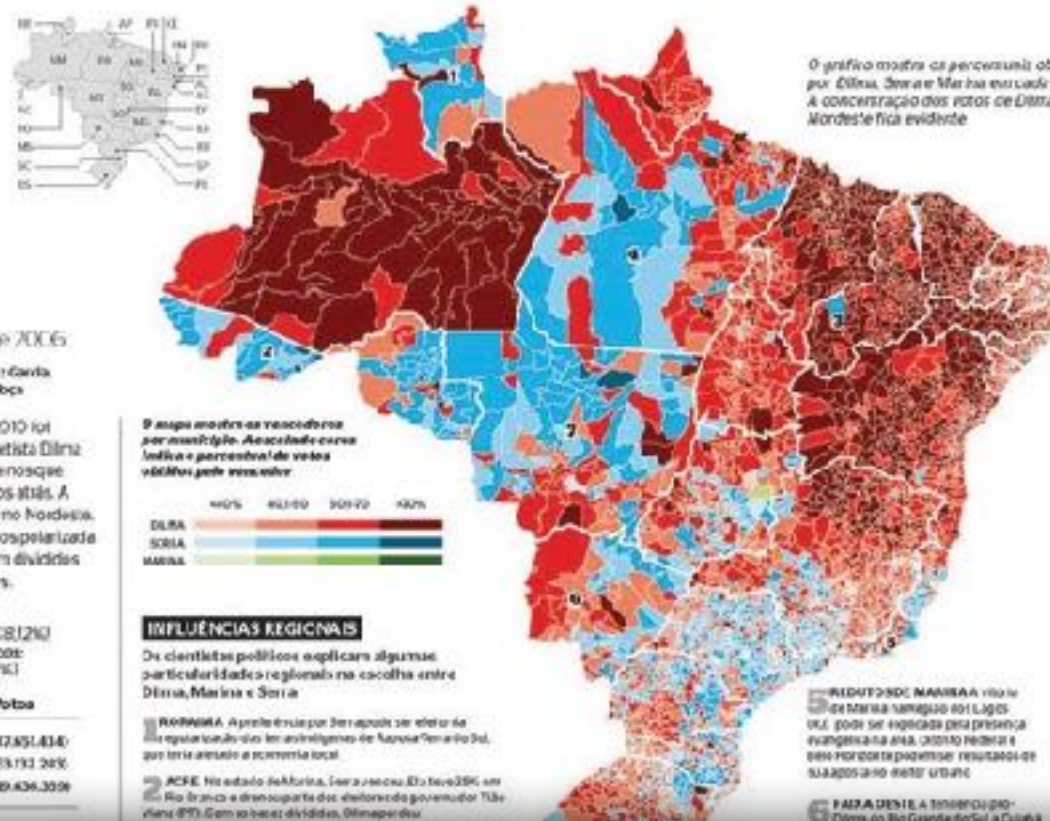
Alberto Campos, Alexandre Moraes, Carlos Eduardo Cruz, Carlos Elias, Daniel Junior, Marco Vinícius e Ricardo Mendes

O **PROFUNDAMENTO** da eleição presidencial de 2010 foi muito parecido com a disputa de 2006. Apesar de Dilma Rousseff ter apenas 17 pontos percentuais menos que o índice obtido pelo presidente Lula quatro anos atrás, a concentração maior de seus votos também foi no Nordeste. Dessa vez, porém, a disputa foi um pouco menos polarizada. Os votos que se dividiram segundo turno foram divididos entre o tucano José Serra e a verde Marina Silva.

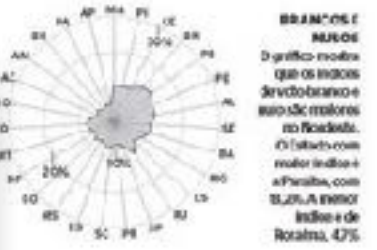
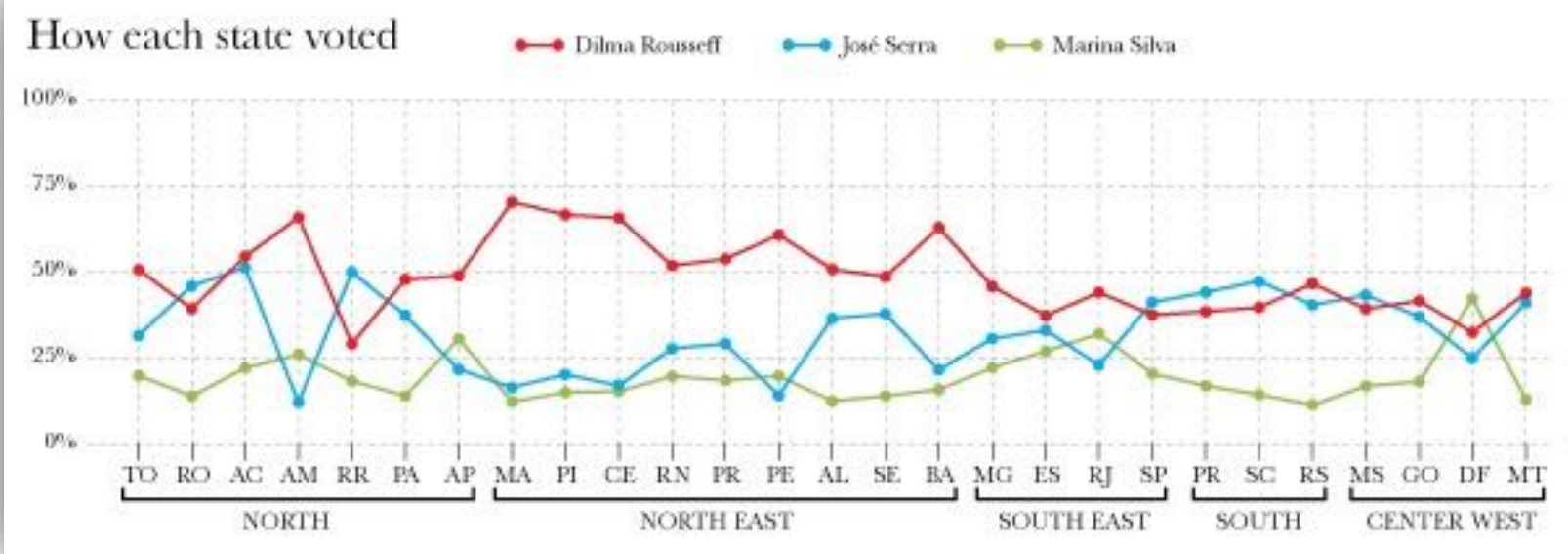
**Votos:** 135.904.433, **abstenção:** 24.610.296 (18,12%)  
**votos válidos:** 101.590.157 (91,36%), **votos brancos:** 3.479.340 (3,39%) e **votos nulos:** 4.510.754 (4,51%)

Candidato	votos	Porcentagem
Dilma Rousseff (PT)	47.851.434	46,9%
José Serra (PSDB)	33.732.289	33,0%
Marina Silva (PV)	19.426.309	19,3%

**Outros candidatos**  
 Nilson Freixo (PSB)  
 José Maria Cyrus (PMDB)  
 Sá Maria (PPS)  
 Ney Nóbrega (PPM)  
 Ivan Pinheiro (PSC)  
 Rui Costa Pinheiro (PSTU)



original  
difficult to interpret



redesign for  
rectilinear

# Idioms: pie chart, coxcomb chart

- pie chart

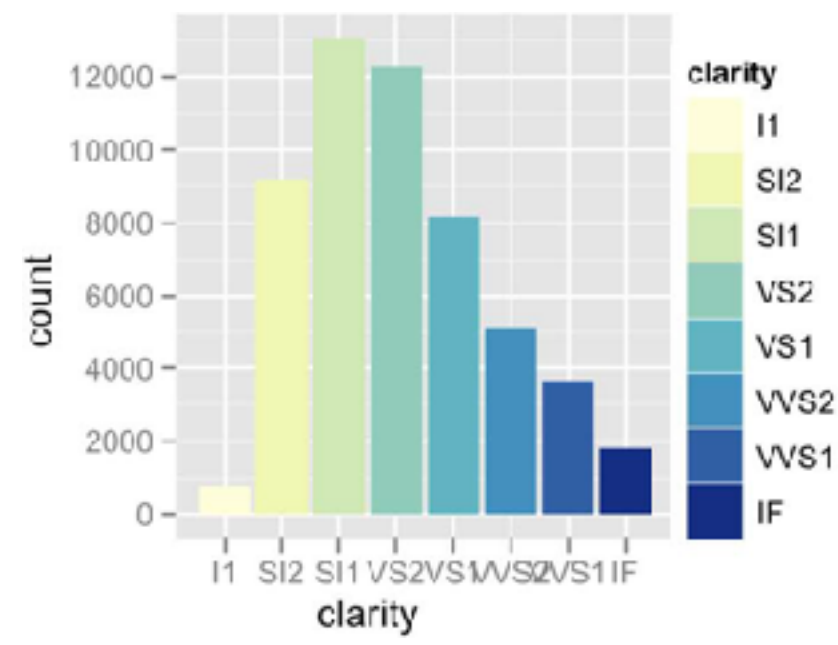
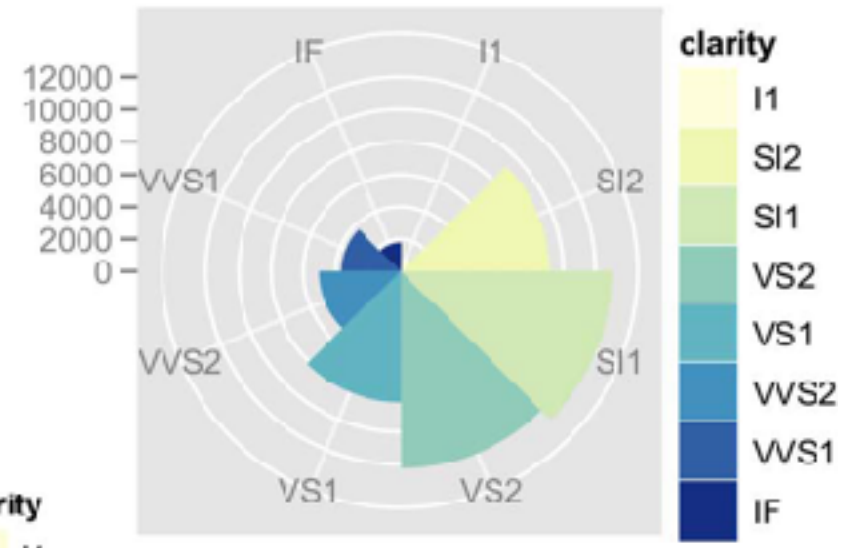
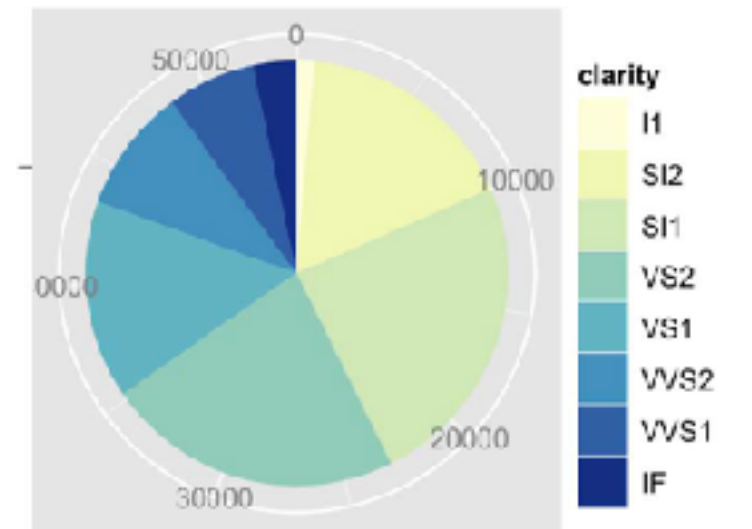
- **interlocking area** marks with angle channel: **2D area varies**
  - separated & ordered radially, uniform height
- accuracy: area less accurate than rectilinear aligned line length
- **task: part-to-whole judgements**

- coxcomb chart

- line marks with length channel: **ID length varies**
  - separated & ordered radially, uniform width
- direct analog to radial bar charts

- data

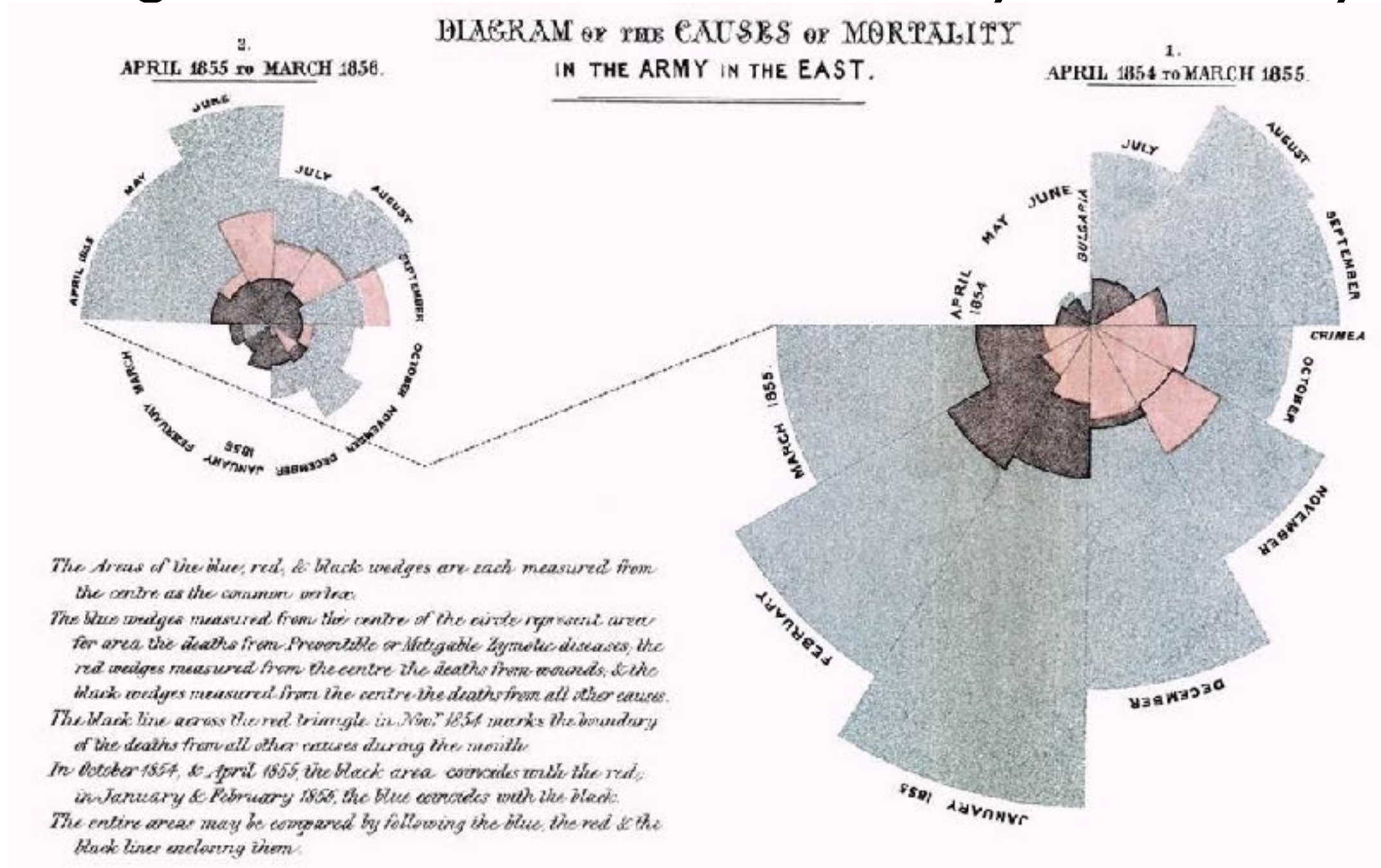
- I categ key attrib, I quant value attrib



[A layered grammar of graphics. Wickham. Journ. Computational and Graphical Statistics 19:1 (2010), 3–28.]

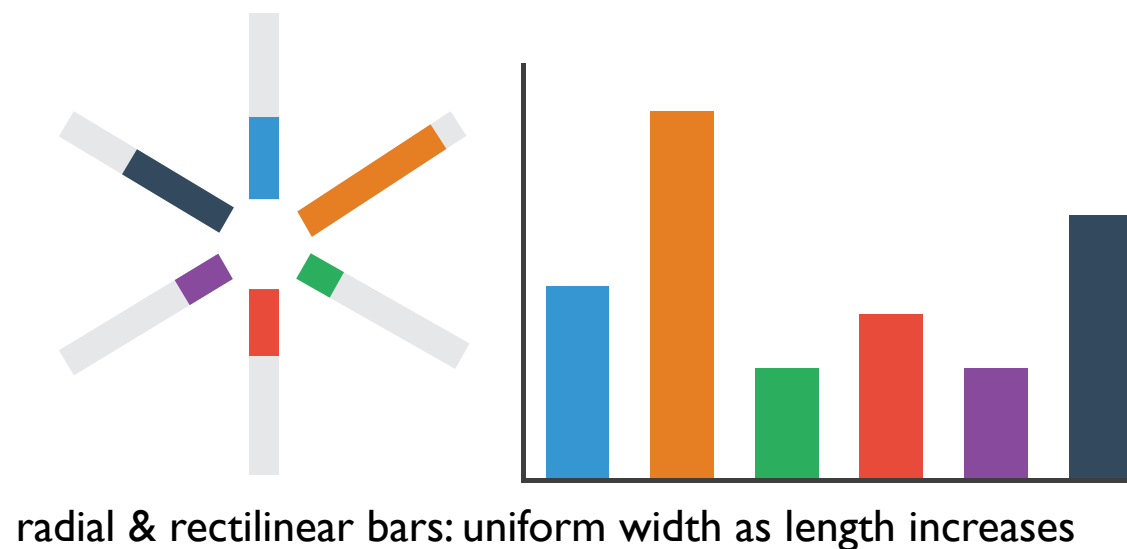
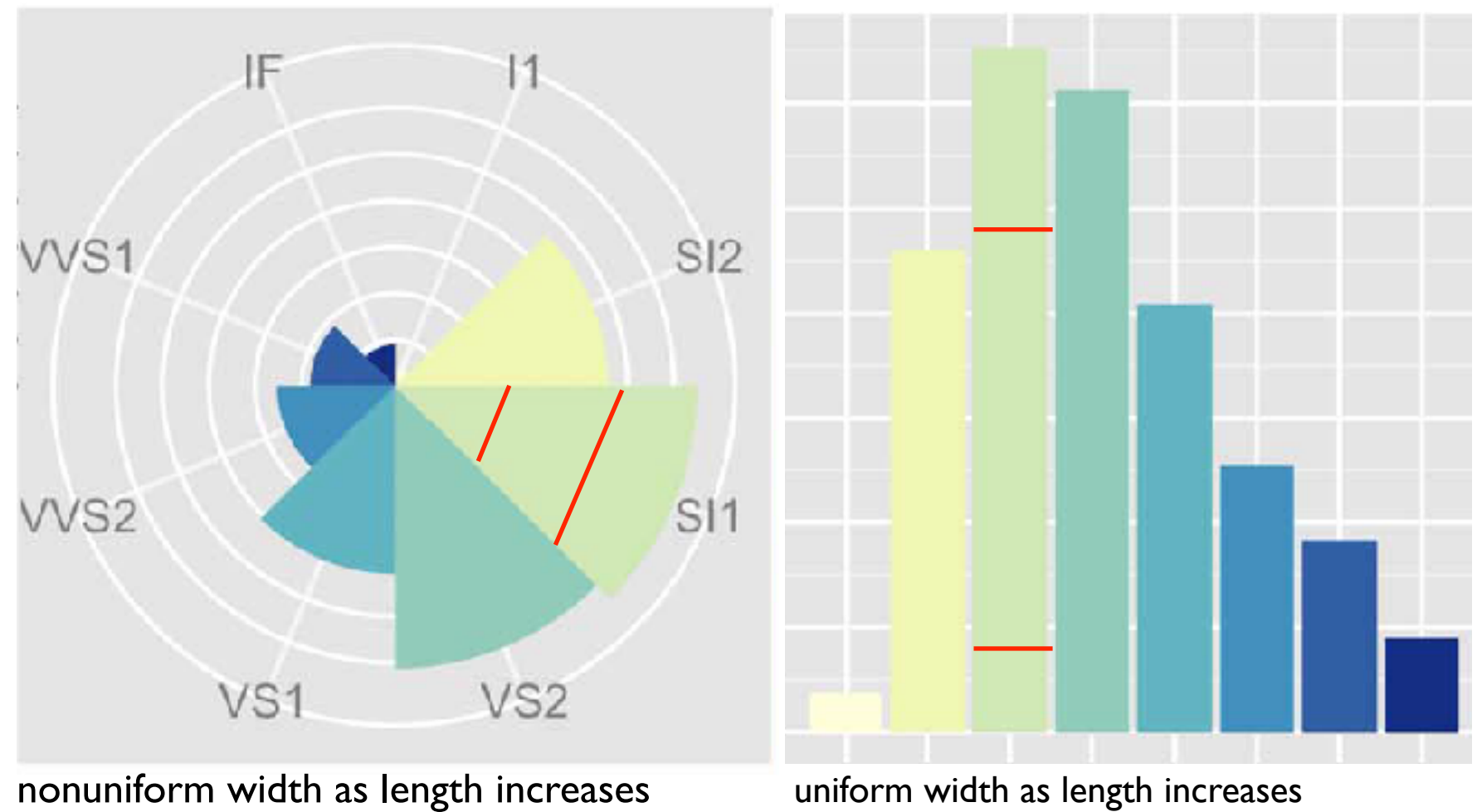
# Coxcomb / nightingale rose / polar area chart

- invented by Florence Nightingale:  
Diagram of the Causes of Mortality in the Army in the East



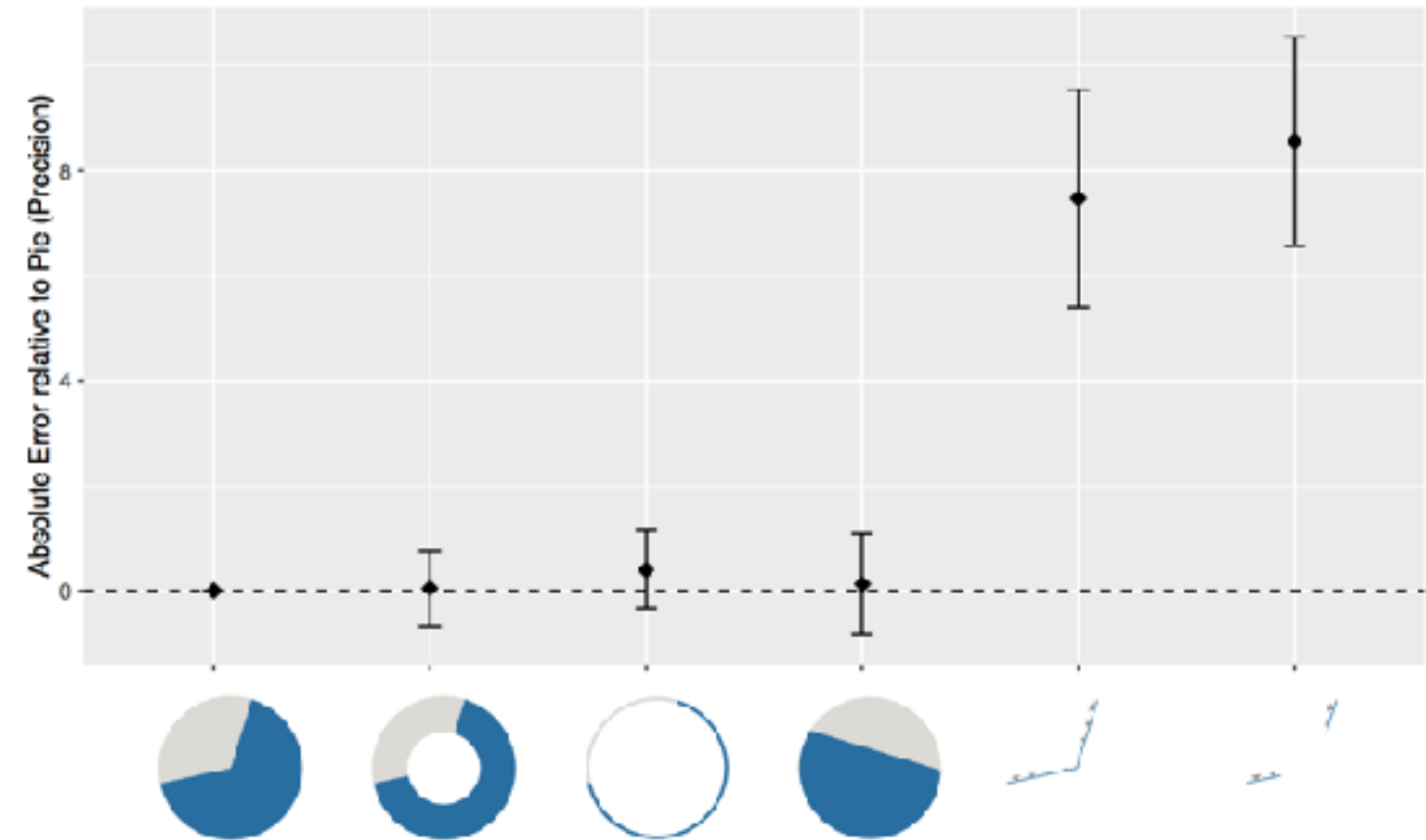
# Coxcomb: perception

- encode: **ID length**
- decode/perceive: **2D area**
- nonuniform line/sector width as length increases
  - so area variation is nonlinear wrt line mark length!
- bar chart safer: uniform width, so area is linear with line mark length
  - **both radial & rectilinear cases**



# Pie charts: perception

- some empirical evidence that people respond to arc length
  - decode/perceive: not angles
  - maybe also areas?...
- donut charts no worse than pie charts



[Arcs, Angles, or Areas: Individual Data Encodings in Pie and Donut Charts.  
Skau and Kosara. Proc. EuroVis 2016.]

<https://eagereyes.org/blog/2016/an-illustrated-tour-of-the-pie-chart-study-results>



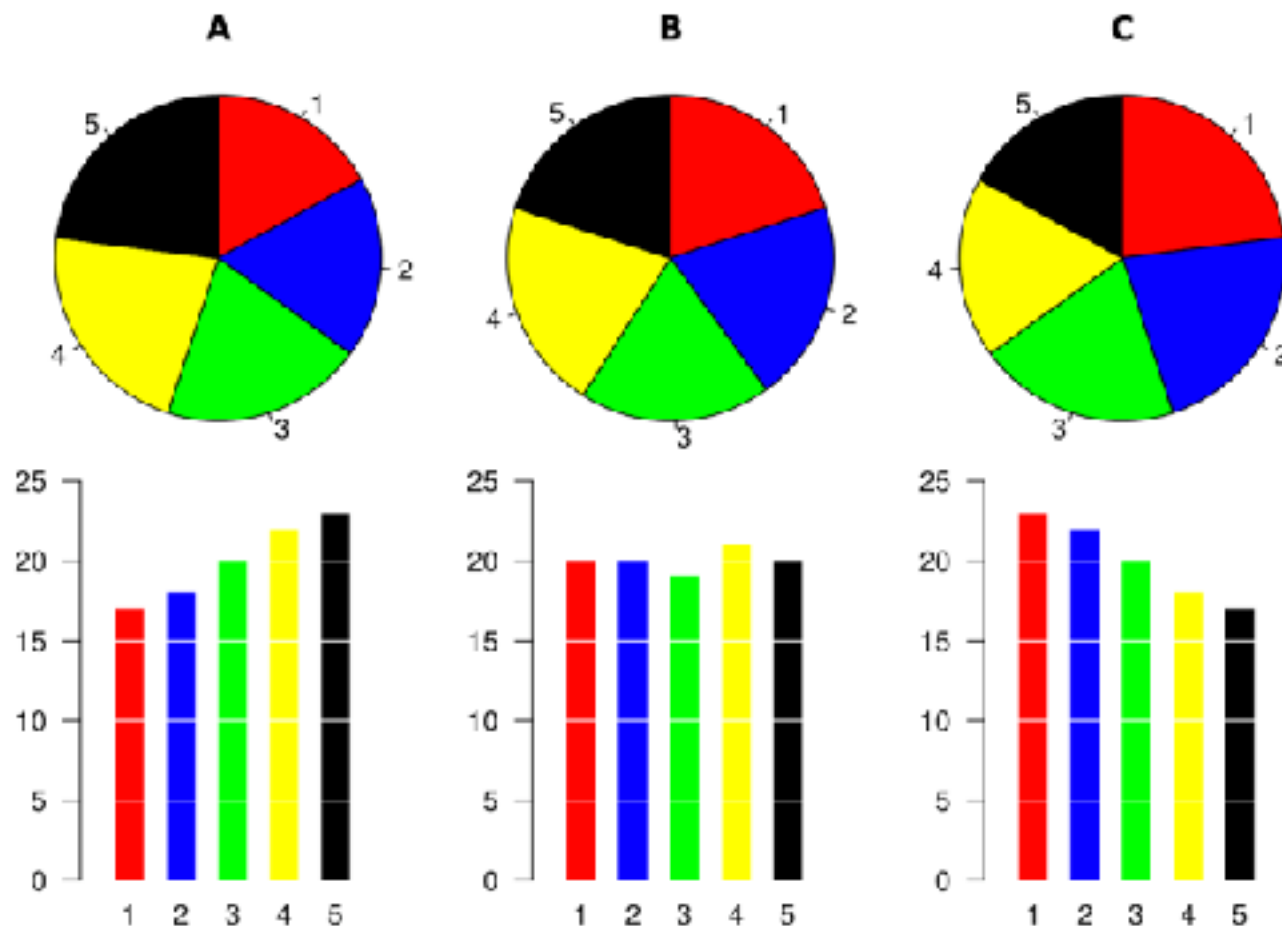
# Pie charts: best practices

- not so bad for two (or few) levels, for part-to-whole task



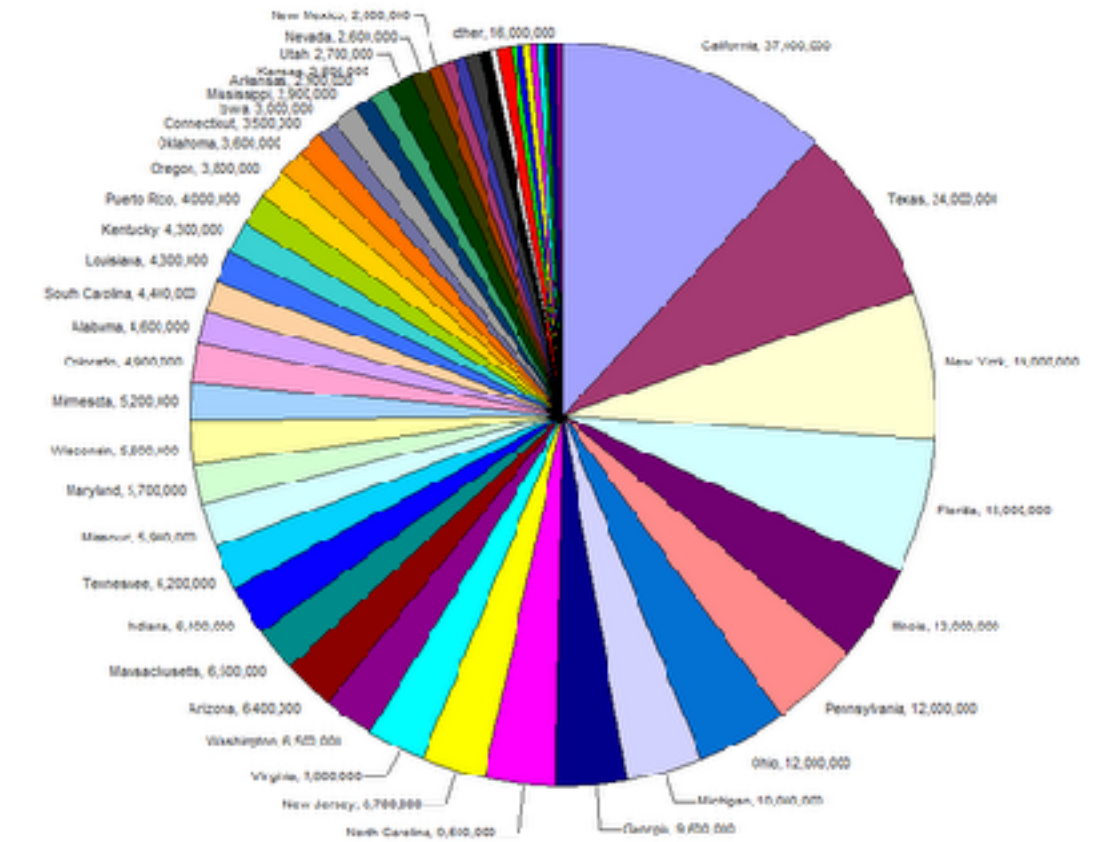
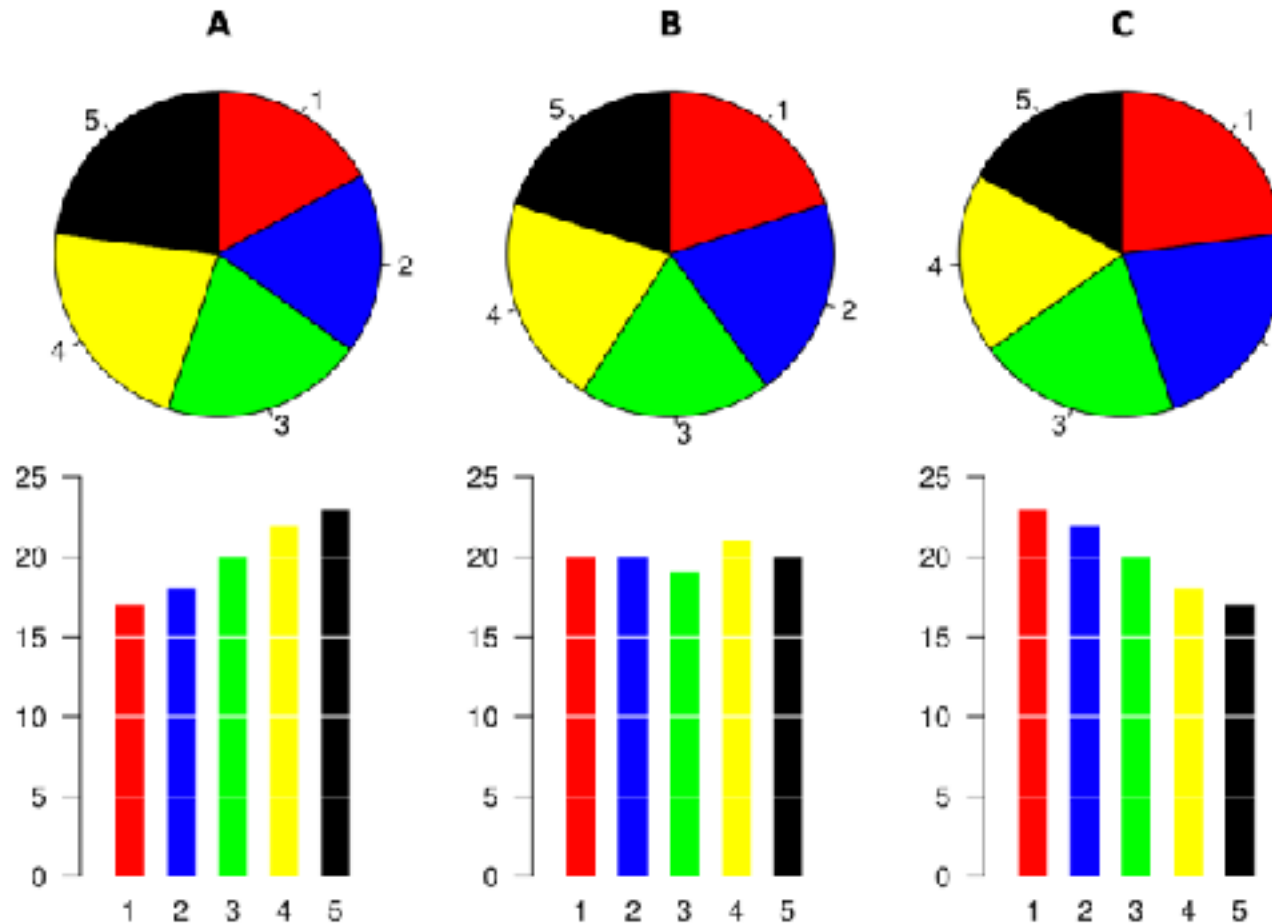
# Pie charts: best practices

- not so bad for two (or few) levels, for part-to-whole task
- dubious for several levels if details matter



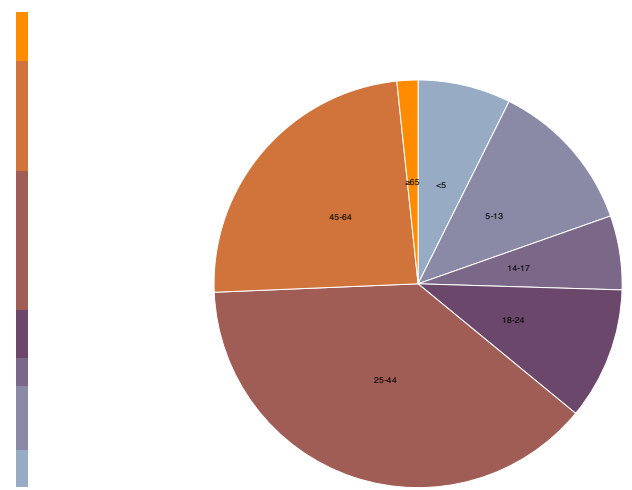
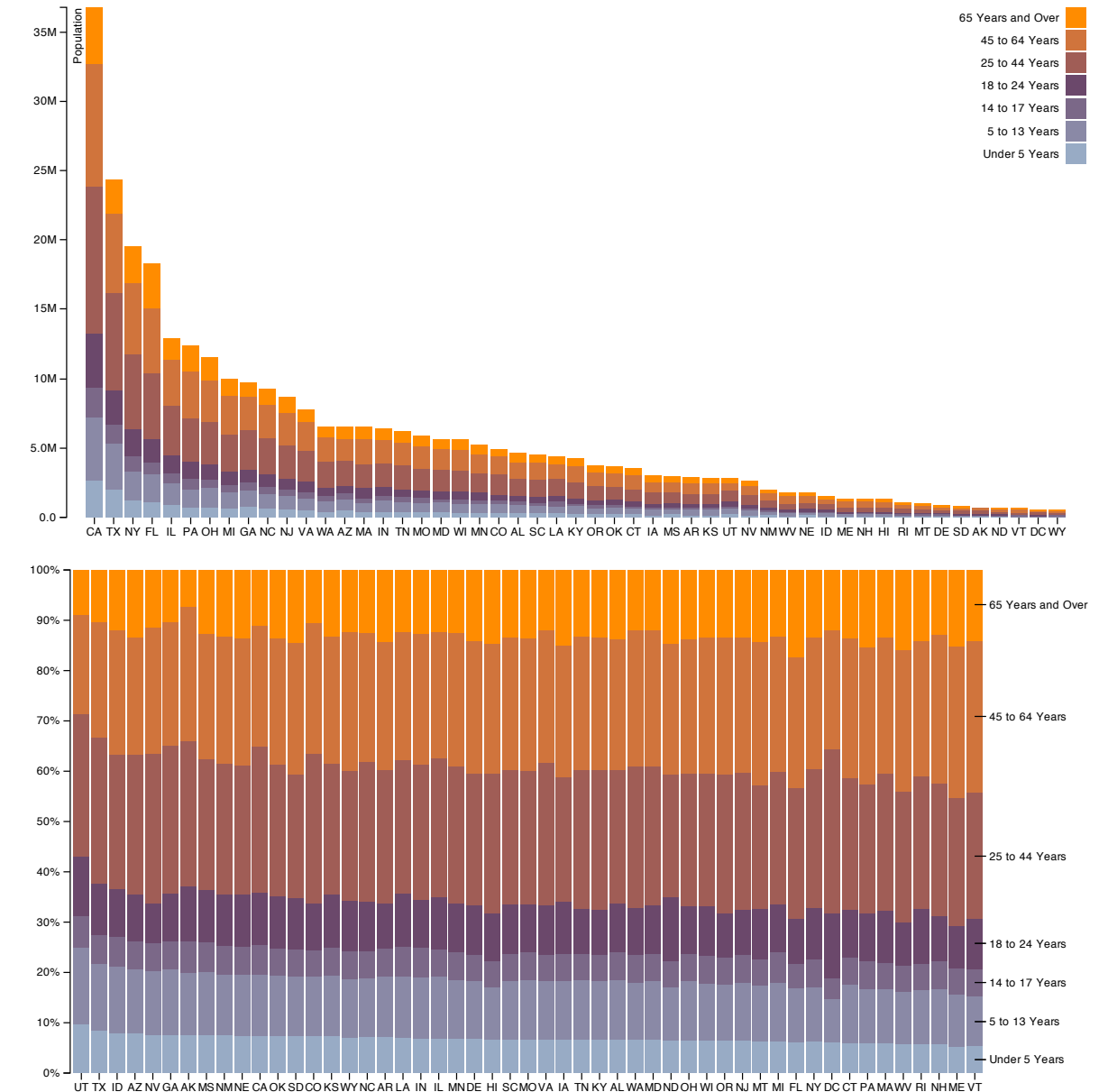
# Pie charts: best practices

- not so bad for two (or few) levels, for part-to-whole task
- dubious for several levels if details matter
- terrible for many levels



# Idioms: **normalized stacked bar chart**

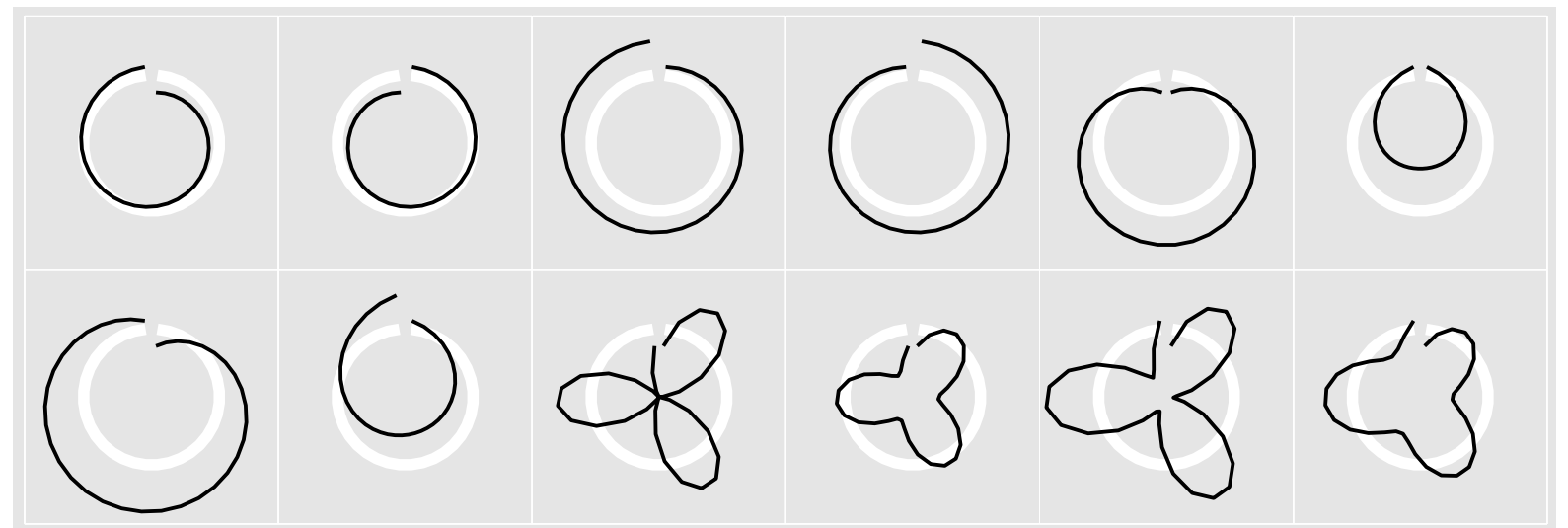
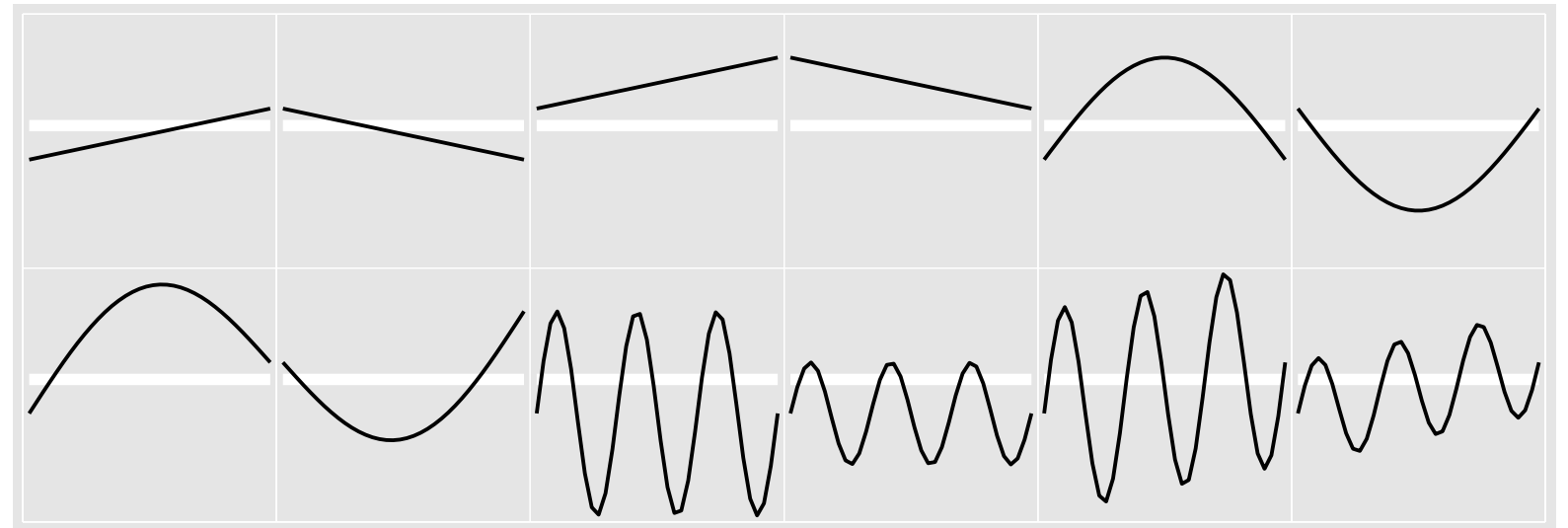
- task
  - part-to-whole judgements
- **normalized stacked bar chart**
  - stacked bar chart, normalized to full vert height
  - single stacked bar equivalent to full pie
    - high information density: requires narrow rectangle
- **pie chart**
  - information density: requires large circle



<http://bl.ocks.org/mbostock/3886208>,  
<http://bl.ocks.org/mbostock/3887235>,  
<http://bl.ocks.org/mbostock/3886394>.

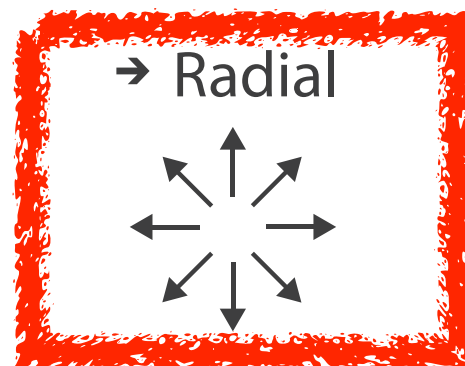
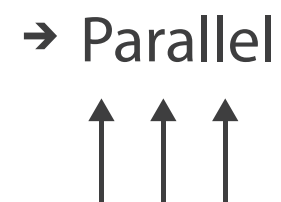
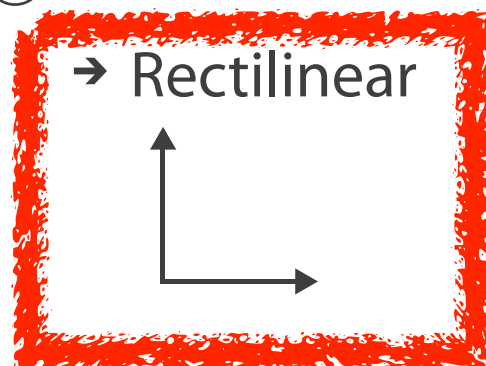
# Idiom: **glyphmaps**

- rectilinear good for linear vs nonlinear trends
- radial good for cyclic patterns
  - evaluating periodicity

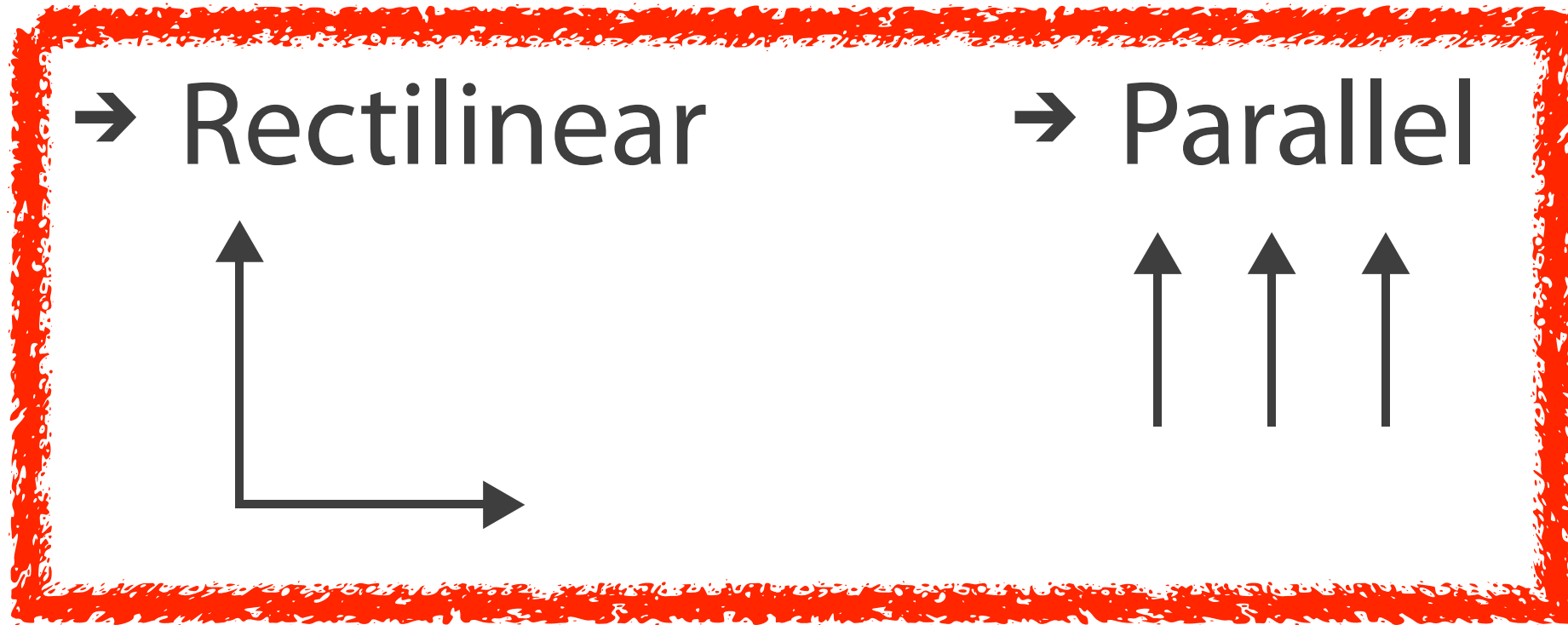


[Glyph-maps for Visually Exploring Temporal Patterns in Climate Data and Models. Wickham, Hofmann, Wickham, and Cook. *Environmetrics* 23:5 (2012), 382–393.]

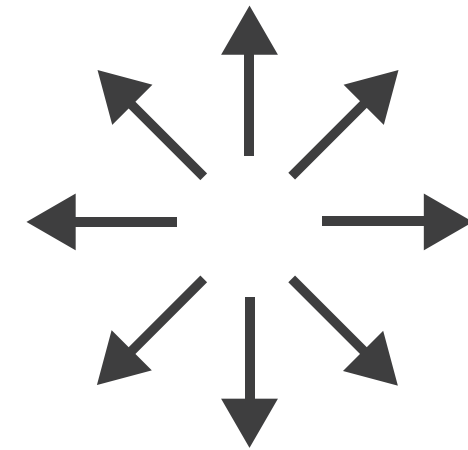
## ➔ Axis Orientation



# → Axis Orientation

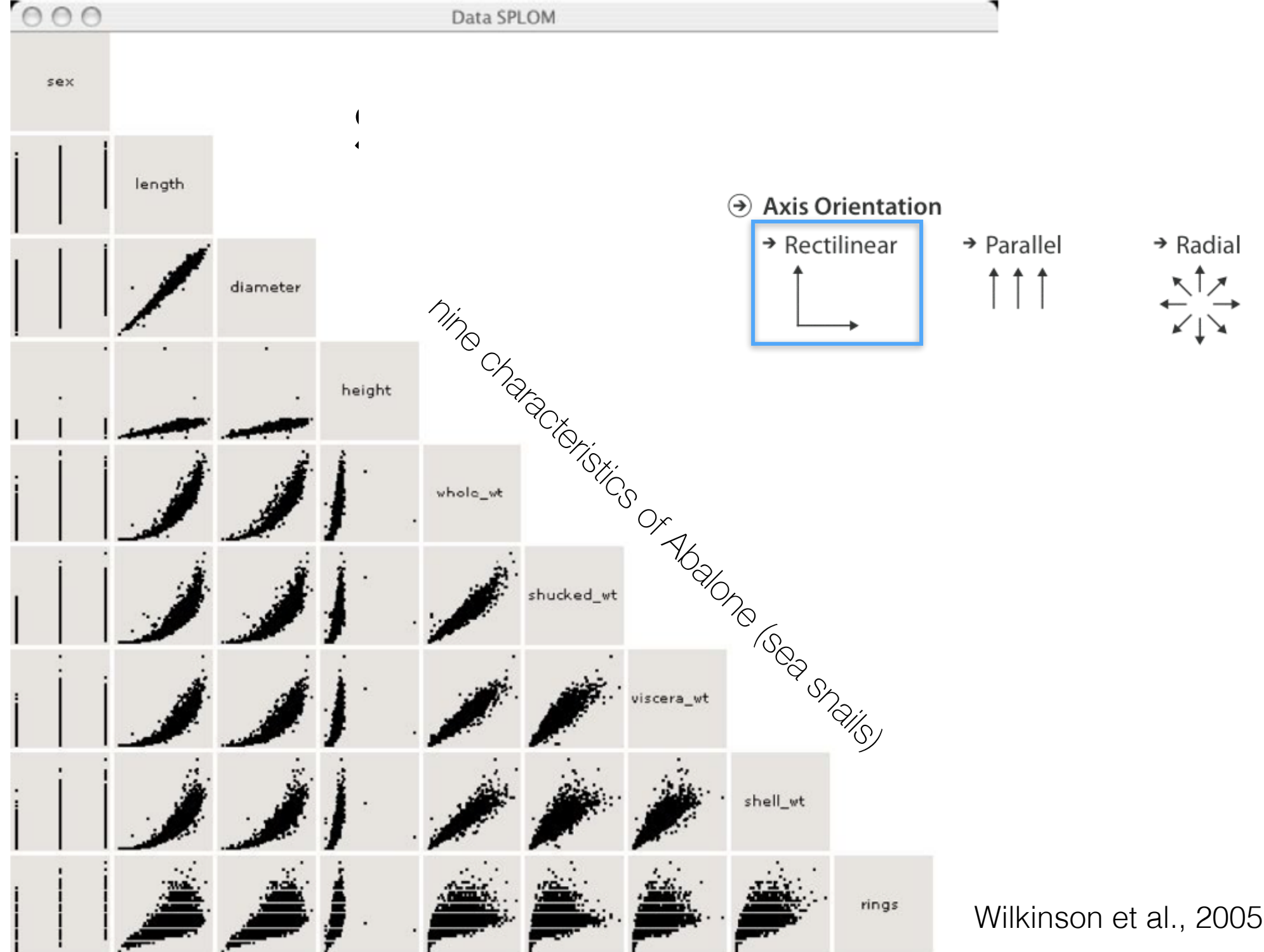


→ Radial



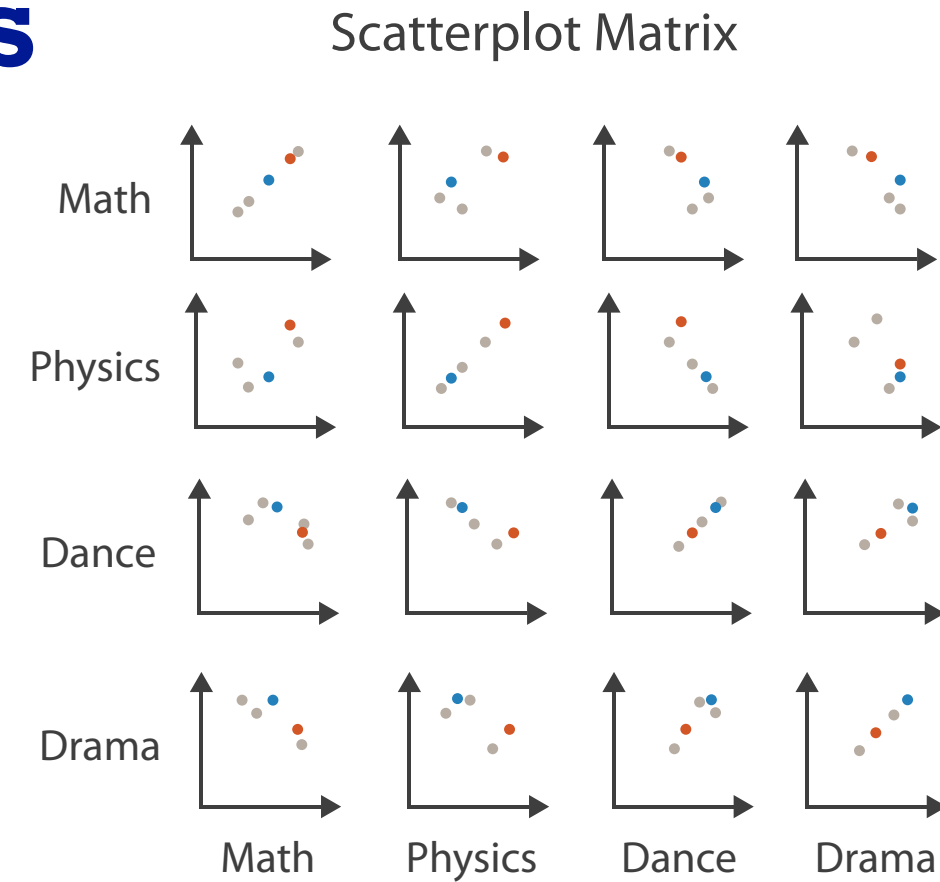
# Idiom: **SPL**OM

- scatterplot matrix (SPL**OM**)
  - rectilinear axes, point mark
  - all possible pairs of axes
  - scalability
    - one dozen attribs
    - dozens to hundreds of items



# Idioms: parallel coordinates

- scatterplot limitation
  - visual representation with orthogonal axes
  - can show only two attributes with spatial position channel



Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

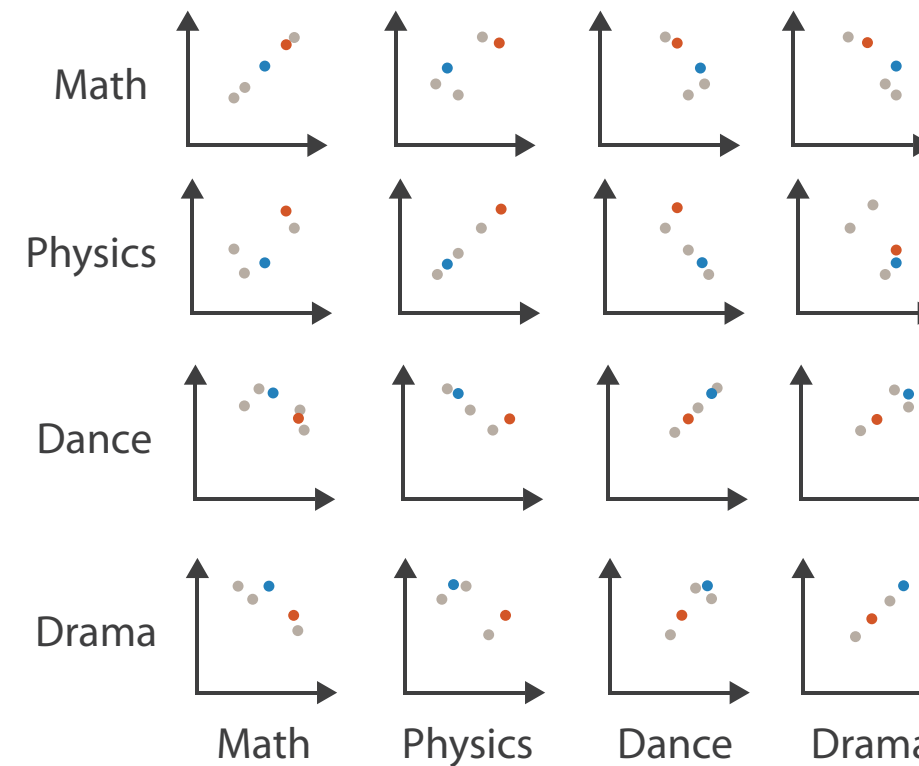
after [Visualization Course Figures. McGuffin, 2014.  
<http://www.michaelmcguffin.com/courses/vis/>]



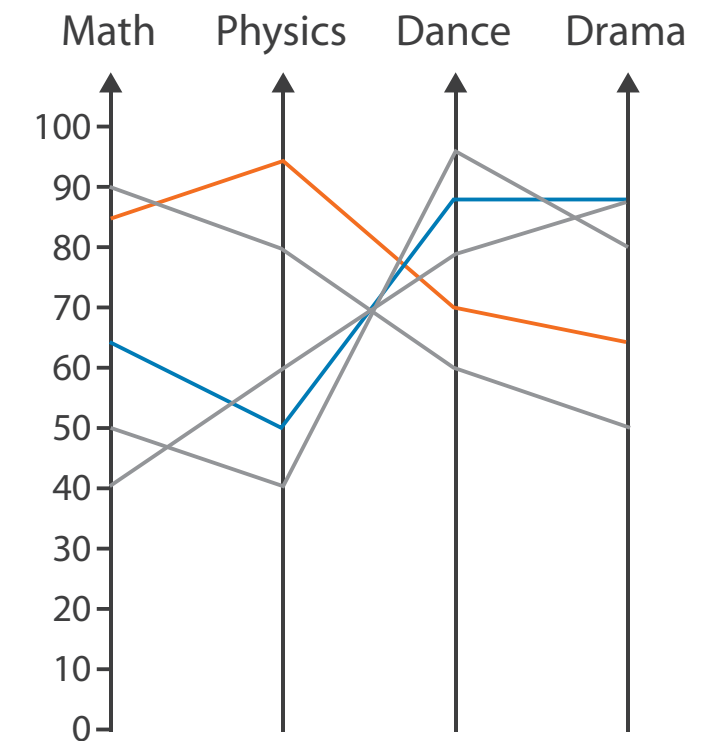
# Idioms: parallel coordinates

- scatterplot limitation
  - visual representation with orthogonal axes
  - can show only two attributes with spatial position channel
- alternative: line up axes in parallel to show many attributes with position
  - item encoded with a line with n segments
  - n is the number of attributes shown
- parallel coordinates
  - parallel axes, jagged line for item
  - rectilinear axes, item as point
    - axis ordering is major challenge
  - scalability
    - dozens of attribs
    - hundreds of items

Scatterplot Matrix



Parallel Coordinates



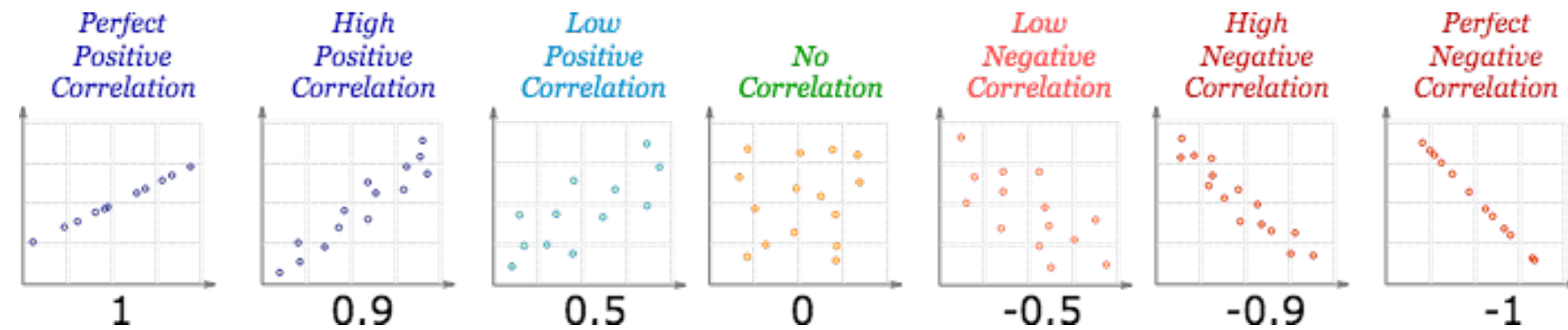
Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

after [Visualization Course Figures. McGuffin, 2014.  
<http://www.michaelmcguffin.com/courses/vis/>]

# Task: Correlation

- scatterplot matrix
  - positive correlation
    - diagonal low-to-high
  - negative correlation
    - diagonal high-to-low
  - uncorrelated: spread out
- parallel coordinates
  - positive correlation
    - parallel line segments
  - negative correlation
    - all segments cross at halfway point
  - uncorrelated
    - scattered crossings



<https://www.mathsisfun.com/data/scatter-xy-plots.html>

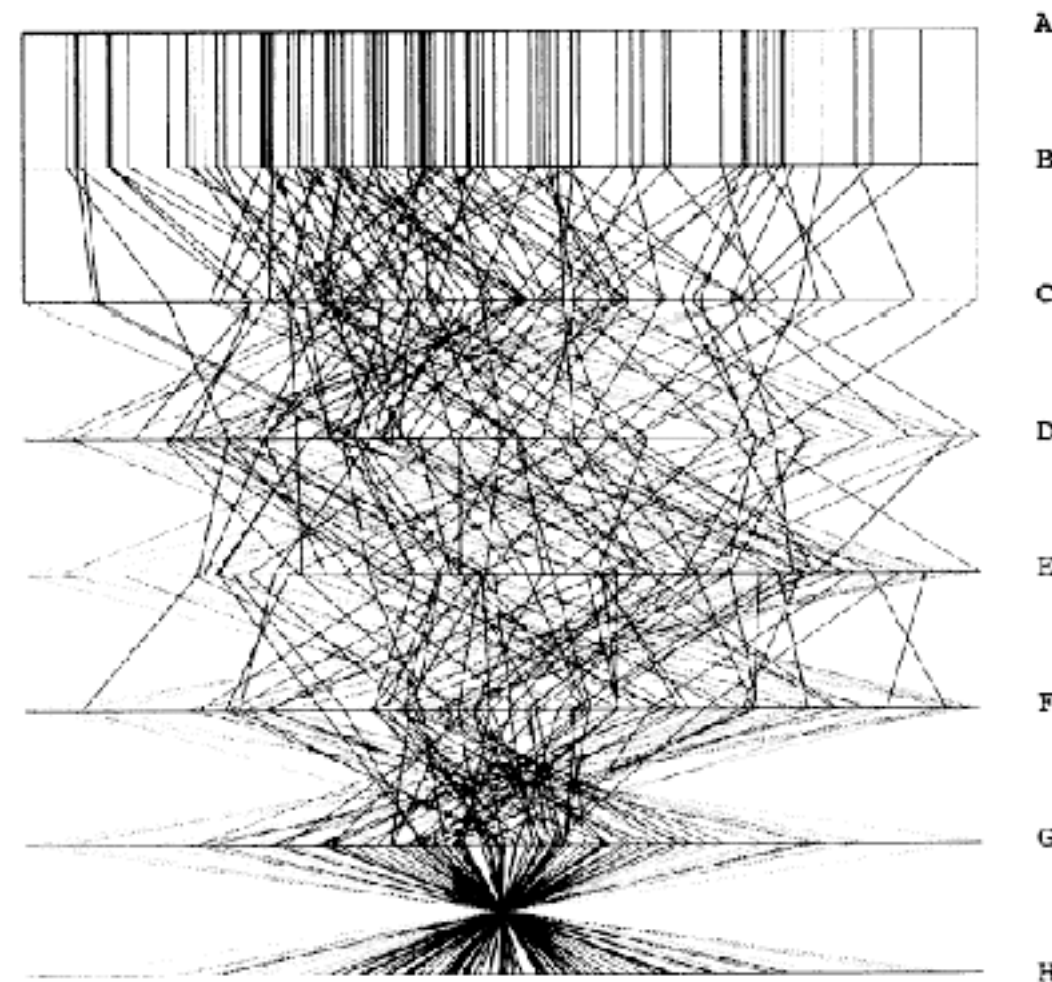
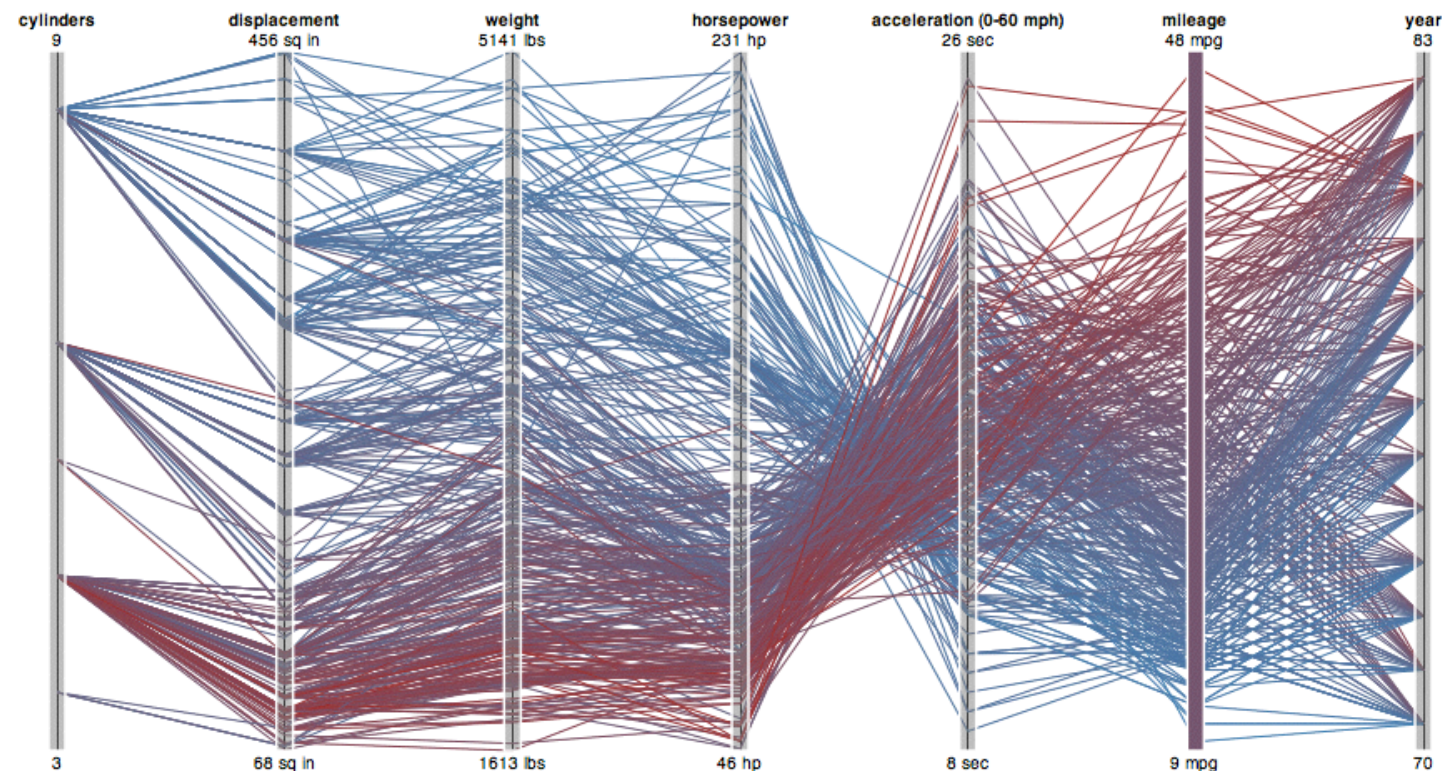


Figure 3. Parallel Coordinate Plot of Six-Dimensional Data Illustrating Correlations of  $\rho = 1, .8, .2, 0, -.2, -.8, \text{ and } -1$ .

[Hyperdimensional Data Analysis Using Parallel Coordinates. Wegman. *Journ. American Statistical Association* 85:411 (1990), 664–675.]

# Parallel coordinates, limitations

- visible patterns only between neighboring axis pairs
- how to pick axis order?
  - usual solution: reorderable axes, interactive exploration
  - same weakness as many other techniques
    - downside of interaction: human-powered search
  - some algorithms proposed, none fully solve

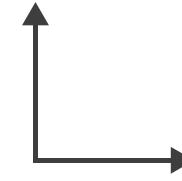


# Orientation limitations

- rectilinear: scalability wrt #axes
  - 2 axes best, 3 problematic, 4+ impossible

⊕ Axis Orientation

→ Rectilinear

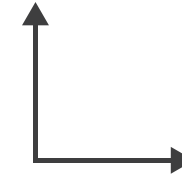


# Orientation limitations

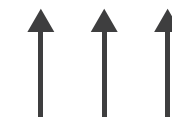
- **rectilinear: scalability wrt #axes**
  - 2 axes best, 3 problematic, 4+ impossible
- **parallel: unfamiliarity, training time**

## ⊕ Axis Orientation

→ Rectilinear



→ Parallel

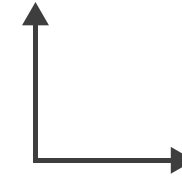


# Orientation limitations

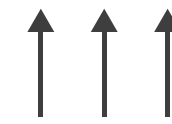
- **rectilinear: scalability wrt #axes**
  - 2 axes best, 3 problematic, 4+ impossible
- **parallel: unfamiliarity, training time**
- **radial: perceptual limits**
  - polar coordinate asymmetry
    - angles lower precision than length
    - nonuniform sector width/size depending on radial distance
  - frequently problematic
    - but sometimes can be deliberately exploited!
      - for 2 attribs of very unequal importance

## ➔ Axis Orientation

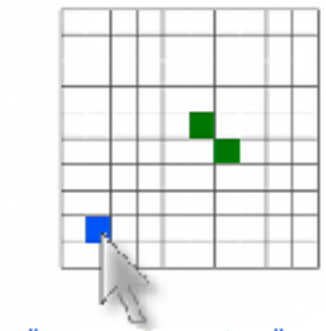
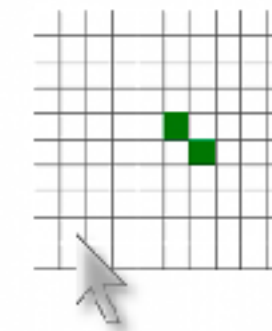
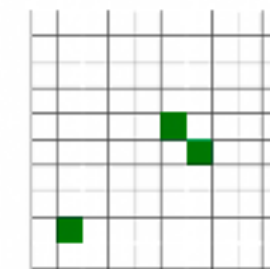
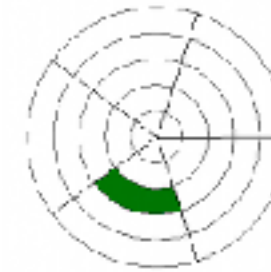
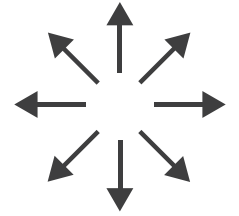
➔ Rectilinear



➔ Parallel



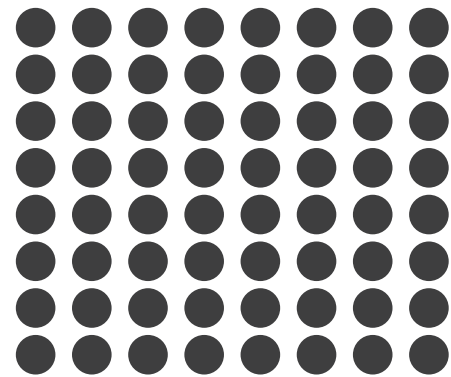
➔ Radial



# Layout density

## ① Layout Density

→ Dense



→ Space-Filling

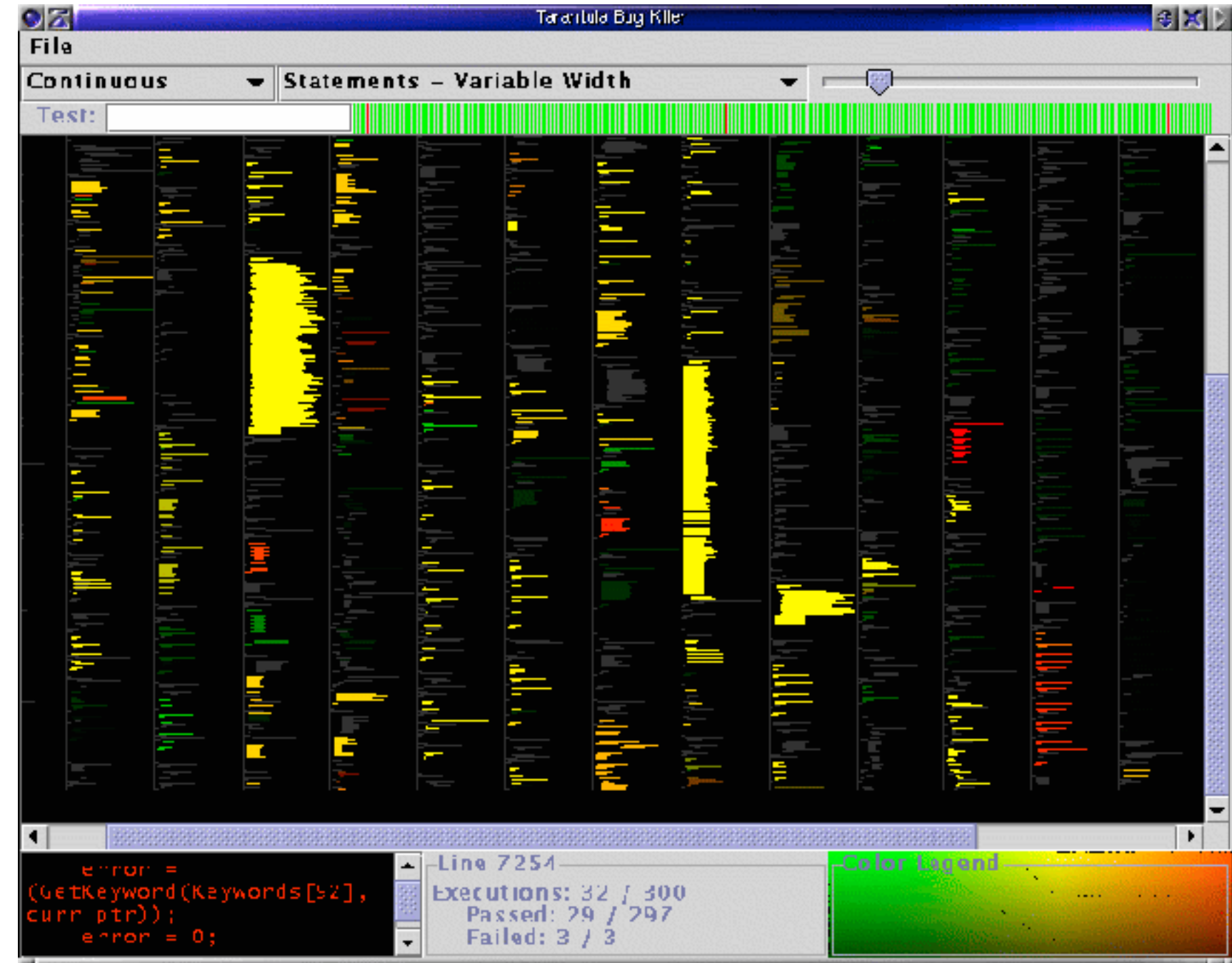
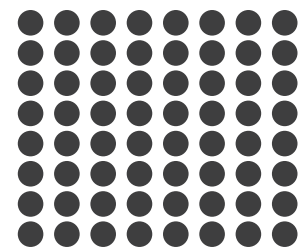


# Idiom: Dense software overviews

- data: text
  - text + 1 quant attrib per line
- derived data:
  - one pixel high line
  - length according to original
- color line by attrib
- scalability
  - 10K+ lines

➔ Layout Density

➔ Dense





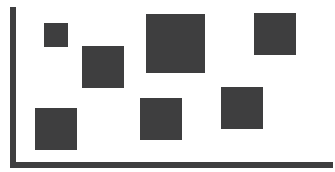
# Arrange tables

## ① Express Values

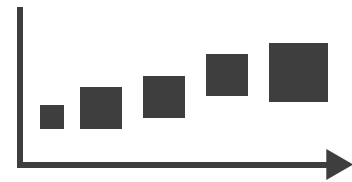


## ② Separate, Order, Align Regions

→ Separate



→ Order



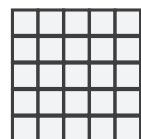
→ Align



→ 1 Key  
*List*

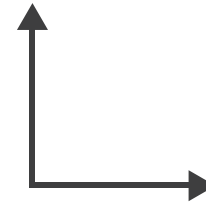


→ 2 Keys  
*Matrix*

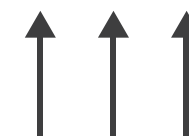


## ③ Axis Orientation

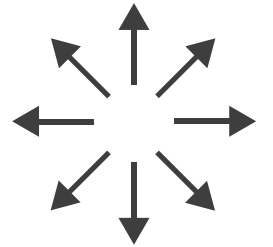
→ Rectilinear



→ Parallel

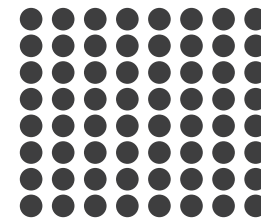


→ Radial



## ④ Layout Density

→ Dense



# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



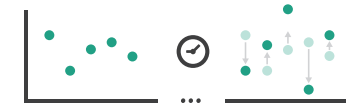
→ Motion

*Direction, Rate, Frequency, ...*

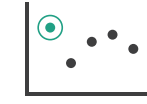


## Manipulate

### → Change



### → Select



### → Navigate

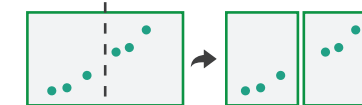


## Facet

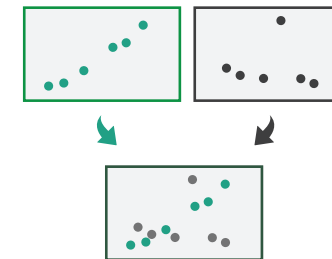
### → Juxtapose



### → Partition



### → Superimpose

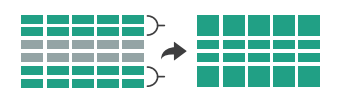


## Reduce

### → Filter



### → Aggregate



### → Embed



What?

Why?

How?

# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



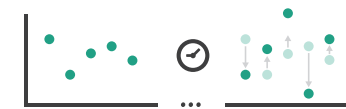
→ Motion

*Direction, Rate, Frequency, ...*

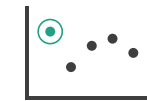


## Manipulate

### → Change



### → Select



### → Navigate

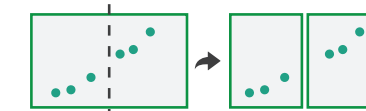


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

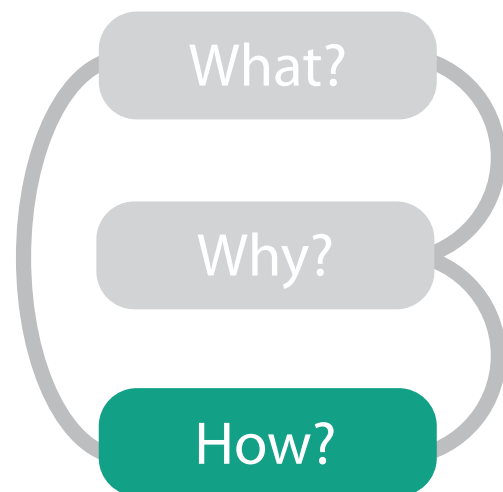
### → Filter



### → Aggregate



### → Embed



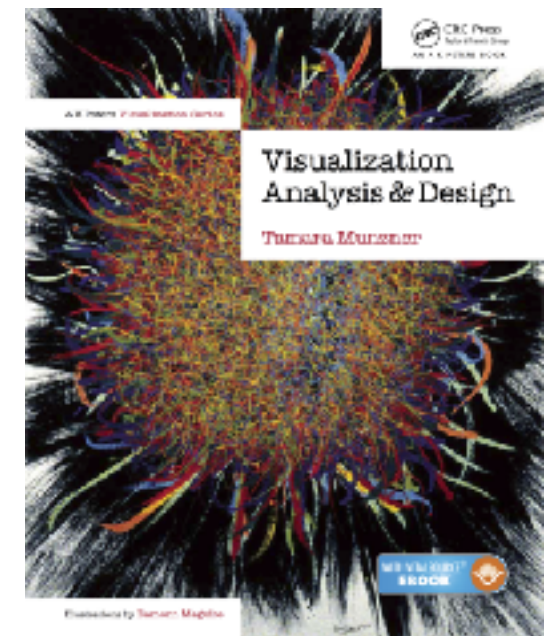
# Visualization Analysis & Design

## *Network Data (Ch 9)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# Network data

- networks

- model relationships between things

- aka graphs

- two kinds of items, both can have attributes

- nodes

- links

- tree

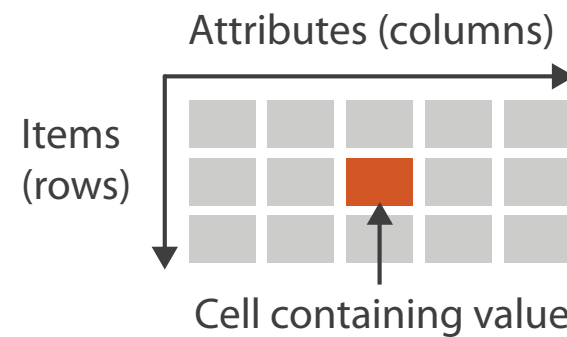
- special case

- no cycles

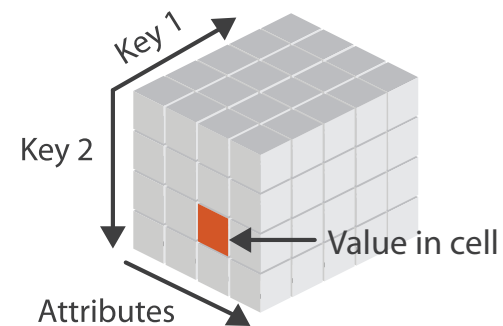
- one parent per node

## ➔ Dataset Types

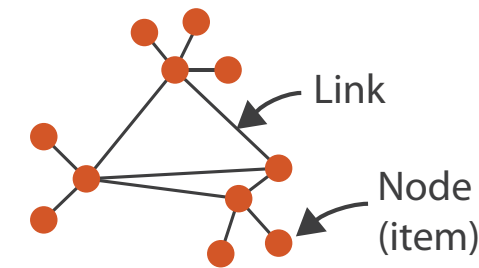
➔ Tables



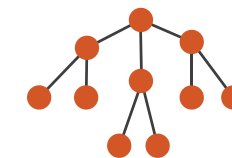
➔ *Multidimensional Table*



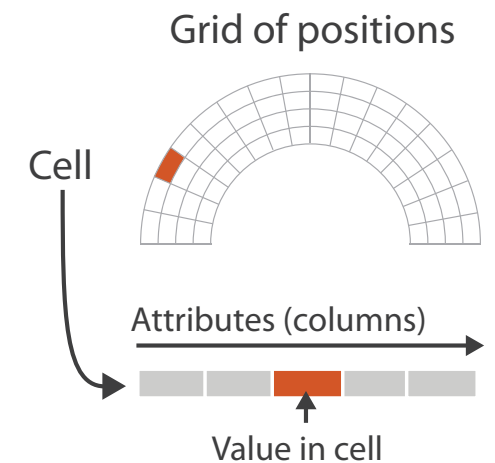
➔ Networks



➔ Trees

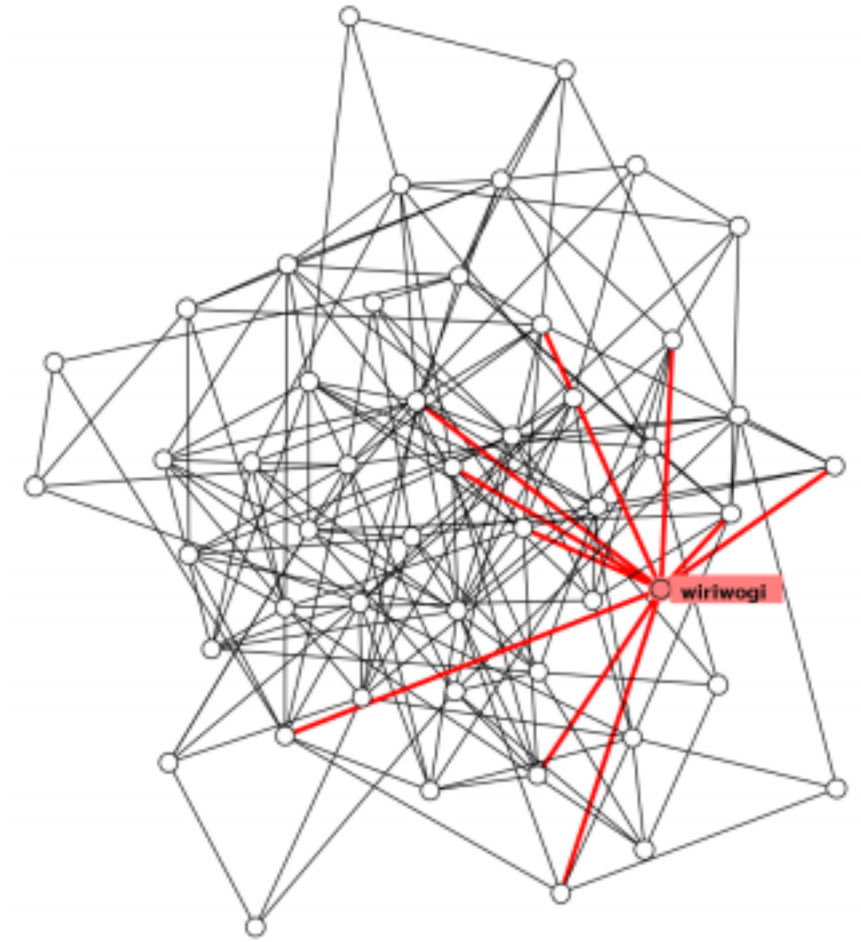


➔ Spatial  
➔ Fields (Continuous)



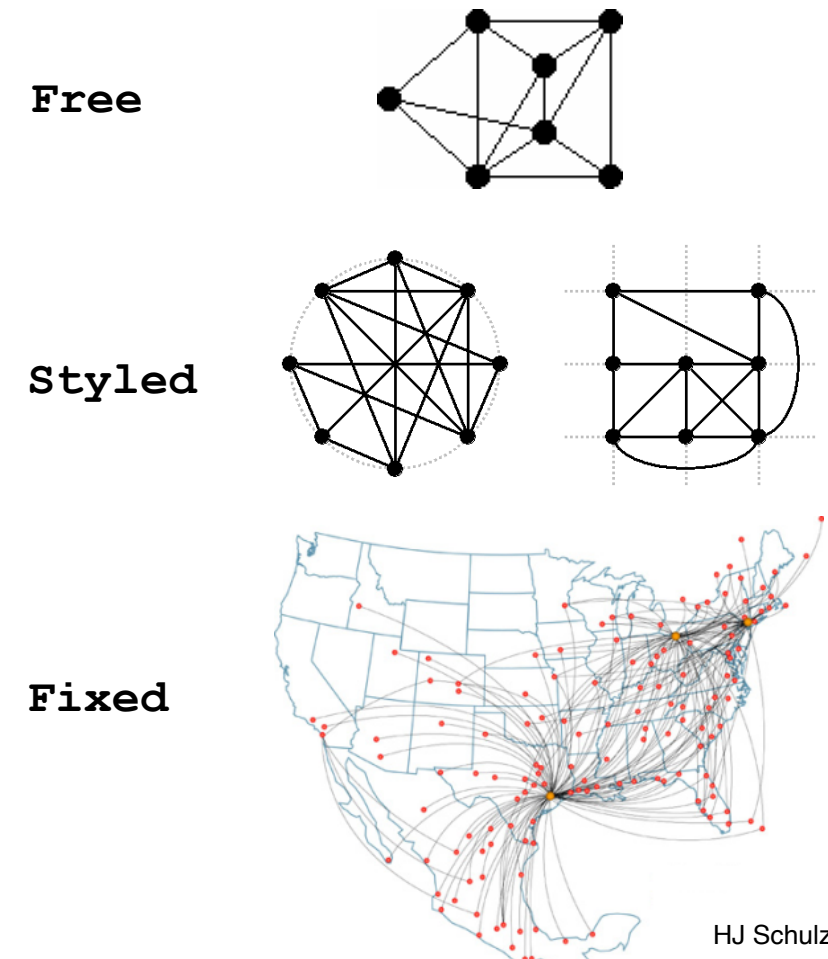
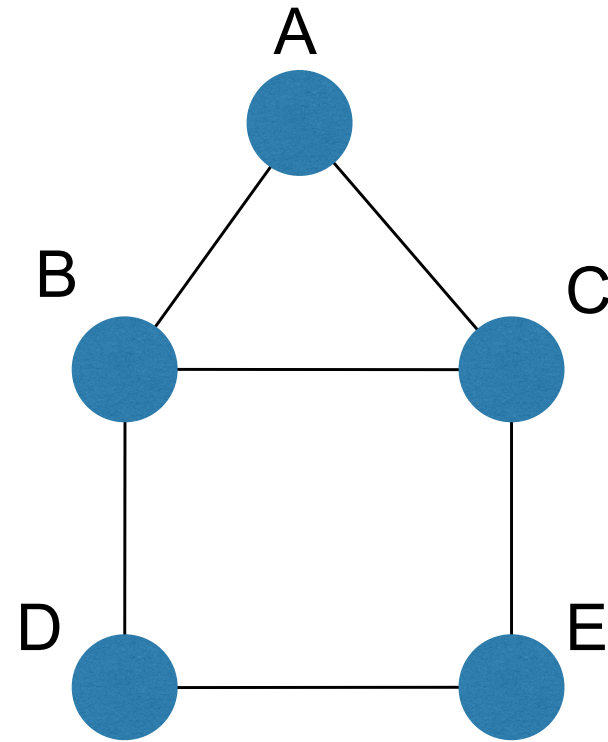
# Network tasks: topology-based and attribute-based

- topology based tasks
  - find paths
  - find (topological) neighbors
  - compare centrality/importance measures
  - identify clusters / communities
- attribute based tasks (similar to table data)
  - find distributions, ...
- combination tasks, incorporating both
  - example: find friends-of-friends who like cats
    - topology: find all adjacent nodes of given node
    - attributes: check if has-pet (node attribute) == cat



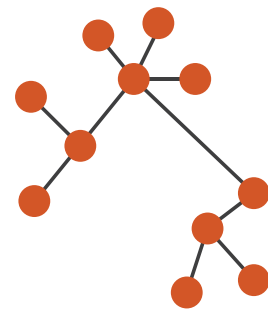
# Node-link diagrams

- nodes: point marks
- links: line marks
  - straight lines or arcs
  - connections between nodes
- intuitive & familiar
  - most common
  - many, many variants



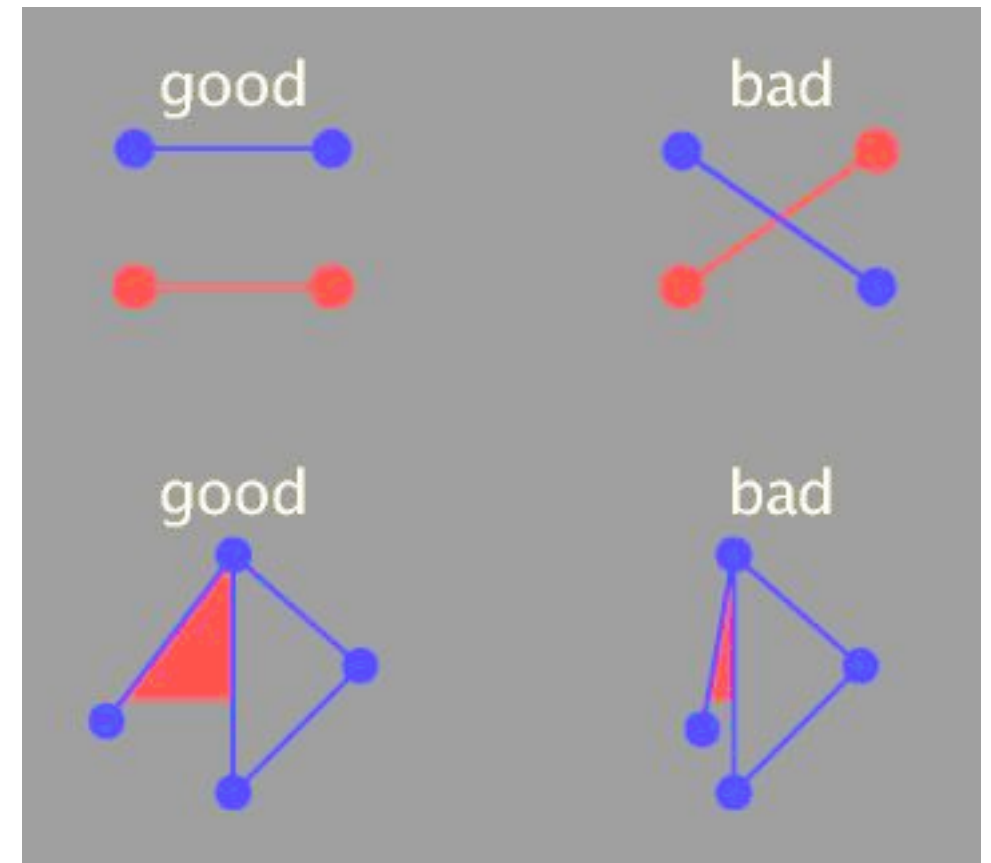
## ➔ Node-Link Diagrams Connection Marks

✓ NETWORKS    ✓ TREES



# Criteria for good node-link layouts

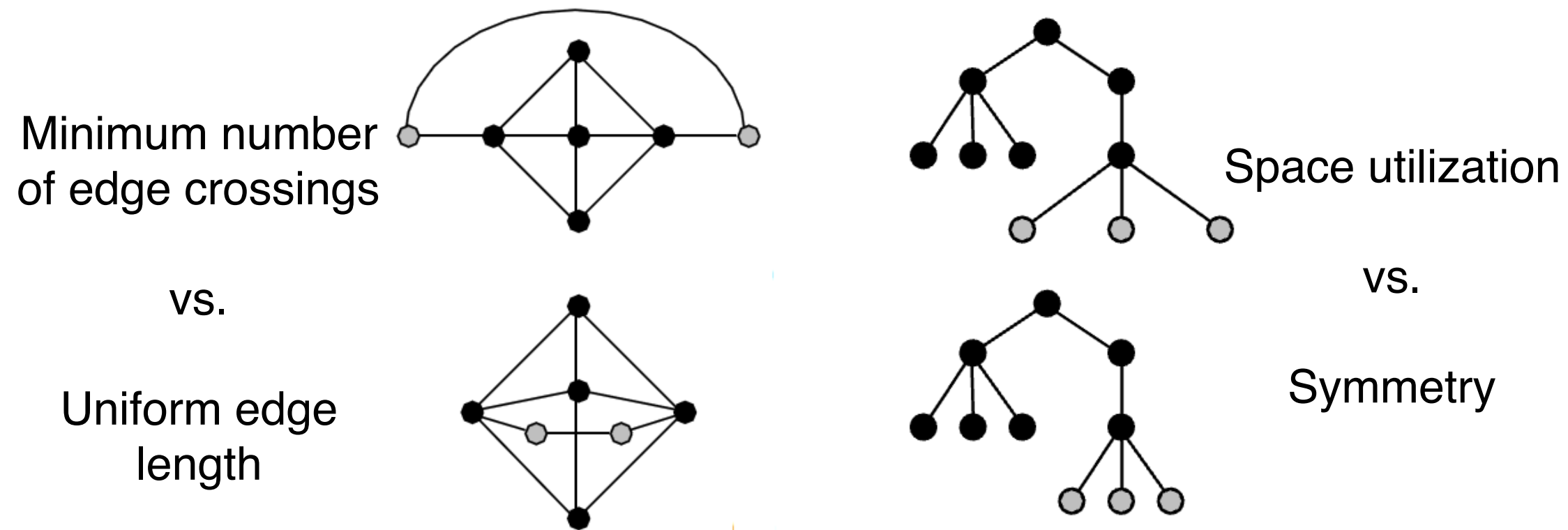
- minimize
  - edge crossings, node overlaps
  - distances between topological neighbor nodes
  - total drawing area
  - edge bends
- maximize
  - angular distance between different edges
  - aspect ratio disparities
- emphasize symmetry
  - similar graph structures should look similar in layout





# Criteria conflict

- most criteria NP-hard individually
- many criteria directly conflict with each other



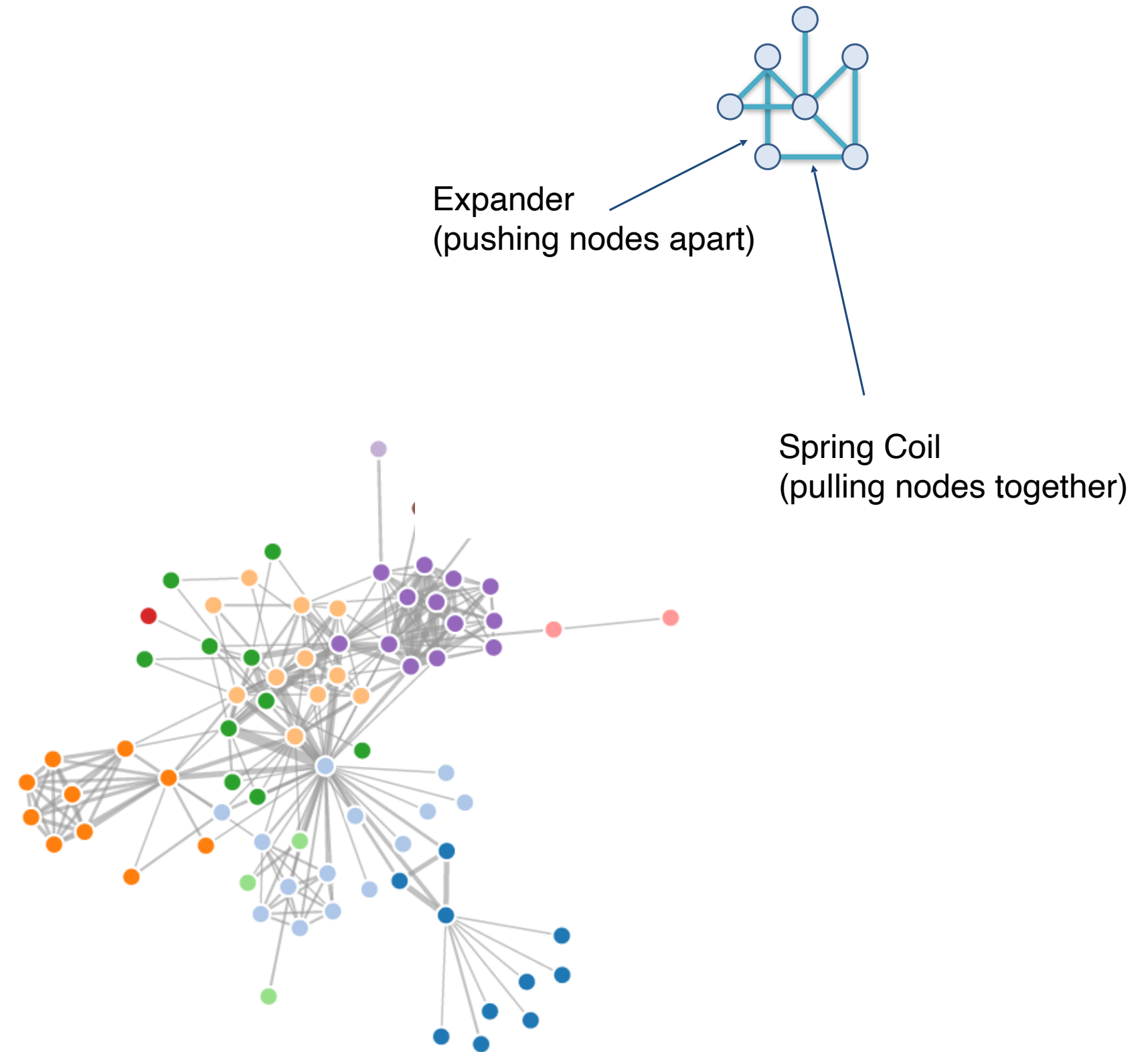
Schulz 2004

# Optimization-based layouts

- formulate layout problem as optimization problem
- convert criteria into weighted cost function
  - $F(\text{layout}) = a * [\text{crossing counts}] + b * [\text{drawing space used}] + \dots$
- use known optimization techniques to find layout at minimal cost
  - energy-based physics models
  - force-directed placement
  - spring embedders

# Force-directed placement

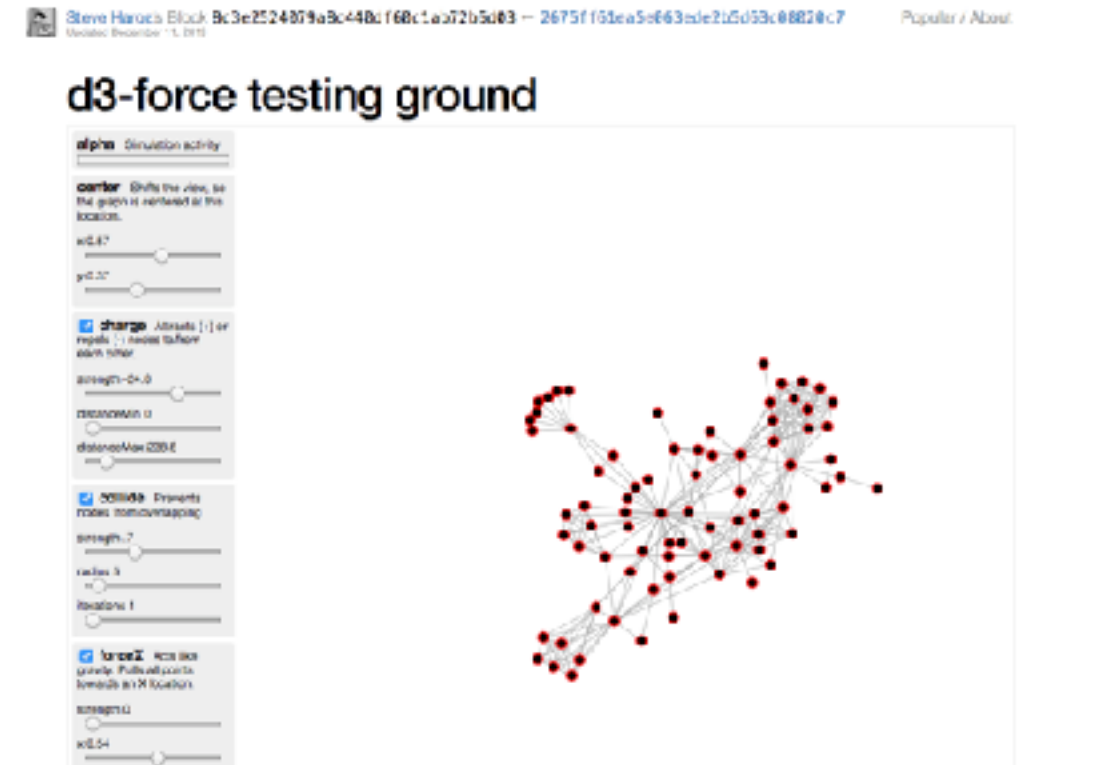
- physics model
  - links = springs pull together
  - nodes = magnets repulse apart
- algorithm
  - place vertices in random locations
  - while not equilibrium
    - calculate force on vertex
      - sum of
        - » pairwise repulsion of all nodes
        - » attraction between connected nodes
    - move vertex by  $c * \text{vertex\_force}$



<http://mbostock.github.com/d3/ex/force.html>

# Force-directed placement properties

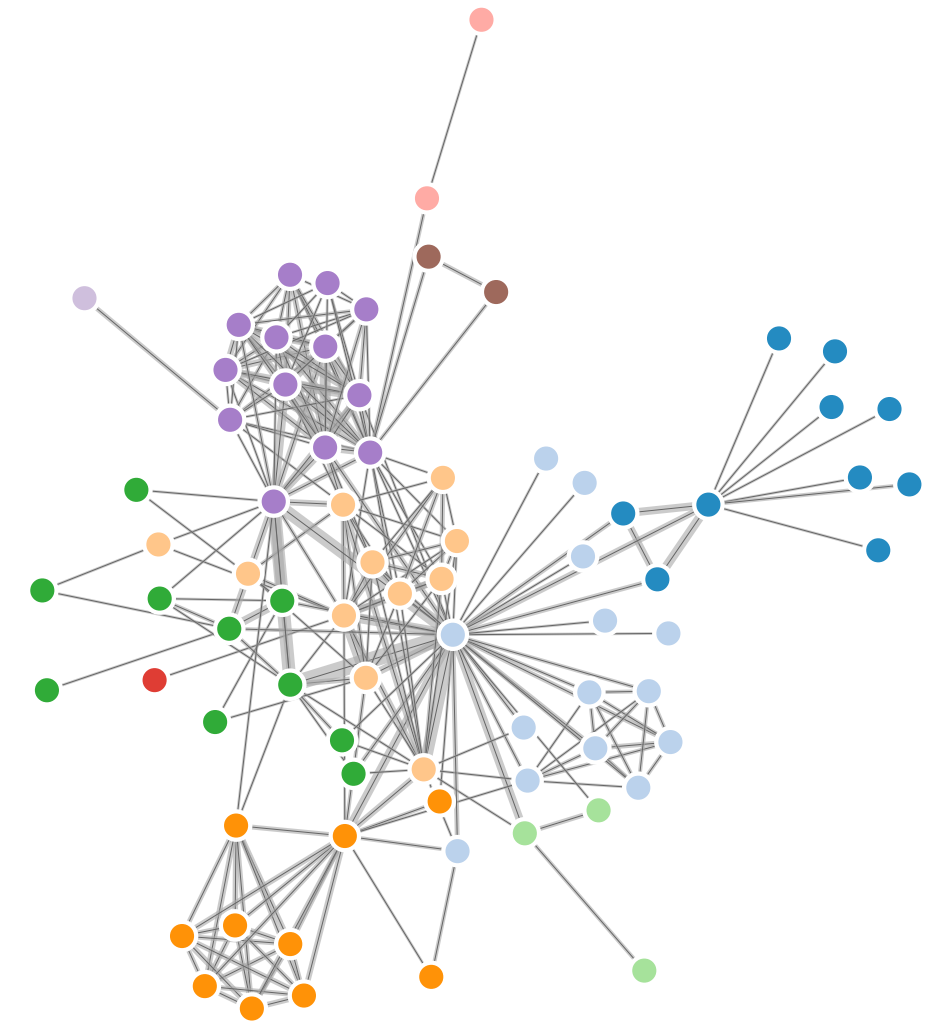
- strengths
  - reasonable layout for small, sparse graphs
  - clusters typically visible
  - edge length uniformity
- weaknesses
  - nondeterministic
  - computationally expensive:  $O(n^3)$  for  $n$  nodes
    - each step is  $n^2$ , takes  $\sim n$  cycles to reach equilibrium
  - naive FD doesn't scale well beyond 1K nodes
  - iterative progress: engaging but distracting



<https://bl.ocks.org/steveharoz/8c3e2524079a8c440df60c1ab72b5d03>

# Idiom: **force-directed placement**

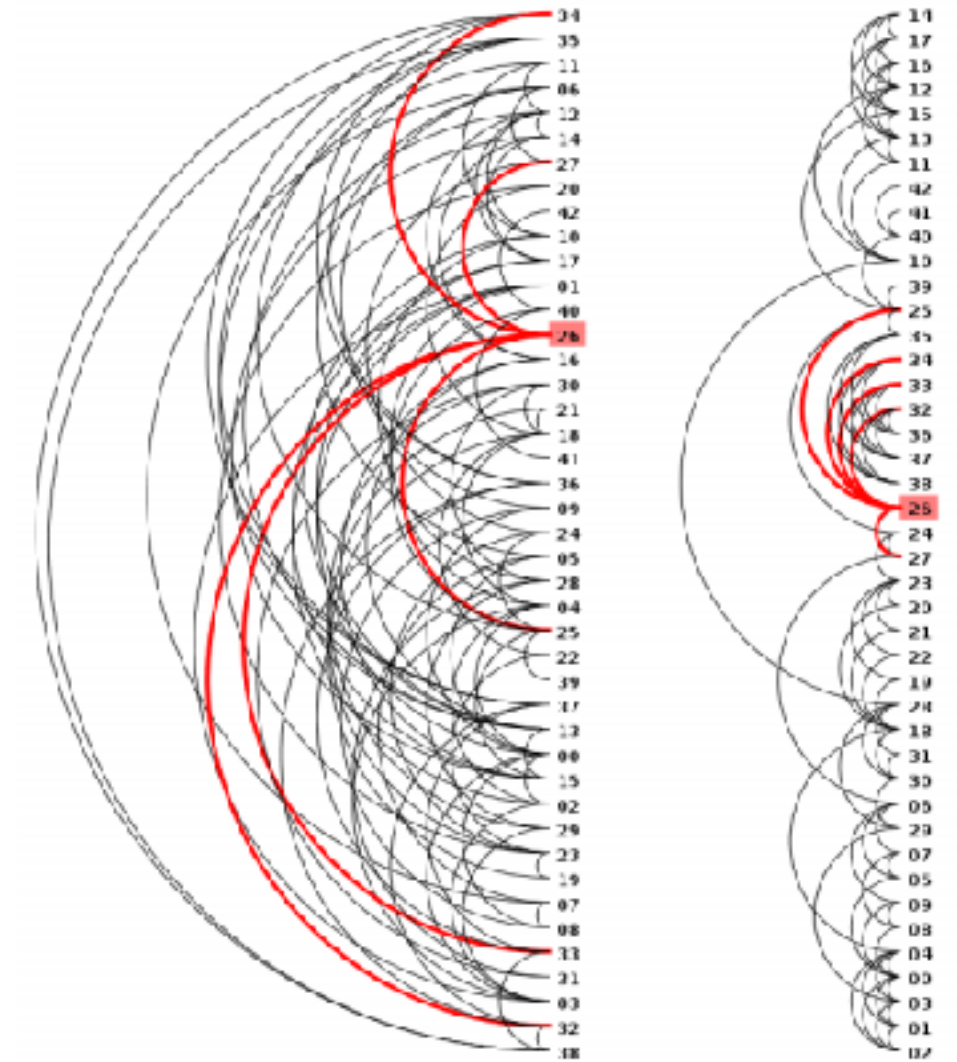
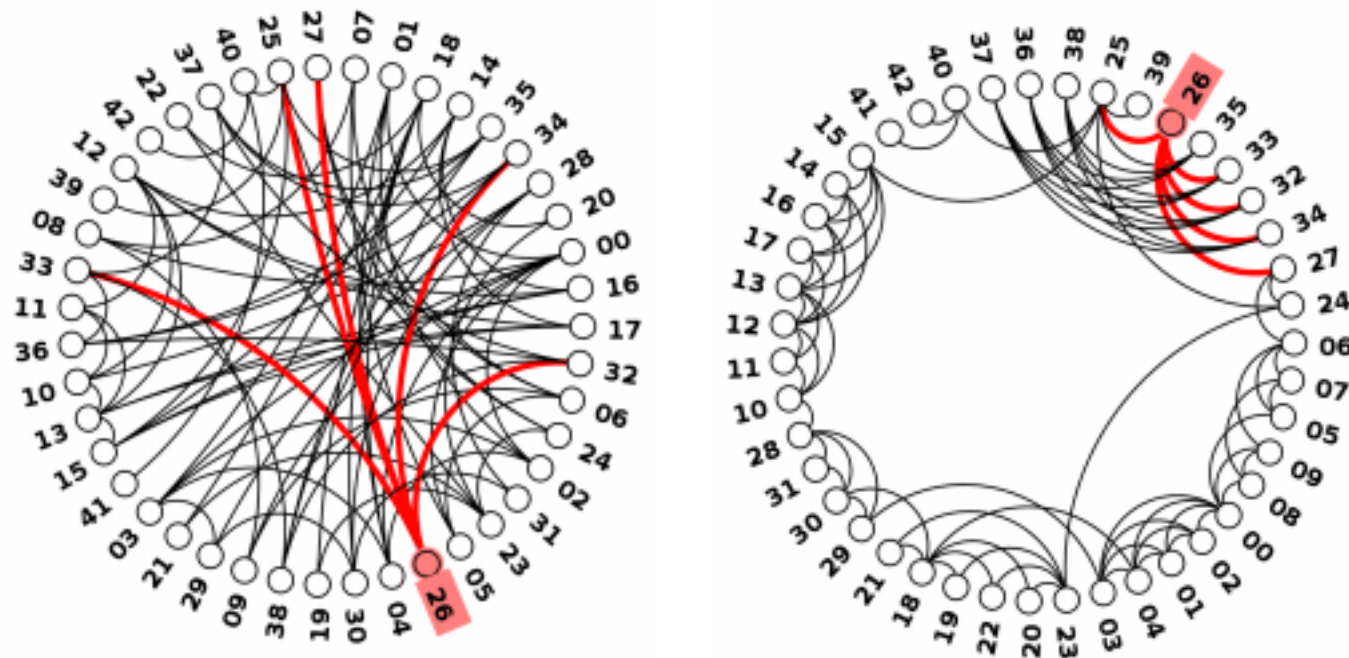
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short
- tasks
  - explore topology; locate paths, clusters
- scalability
  - node/edge density  $E < 4N$



<http://mbostock.github.com/d3/ex/force.html>

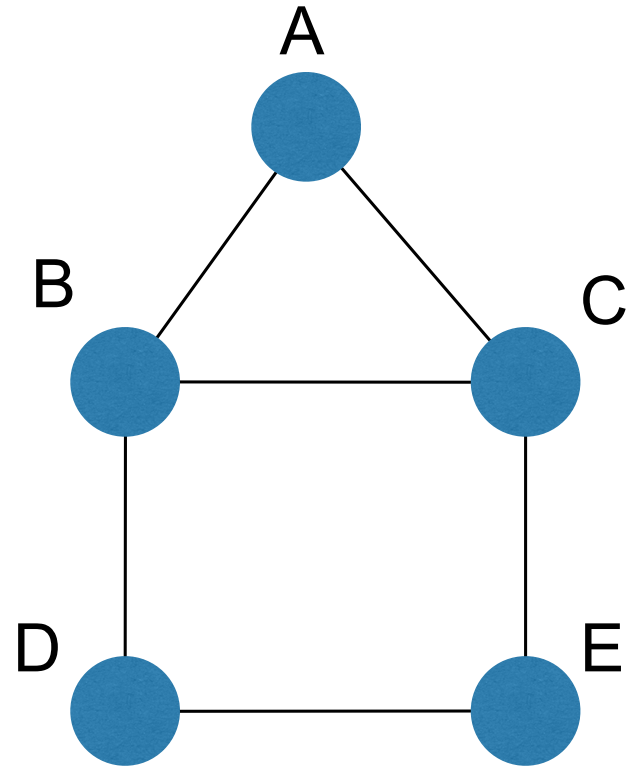
# Idiom: **circular layouts / arc diagrams (node-link)**

- restricted node-link layouts: lay out nodes around circle or along line
- data
  - original: network
  - derived: node ordering attribute (global computation)
- considerations: node ordering crucial to avoid excessive clutter from edge crossings
  - examples: before & after barycentric ordering



# Adjacency matrix representations

- derive adjacency matrix from network



	A	B	C	D	E
A		■	■		
B	■		■	■	
C	■	■			■
D		■			■
E			■	■	

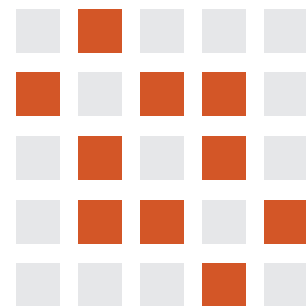


## Adjacency Matrix

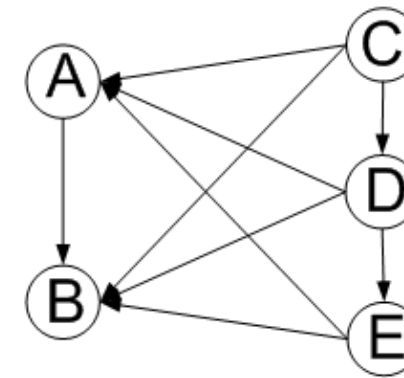
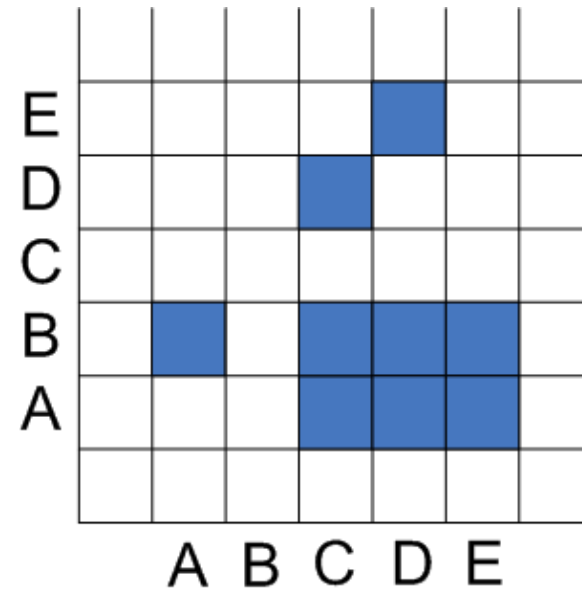
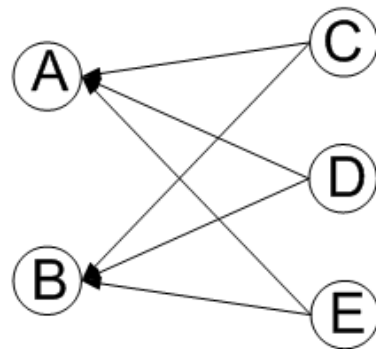
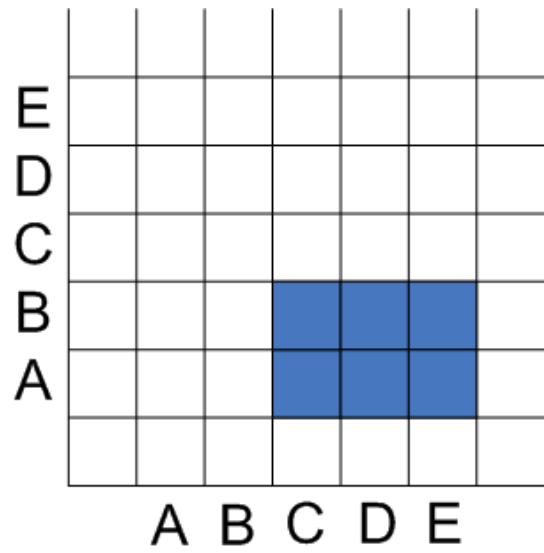
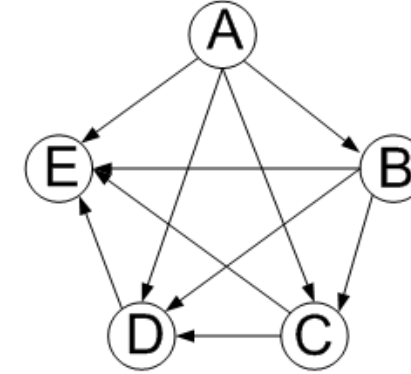
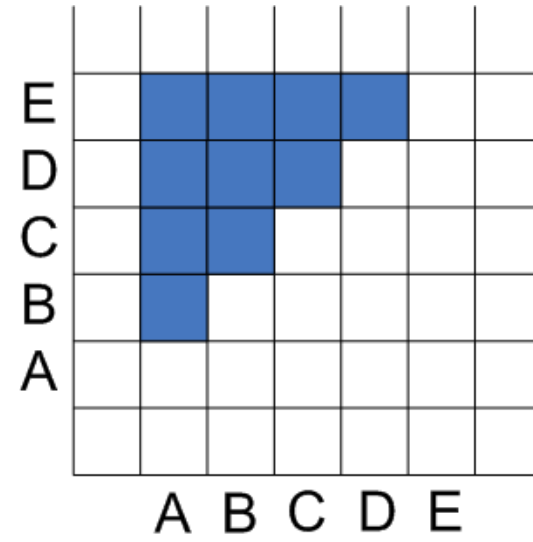
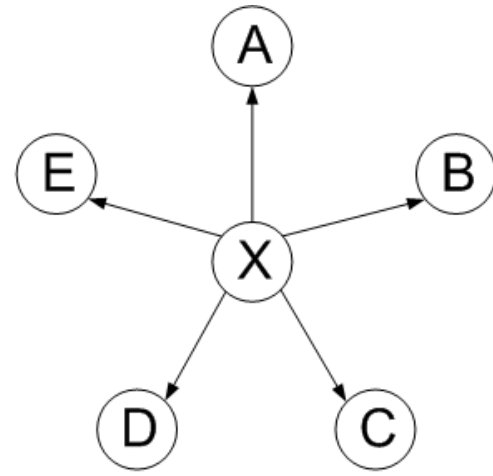
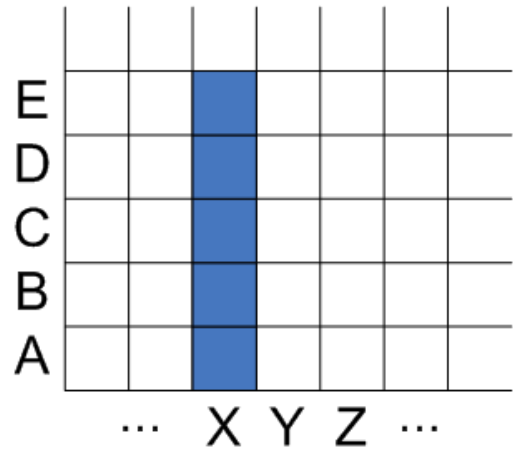
Derived Table

✓ NETWORKS

✓ TREES

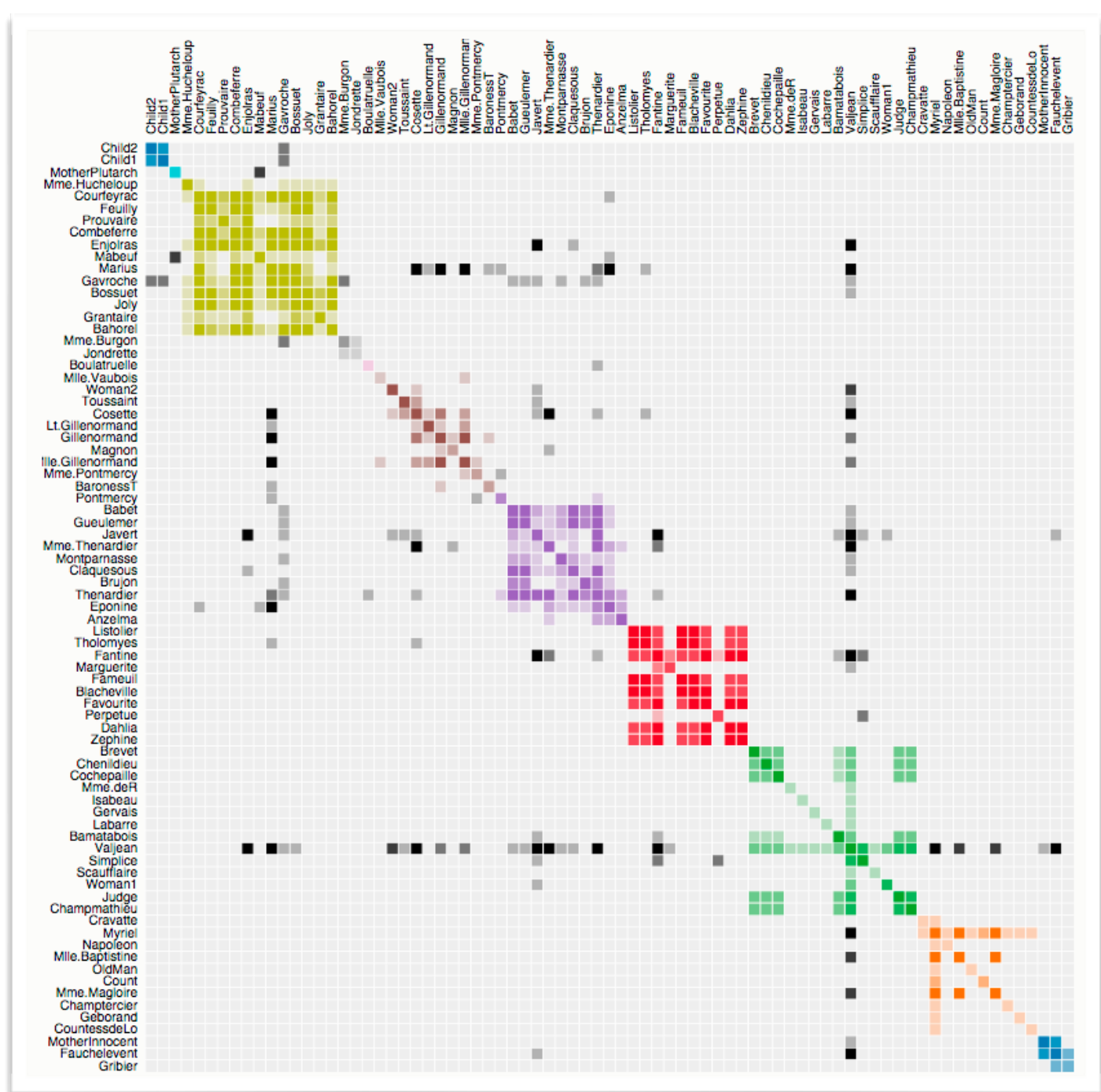
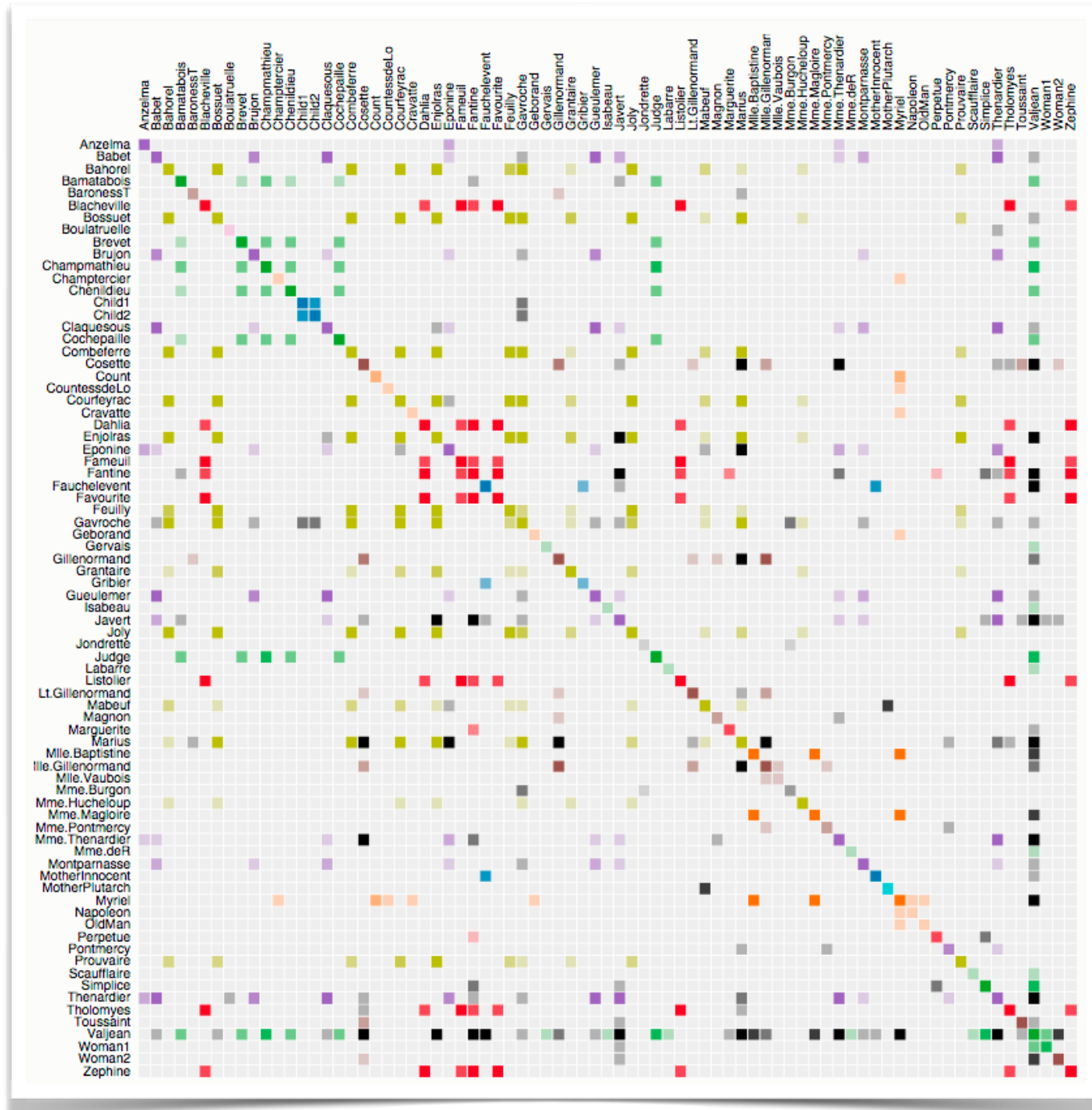


# Adjacency matrix examples





# Node order is crucial: Reordering

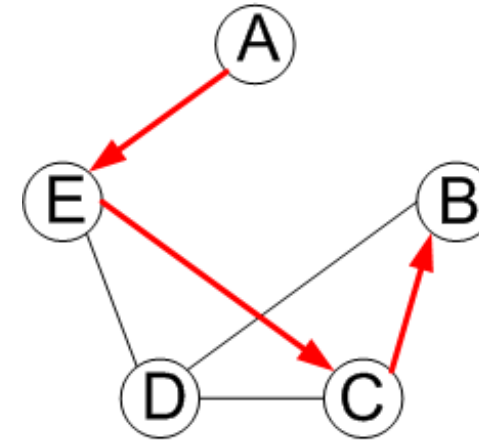


<https://bost.ocks.org/mike/miserables/>

# Adjacency matrix

		TO							
		A	B	C	D	E	F	G	H
FROM	A		■	■					
	B	■		■	■				
	C		■				■		
	D								■
	E				■		■	■	
	F					■		■	
	G						■		■
	H					■		■	

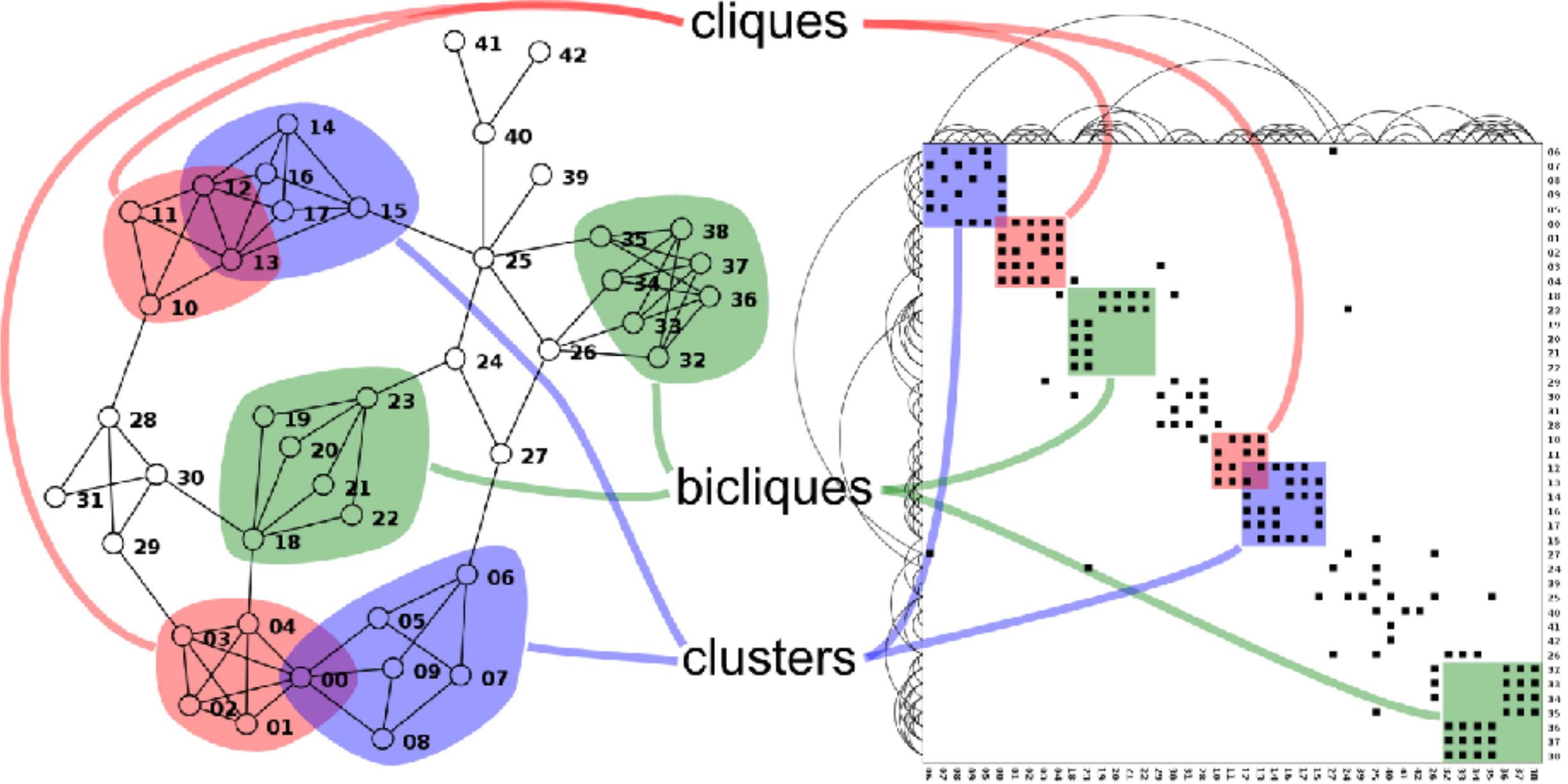
good for topology tasks  
related to neighborhoods  
(node 1-hop neighbors)



E		■	■	■	
D		■	■		■
C			■	■	■
B			■	■	
A					■
	A	B	C	D	E

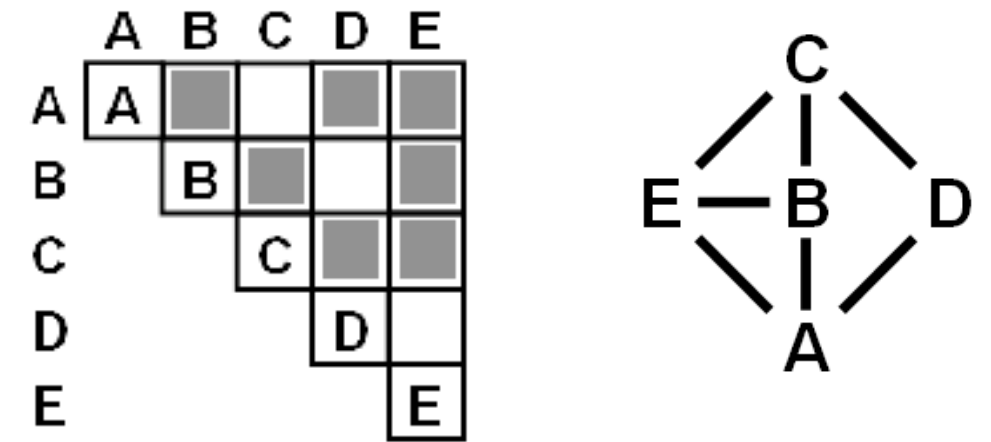
bad for topology tasks  
related to paths

# Structures visible in both

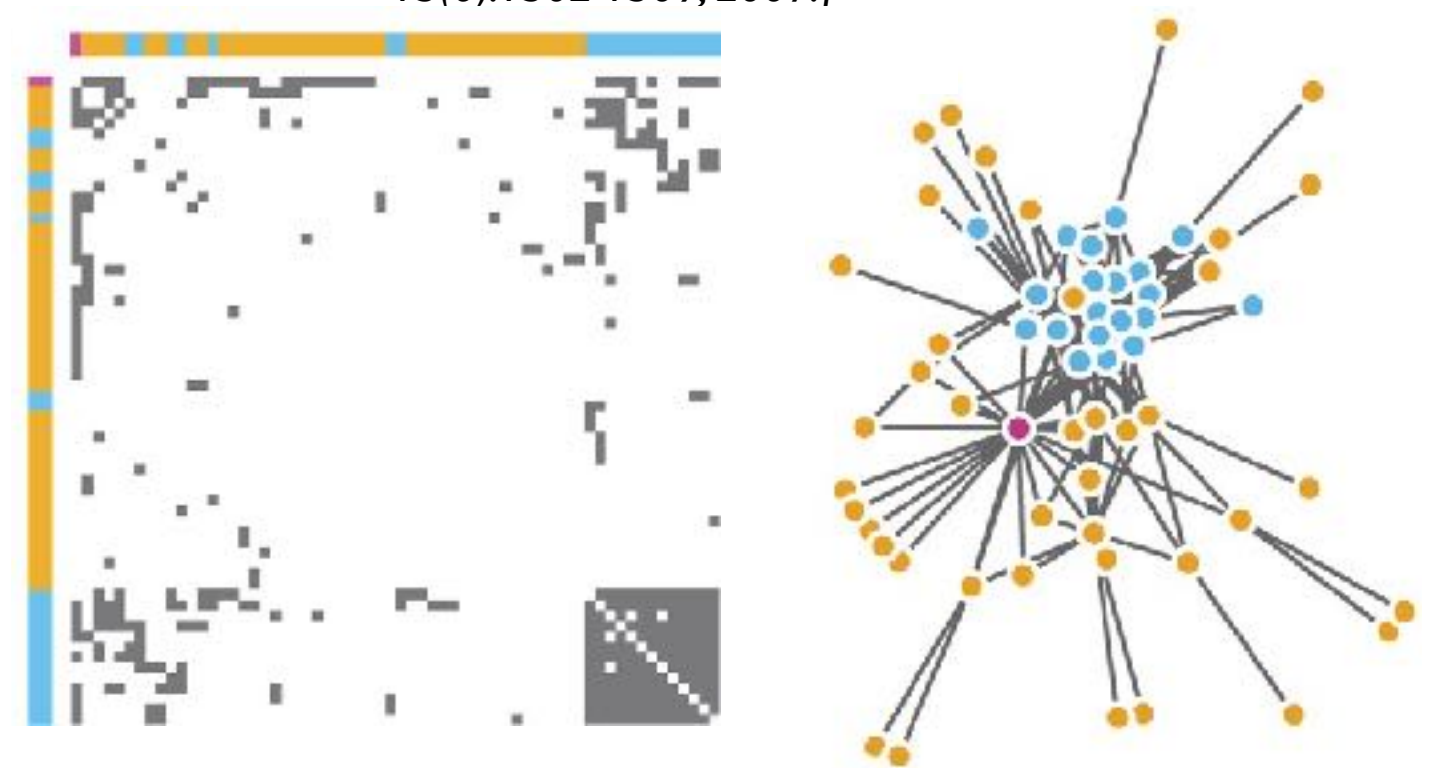


# Idiom: adjacency matrix view

- data: network
  - transform into same data/encoding as heatmap
- derived data: table from network
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- visual encoding
  - cell shows presence/absence of edge
- scalability
  - 1K nodes, 1M edges



[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]

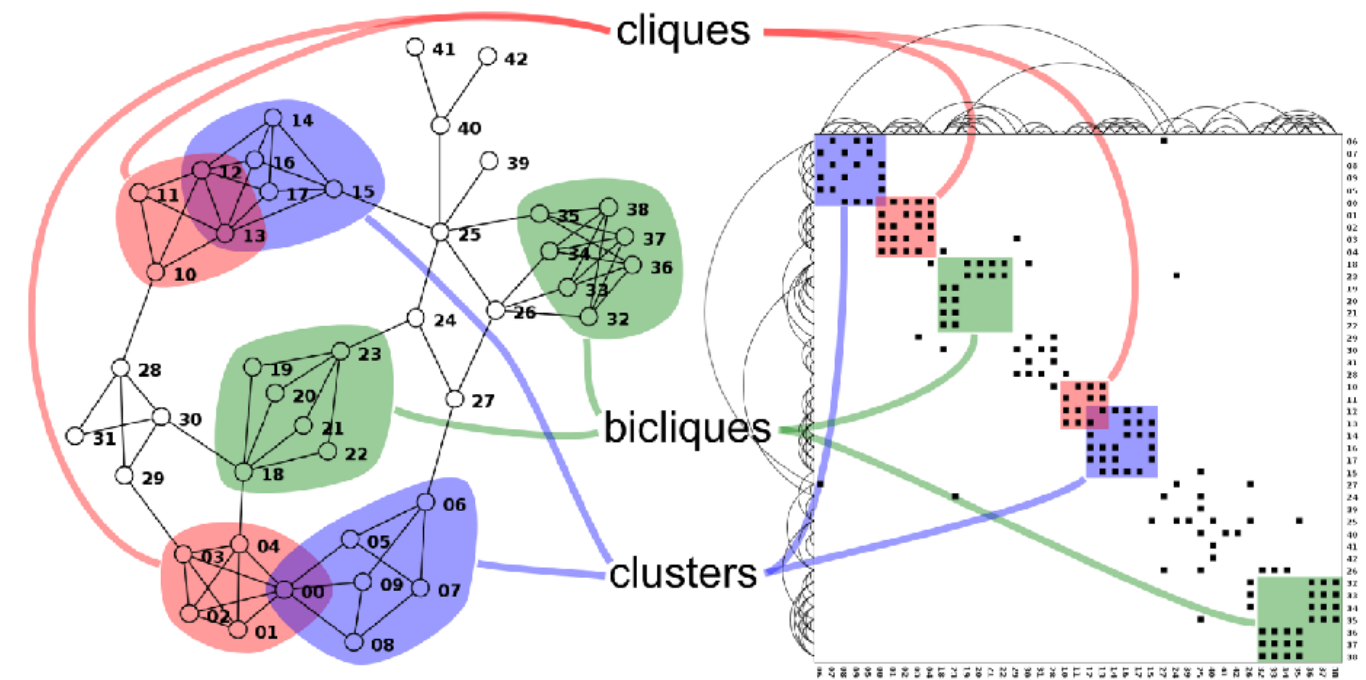


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

# Node-link vs. matrix comparison

- node-link diagram strengths
  - topology understanding, path tracing
  - intuitive, flexible, no training needed
- adjacency matrix strengths
  - focus on edges rather than nodes
  - layout straightforward (reordering needed)
  - predictability, scalability
  - some topology tasks trainable
- empirical study
  - node-link best for small networks
  - matrix best for large networks
    - if tasks don't involve path tracing!

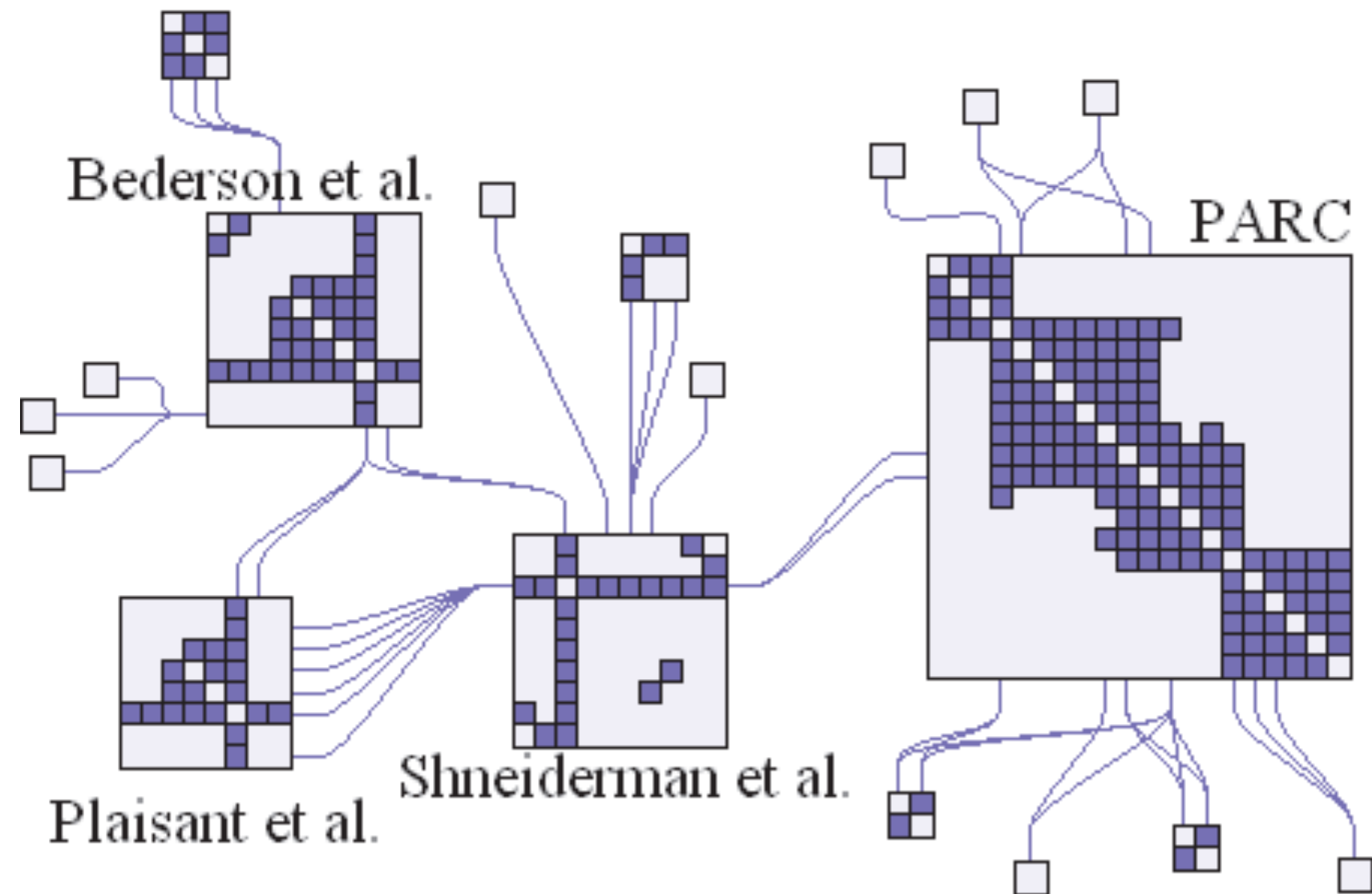
*[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]*



<http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png>

# Idiom: NodeTrix

- hybrid nodelink/matrix
- capture strengths of both



[NodeTrix: a Hybrid Visualization of Social Networks.  
Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis)  
13(6):1302-1309, 2007.]

# Trees

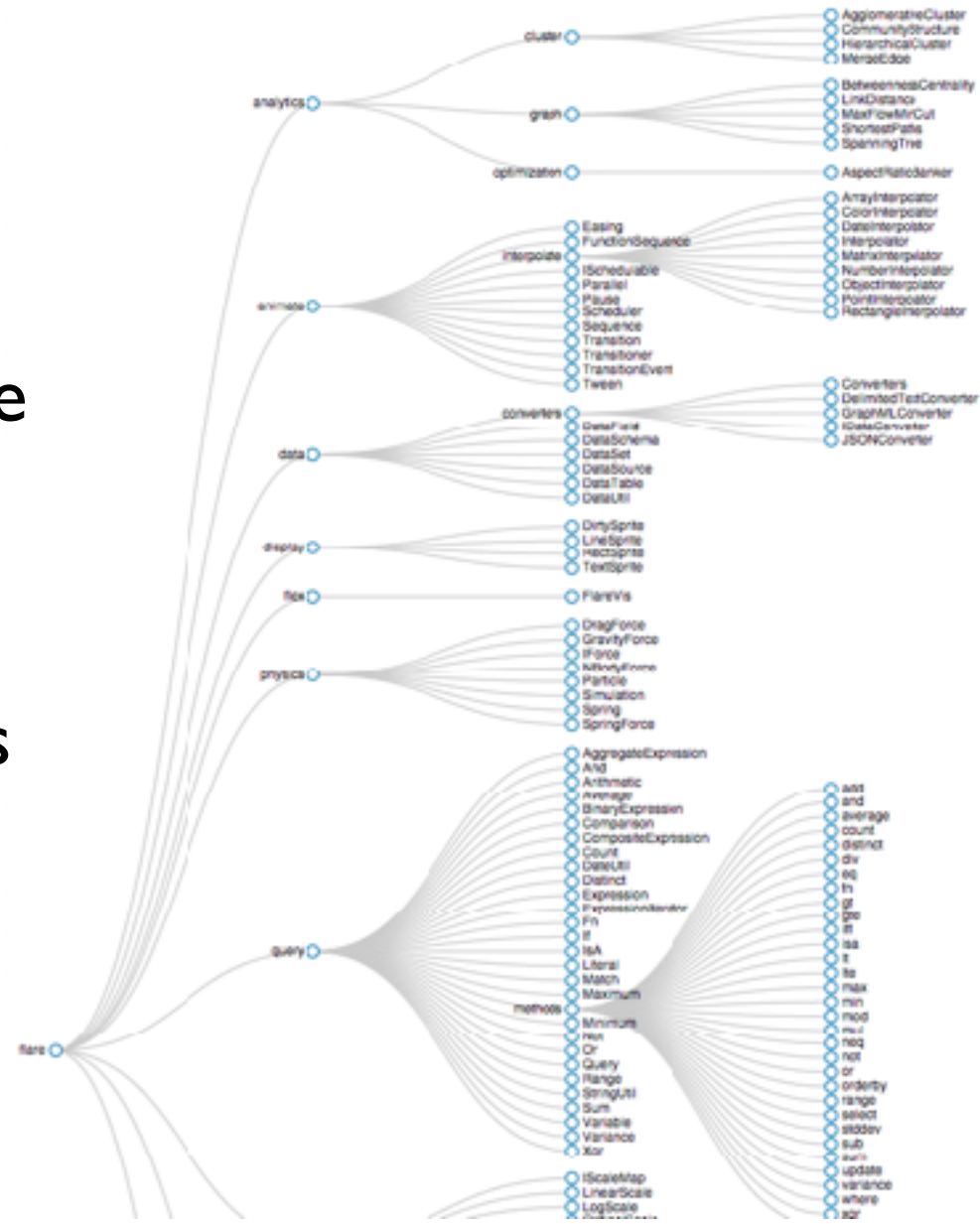
# Node-link trees

- Reingold-Tilford
  - tidy drawings of trees
    - exploit parent/child structure
  - allocate space: compact but without overlap
    - rectilinear and radial variants

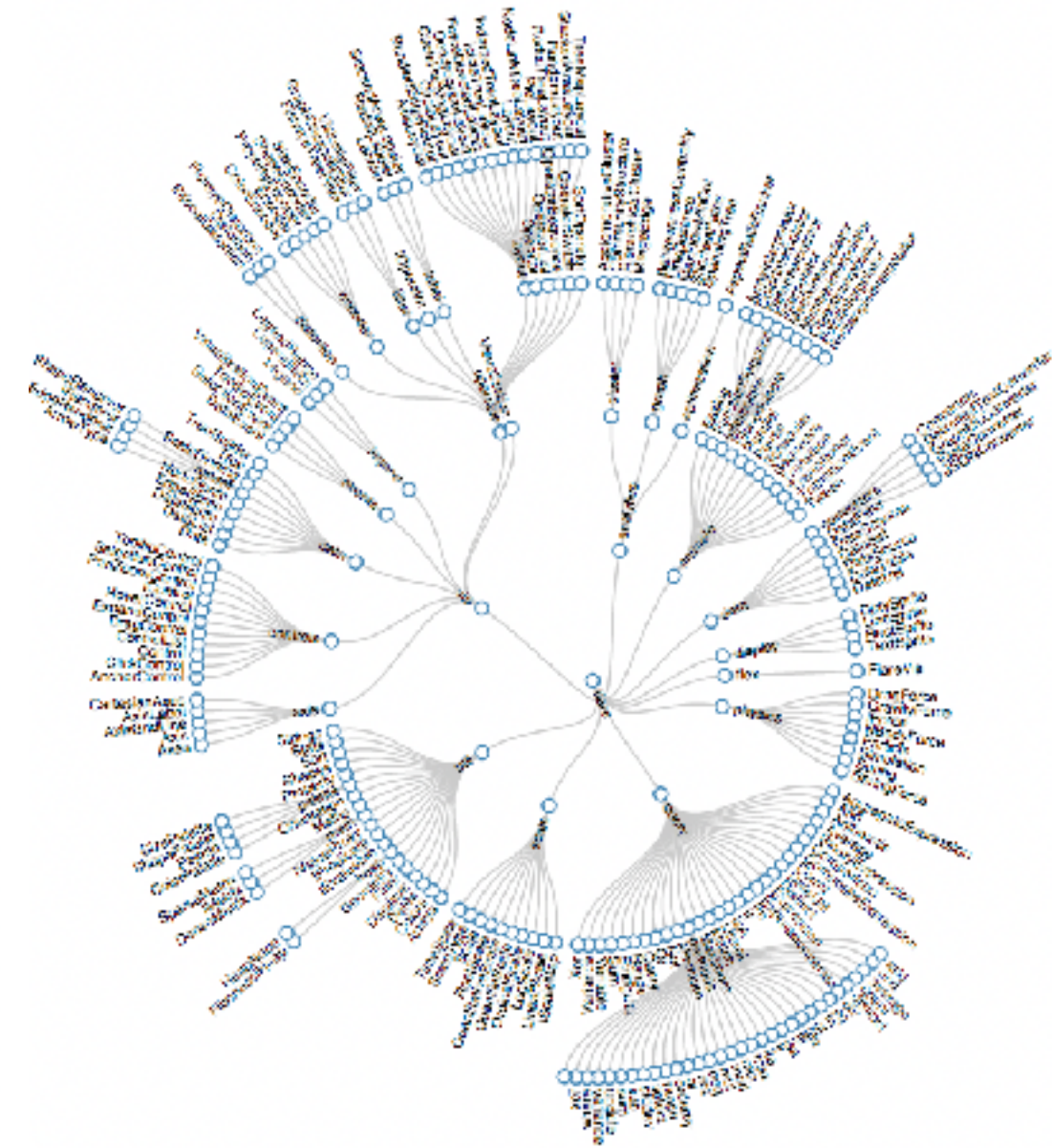
[Tidier drawing of trees. Reingold and Tilford. IEEE Trans. Software Eng., SE-7(2):223–228, 1981.]

- nice algorithm writeup

<http://billmill.org/pymag-trees/>



<http://bl.ocks.org/mbostock/4339184>



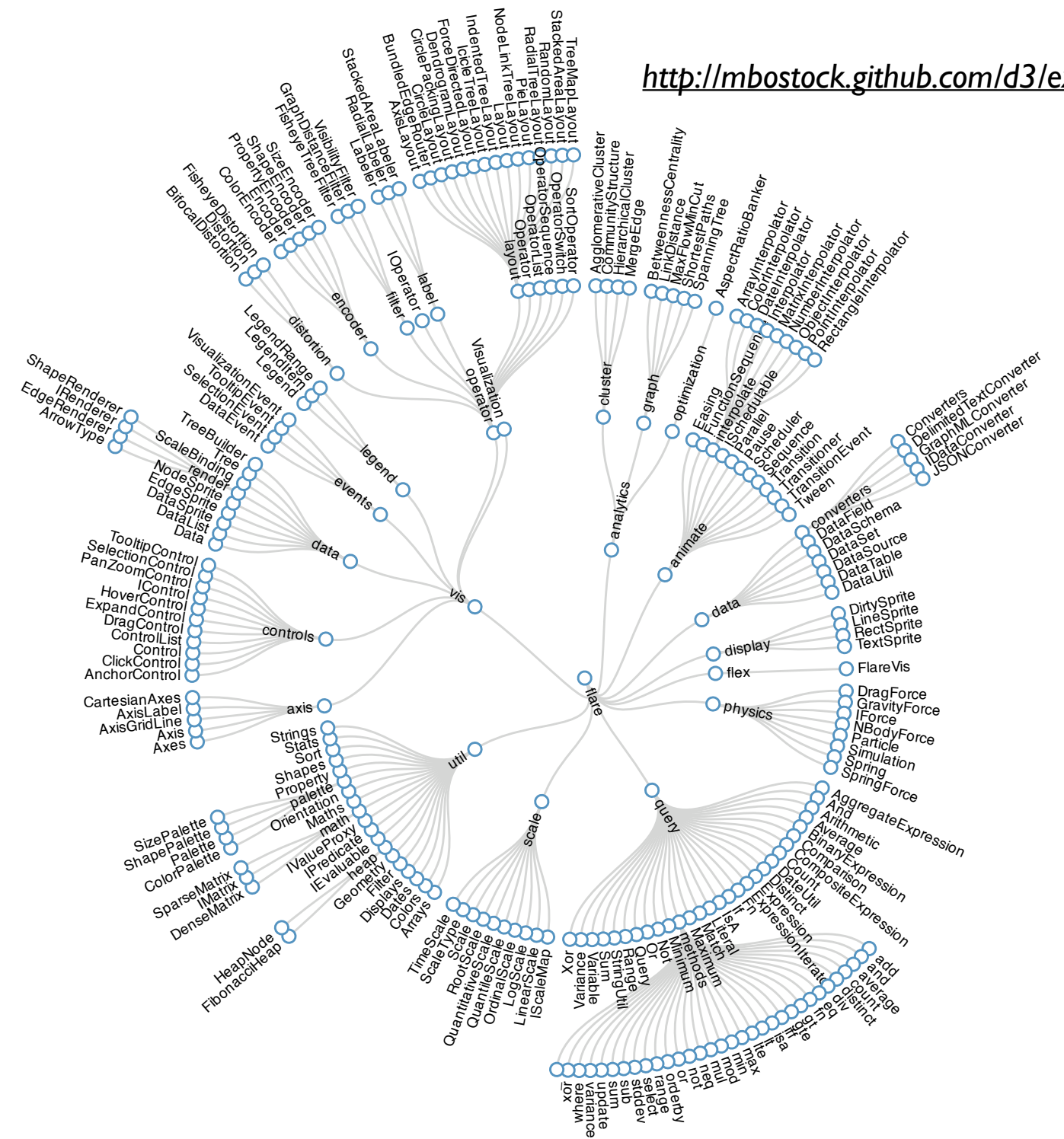
<http://bl.ocks.org/mbostock/4063550>



# Idiom: radial node-link tree

<http://mbostock.github.com/d3/ex/tree.html>

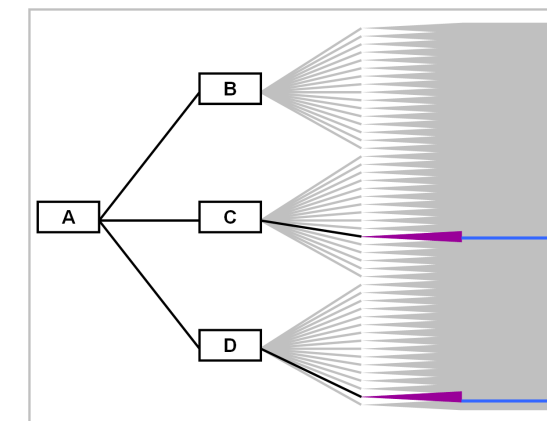
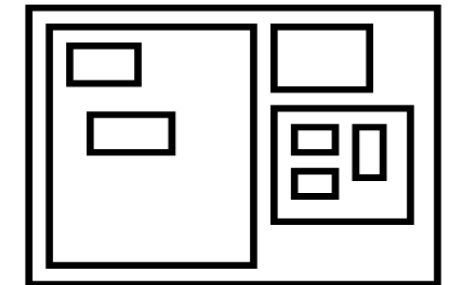
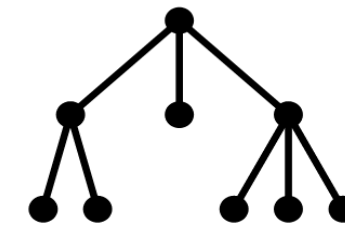
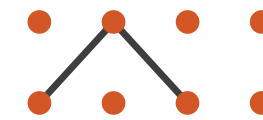
- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes (with/without labels)



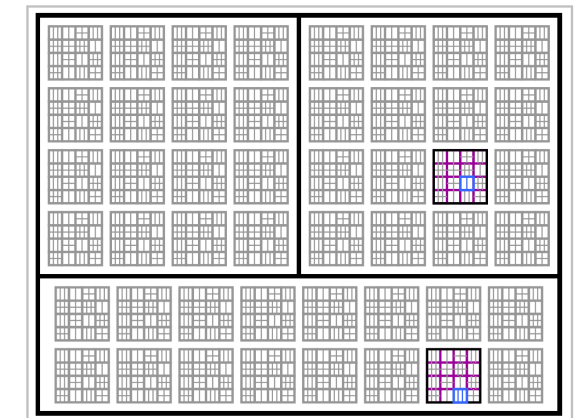
# Link marks: Connection and containment

- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
    - only trees

→ Connection → Containment



**Node-Link Diagram**



**Treemap**

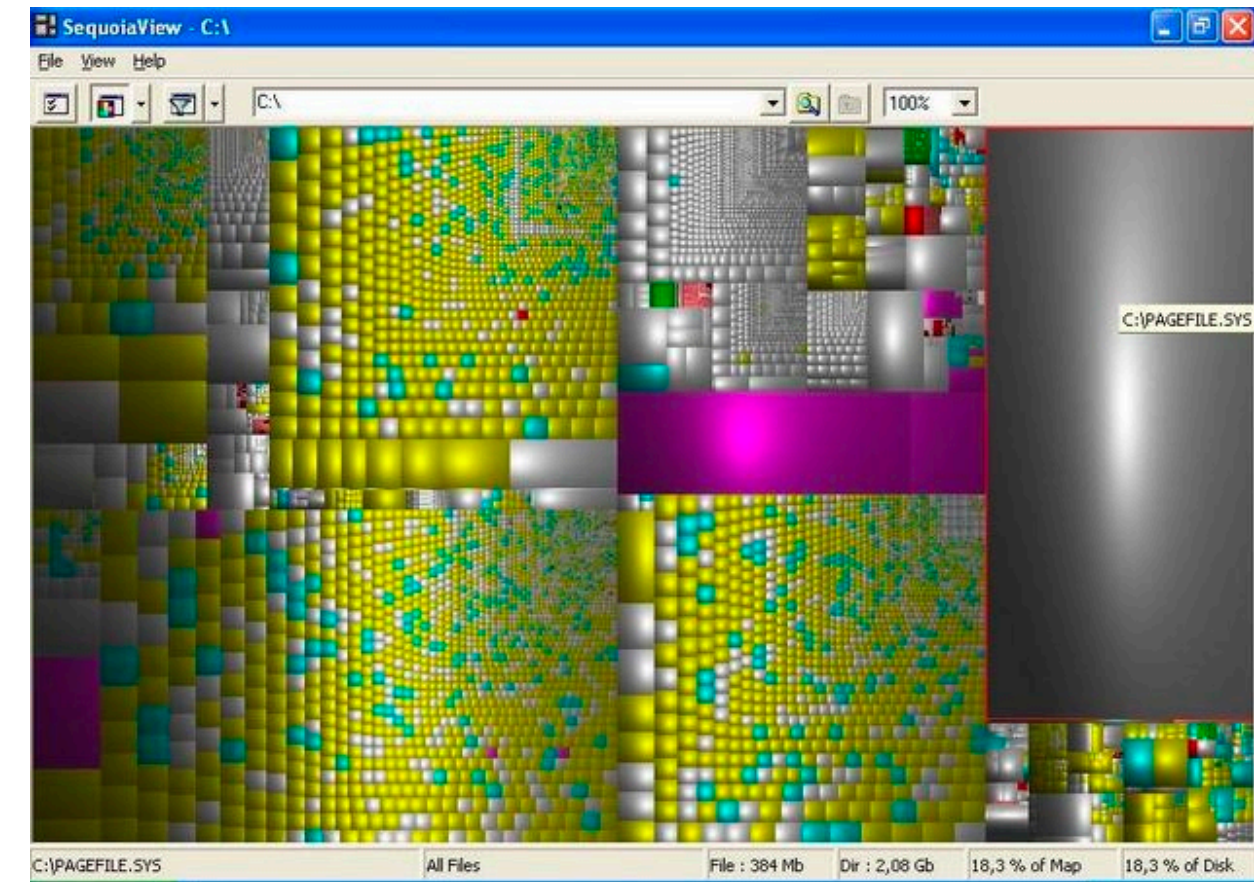
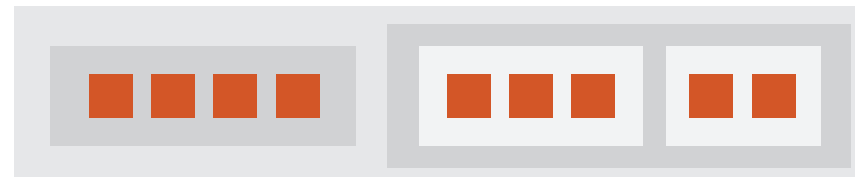
[Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams.  
Dong, McGuffin, and Chignell. Proc. InfoVis 2005, p. 57-64.]

# Idiom: treemap

- data
  - tree
  - 1 quant attrib at leaf nodes
- encoding
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- tasks
  - query attribute at leaf nodes
  - ex: disk space usage within filesystem
- scalability
  - IM leaf nodes

➔ **Enclosure**  
Containment Marks

NETWORKS  TREES



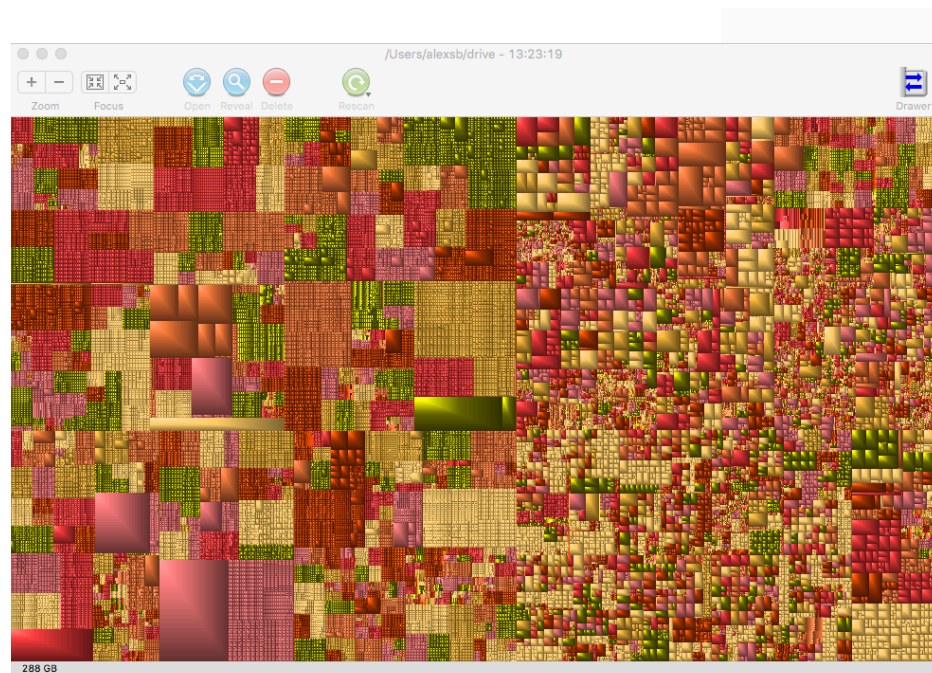
<https://www.win.tue.nl/sequoiaview/>

[Cushion Treemaps. van Wijk and van de Wetering.  
Proc. Symp. InfoVis 1999, 73-78.]

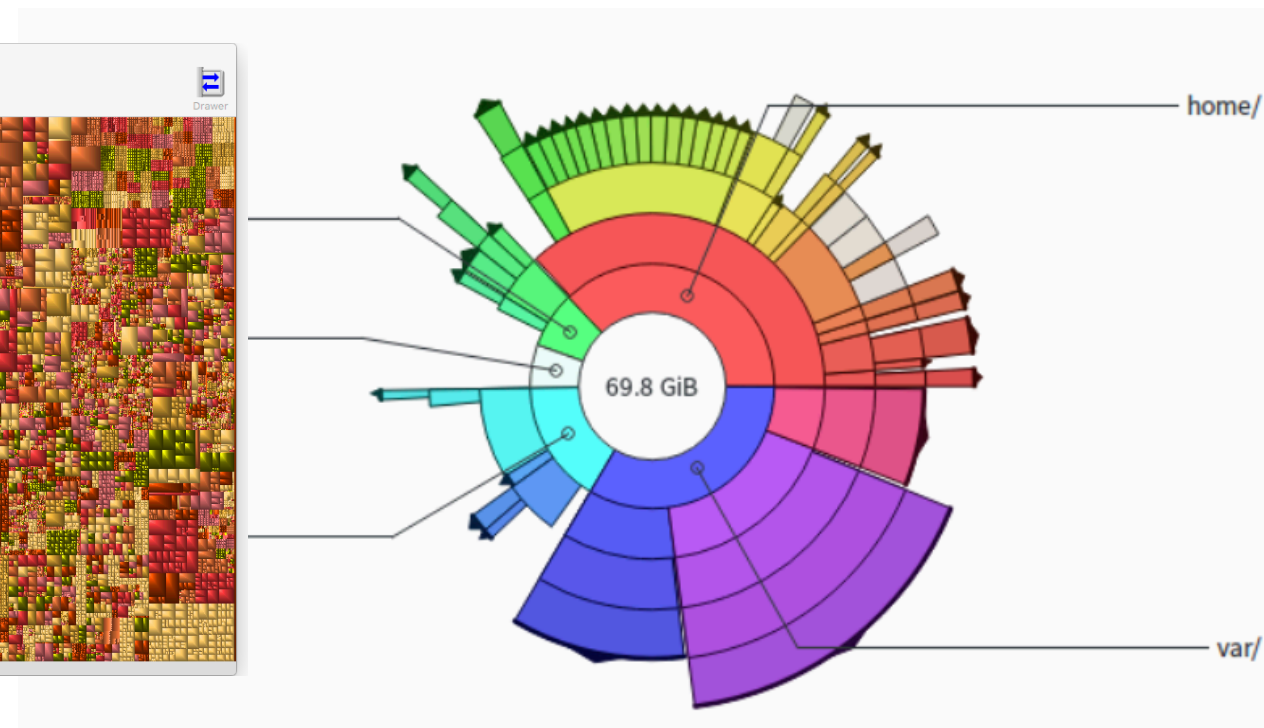
# Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

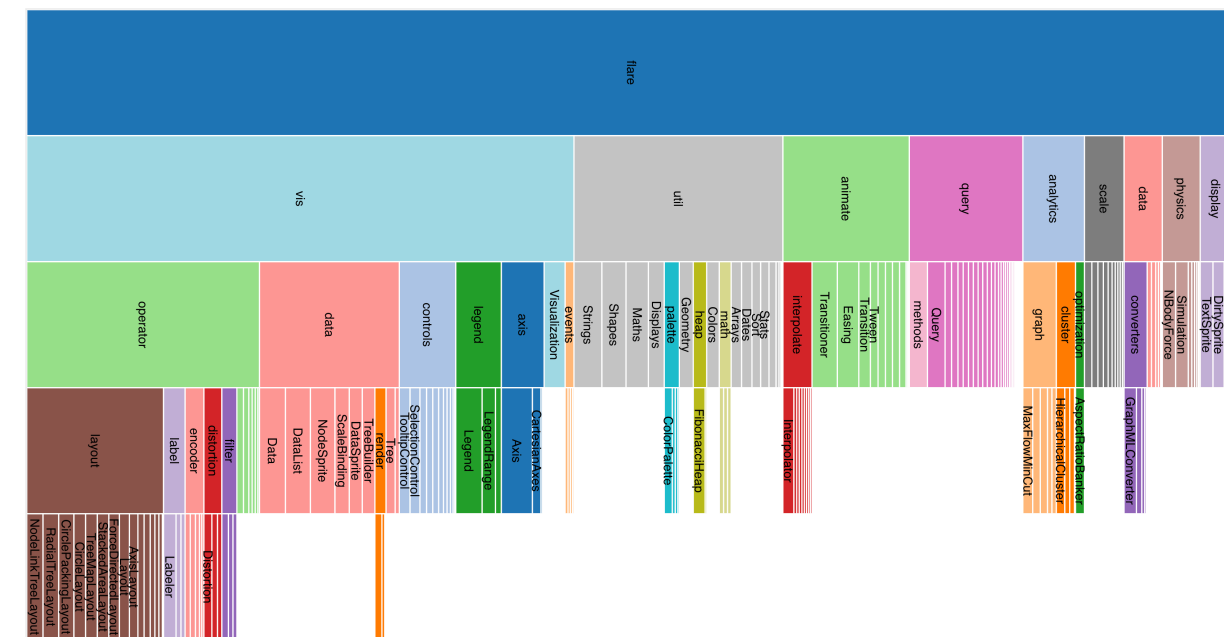
Treemap  
containment



Sunburst  
position (radial)



Icicle Plot  
position (rectilinear)





# Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment:
  - show parent-child relationships only through relative positions

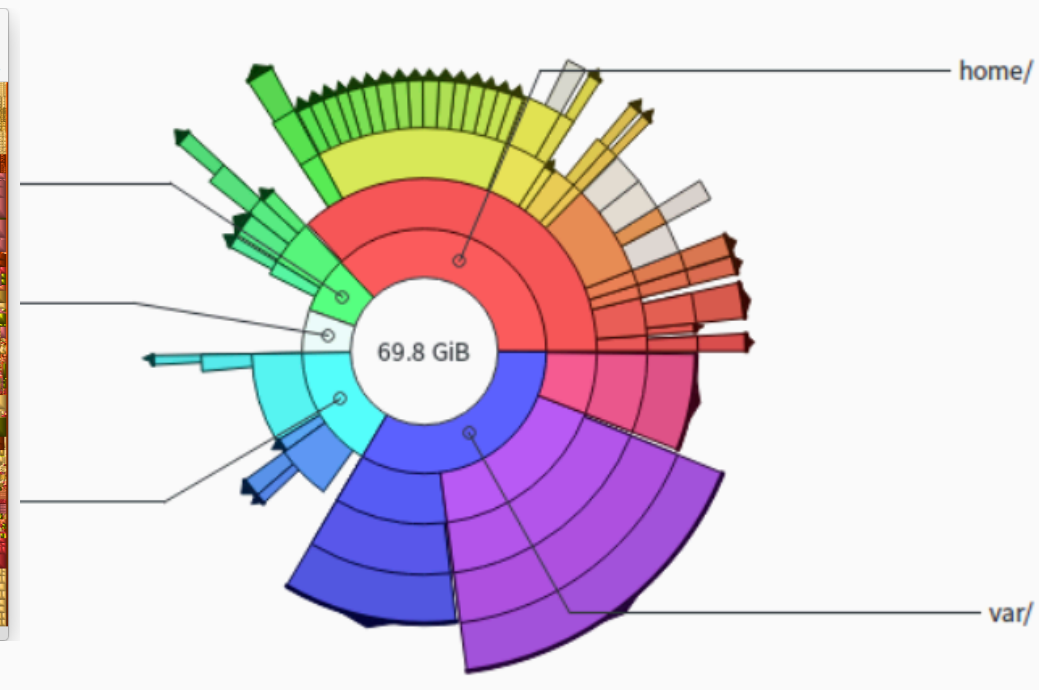
## Treemap

containment  
only leaves visible



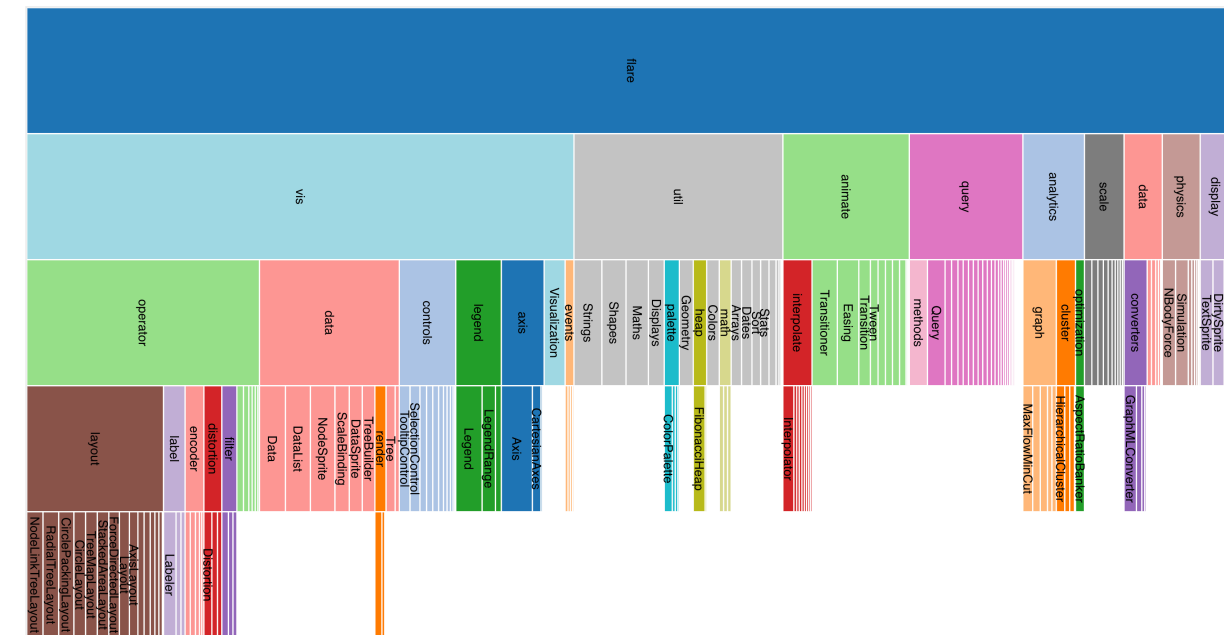
## Sunburst

position (radial)  
inner nodes & leaves visible



## Icicle Plot

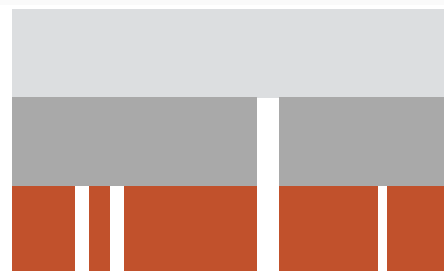
position (rectilinear)  
inner nodes & leaves visible



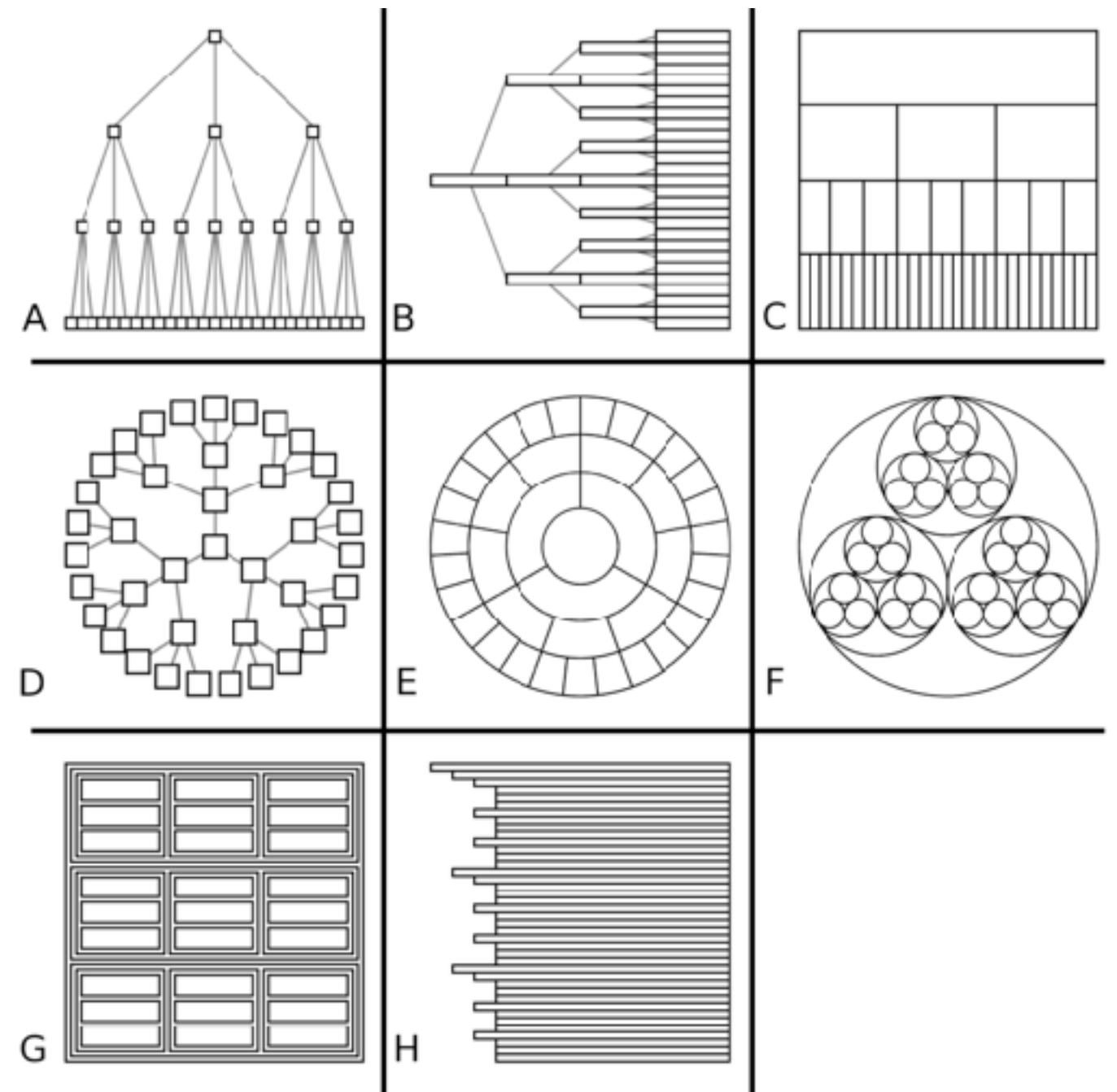
➔ **Implicit**  
Spatial Position

✗ NETWORKS

✓ TREES

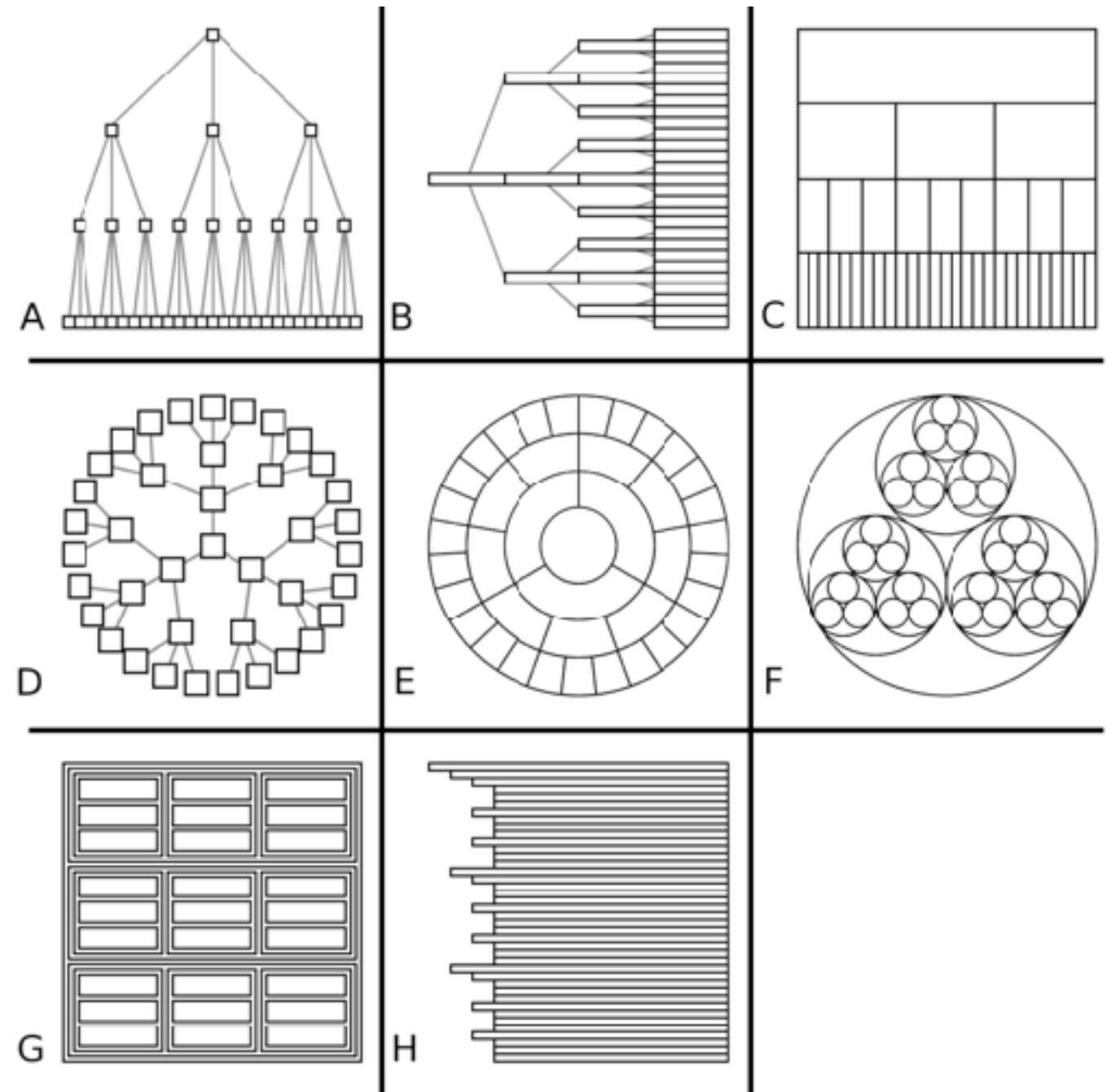


# Tree drawing idioms comparison



# Comparison: tree drawing idioms

- data shown
  - link relationships
  - tree depth
  - sibling order

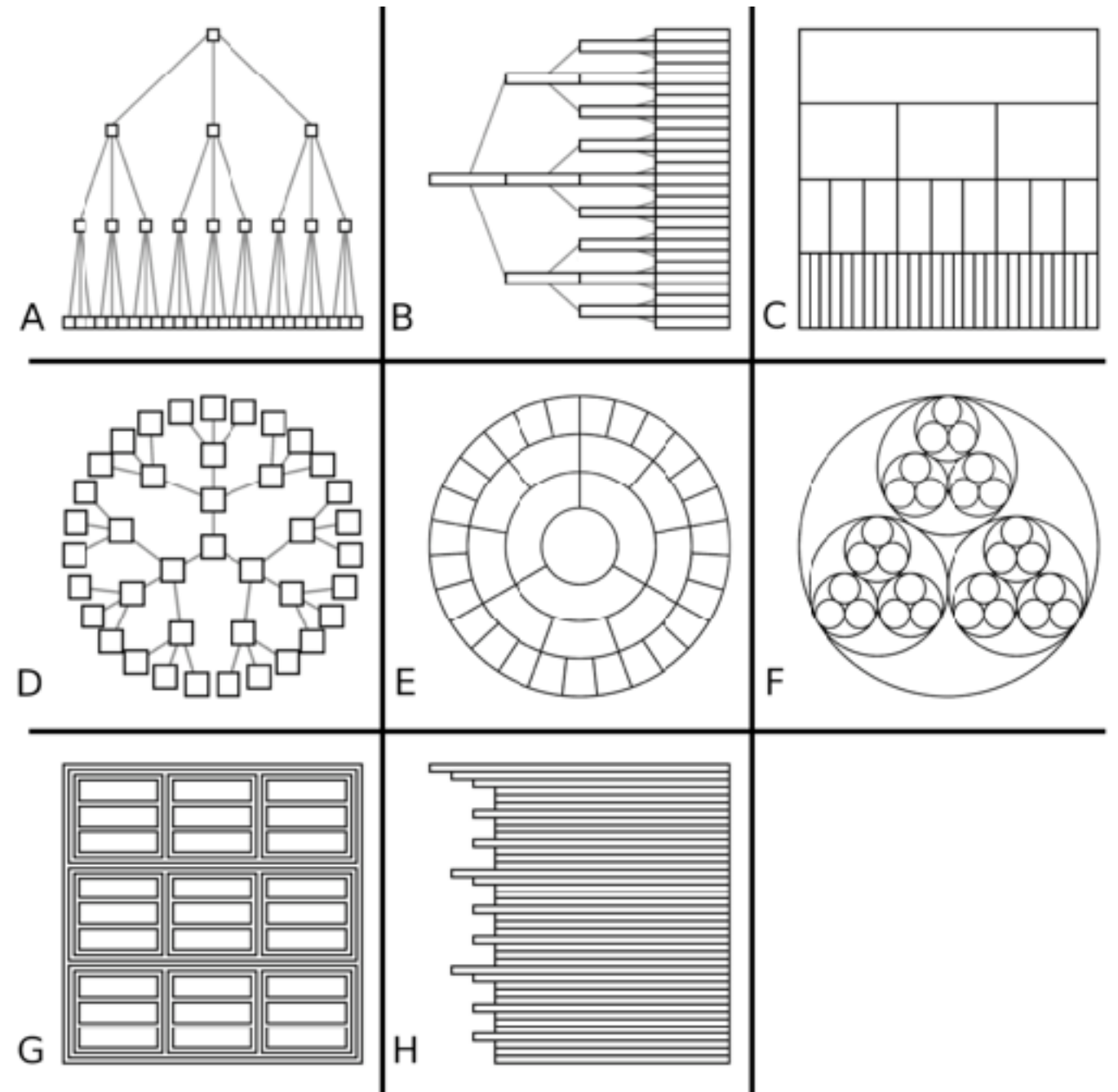


[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees.  
McGuffin and Robert. *Information Visualization* 9:2 (2010), 115–140.]



# Comparison: tree drawing idioms

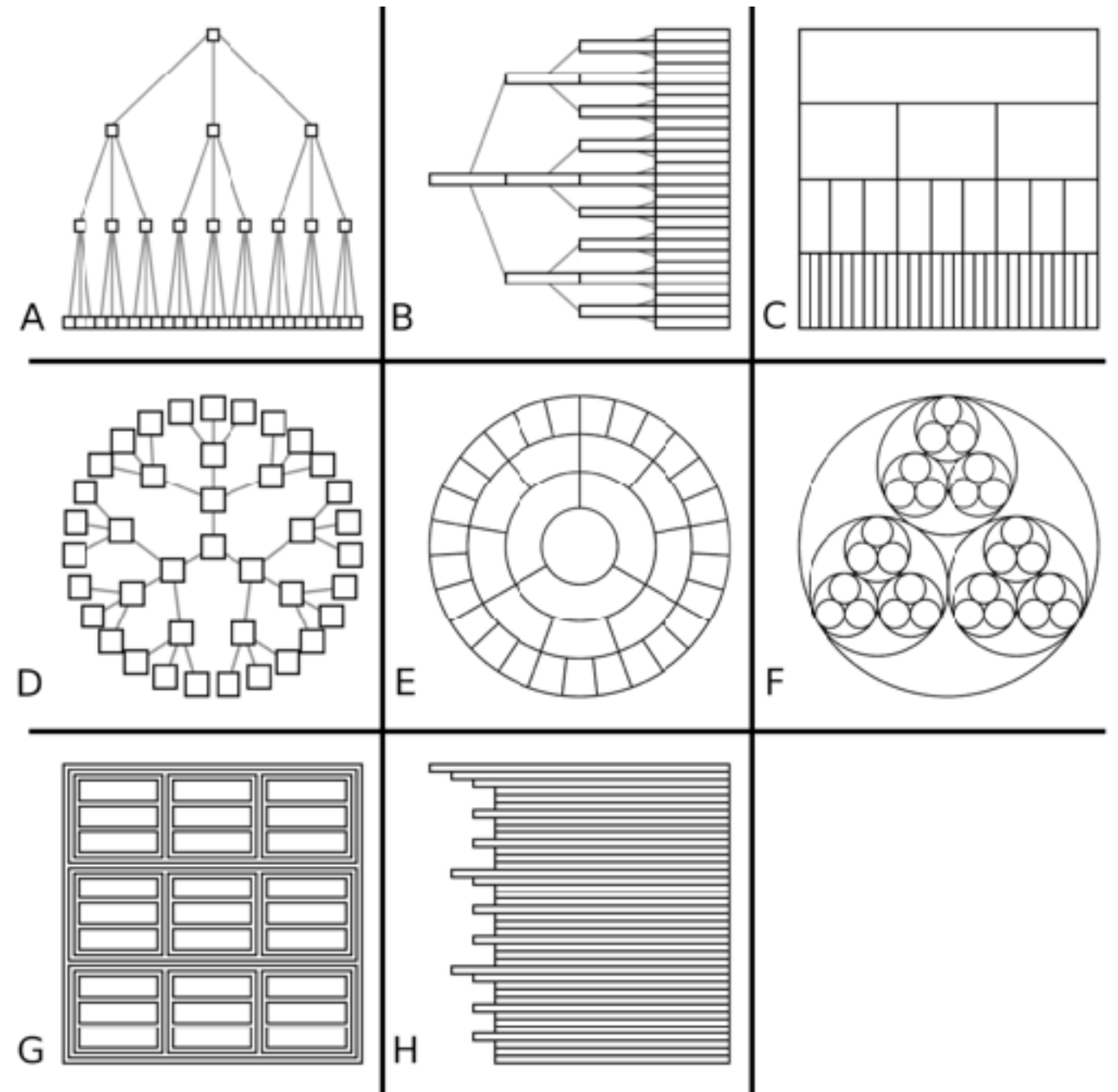
- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees.  
McGuffin and Robert. *Information Visualization* 9:2 (2010), 115–140.]

# Comparison: tree drawing idioms

- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels
- considerations
  - redundant? arbitrary?
  - information density?
    - avoid wasting space
    - consider where to fit labels!









[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees.  
McGuffin and Robert. *Information Visualization* 9:2 (2010), 115–140.]




# [treevis.net](http://treevis.net): Many, many options!

How to cite this site? treevis.net - A Visual Bibliography of Tree Visualization 2.0 by Hans-Jörg Schulz v.21-OCT-2014

Check out other surveys!

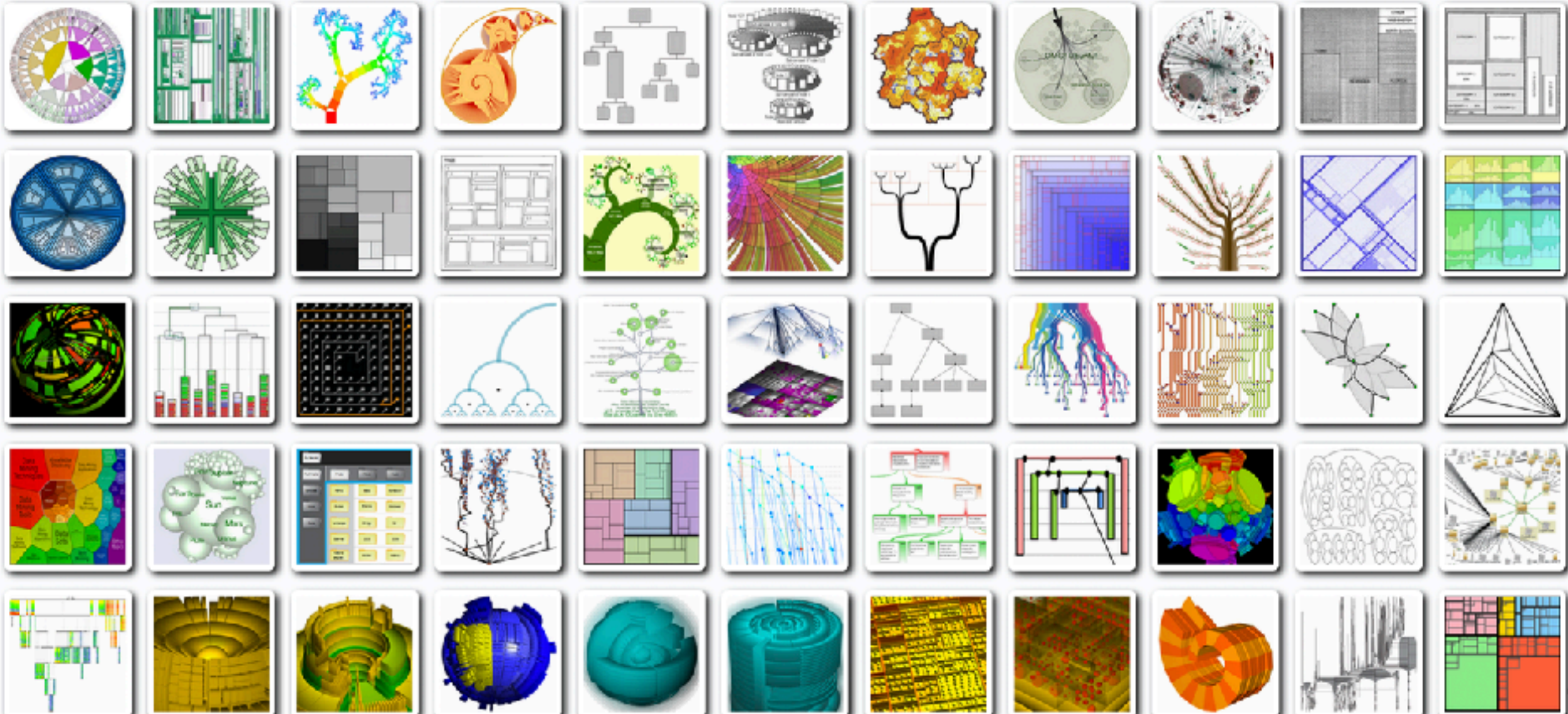
Dimensionality: All   

Representation: All   

Alignment: All   

Fulltext Search:  ×

Techniques Shown: 277

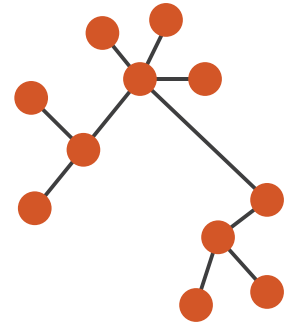


<https://treevis.net/>

# Arrange networks and trees

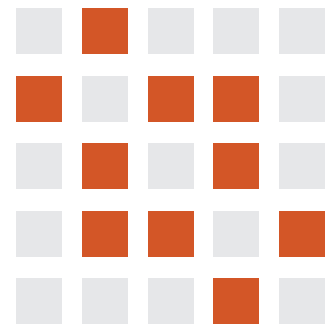
## → Node–Link Diagrams Connection Marks

✓ NETWORKS ✓ TREES



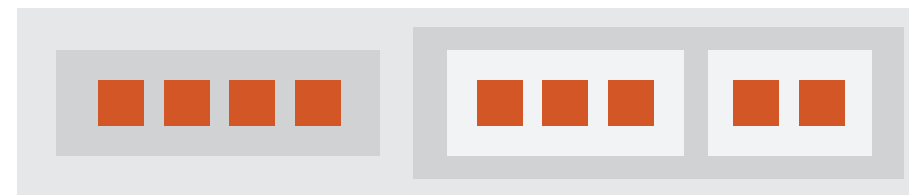
## → Adjacency Matrix Derived Table

✓ NETWORKS ✓ TREES



## → Enclosure Containment Marks

✗ NETWORKS ✓ TREES



## → Implicit Spatial Position

✗ NETWORKS ✓ TREES



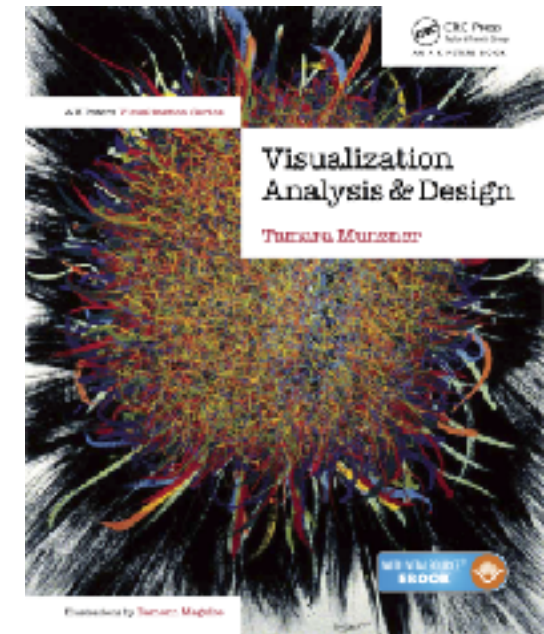
# Visualization Analysis & Design

## *Spatial Data (Ch 9)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

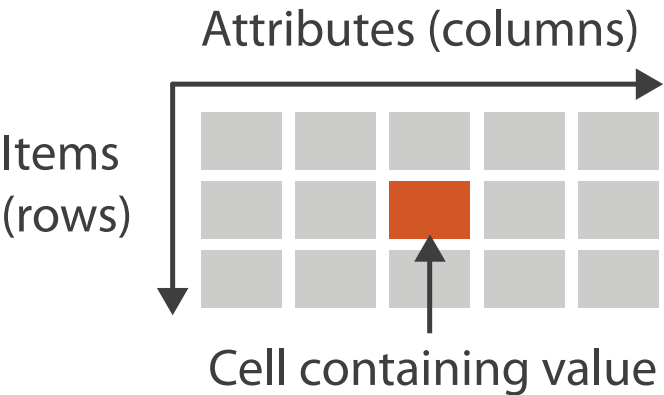
[@tamaramunzner](#)



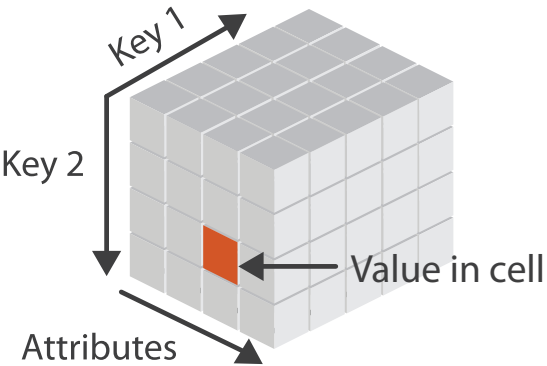
# Focus on Spatial

## → Dataset Types

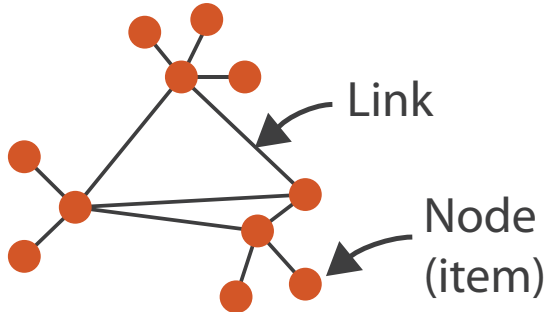
### → Tables



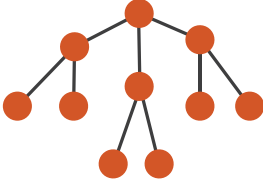
### → Multidimensional Table



### → Networks

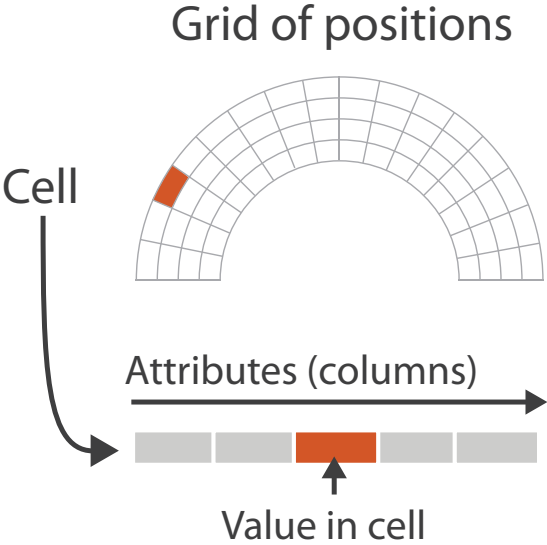


### → Trees

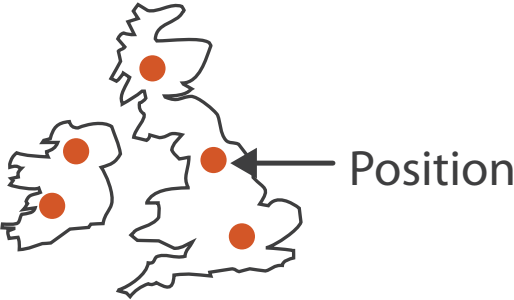


### → Spatial

#### → Fields (Continuous)



#### → Geometry (Spatial)



# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



What?

Why?

How?

### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



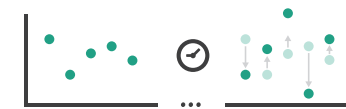
→ Motion

*Direction, Rate, Frequency, ...*

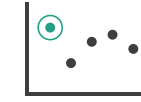


## Manipulate

### → Change



### → Select



### → Navigate

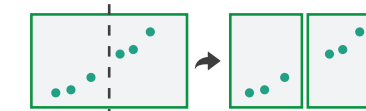


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

### → Filter



### → Aggregate



### → Embed



# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



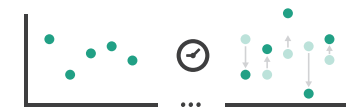
→ Motion

*Direction, Rate, Frequency, ...*

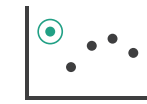


## Manipulate

### → Change



### → Select



### → Navigate

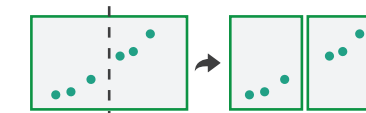


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

### → Filter



### → Aggregate



### → Embed



What?

Why?

How?



# Spatial data

- use given spatial position
- when?
  - dataset contains spatial attributes and they have primary importance
  - central tasks revolve around understanding spatial relationships
- examples
  - geographical/cartographic data
  - sensor/simulation data

# Geographic Maps

# Geographic Map



## Interlocking marks

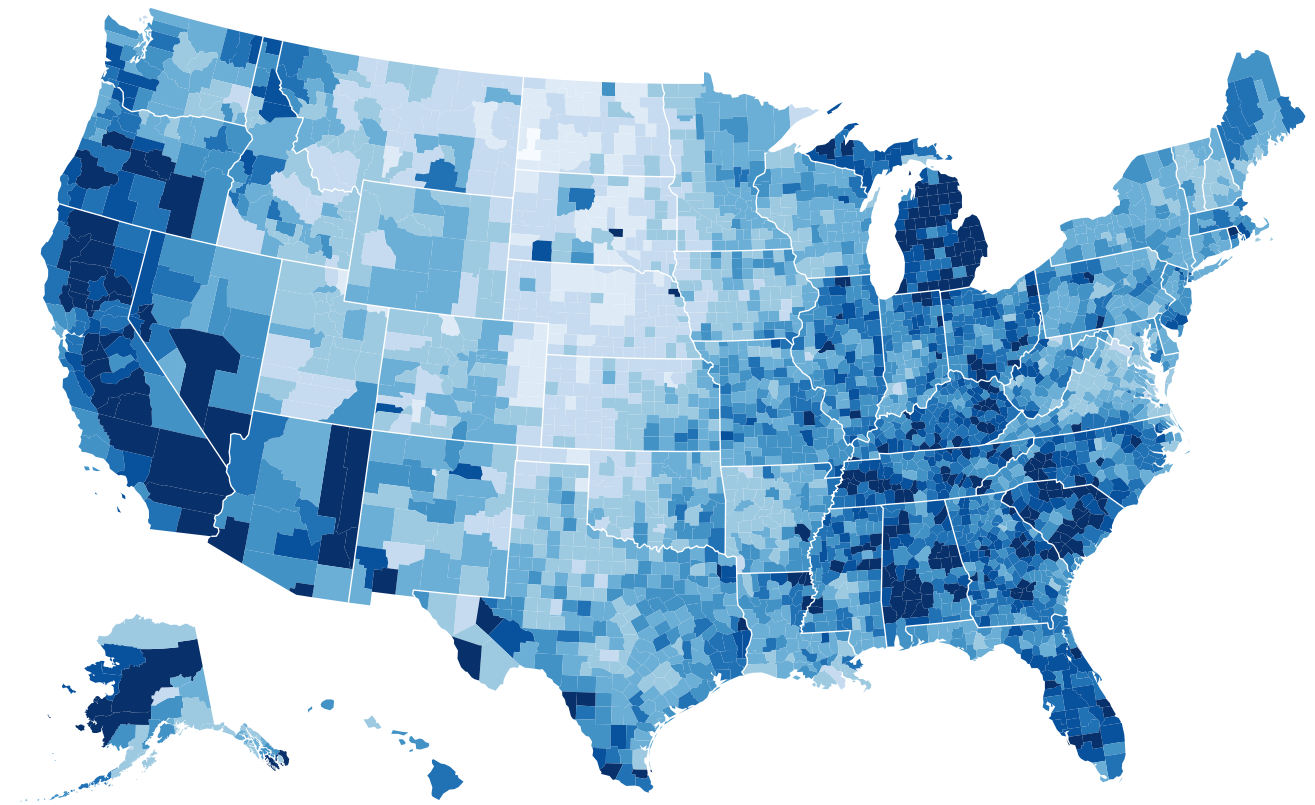
- **shape coded**
  - **area coded**
  - **position coded**
- 
- cannot encode another attribute with these channels, they're "taken"

# Thematic maps

- show spatial variability of attribute ("theme")
  - combine geographic / reference map with (simple, flat) tabular data
  - join together
    - region: interlocking area marks (provinces, countries with outline shapes)
      - also could have point marks (cities, locations with 2D lat/lon coords)
    - region: categorical key attribute in table
      - use to look up value attributes
- major idioms
  - choropleth
  - symbol maps
  - cartograms
  - dot density maps

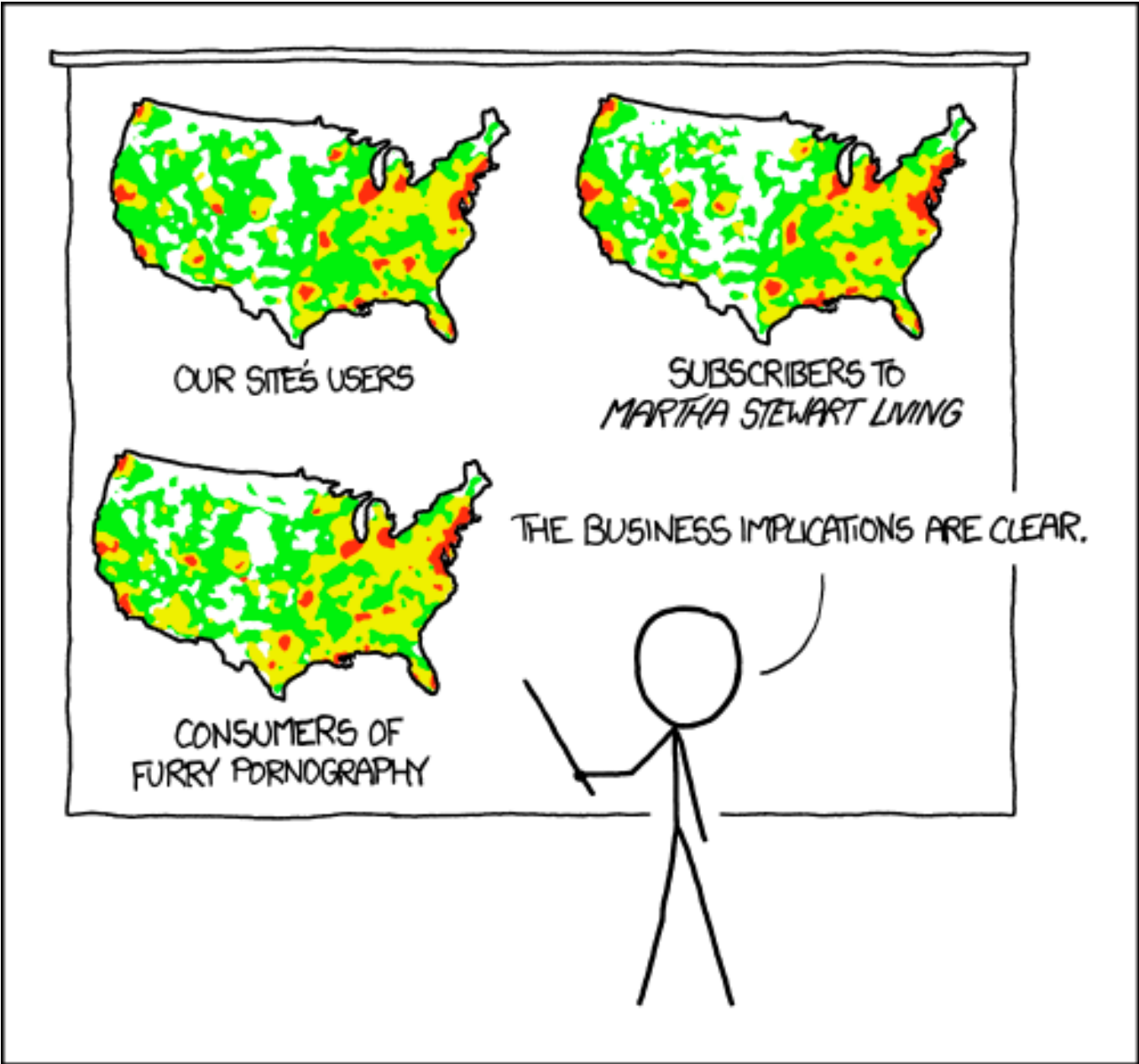
# Idiom: **choropleth map**

- use given spatial data
  - when central task is understanding spatial relationships
- data
  - geographic geometry
  - table with 1 quant attribute per region
- encoding
  - position:
    - use given geometry for area mark boundaries
  - color:
    - sequential segmented colormap



<http://bl.ocks.org/mbostock/4060606>

# Beware: Population maps trickiness!

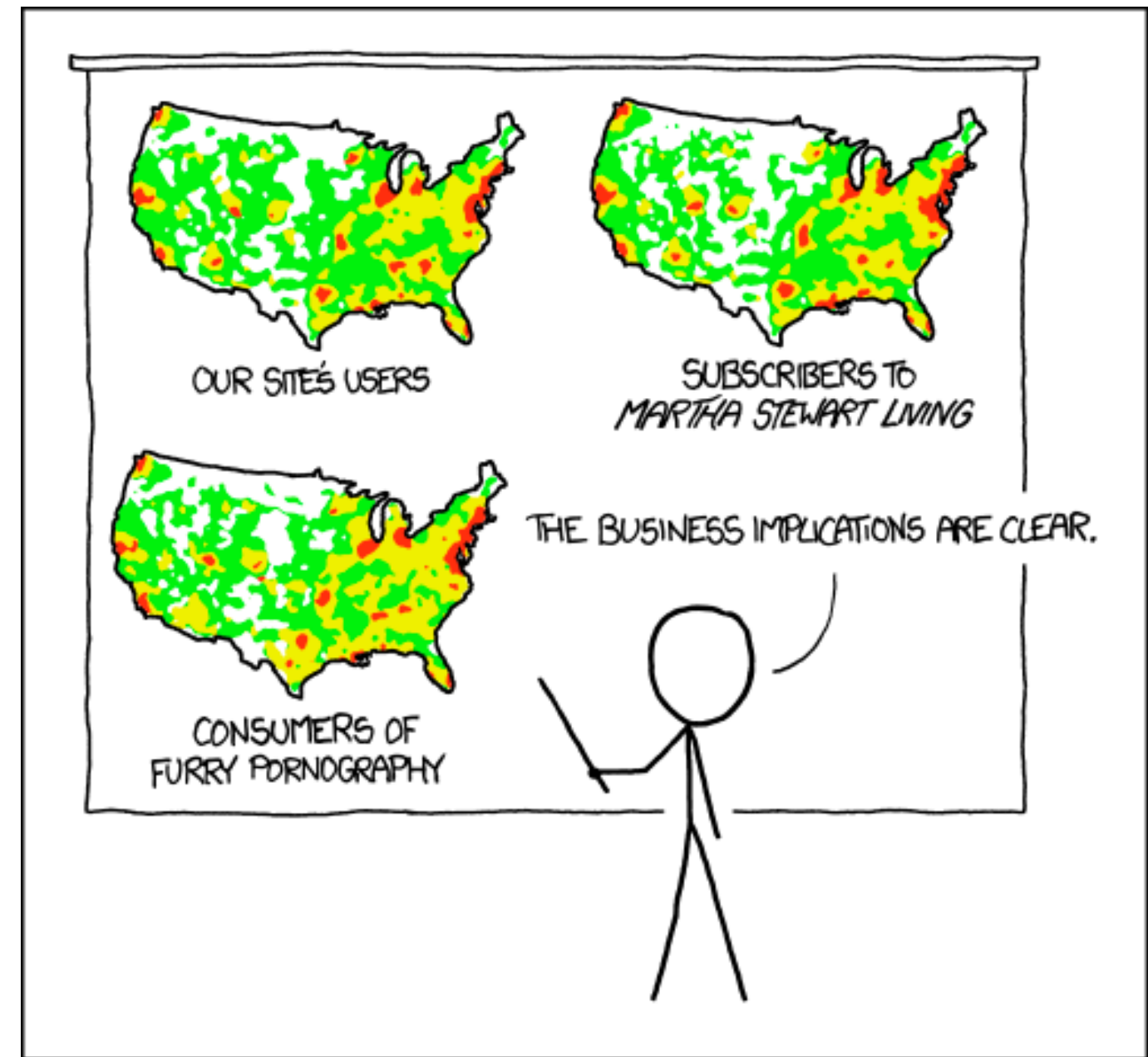


PET PEEVE #208:  
GEOGRAPHIC PROFILE MAPS WHICH ARE  
BASICALLY JUST POPULATION MAPS

[ <https://xkcd.com/1138> ]

# Beware: Population maps trickiness!

- spurious correlations: most attributes just show where people live

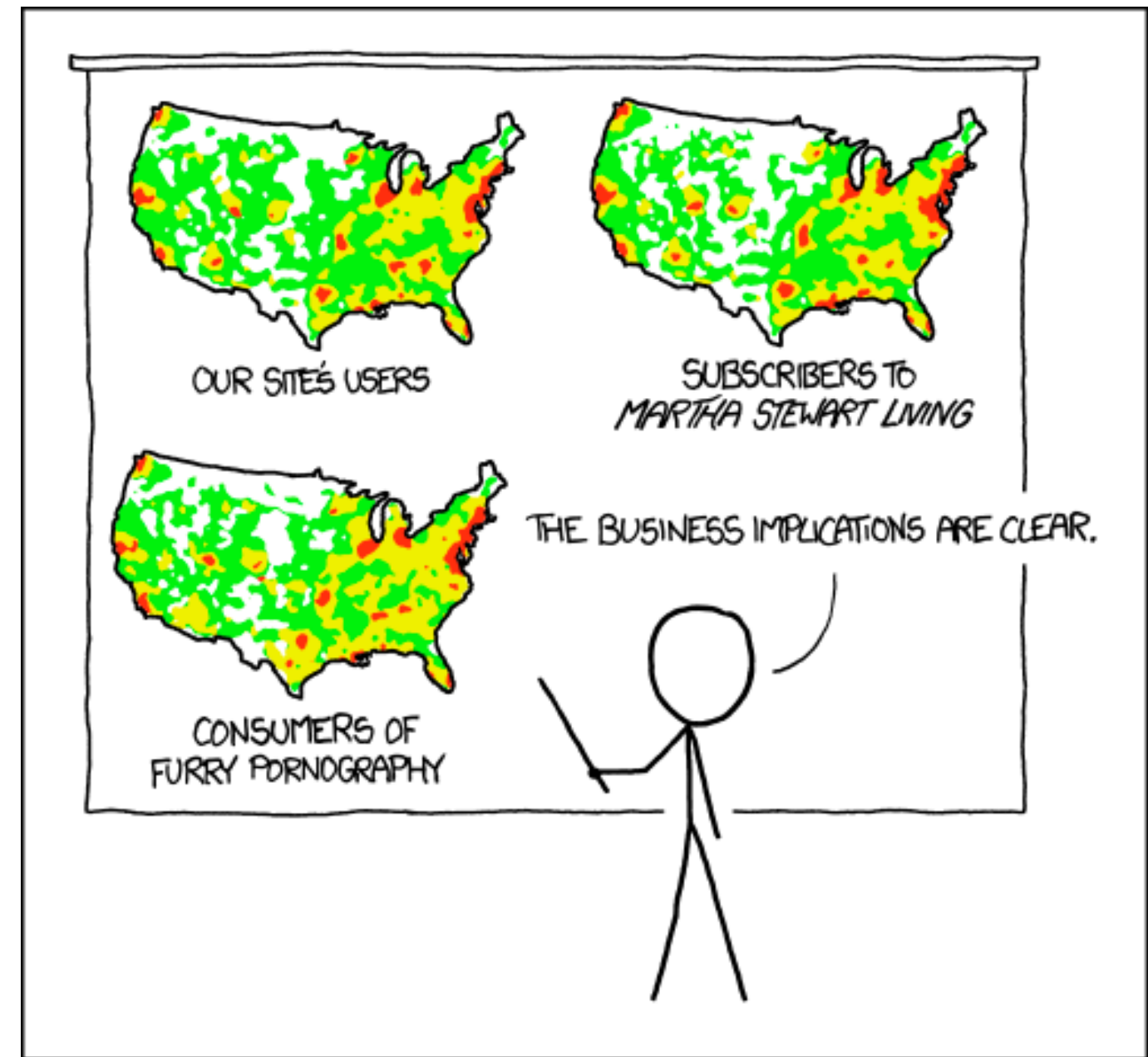


PET PEEVE #208:  
GEOGRAPHIC PROFILE MAPS WHICH ARE  
BASICALLY JUST POPULATION MAPS

[ <https://xkcd.com/1138> ]

# Beware: Population maps trickiness!

- spurious correlations: most attributes just show where people live
- consider when to normalize by population density
  - encode raw data values
    - tied to underlying population
  - but should use normalized values
    - unemployed people per 100 citizens, mean family income



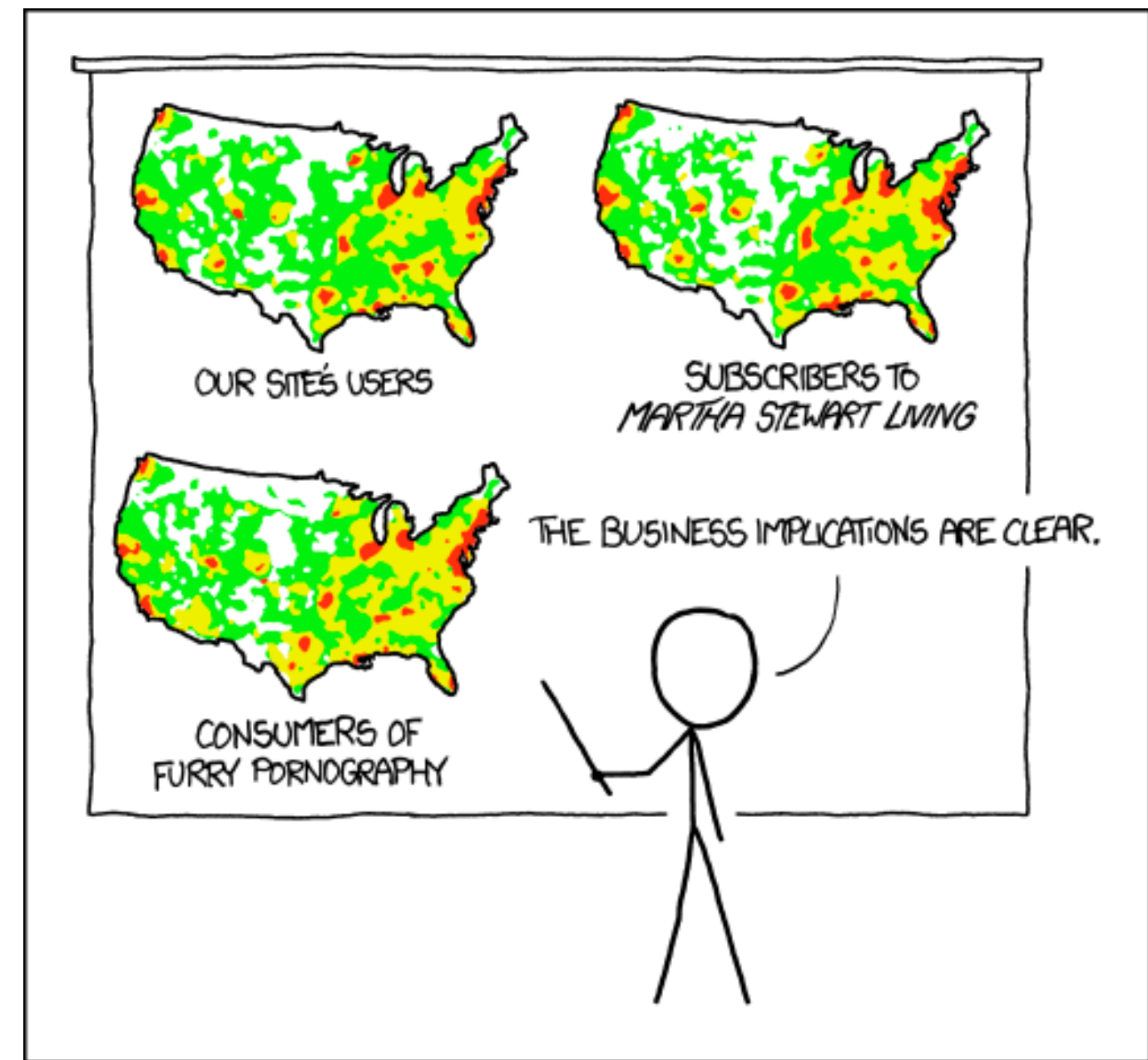
PET PEEVE #208:  
GEOGRAPHIC PROFILE MAPS WHICH ARE  
BASICALLY JUST POPULATION MAPS

[ <https://xkcd.com/1138> ]



# Beware: Population maps trickiness!

- spurious correlations: most attributes just show where people live
- consider when to normalize by population density
  - encode raw data values
    - tied to underlying population
  - but should use normalized values
    - unemployed people per 100 citizens, mean family income
- general issue
  - absolute counts vs relative/normalized data
  - failure to normalize is common error



PET PEEVE #208:  
GEOGRAPHIC PROFILE MAPS WHICH ARE  
BASICALLY JUST POPULATION MAPS

[ <https://xkcd.com/1138> ]

# Choropleth maps: Recommendations

- only use when central task is understanding spatial relationships
- show only one variable at a time
- normalize when appropriate
- be careful when choosing colors & bins
- best case: regions are roughly equal sized

# Choropleth map: Pros & cons

- pros

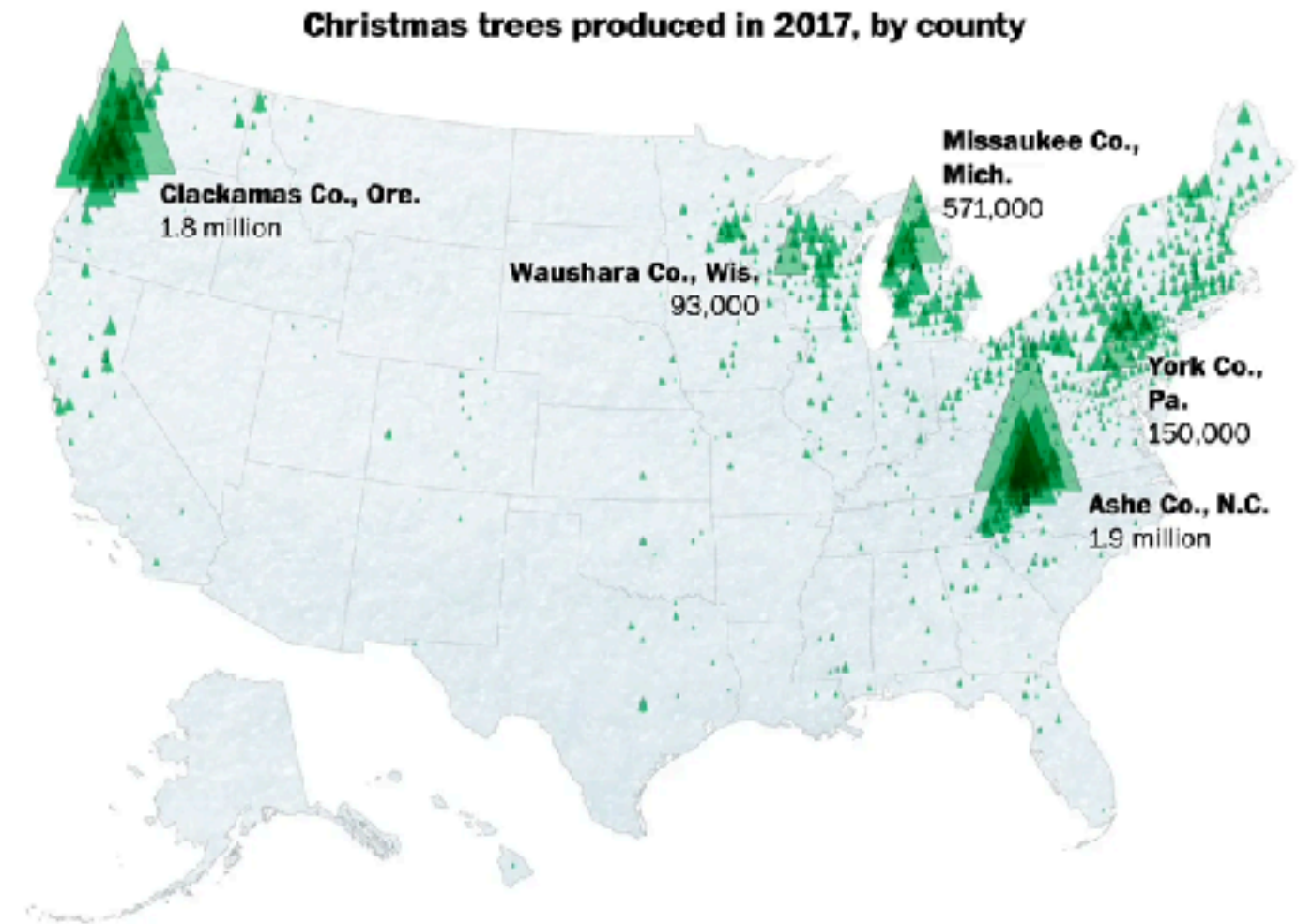
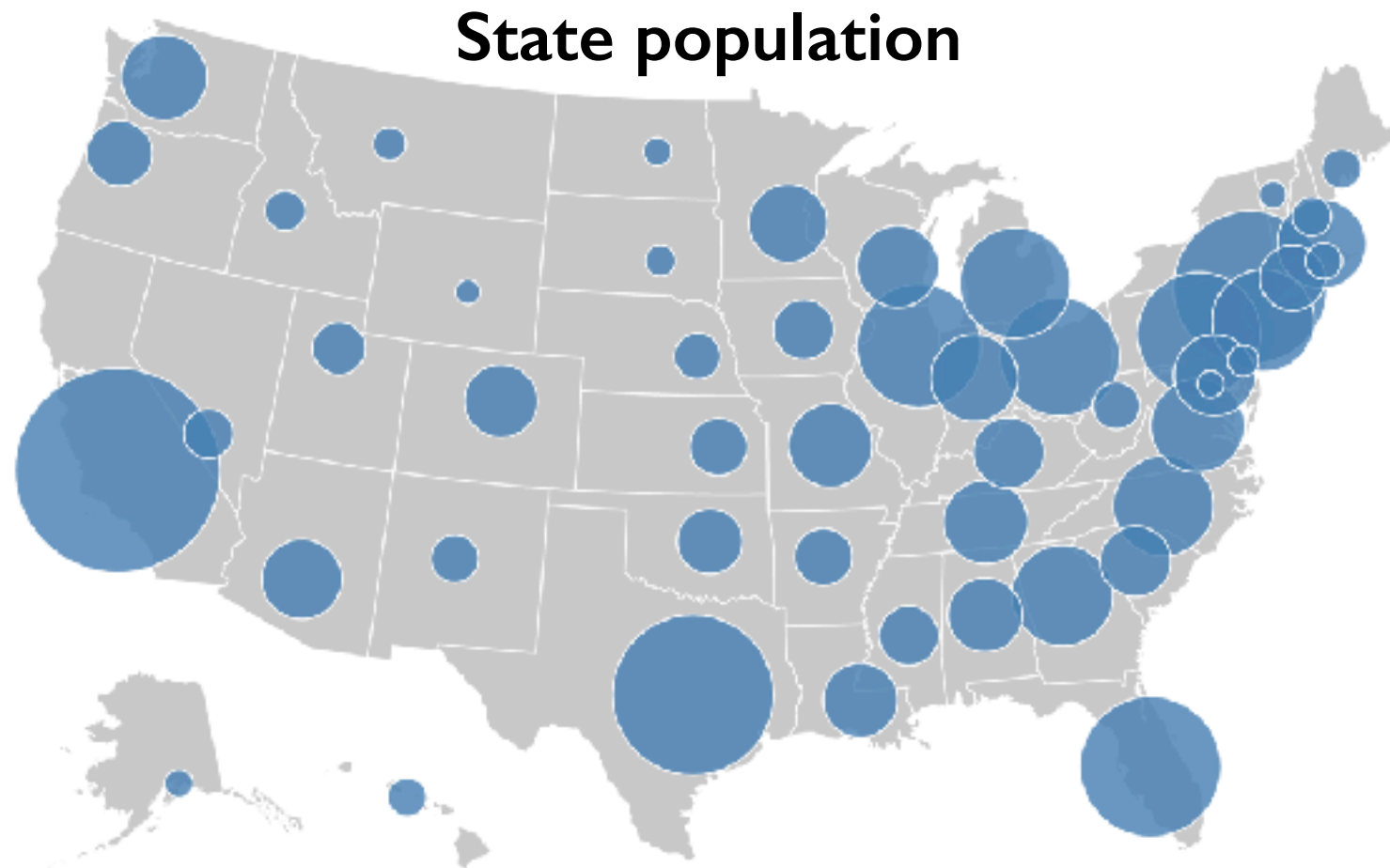
- easy to read and understand
- well established visualization (no learning curve)
- data is often collected and aggregated by geographical regions

- cons

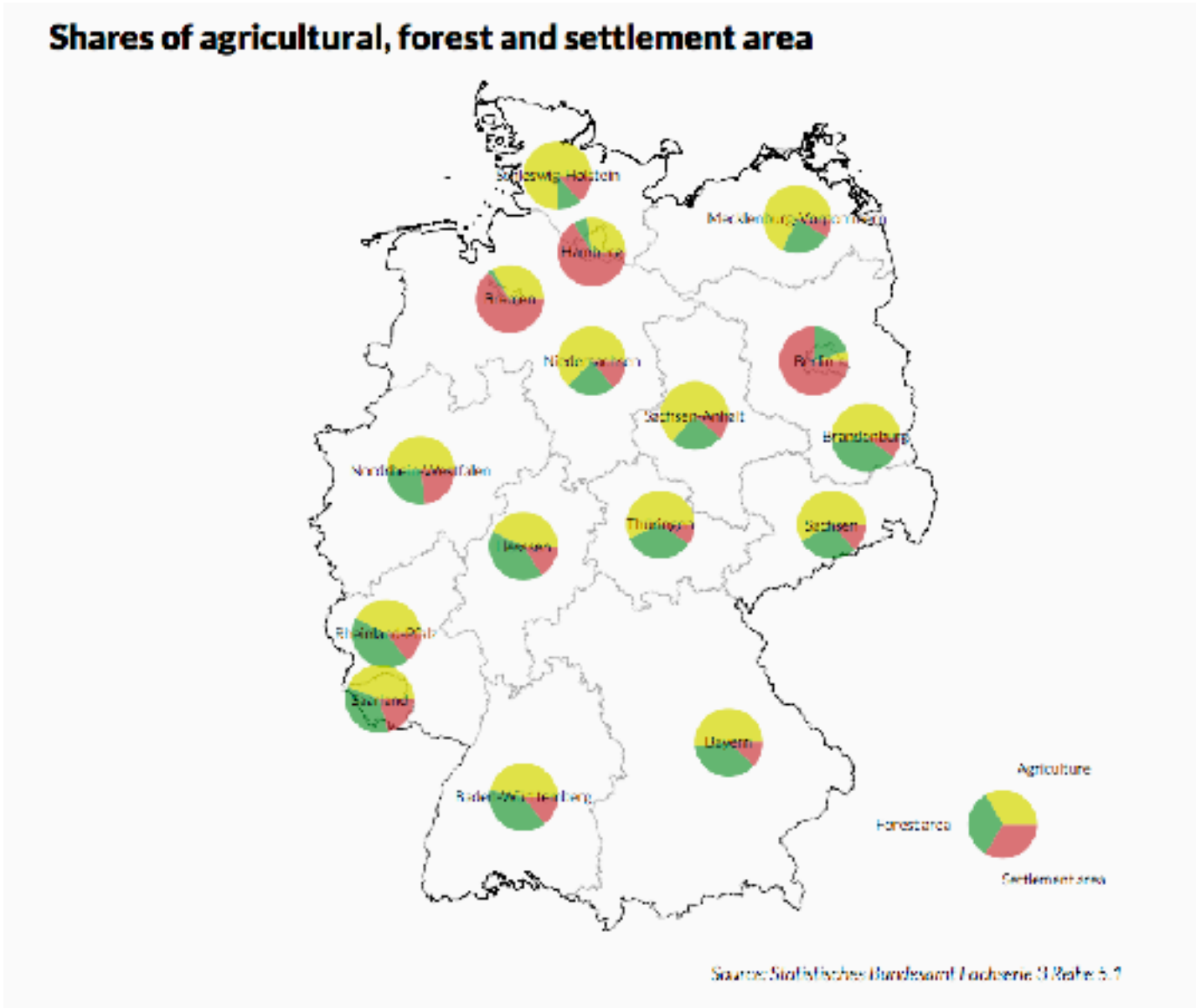
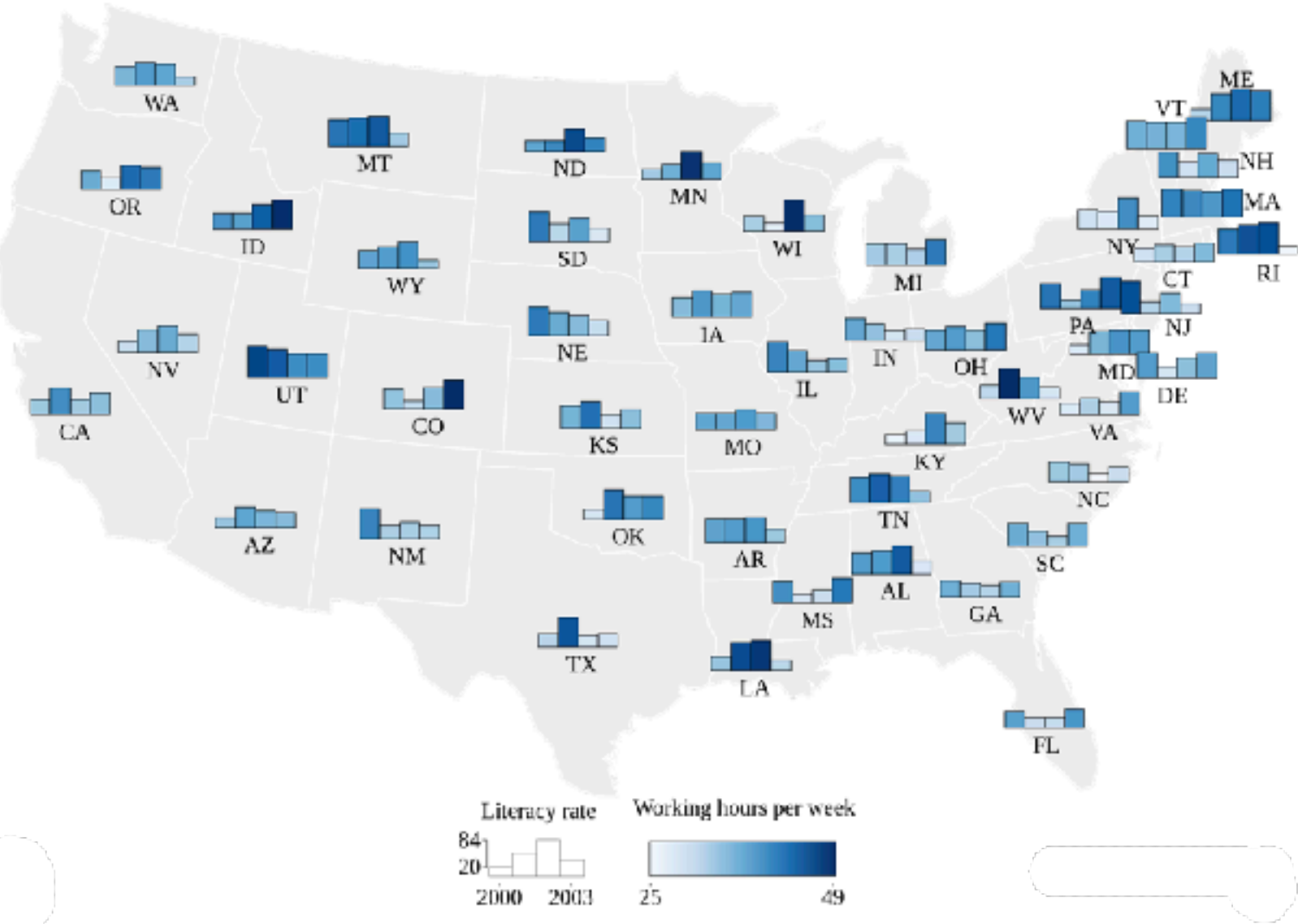
- most effective visual variable used for geographic location
- visual salience depends on region size, not true importance wrt attribute value
  - large regions appear more important than small ones
- color palette choice has a huge influence on the result

# Idiom: **Symbol maps**

- symbol is used to represent aggregated data (mark or glyph)
  - allows use of size and shape and color channels
    - aka proportional symbol maps, graduated symbol maps
- keep original spatial geometry in the background
- often a good alternative to choropleth maps



# Symbol maps with glyphs



# Symbol map: Pros & cons

- pros

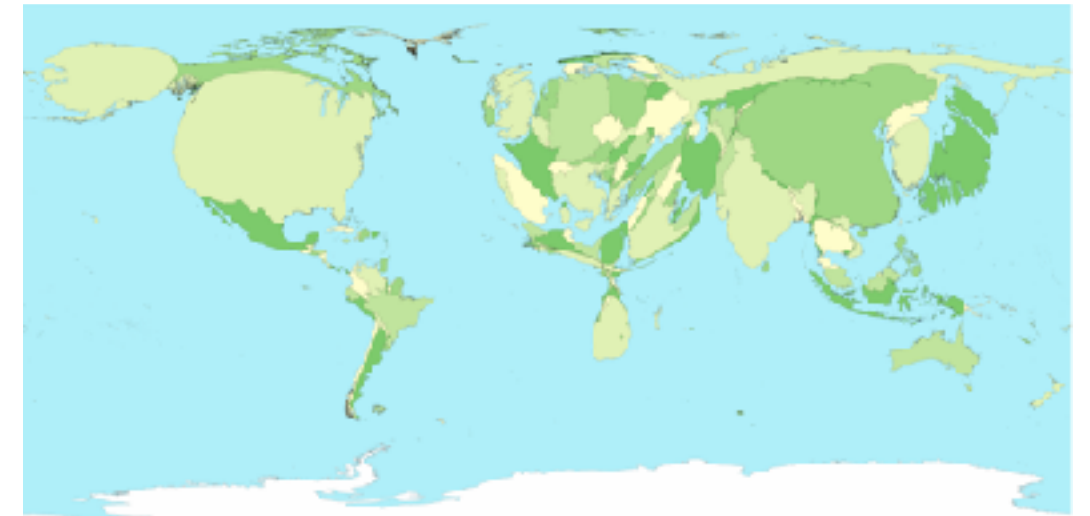
- somewhat intuitive to read and understand
- mitigate problems with region size vs data salience
  - marks: symbol size follows attribute value
  - glyphs: symbol size can be uniform

- cons

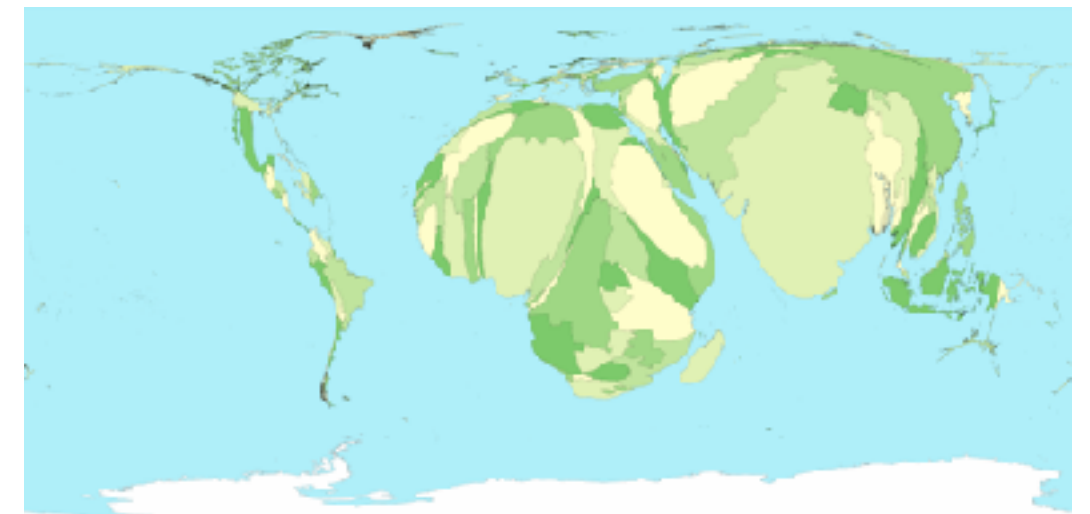
- possible occlusion / overlap
  - symbols could overlap each other
  - symbols could occlude region boundaries
- complex glyphs may require explanation / training

# Idiom: **Contiguous cartogram**

- interlocking marks:  
shape, area, and position coded
- derive new interlocking marks
  - based on combination of original interlocking marks and new quantitative attribute
- algorithm to create new marks
  - input: target size
  - goal: shape as close to the original as possible
  - requirement: maintain constraints
    - relative position
    - contiguous boundaries with their neighbours

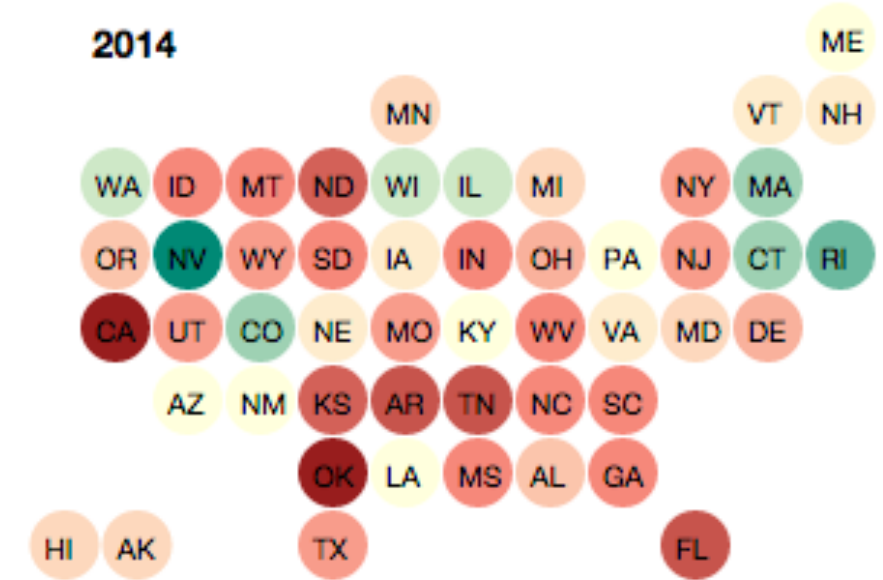
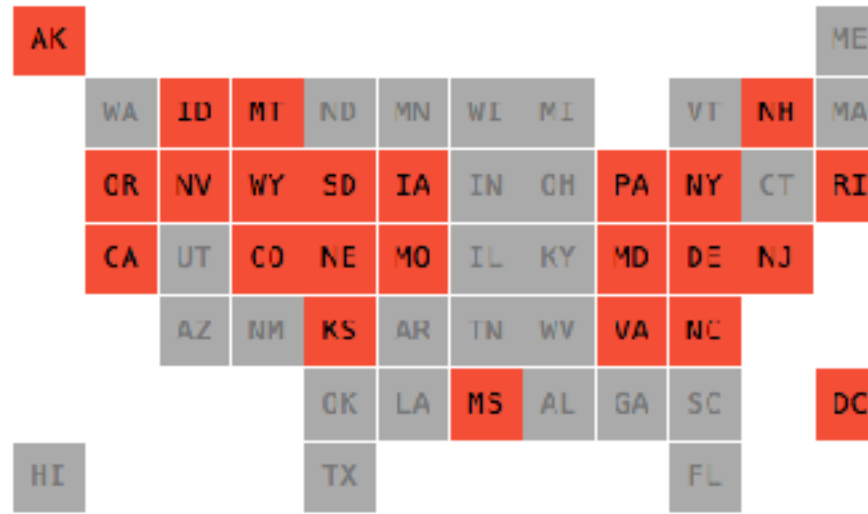
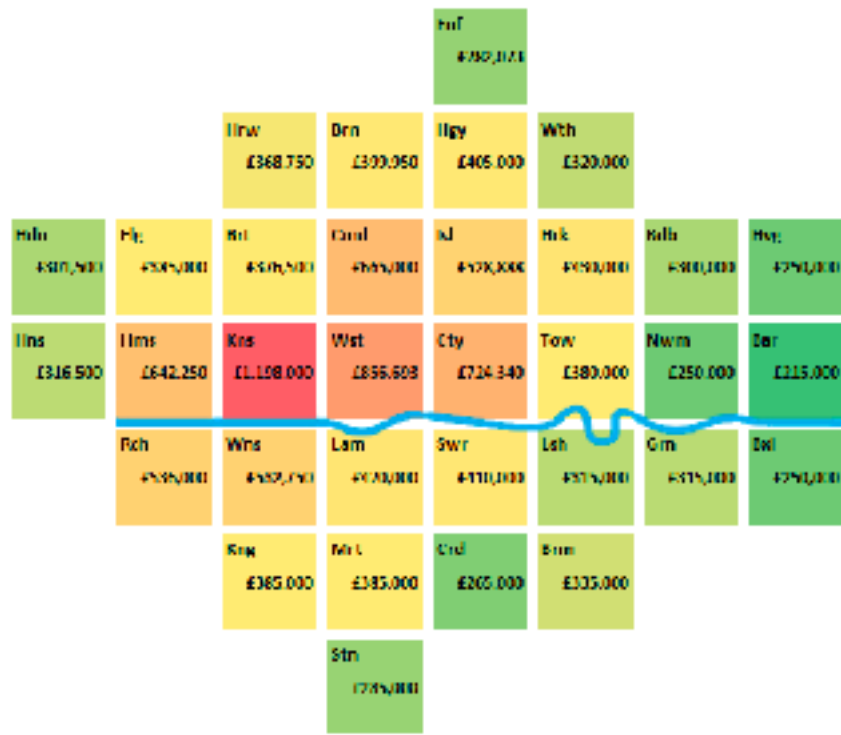


Greenhouse Emissions



Child Mortality

# Idiom: Grid Cartogram



- uniform-sized shapes arranged in rectilinear grid
- maintain approximate spatial position and arrangement



# Cartogram: Pros & cons

- pros

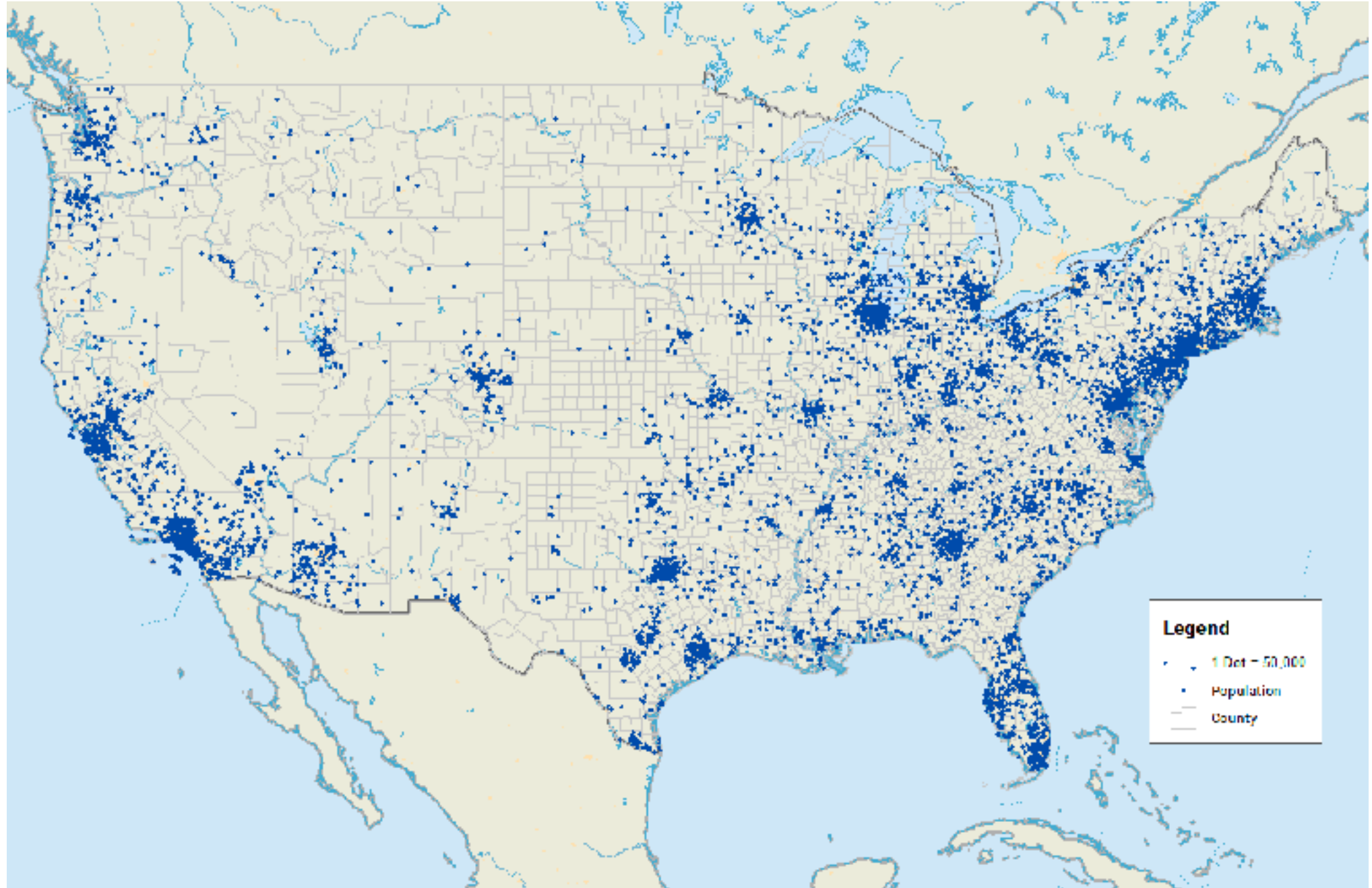
- can be intriguing and engaging
- best case: strong and surprising size disparities
- non-contiguous cartograms often easier to understand

- cons

- require substantial familiarity with original dataset & use of memory
  - compare distorted marks to memory of original marks
  - mitigation strategies: transitions or side by side views
- major distortion is problematic
  - may be aesthetically displeasing
  - may result in unrecognizable marks
- difficult to extract exact quantities

# Idiom: **Dot density maps**

- visualize distribution of a phenomenon by placing dots
- one symbol represents a constant number of items
  - dots have uniform size & shape
  - allows use of color channel
- task:  
show spatial patterns, clusters



# Dot density maps: Pros and cons

- pros

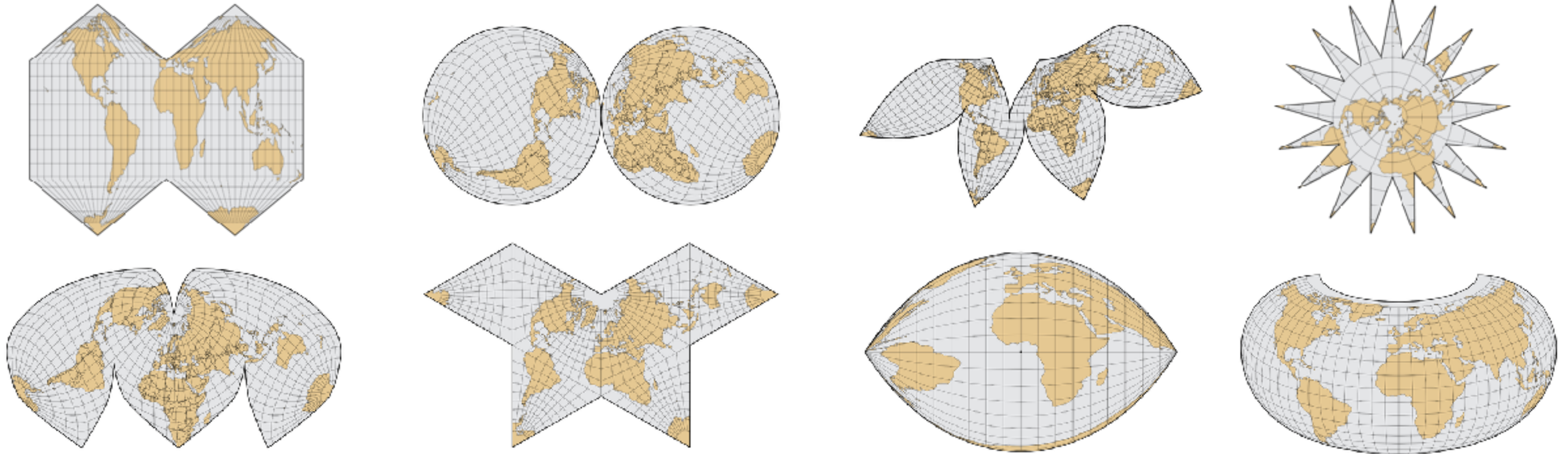
- straightforward to understand
- avoids choropleth non-uniform region size problems

- cons

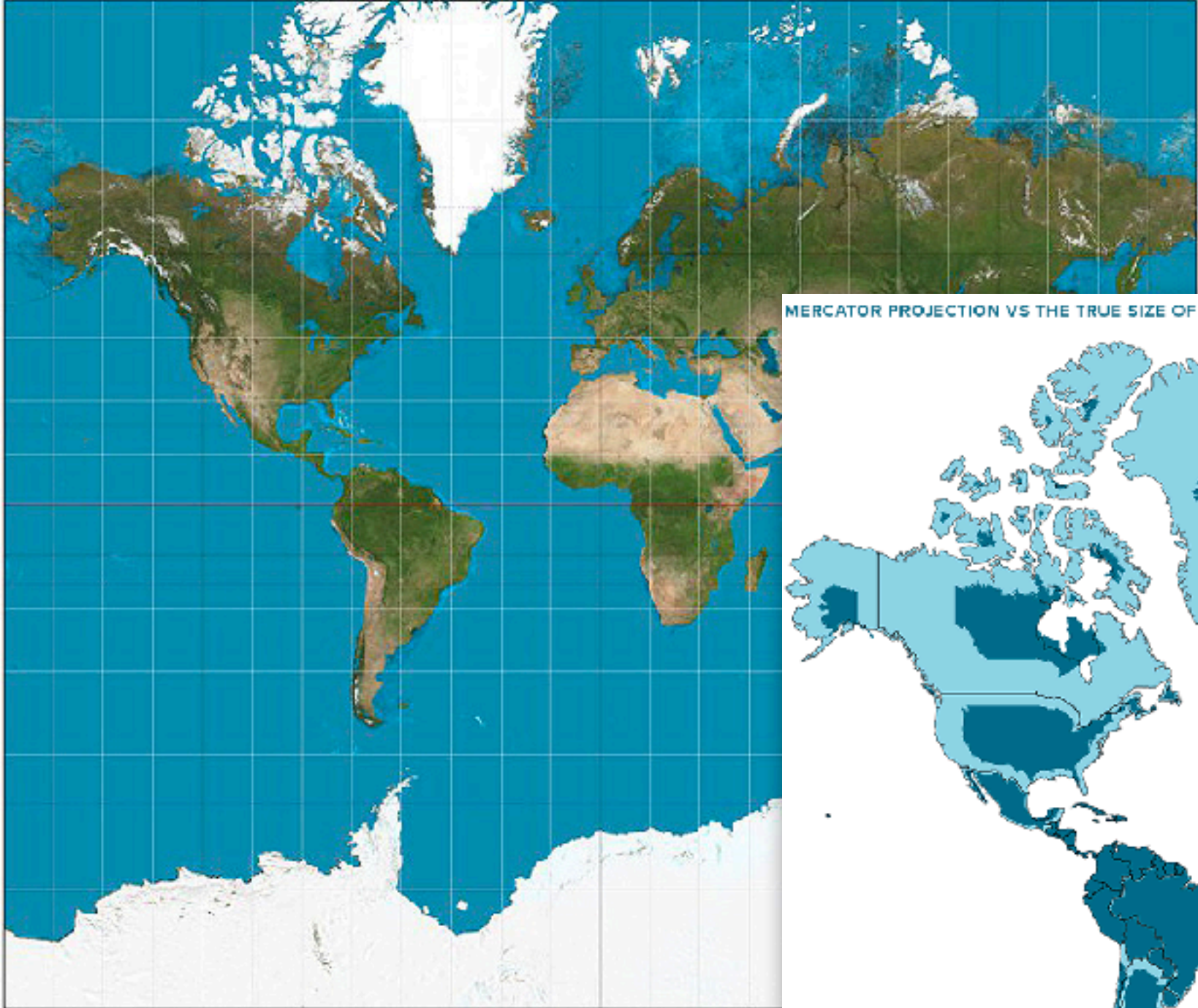
- challenge: normalization, just like choropleths
  - show population density (correlated with attribute), not effect of interest
- perceptual disadvantage:  
difficult to extract quantities
- performance disadvantage:  
rendering many dots can be slow

# Map Projections

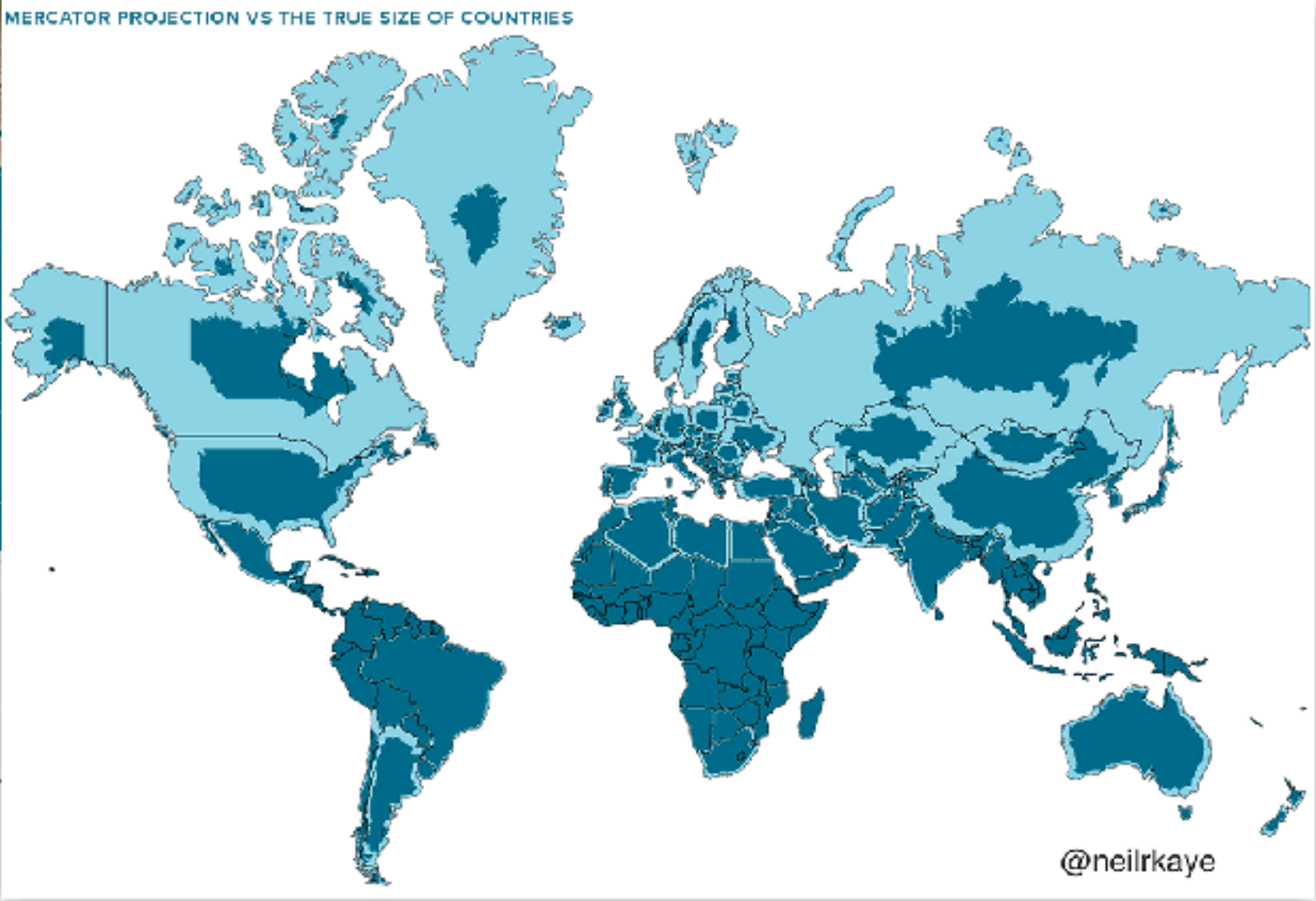
- mathematical functions that map 3D surface geometry of the Earth to 2D maps
- all projections of sphere on plane necessarily distort surface in some way
- **interactive:** [philogb.github.io/page/myriahedral/](http://philogb.github.io/page/myriahedral/) and [jasondavies.com/maps/](http://jasondavies.com/maps/)



# Mercator Projection



» Heavily distorts country sizes; particularly close to the poles.



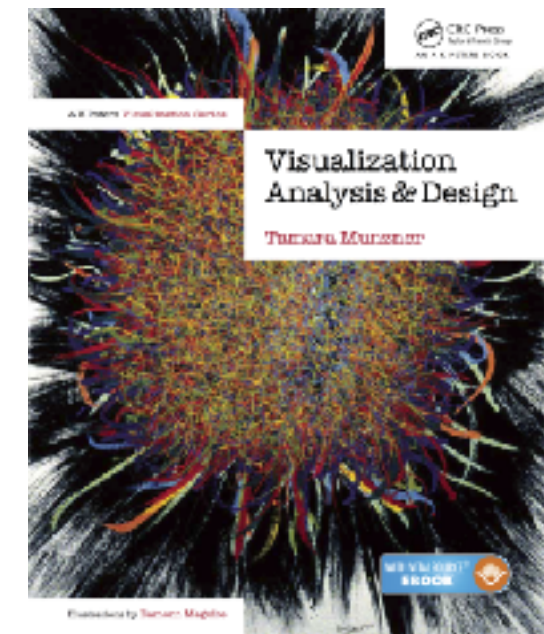
# Visualization Analysis & Design

## *Spatial Data (Ch 9) II*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

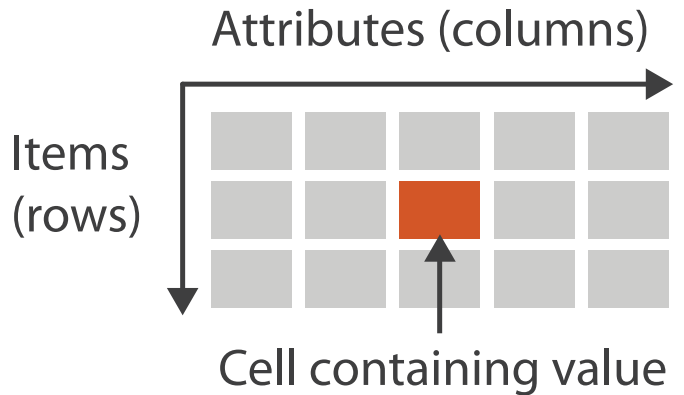
[@tamaramunzner](#)



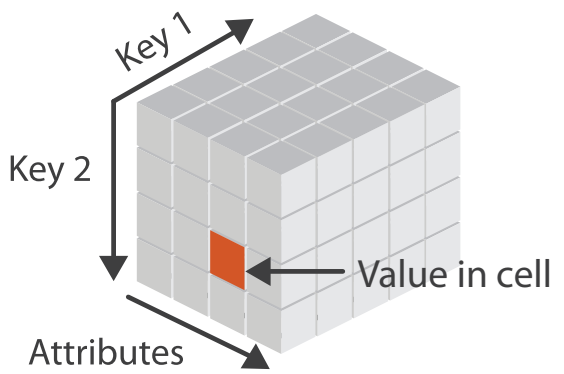
# Focus on Spatial

## → Dataset Types

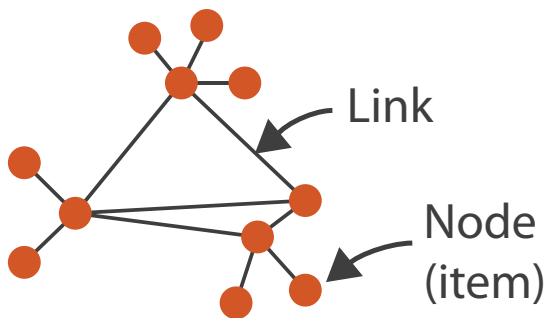
### → Tables



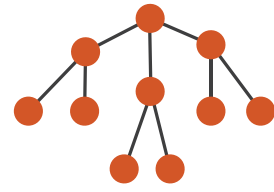
### → Multidimensional Table



### → Networks

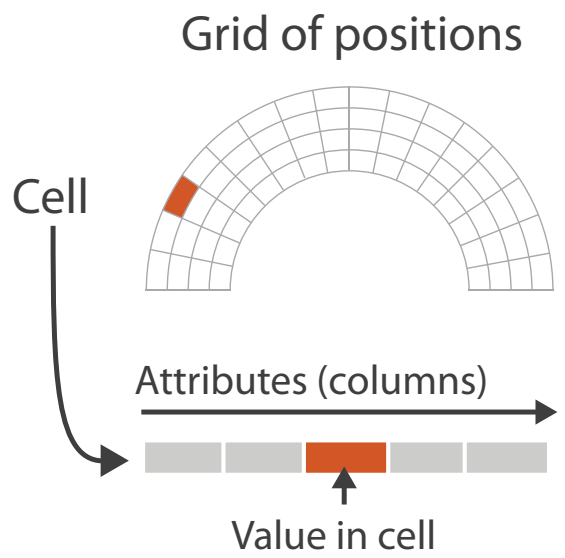


### → Trees

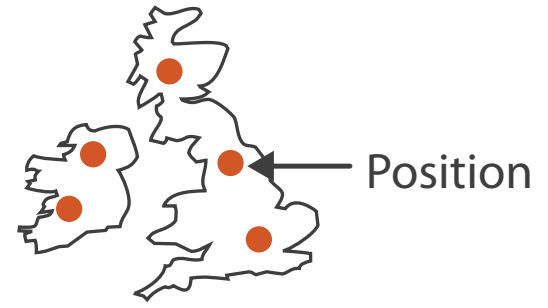


### → Spatial

#### → Fields (Continuous)



#### → Geometry (Spatial)

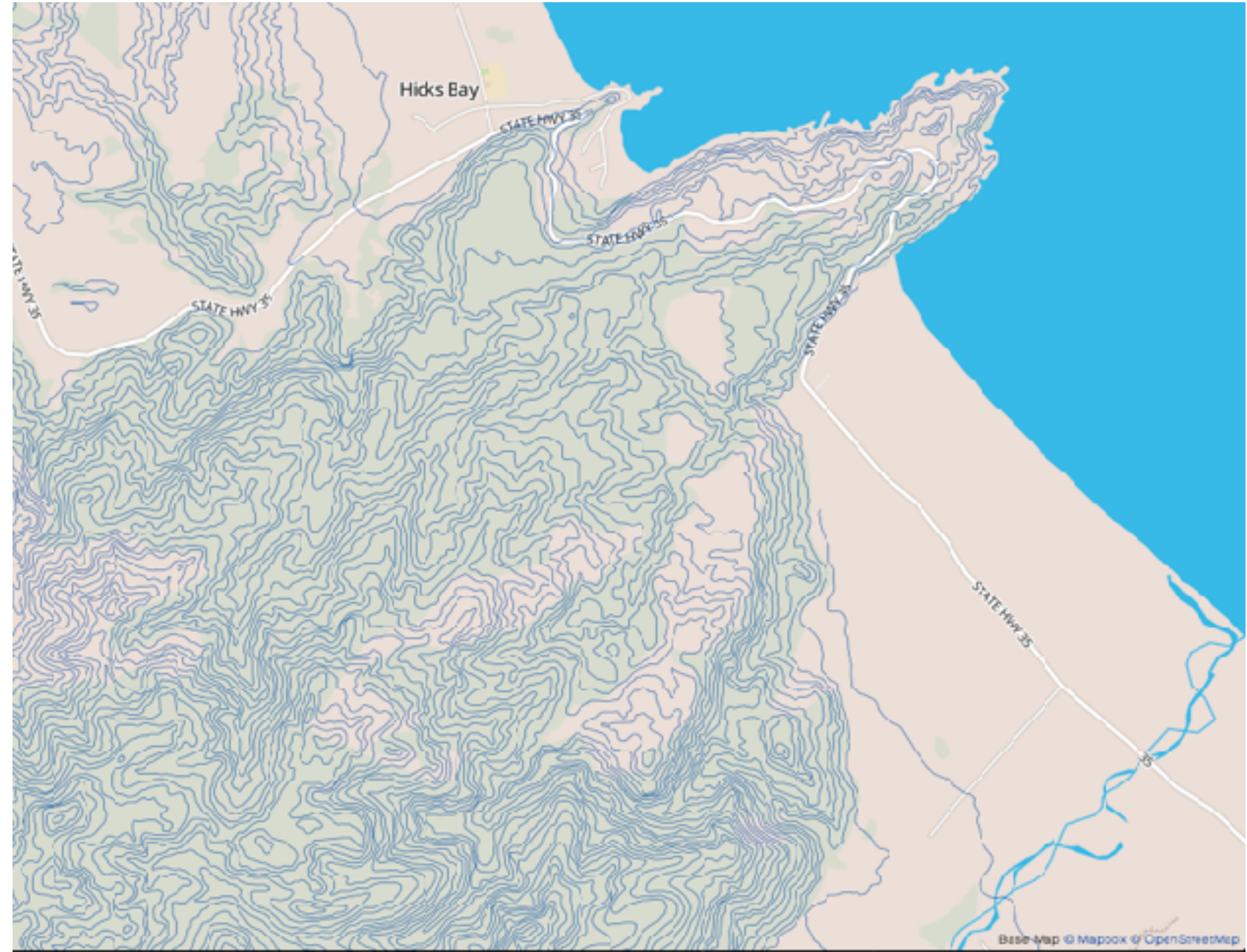


# Spatial Fields



# Idiom: **topographic map**

- data
  - geographic geometry
  - scalar spatial field
    - 1 quant attribute per grid cell
- derived data
  - isoline geometry
    - isocontours computed for specific levels of scalar values
- task
  - understanding terrain shape
    - densely lined regions = steep
- pros
  - use only 2D position, avoid 3D challenges
  - color channel available for other attributes
- cons
  - significant clutter from additional lines



Land Information New Zealand Data Service

# Idioms: **isosurfaces**, **direct volume rendering**

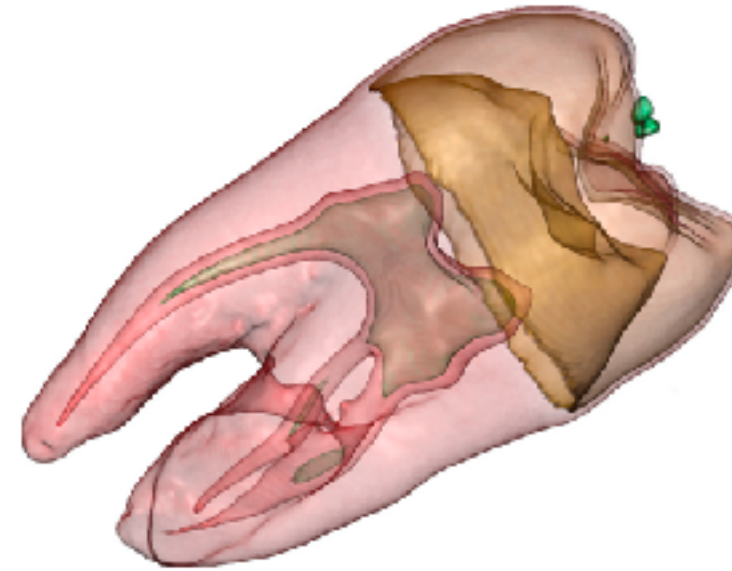
- **data**
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- **task**
  - shape understanding, spatial relationships

*[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]*

*[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]*

# Idioms: **isosurfaces**, **direct volume rendering**

- **data**
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- **task**
  - shape understanding, spatial relationships
- **isosurface**
  - derived data: isocontours computed for specific levels of scalar values

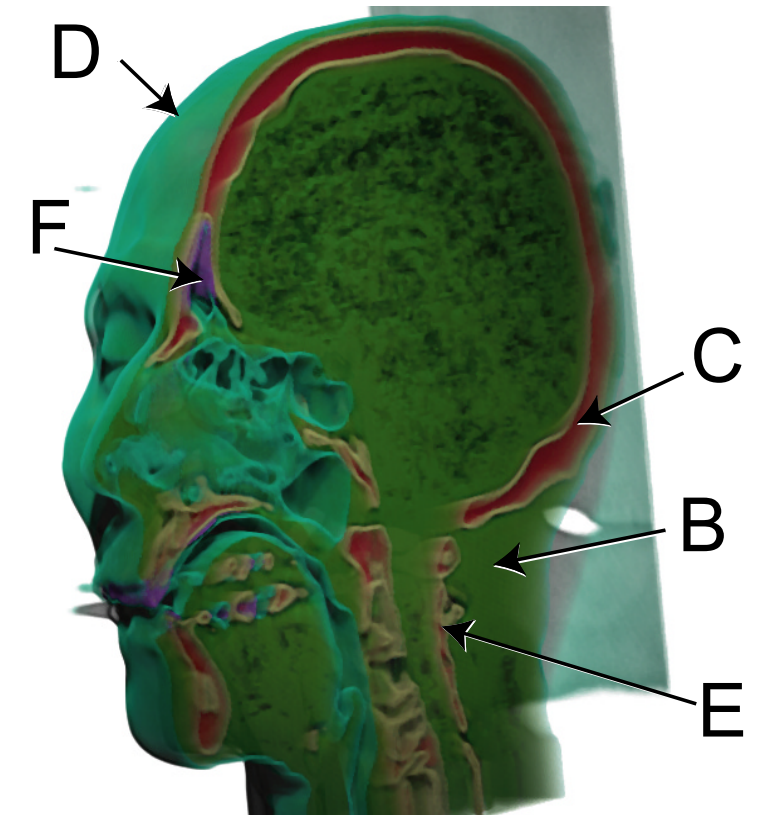
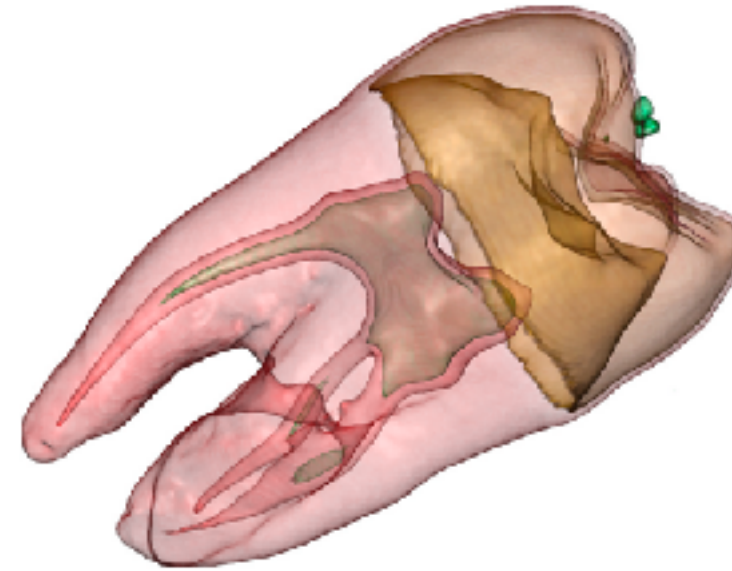


*[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]*

*[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]*

# Idioms: **isosurfaces**, **direct volume rendering**

- **data**
  - scalar spatial field (3D volume)
    - 1 quant attribute per grid cell
- **task**
  - shape understanding, spatial relationships
- **isosurface**
  - derived data: isocontours computed for specific levels of scalar values
- **direct volume rendering**
  - transfer function maps scalar values to color, opacity
    - no derived geometry

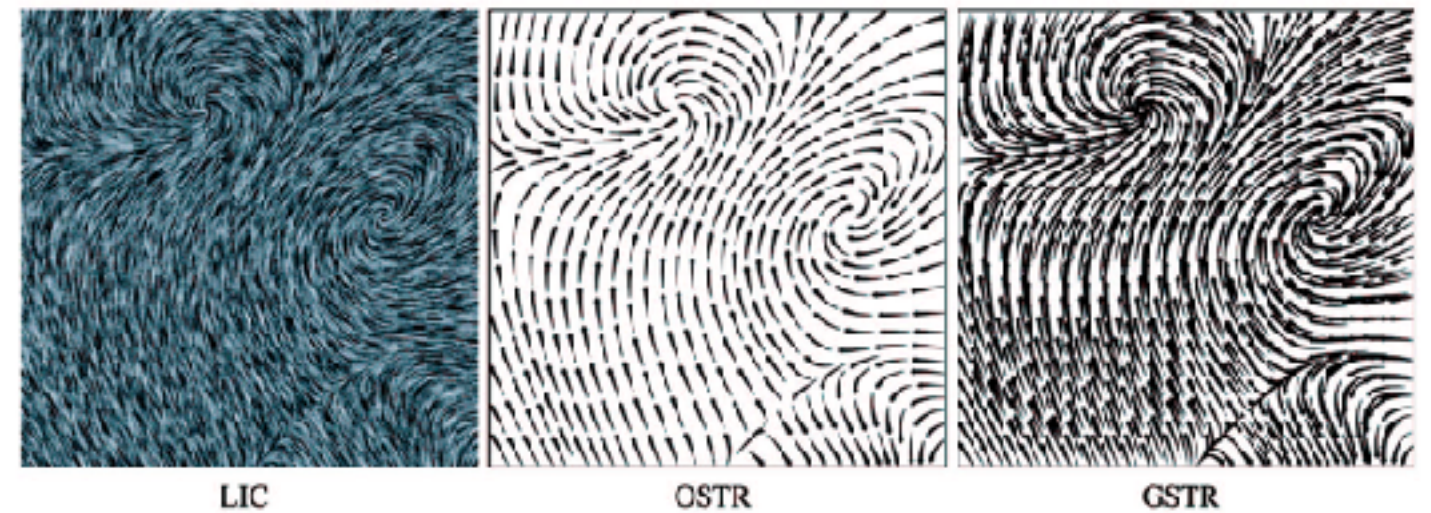
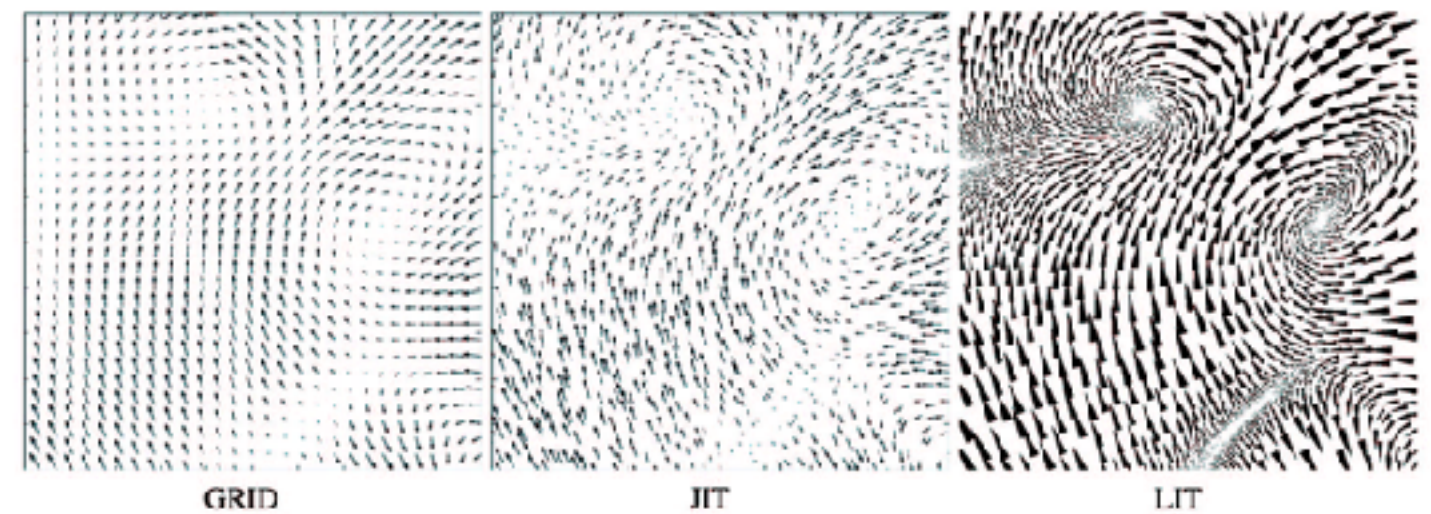


[Interactive Volume Rendering Techniques. Kniss. Master's thesis, University of Utah Computer Science, 2002.]

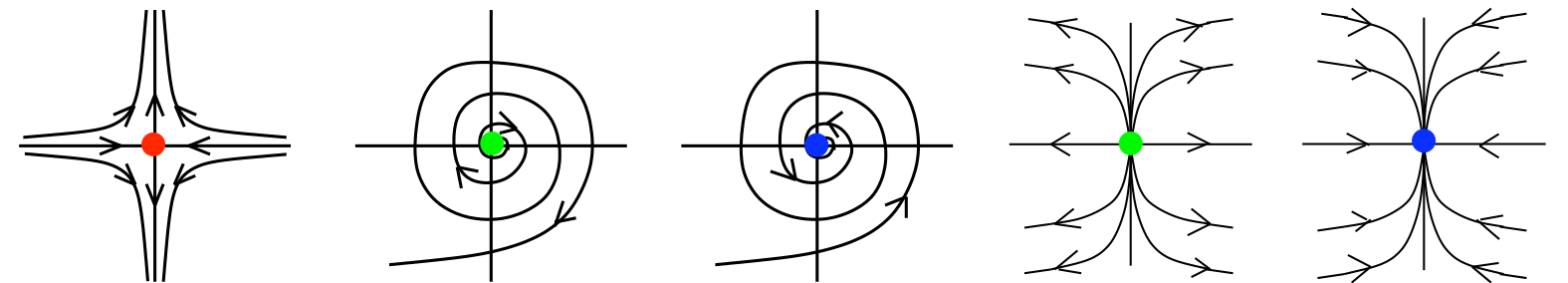
[Multidimensional Transfer Functions for Volume Rendering. Kniss, Kindlmann, and Hansen. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 189–210. Elsevier, 2005.]

# Vector and tensor fields

- data
  - multiple attribs per cell (vector: 2)
- idiom families
  - flow *glyphs*
    - purely local
  - *geometric flow*
    - derived data from tracing particle trajectories
    - sparse set of seed points
  - *texture flow*
    - derived data, dense seeds
  - *feature flow*
    - global computation to detect features



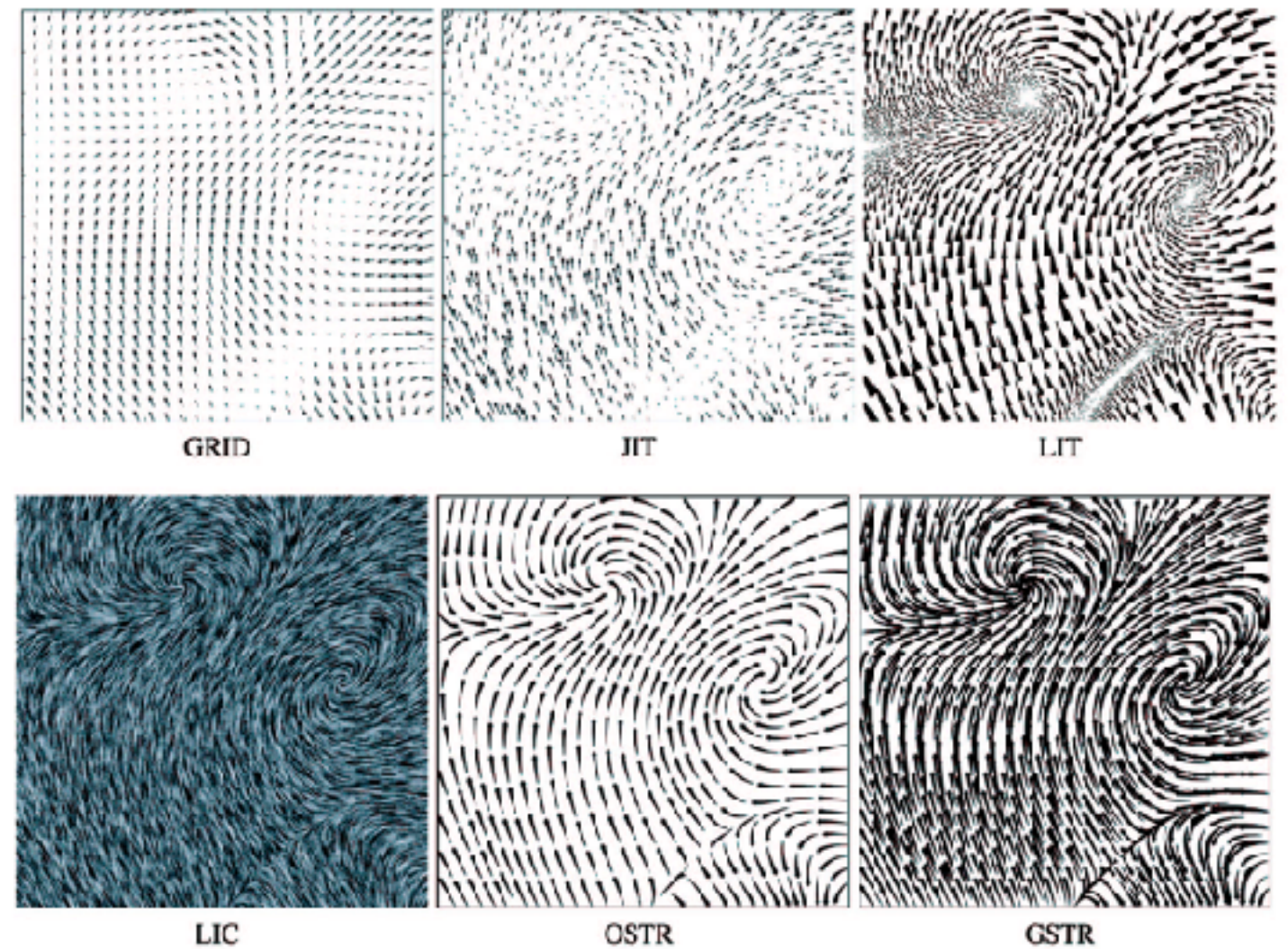
[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. IEEE Trans. Visualization and Computer Graphics (TVCG) 11:1 (2005), 59–70.]



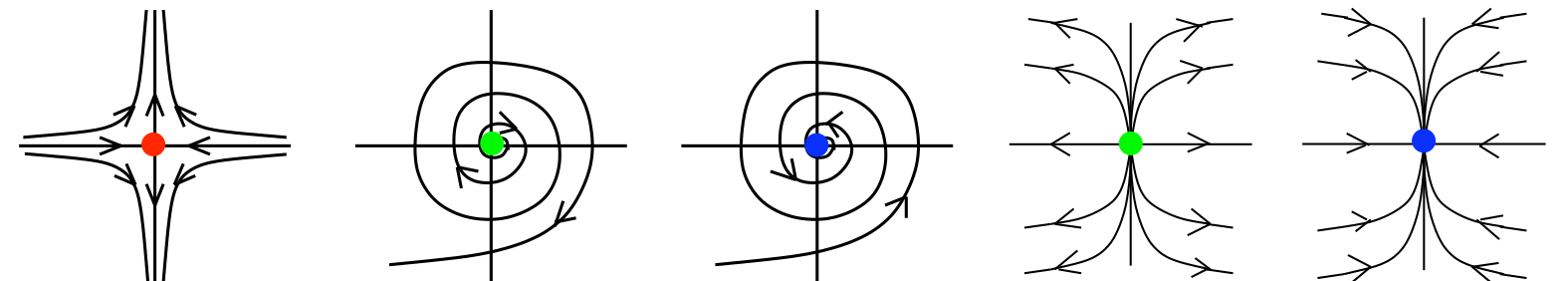
[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. Computers & Graphics 26:2 (2002), 249–257.]

# Vector fields

- empirical study tasks
  - finding critical points, identifying their types
  - identifying what type of critical point is at a specific location
  - predicting where a particle starting at a specified point will end up (advection)



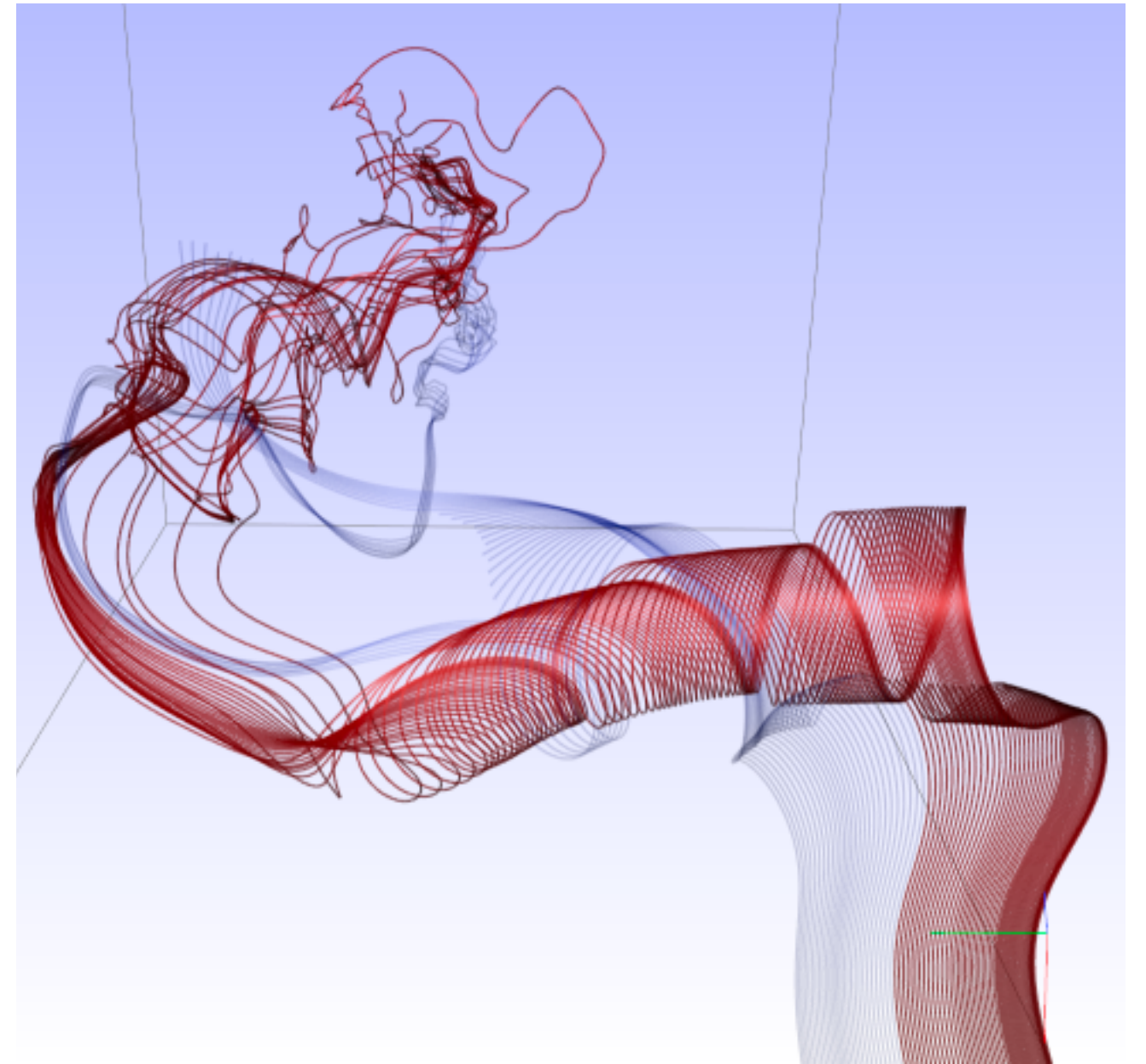
[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. *IEEE Trans. Visualization and Computer Graphics (TVCG)* 11:1 (2005), 59–70.]



[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. *Computers & Graphics* 26:2 (2002), 249–257.]

# Idiom: **similarity-clustered streamlines**

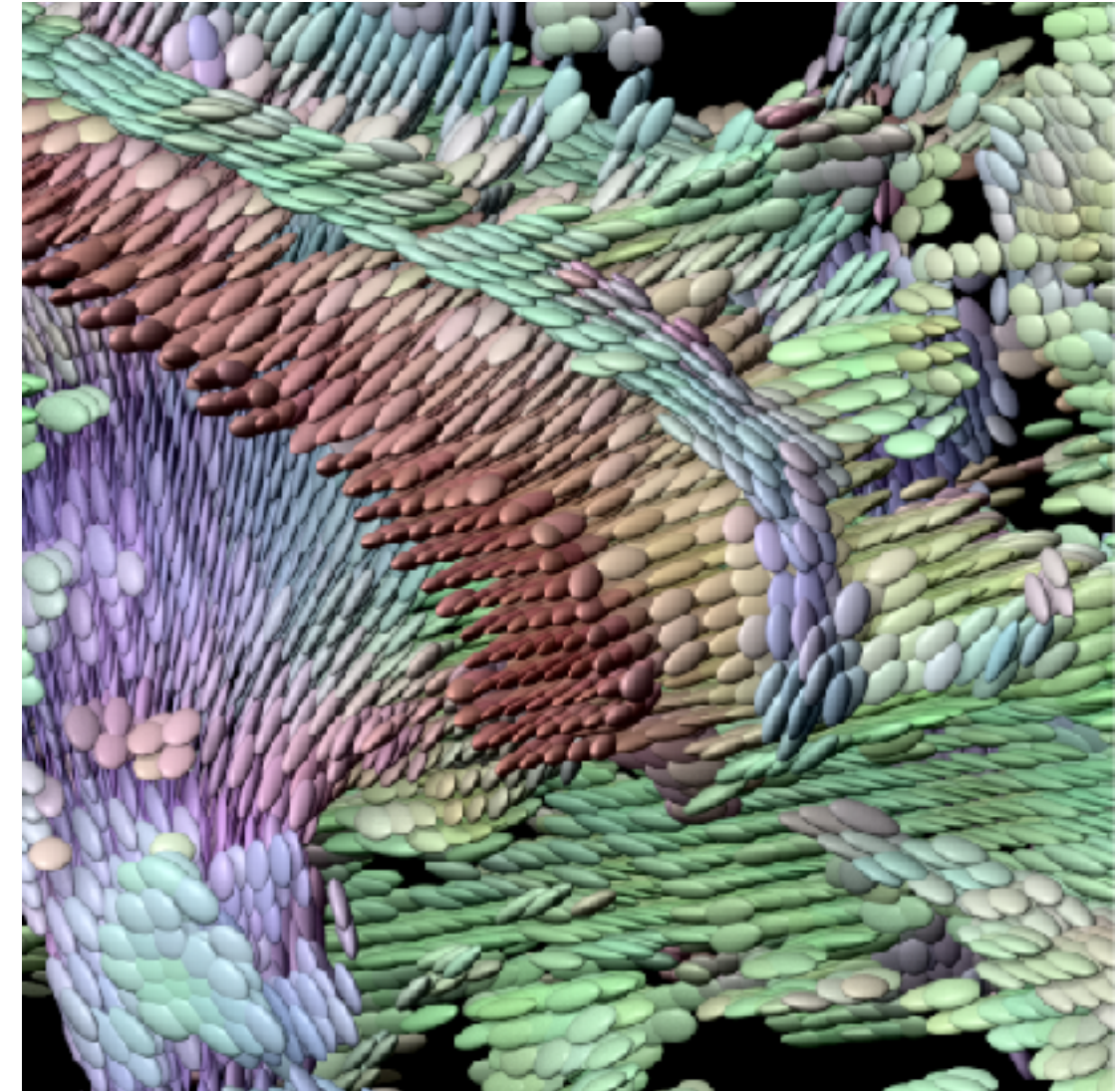
- data
  - 3D vector field
- derived data (from field)
  - streamlines: trajectory particle will follow
- derived data (per streamline)
  - curvature, torsion, tortuosity
  - signature: complex weighted combination
  - compute cluster hierarchy across all signatures
  - encode: color and opacity by cluster
- tasks
  - find features, query shape
- scalability
  - millions of samples, hundreds of streamlines



*[Similarity Measures for Enhancing Interactive Streamline Seeding. McLoughlin, Jones, Laramee, Malki, Masters, and Hansen. IEEE Trans. Visualization and Computer Graphics 19:8 (2013), 1342–1353.]*

# Idiom: Ellipsoid Tensor Glyphs

- data
  - tensor field: multiple attributes at each cell (entire matrix)
    - stress, conductivity, curvature, diffusivity...
  - derived data:
    - shape (eigenvalues)
    - orientation (eigenvectors)
- visual encoding
  - glyph: 3D ellipsoid



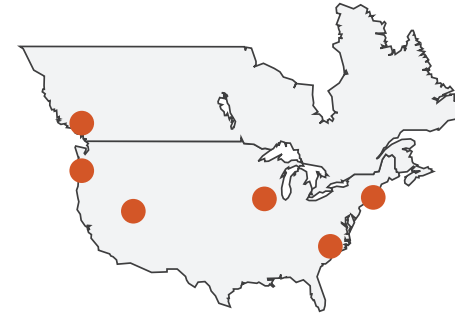


# Arrange spatial data

## → Use Given

### → Geometry

→ *Geographic*

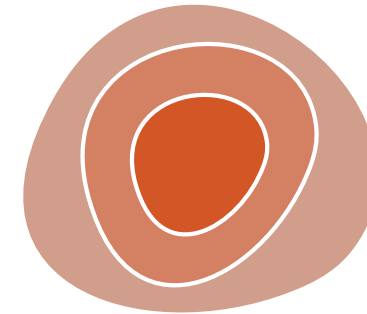


### → Spatial Fields

→ *Scalar Fields (one value per cell)*

→ *Isocontours*

→ *Direct Volume Rendering*



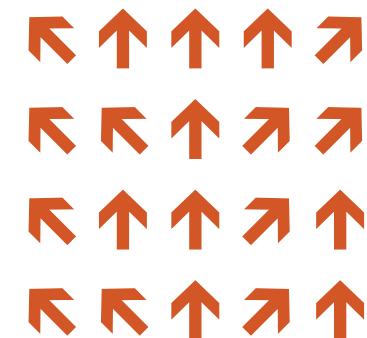
→ *Vector and Tensor Fields (many values per cell)*

→ *Flow Glyphs (local)*

→ *Geometric (sparse seeds)*

→ *Textures (dense seeds)*

→ *Features (globally derived)*



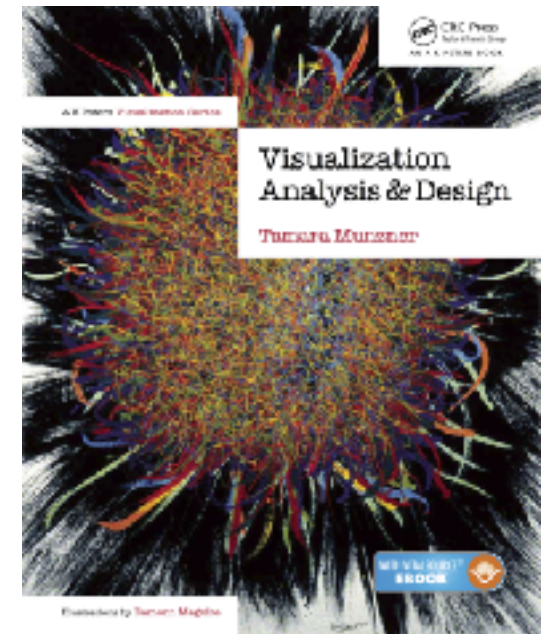
# Visualization Analysis & Design

## *Color (Ch 10)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# Idiom design choices: Visual encoding

## 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, ...*



What?

Why?

How?

# Idiom design choices: Beyond spatial arrangement

## 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, ...*



What?

Why?

How?

# Channels: What's up with color?

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale 

Position on unaligned scale 

Length (1D size) 

Tilt/angle 

Area (2D size) 

Depth (3D position) 

 Color luminance



 Color saturation



Curvature 


Volume (3D size) 

Same Same

## ➔ Identity Channels: Categorical Attributes

Spatial region 

Color hue 

Motion 

Shape 

Best  
Effectiveness  
Least

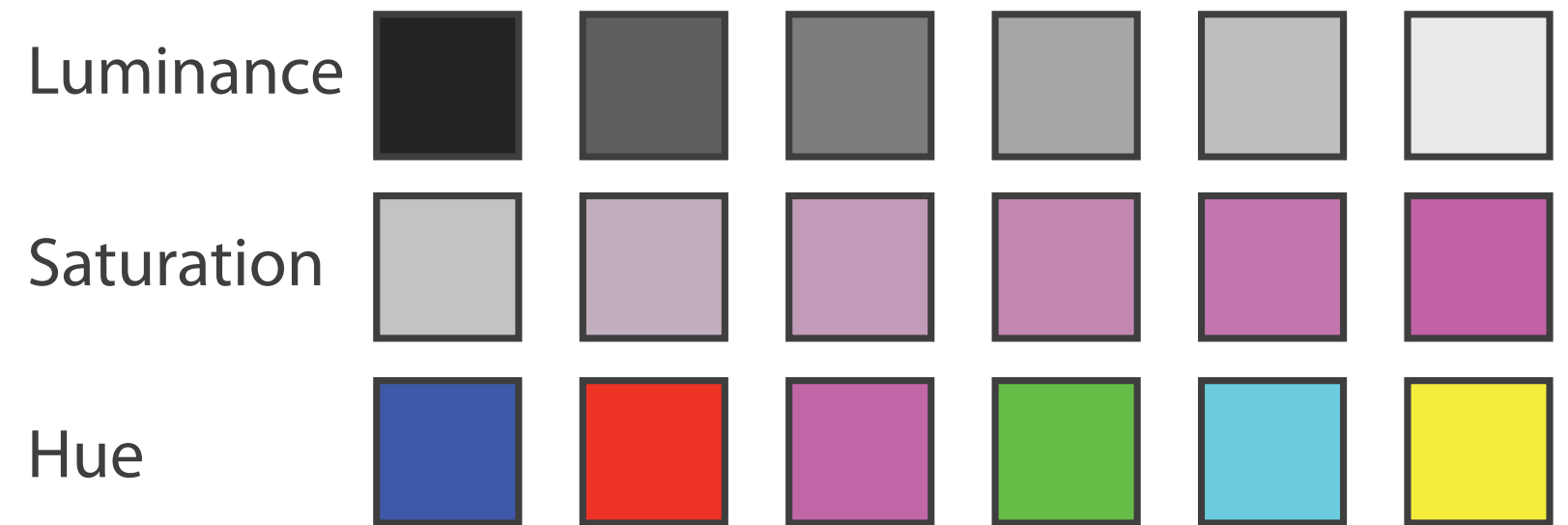
# Decomposing color

# Decomposing color

- first rule of color: do not (just) talk about color!
  - color is confusing if treated as monolithic

# Decomposing color

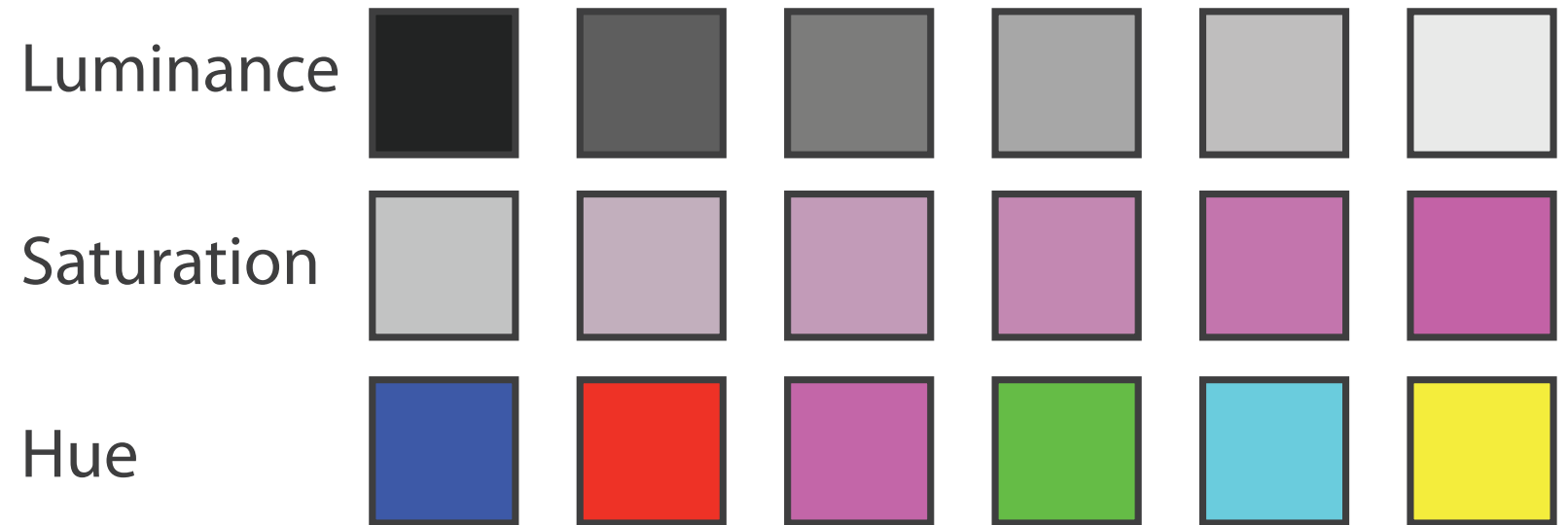
- first rule of color: do not (just) talk about color!
  - color is confusing if treated as monolithic
- decompose into three channels
  - ordered can show magnitude
    - **luminance**: how bright (B/W)
    - **saturation**: how colourful
  - categorical can show identity
    - **hue**: what color





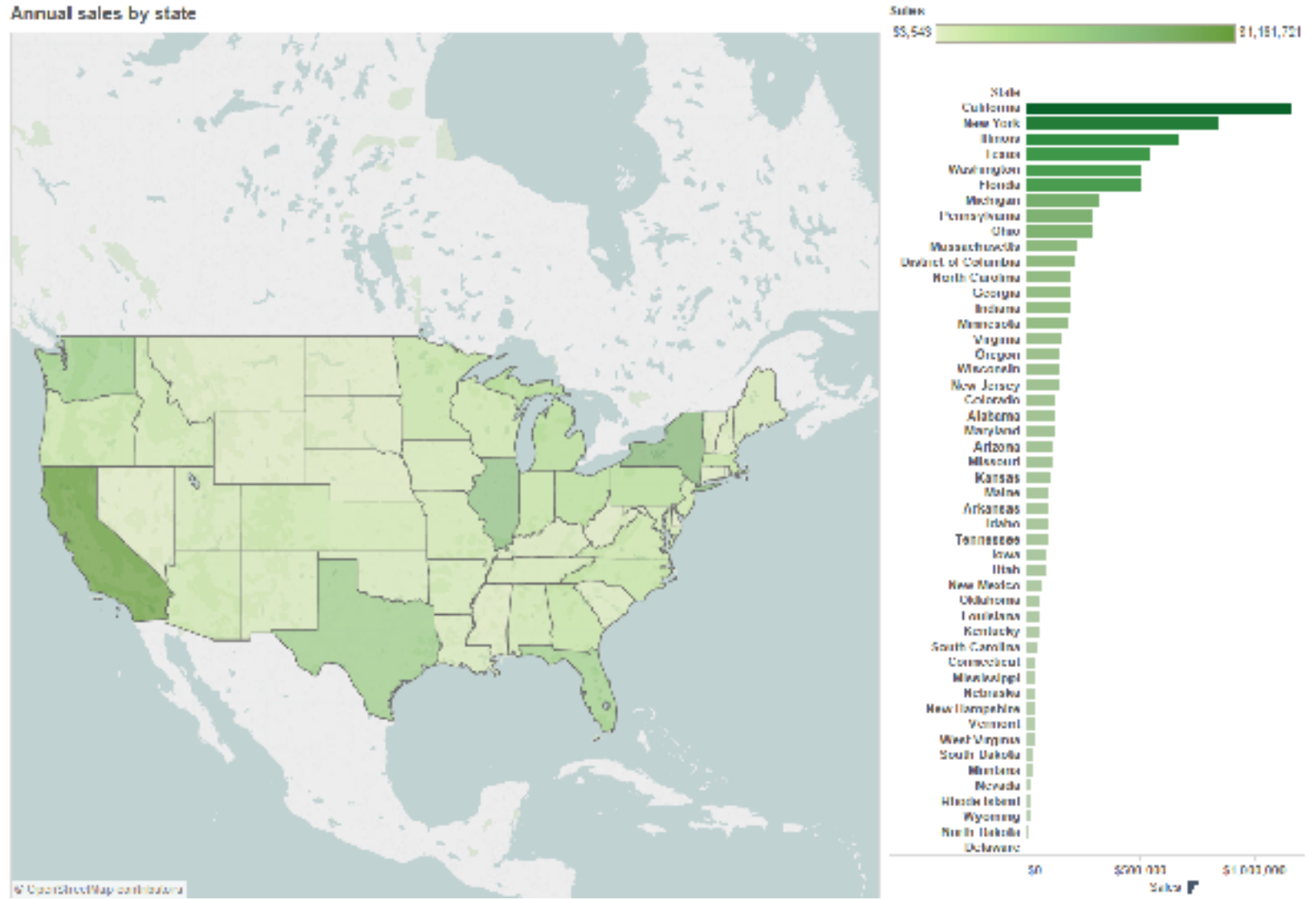
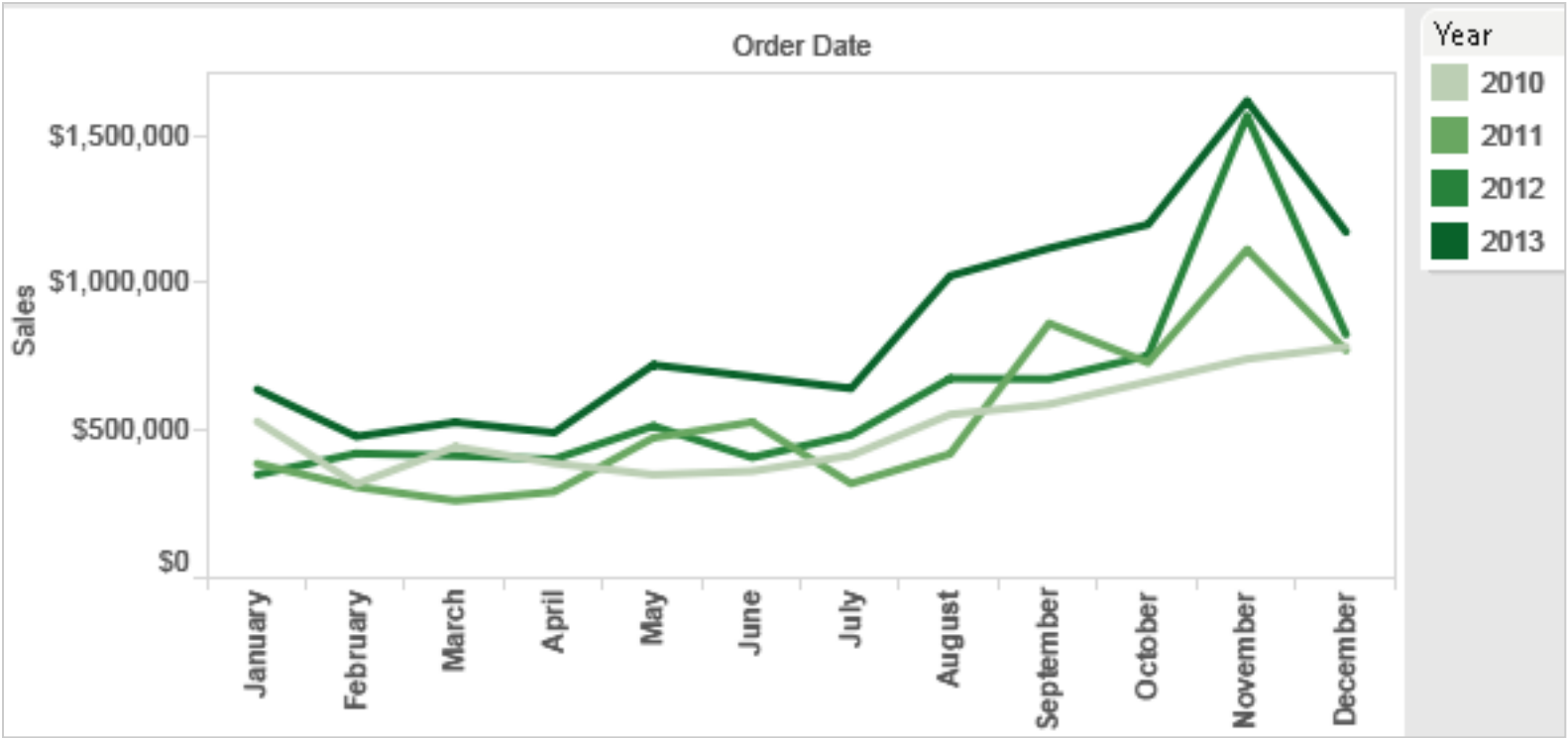
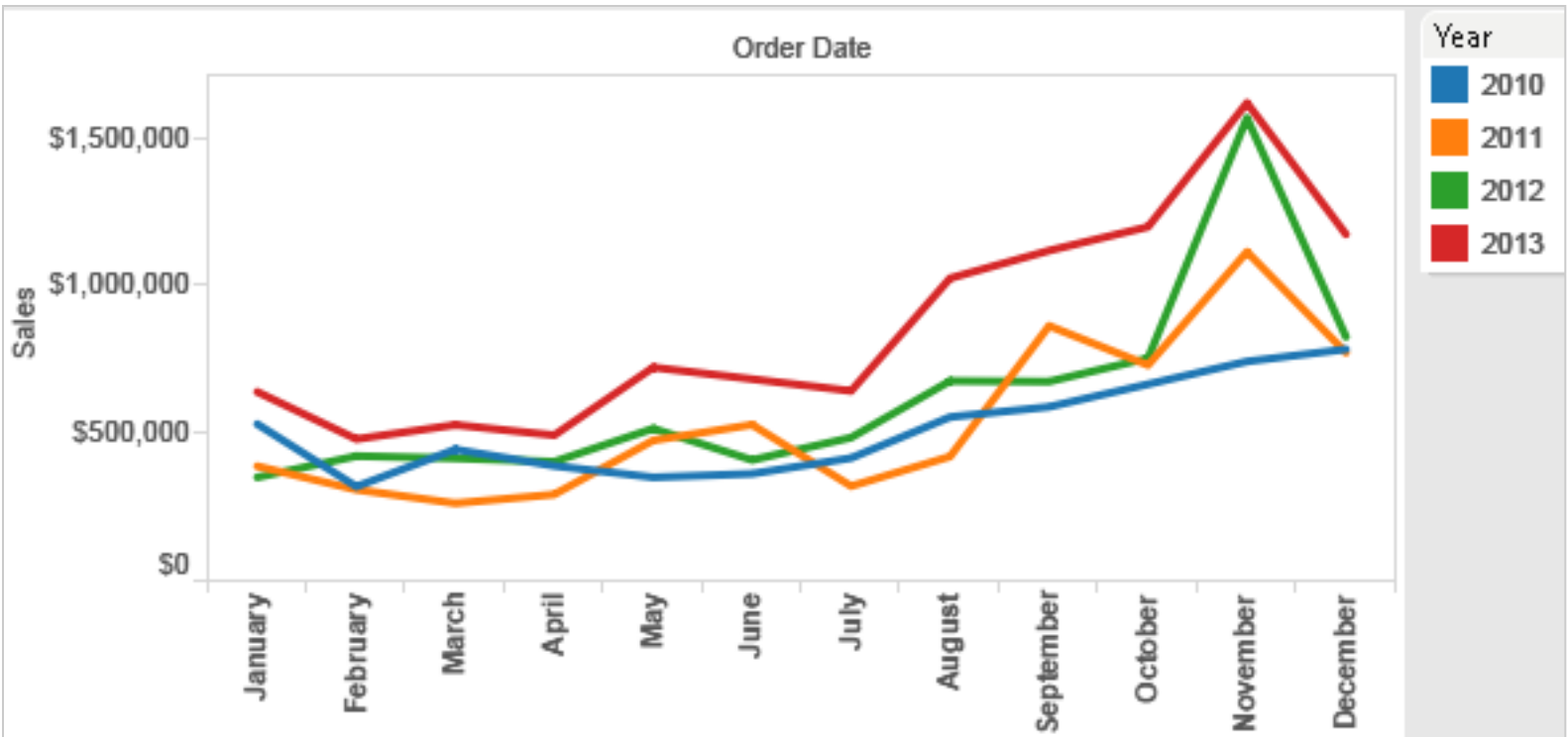
# Decomposing color

- first rule of color: do not (just) talk about color!
  - color is confusing if treated as monolithic
- decompose into three channels
  - ordered can show magnitude
    - **luminance**: how bright (B/W)
    - **saturation**: how colourful
  - categorical can show identity
    - **hue**: what color
- channels have different properties
  - what they convey directly to perceptual system
  - how much they can convey
    - how many discriminable bins can we use?



# Color Channels in Visualization

# Categorical vs ordered color



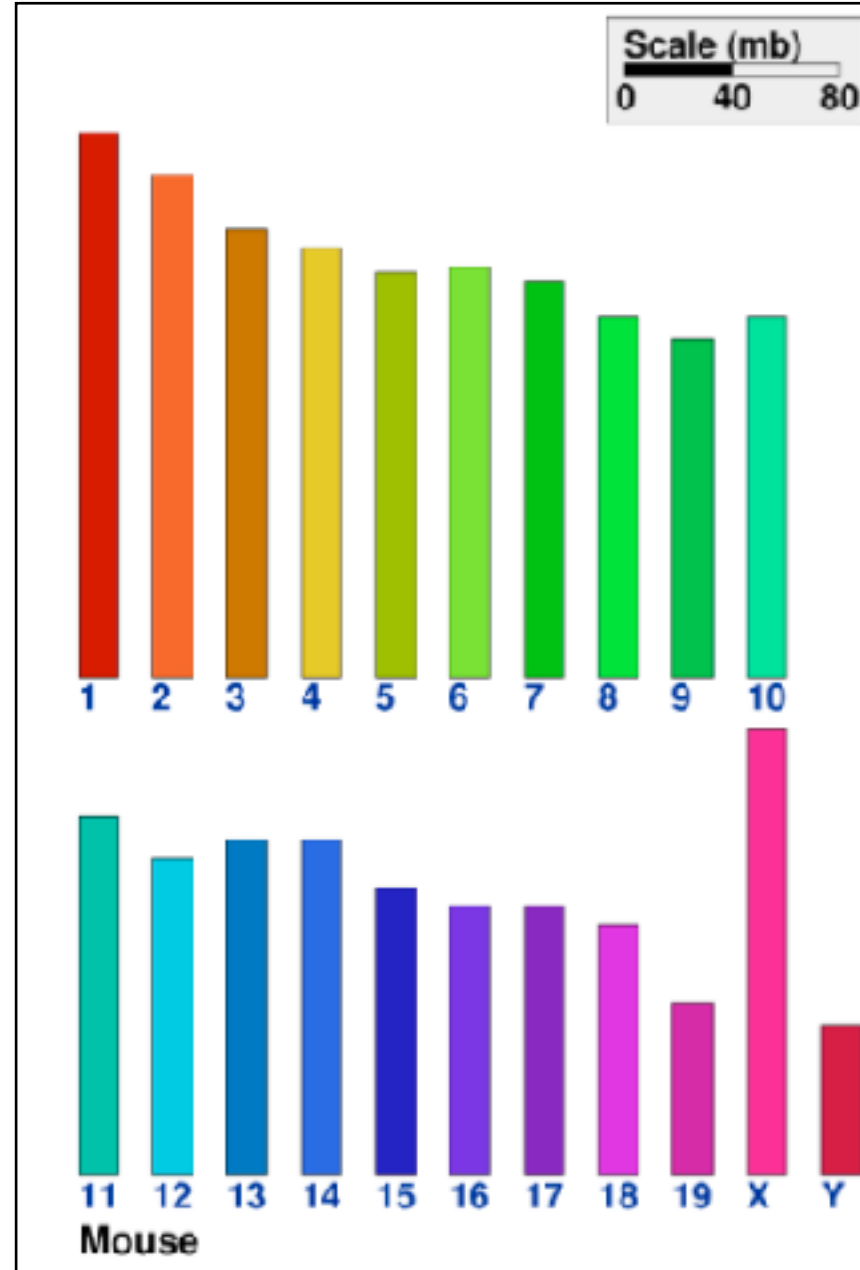
[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]

# Categorical color: limited number of discriminable bins

- human perception built on relative comparisons

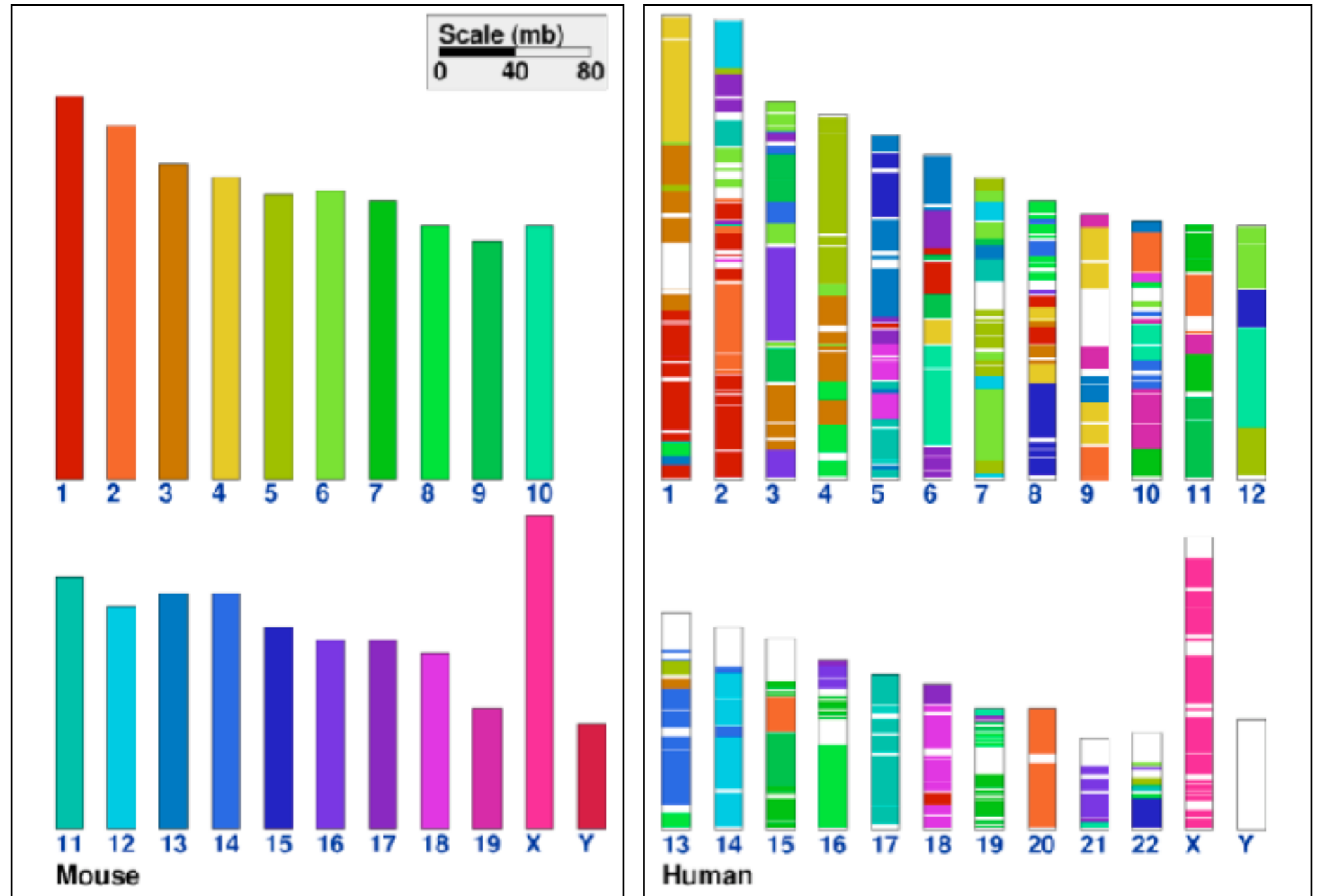
# Categorical color: limited number of discriminable bins

- human perception built on relative comparisons
  - great if color contiguous



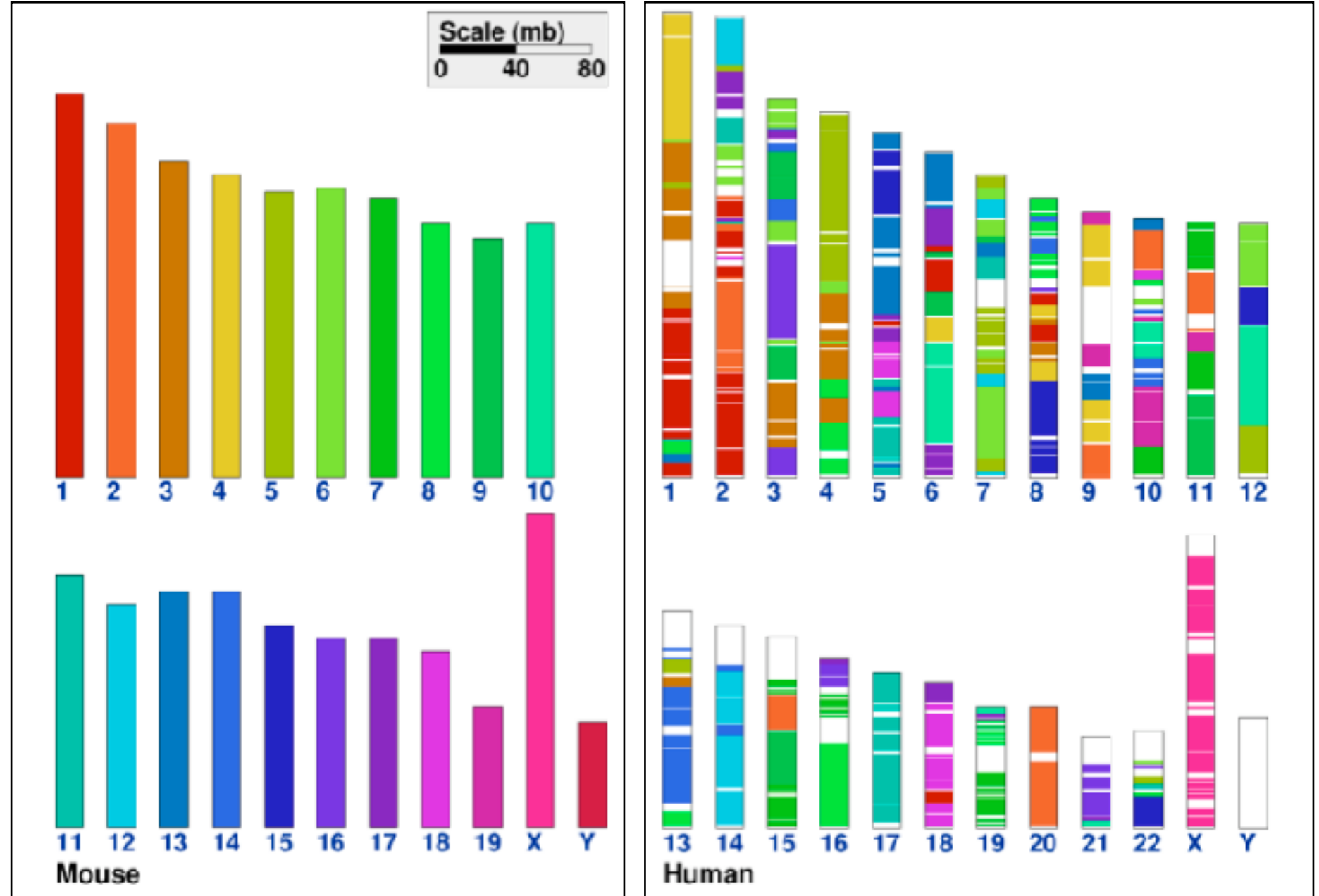
# Categorical color: limited number of discriminable bins

- human perception built on relative comparisons
  - great if color contiguous
  - surprisingly bad for absolute comparisons



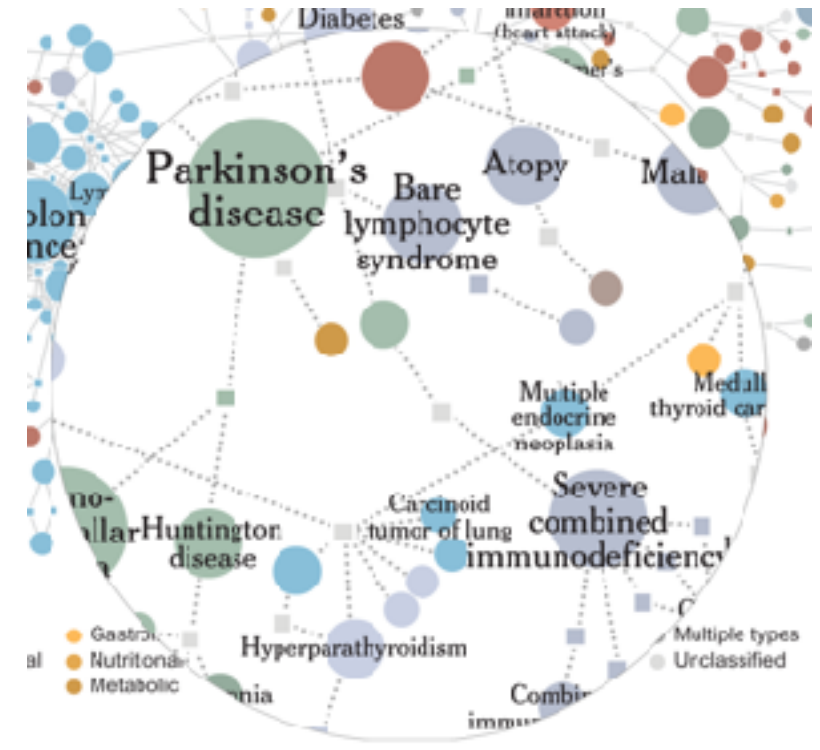
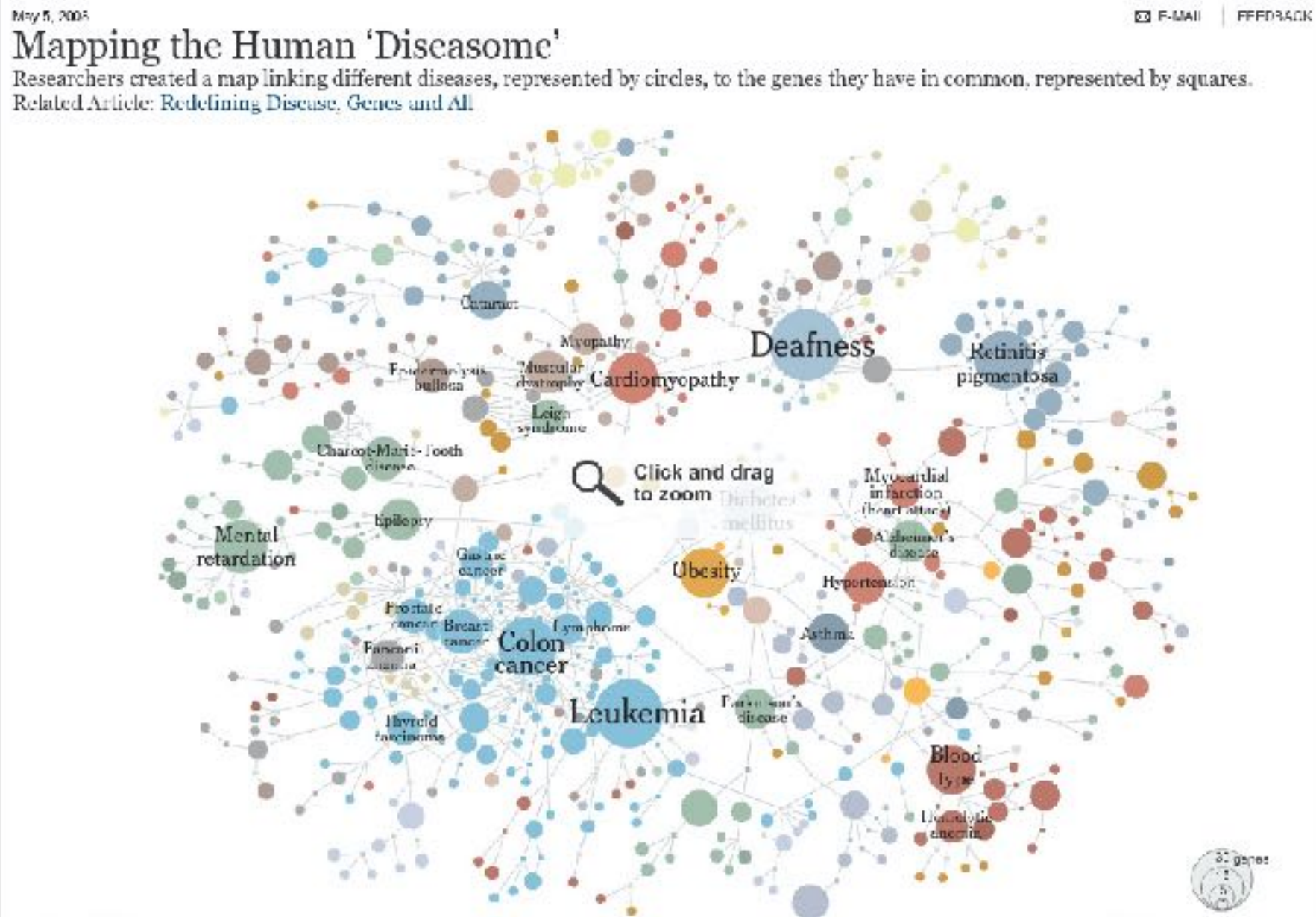
# Categorical color: limited number of discriminable bins

- human perception built on relative comparisons
  - great if color contiguous
  - surprisingly bad for absolute comparisons
- noncontiguous small regions of color
  - fewer bins than you want
  - rule of thumb: 6-12 bins, including background and highlights



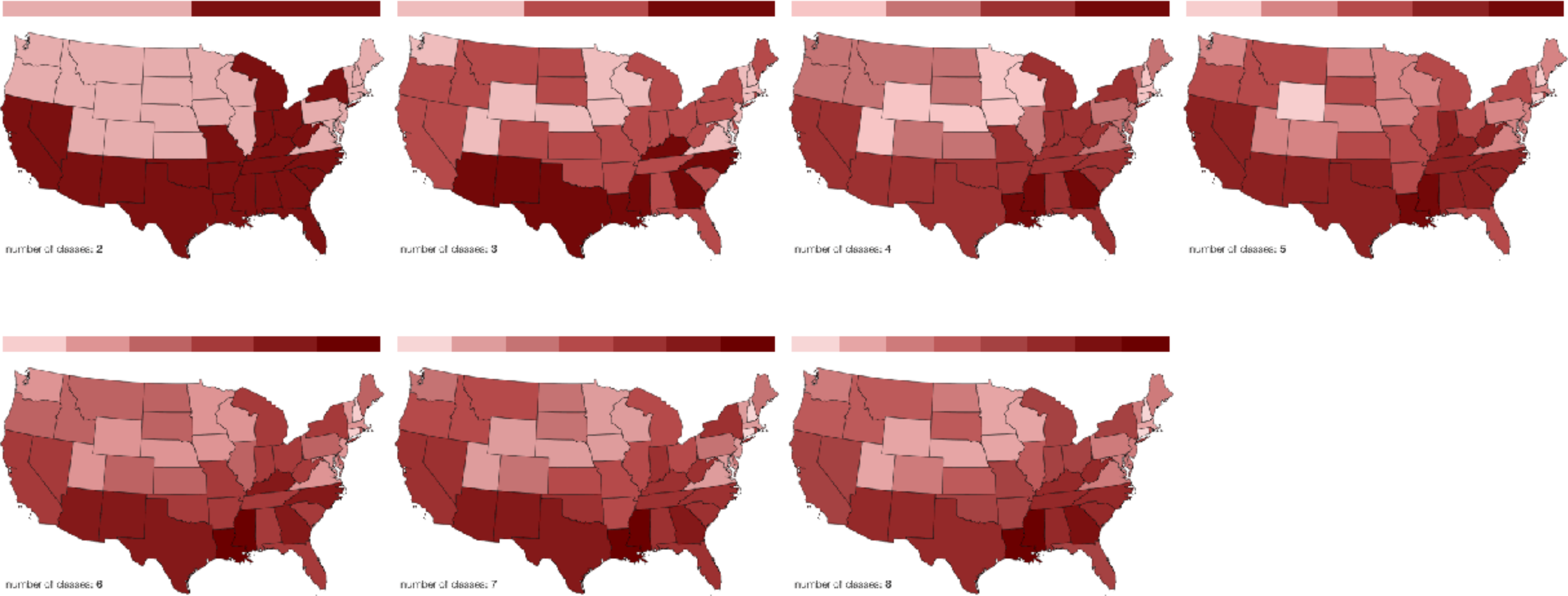
# Categorical color: limited number of discriminable bins

- Cancer
- Connective tissue
- Cardiovascular
- Endocrine
- Gastrointestinal
- Ear, nose, throat
- Developmental
- Multiple types
- Bone
- Muscular
- Hematological
- Immunological
- Nutritional
- Ophthalmological
- Neurological
- Unclassified
- Skeletal
- Dermatological
- Renal
- Metabolic
- Respiratory
- Psychiatric





# Ordered color: limited number of discriminable bins



*Gregor Aisch, [vis4.net/blog/posts/choropleth-maps/](http://vis4.net/blog/posts/choropleth-maps/)*

# Ordered color: Rainbow is poor default

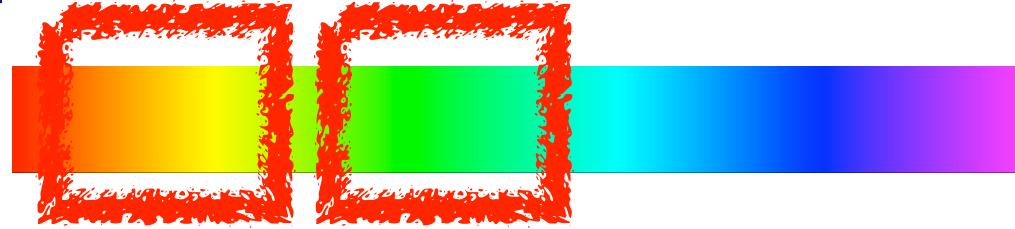
- problems

- perceptually unordered
- perceptually nonlinear



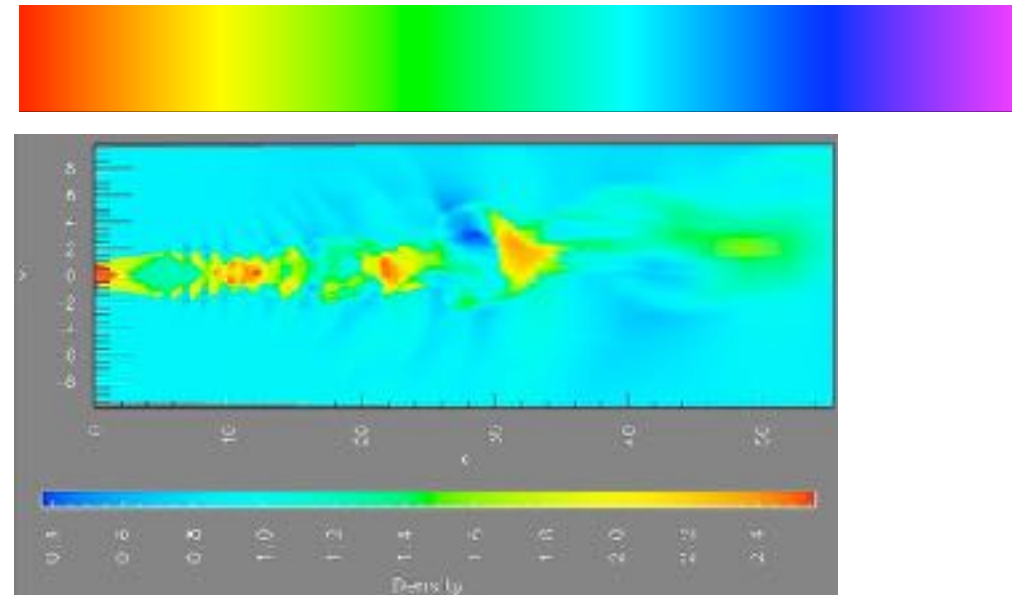
# Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear

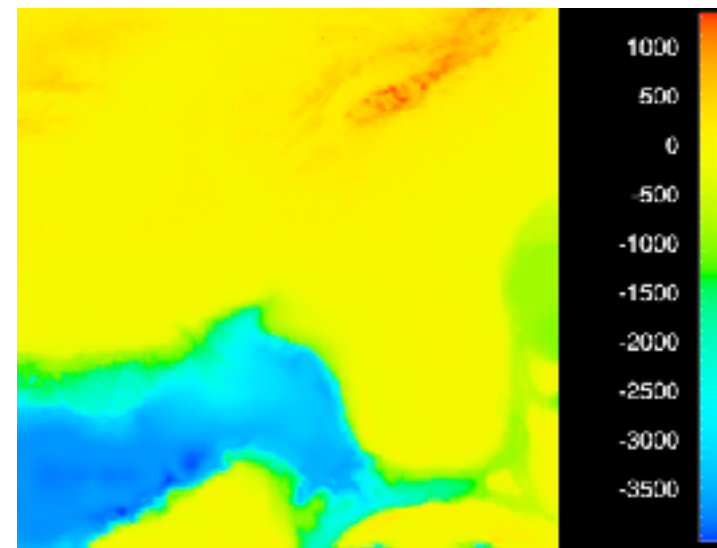


# Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear
- benefits
  - fine-grained structure visible and nameable



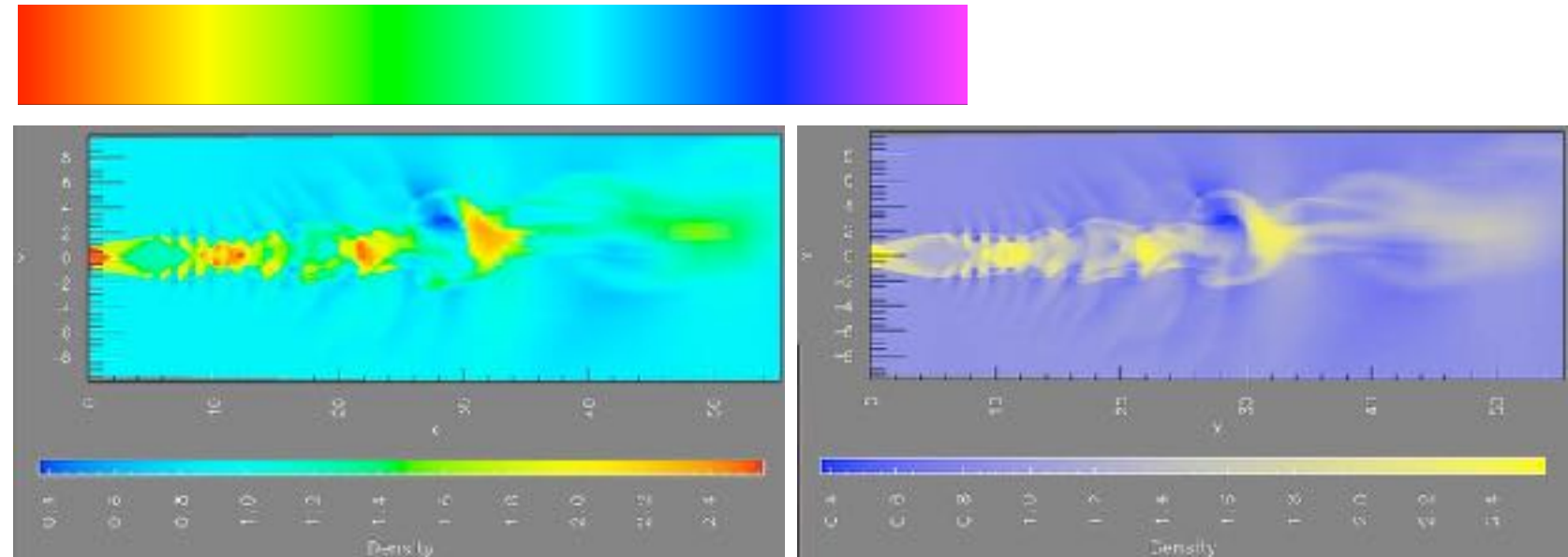
[A Rule-based Tool for Assisting Colormap Selection. Bergman, Rogowitz, and Treinish. Proc. IEEE Visualization (Vis), pp. 118–125, 1995.]



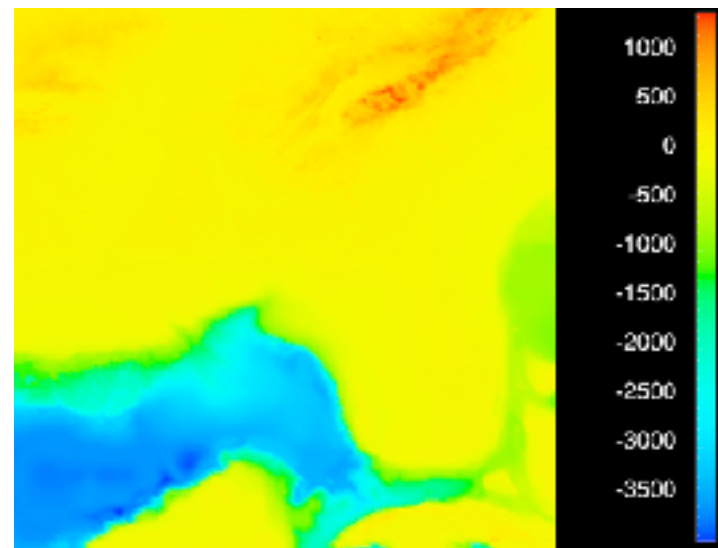
[Why Should Engineers Be Worried About Color? Treinish and Rogowitz 1998. <http://www.research.ibm.com/people/lloyd/color/color.HTM>]

# Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear
- benefits
  - fine-grained structure visible and nameable
- alternatives
  - large-scale structure: fewer hues



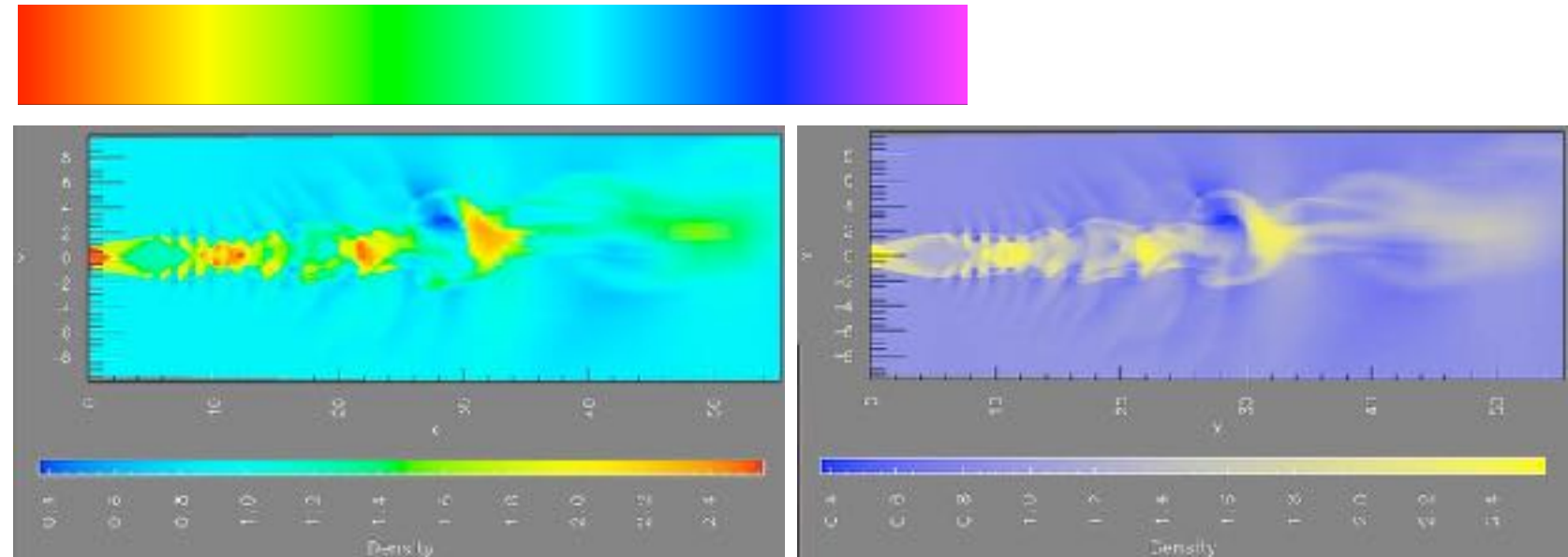
[A Rule-based Tool for Assisting Colormap Selection. Bergman, Rogowitz, and Treinish. Proc. IEEE Visualization (Vis), pp. 118–125, 1995.]



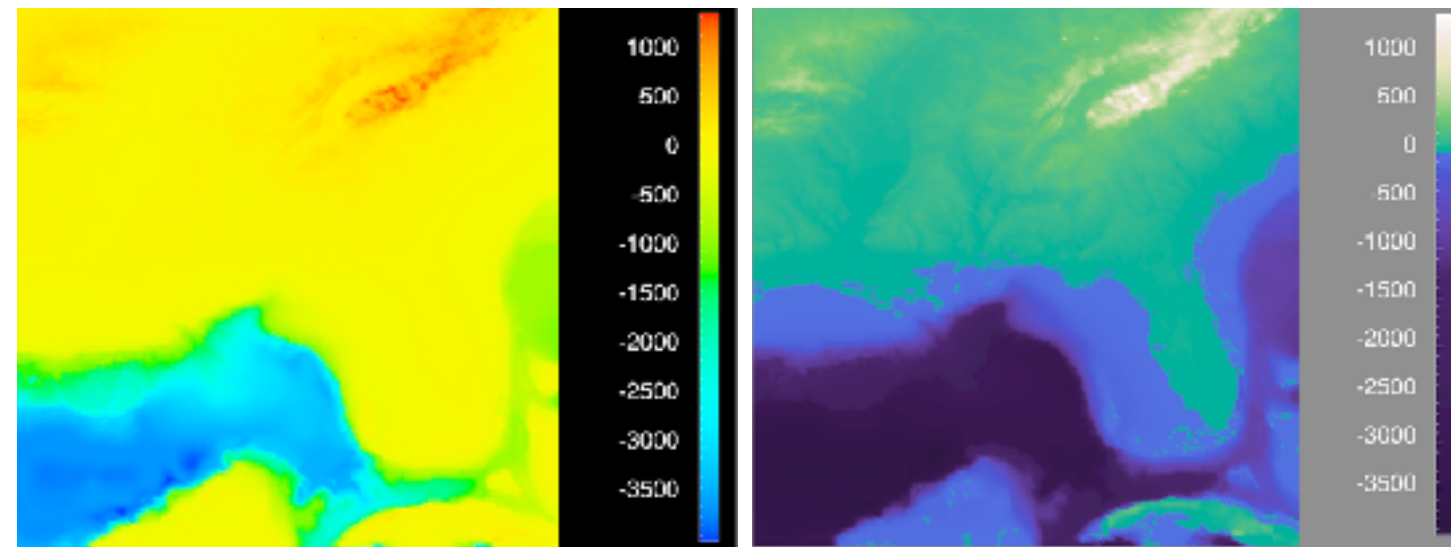
[Why Should Engineers Be Worried About Color? Treinish and Rogowitz 1998. <http://www.research.ibm.com/people/lloyd/color/color.HTM>]

# Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear
- benefits
  - fine-grained structure visible and nameable
- alternatives
  - large-scale structure: fewer hues
  - fine structure: multiple hues with monotonically increasing luminance [eg viridis]



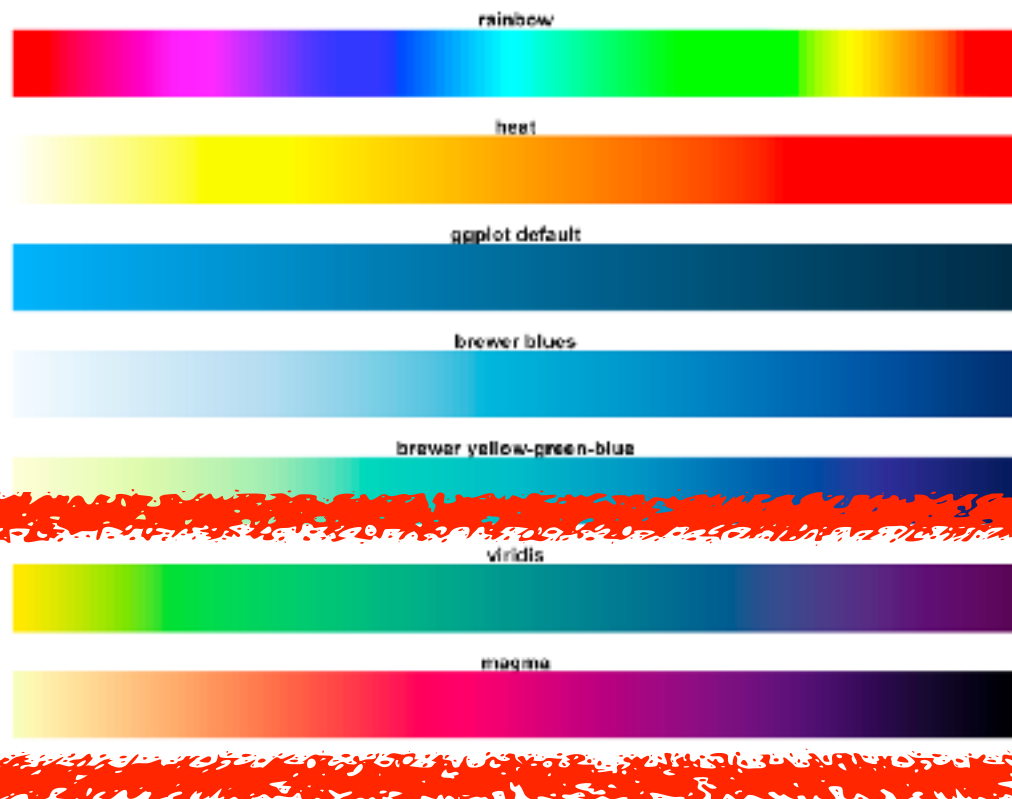
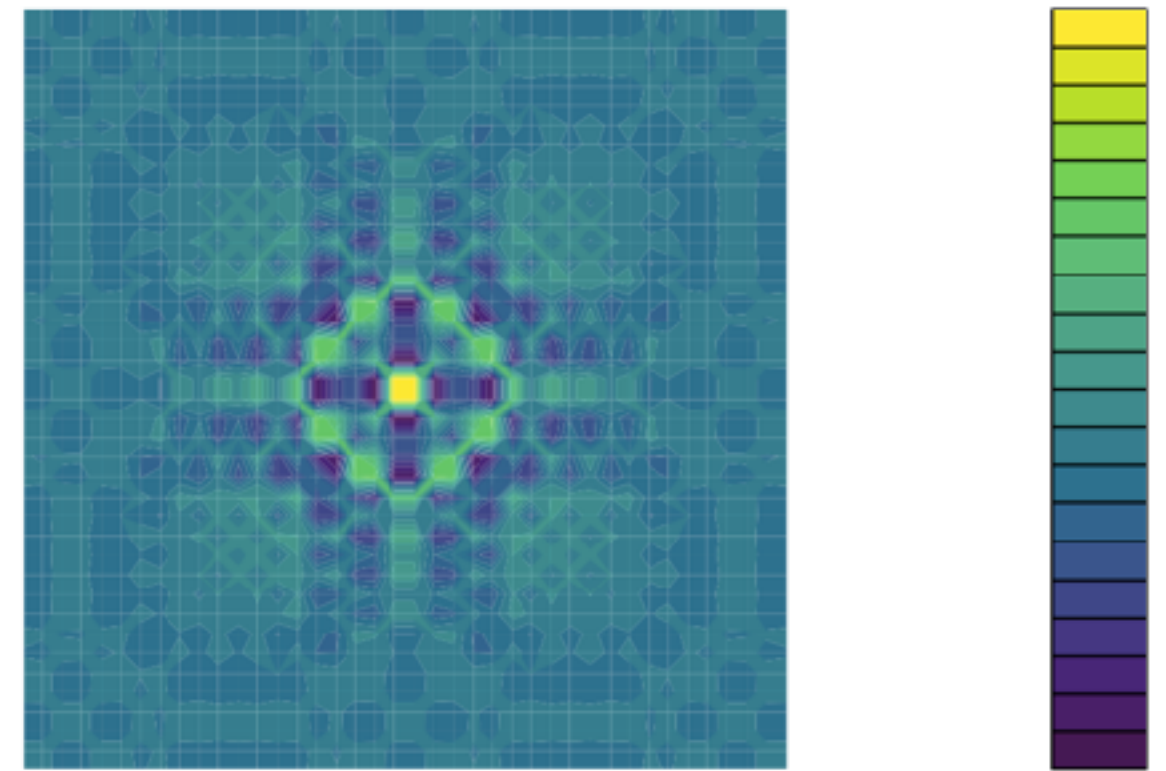
[A Rule-based Tool for Assisting Colormap Selection. Bergman, Rogowitz, and Treinish. Proc. IEEE Visualization (Vis), pp. 118–125, 1995.]



[Why Should Engineers Be Worried About Color? Treinish and Rogowitz 1998. <http://www.research.ibm.com/people/lloyd/color/color.HTM>]

# Viridis / Magma: sequential colormaps

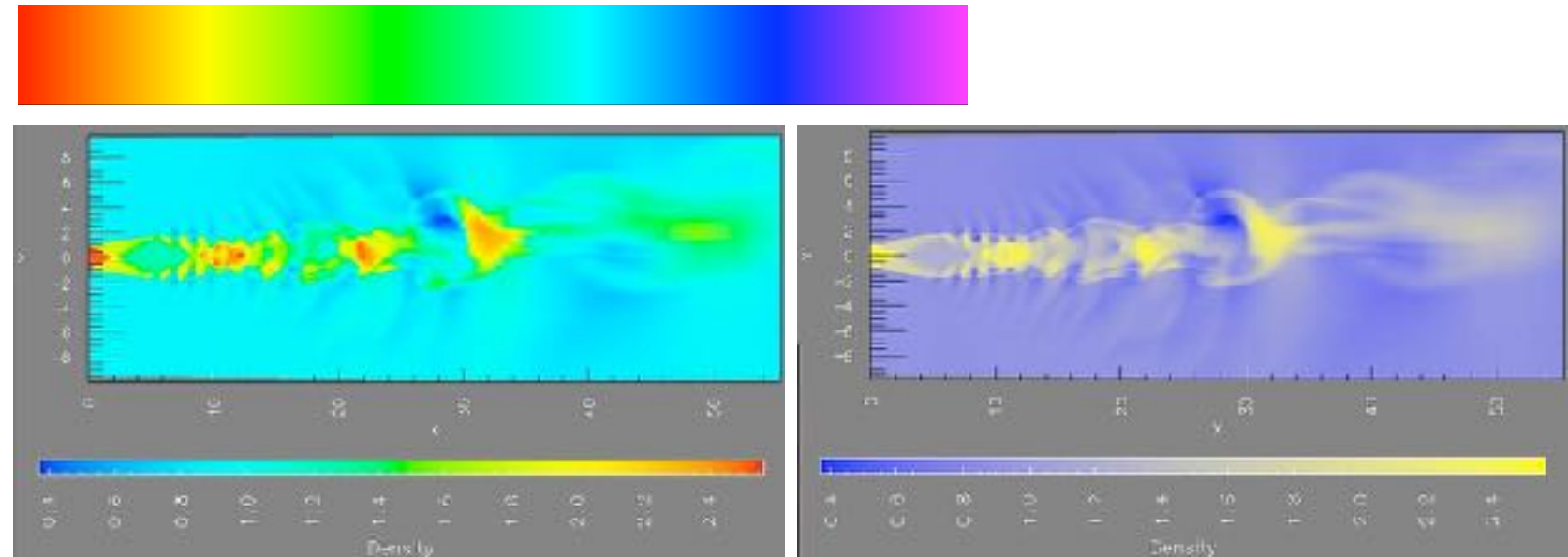
- monotonically increasing luminance, perceptually uniform
- colorful, colorblind-safe
  - R, python, D3



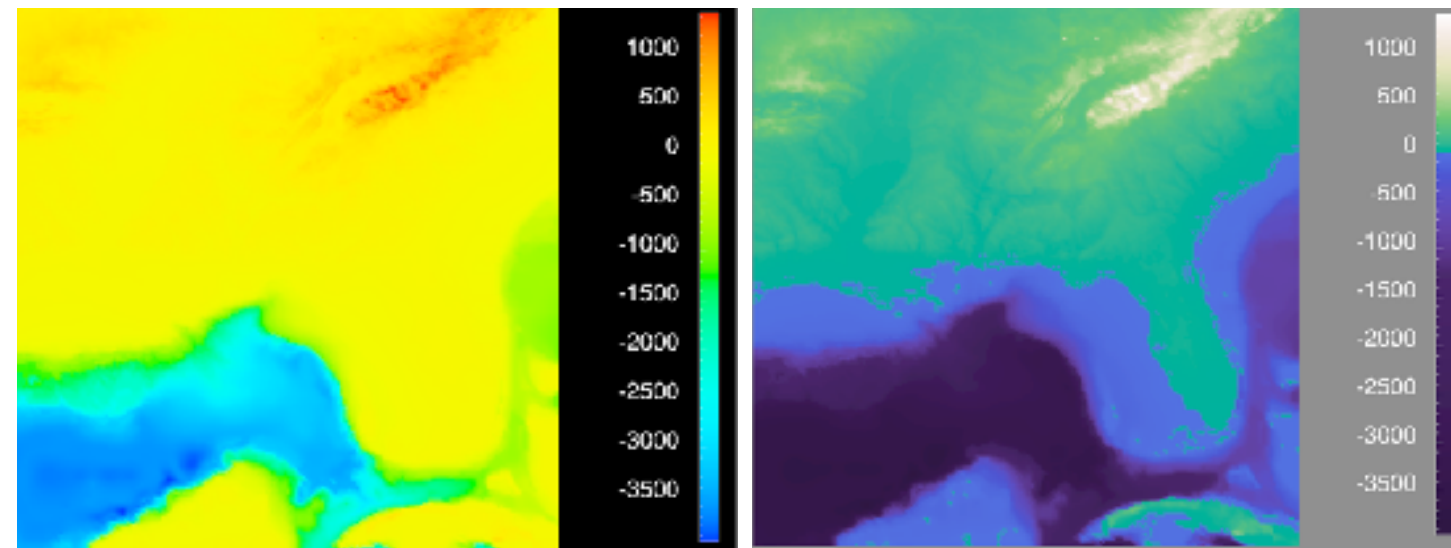
<https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html>

# Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear
- benefits
  - fine-grained structure visible and nameable
- alternatives
  - large-scale structure: fewer hues
  - fine structure: multiple hues with monotonically increasing luminance [eg viridis]
- legit for categorical
  - segmented saturated rainbow is good!



[A Rule-based Tool for Assisting Colormap Selection. Bergman, Rogowitz, and Treinish. Proc. IEEE Visualization (Vis), pp. 118–125, 1995.]



[Why Should Engineers Be Worried About Color? Treinish and Rogowitz 1998. <http://www.research.ibm.com/people/l/lloyd/color/color.HTM>]

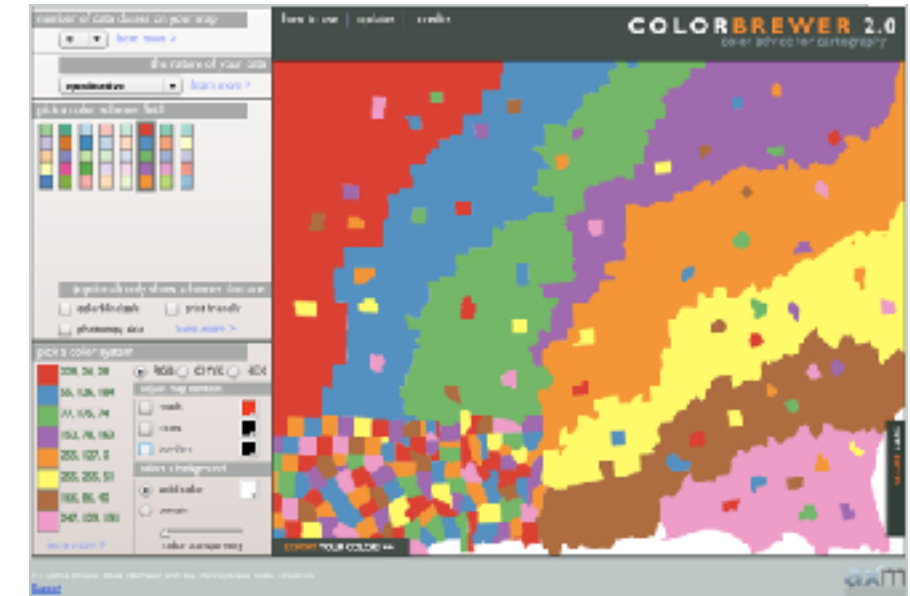
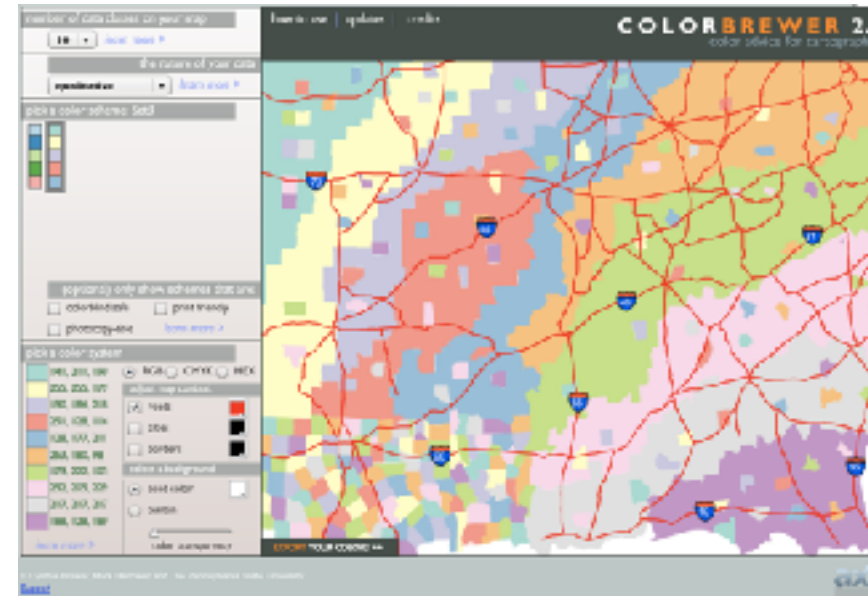


[Transfer Functions in Direct Volume Rendering: Design, Interface, Interaction. Kindlmann. SIGGRAPH 2002 Course Notes]



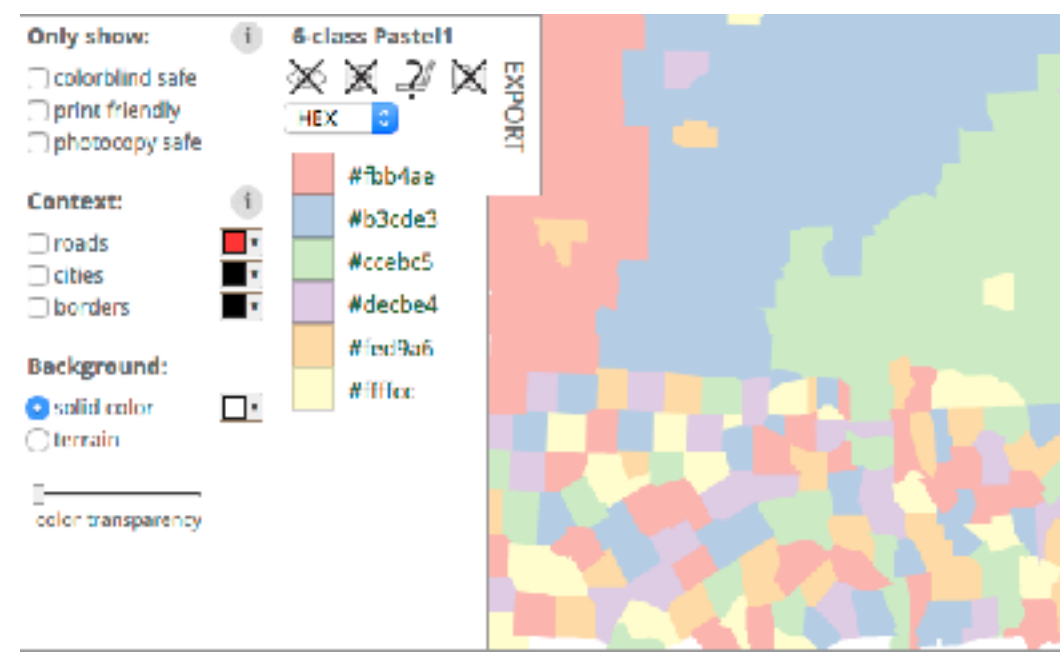
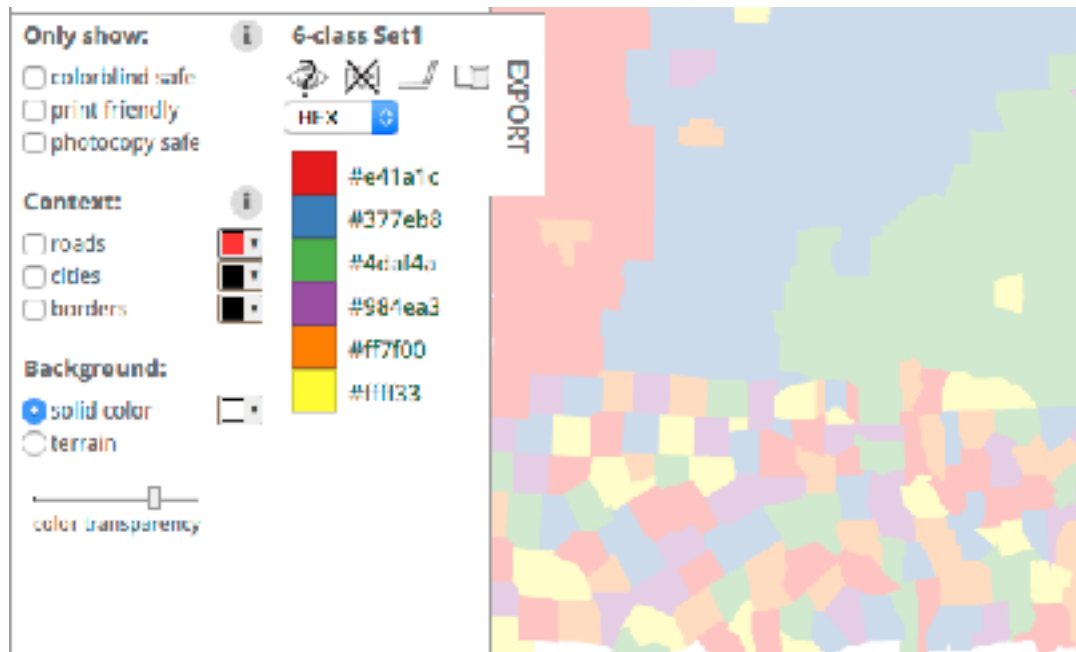
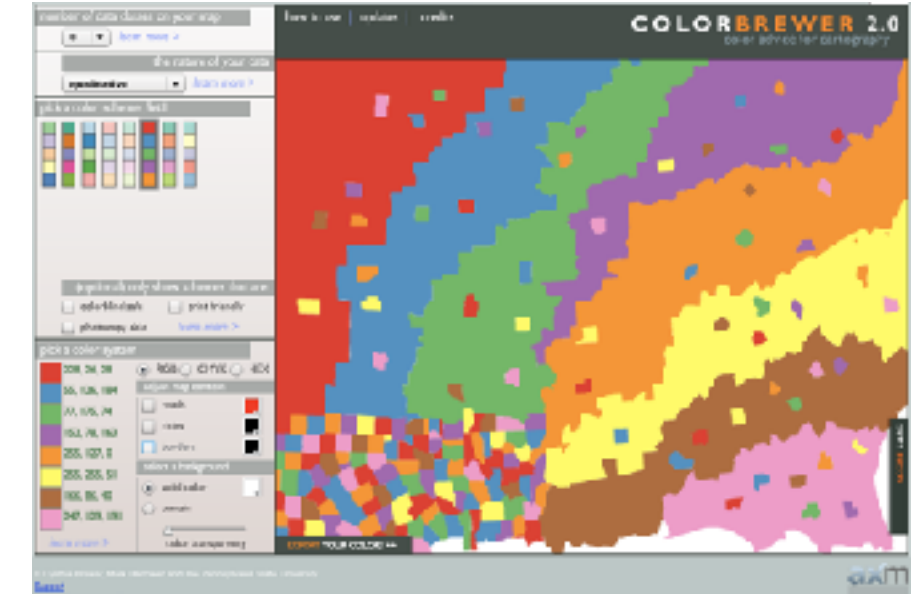
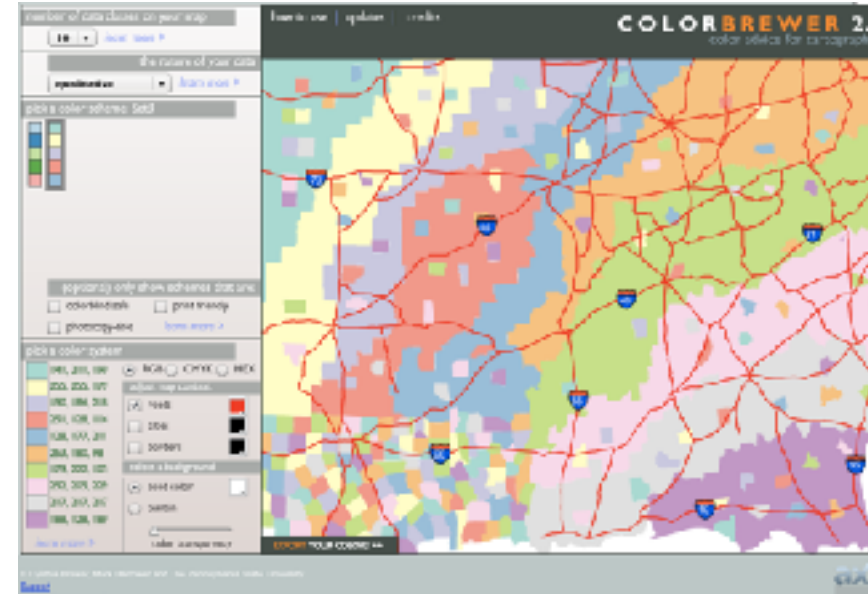
# Interaction between channels: Not fully separable

- color channel interactions
  - size heavily affects salience
  - small regions need high saturation
  - large regions need low saturation



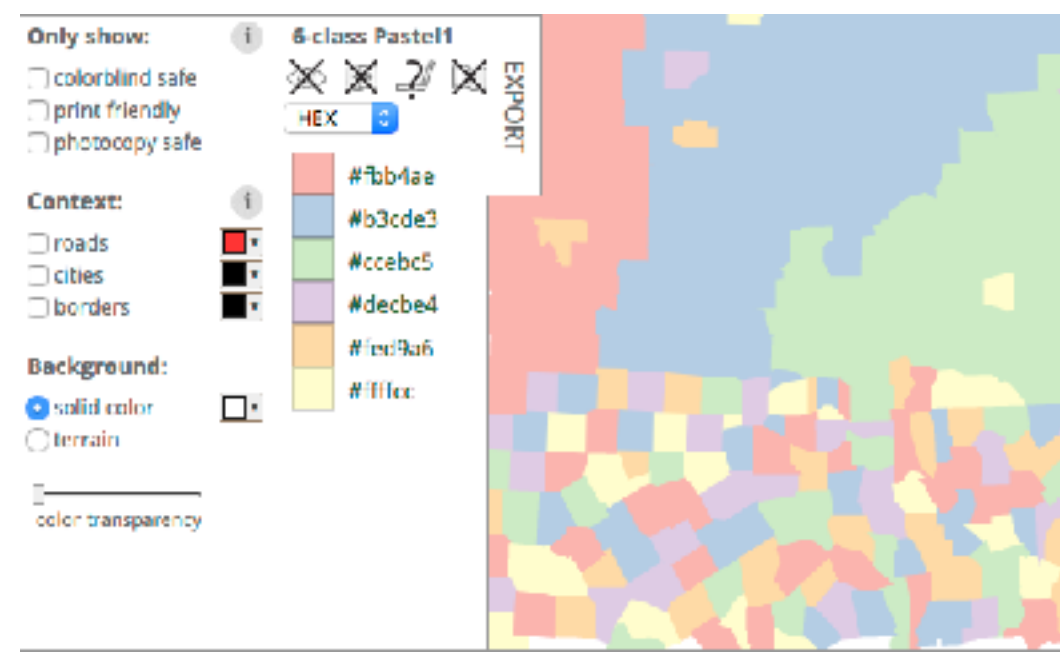
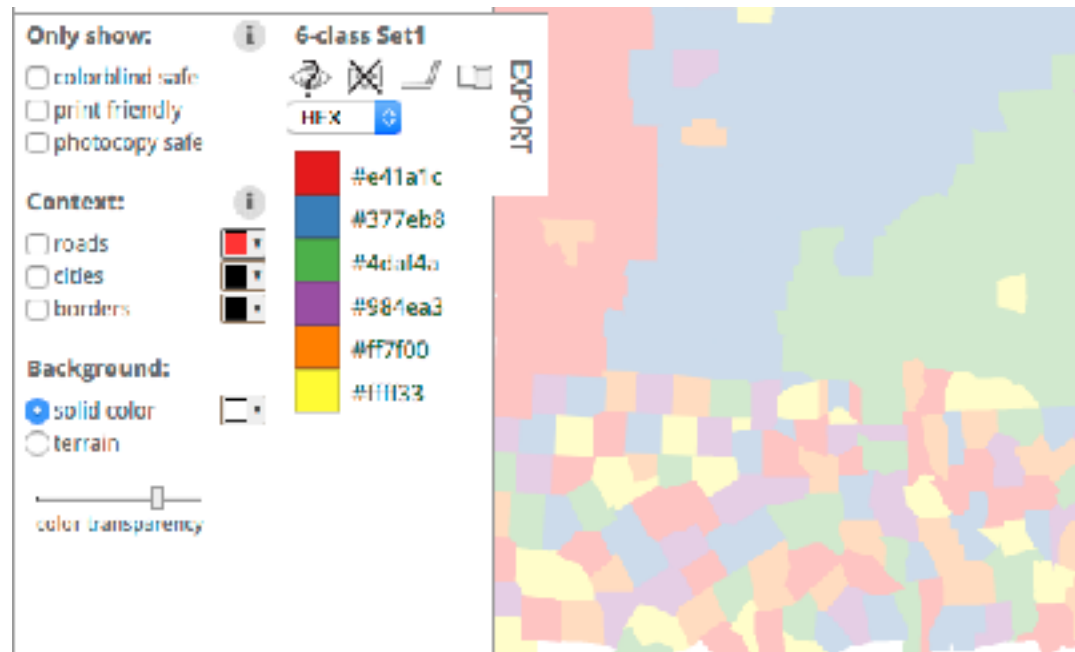
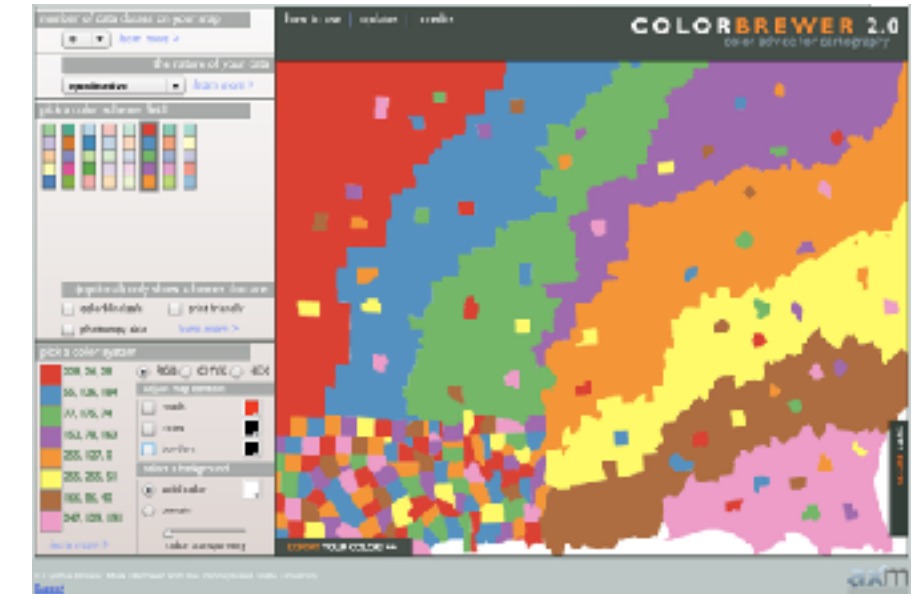
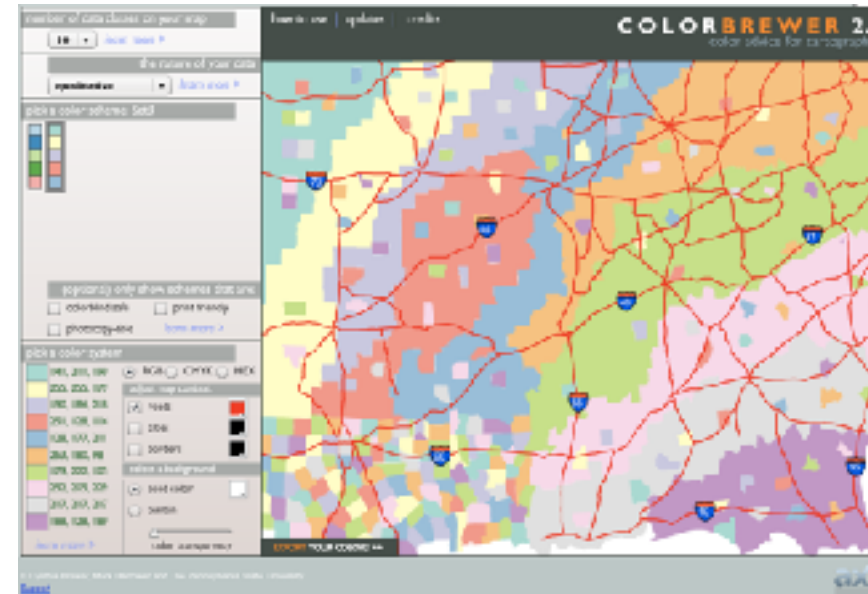
# Interaction between channels: Not fully separable

- color channel interactions
  - size heavily affects salience
  - small regions need high saturation
  - large regions need low saturation
- saturation & luminance:
  - not separable from each other!
  - also not separable from transparency



# Interaction between channels: Not fully separable

- color channel interactions
  - size heavily affects salience
  - small regions need high saturation
  - large regions need low saturation
- saturation & luminance:
  - not separable from each other!
  - also not separable from transparency
  - small separated regions: 2 bins safest (use only one of these channels), 3-4 bins max
  - contiguous regions: many bins (use only one of these channels)



# Color Palettes

# Color palettes: univariate

→ Categorical



- categorical

- aim for maximum distinguishability
- aka *qualitative, nominal*



# Color palettes: univariate

→ Categorical



Categorical

→ Ordered

→ *Sequential*

→ *Diverging*



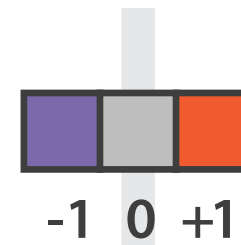
- diverging

- useful when data has meaningful "midpoint"
- use neutral color for midpoint
  - white, yellow, grey
- use saturated colors for endpoints

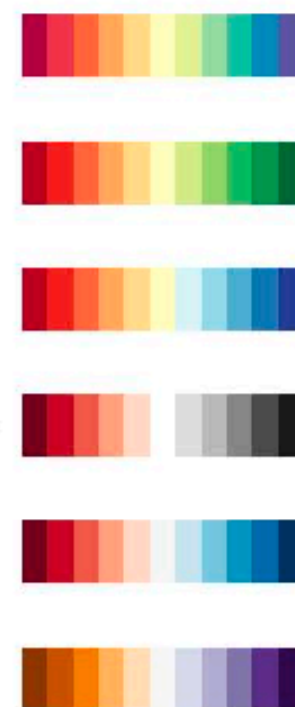
- sequential

- ramp luminance or saturation

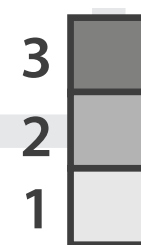
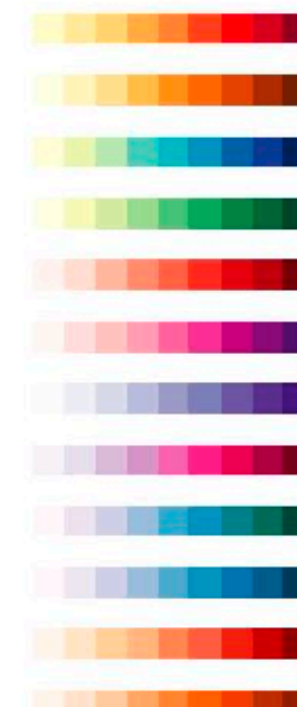
Diverging



diverging



sequential



Sequential

# Color palettes: univariate

→ Categorical

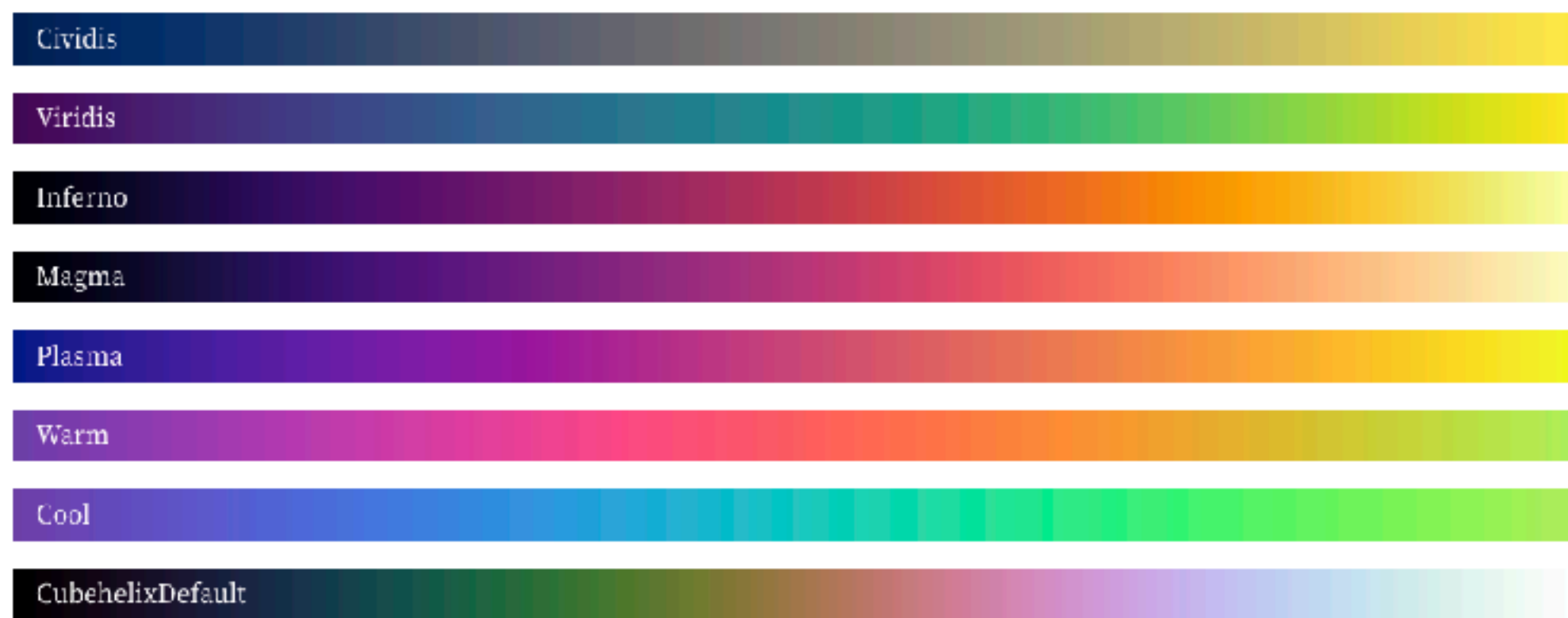


→ Ordered

→ *Sequential*



→ *Diverging*



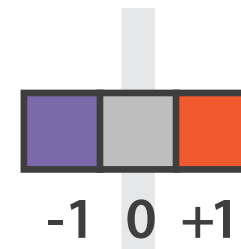
- diverging

- useful when data has meaningful "midpoint"
- use neutral color for midpoint
  - white, yellow, grey
- use saturated colors for endpoints

- sequential

- ramp luminance or saturation
- if multi-hue, good to order by luminance

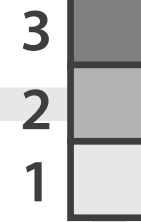
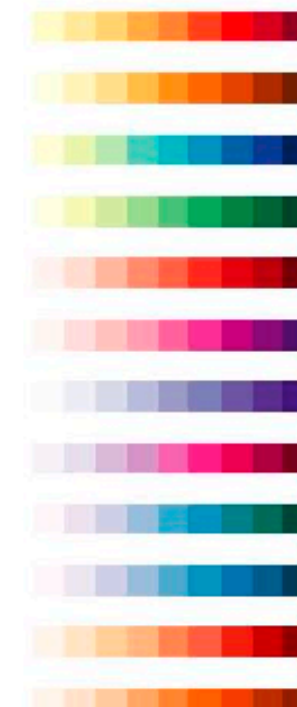
Diverging



diverging



sequential



Sequential

# Color palettes: univariate

→ Categorical



→ Ordered

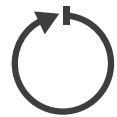
→ *Sequential*



→ *Diverging*



→ Cyclic



cyclic multihue



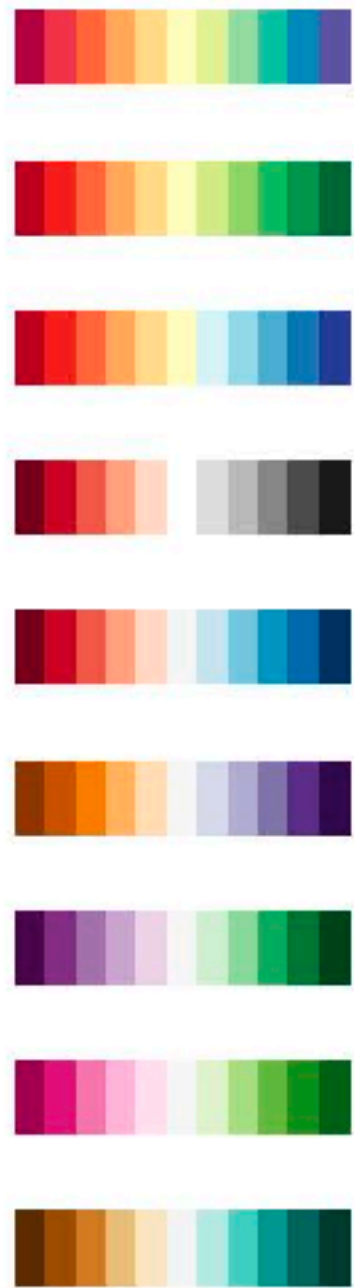
<https://github.com/d3/d3-scale-chromatic>



# Color palette design considerations: univariate

segmented

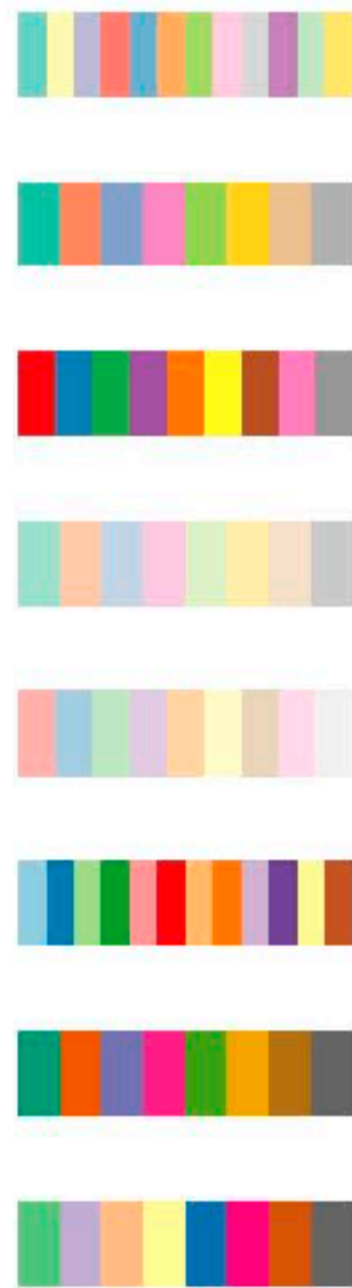
diverging



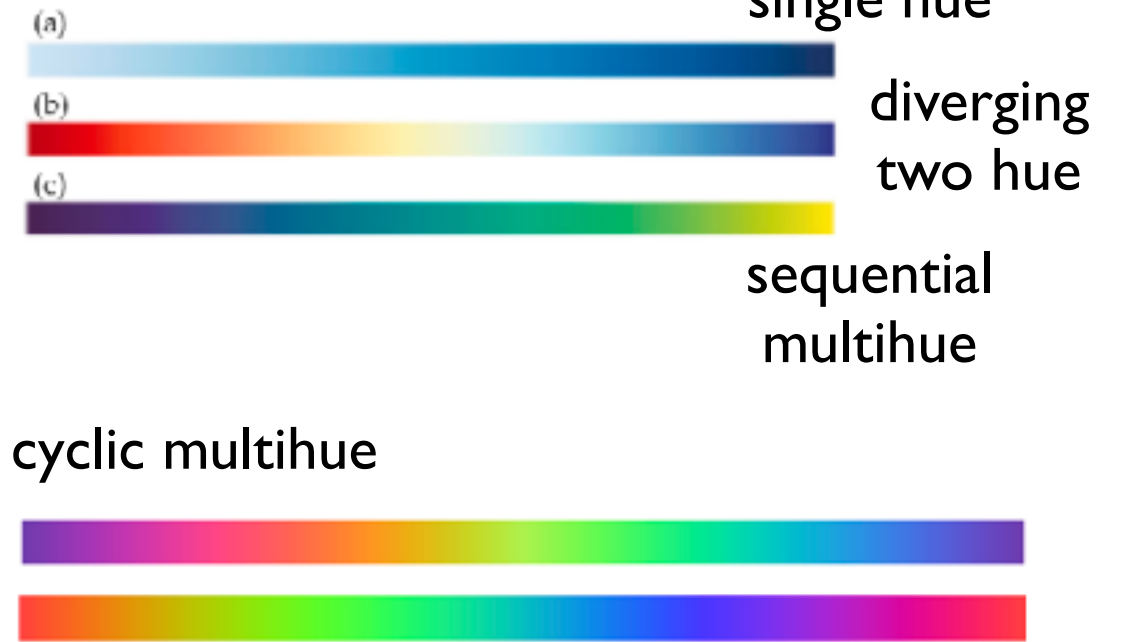
sequential



categorical



continuous



- segmented or continuous?
- diverging or sequential or cyclic?
- single-hue or two-hue or multi-hue?
- perceptually linear?
- ordered by luminance?
- colorblind safe?

# Colormaps: bivariate

→ Categorical



→ Ordered

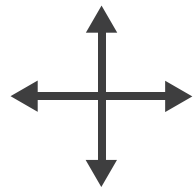
→ *Sequential*



→ *Diverging*



→ Bivariate

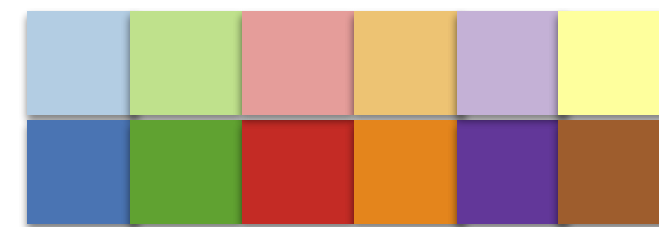
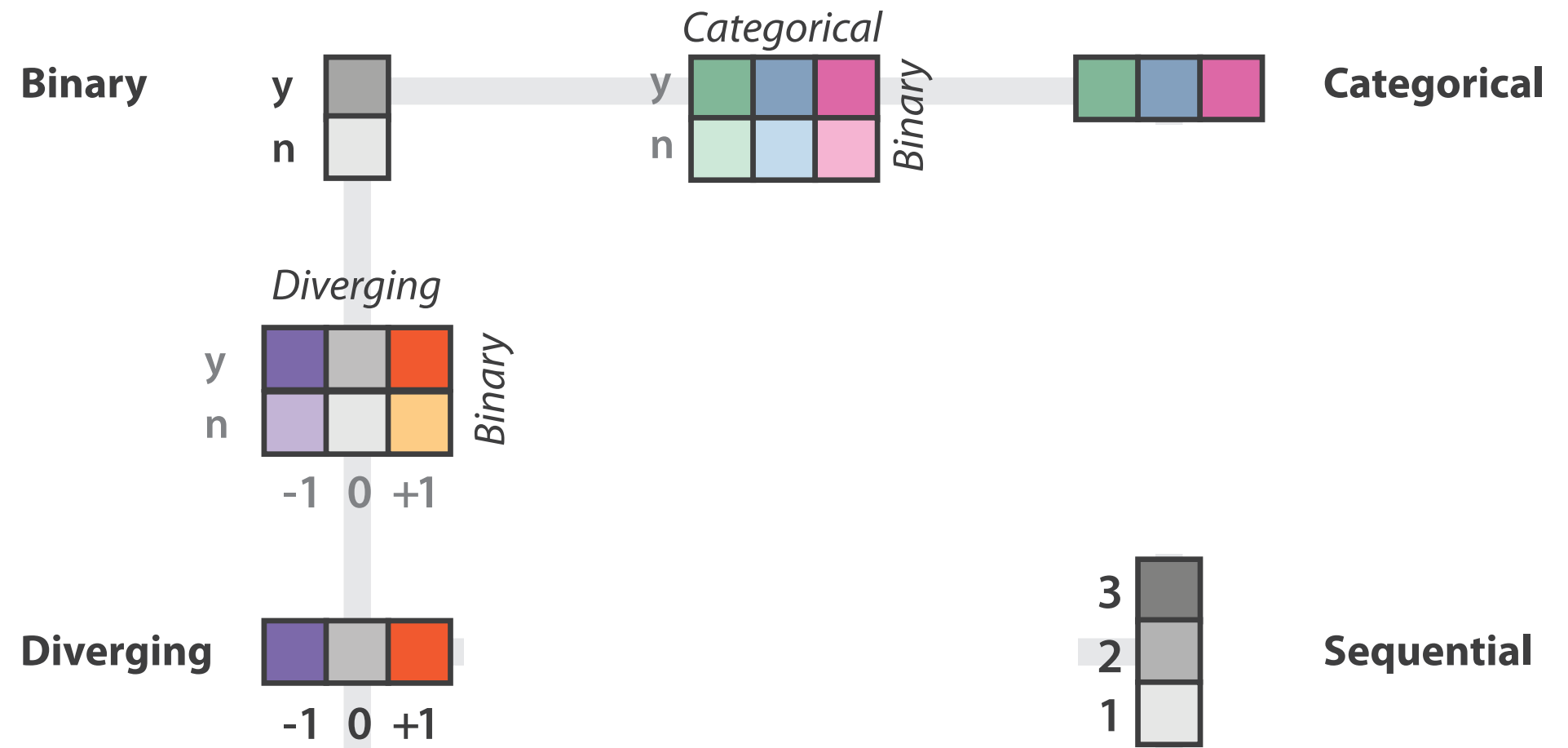


• bivariate best case

• binary in one of the directions



# d3.schemePaired <>



binary saturation

categorical hue

# Colormaps: bivariate

→ Categorical



→ Ordered

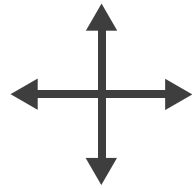
→ *Sequential*



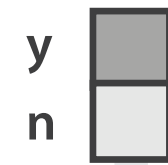
→ *Diverging*



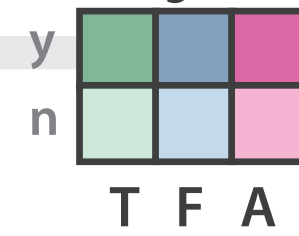
→ Bivariate



**Binary**



*Categorical*

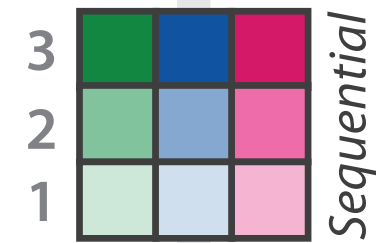


*Binary*

**Categorical**

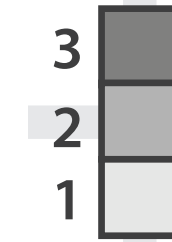


*Categorical*



*Sequential*

**Sequential**

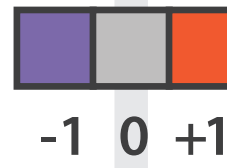


*Diverging*



*Binary*

**Diverging**



# Colormaps

→ Categorical



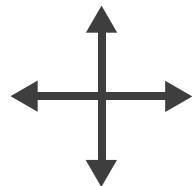
→ Ordered

→ Sequential

→ Diverging

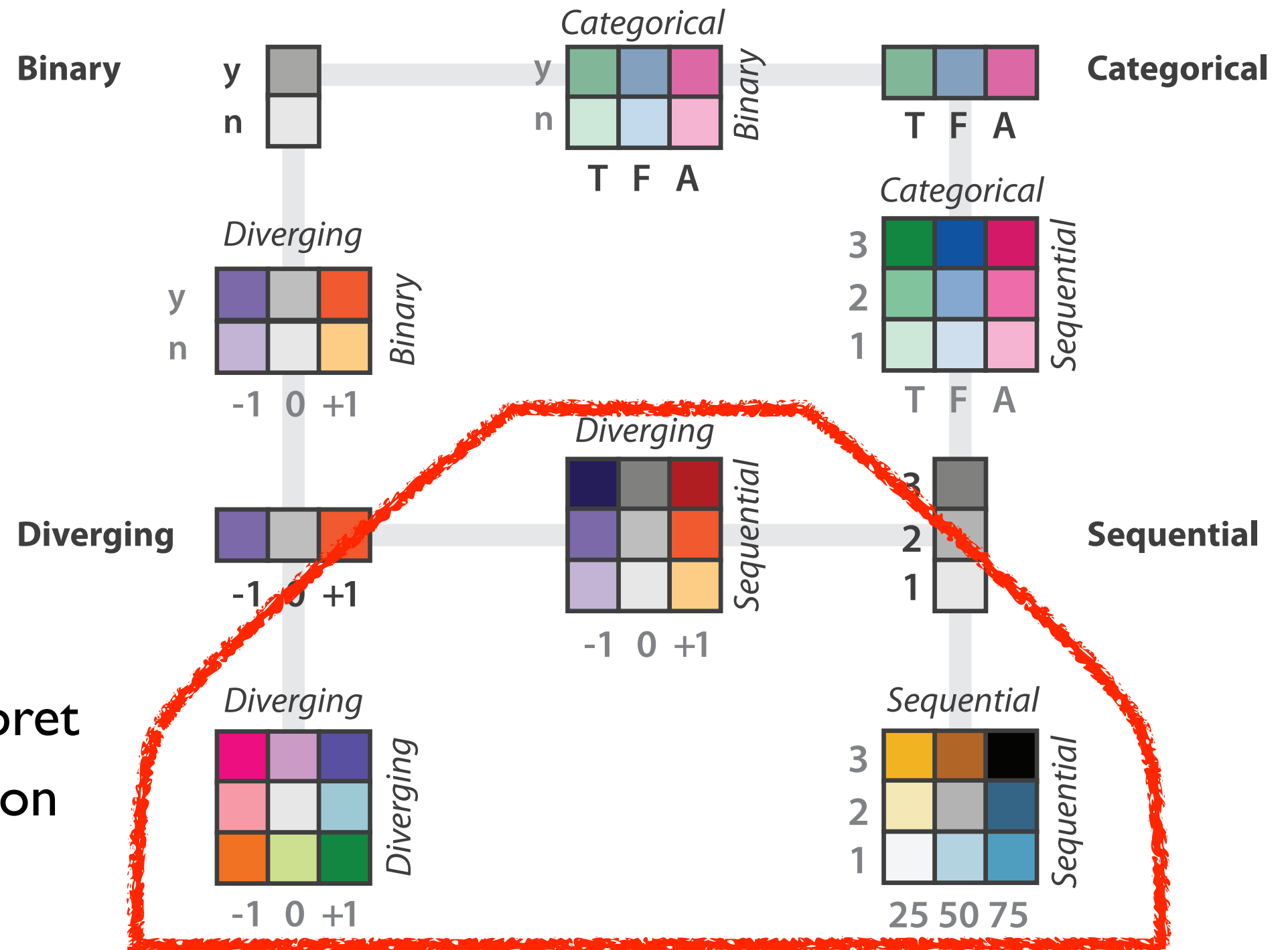


→ Bivariate



use with care!

- bivariate can be very difficult to interpret
  - when multiple levels in each direction



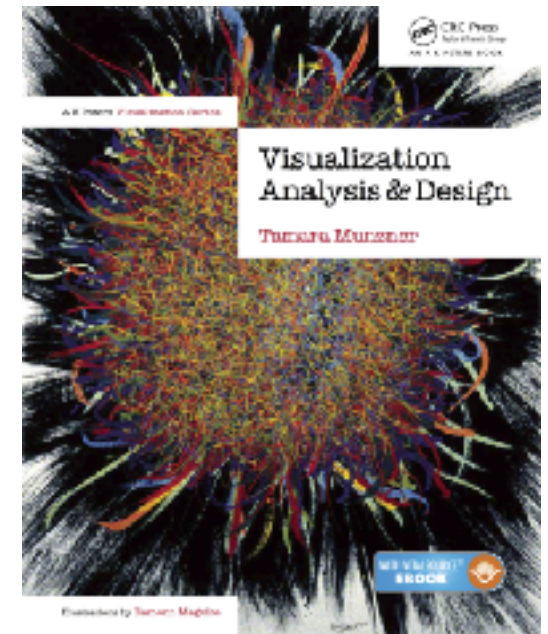
# Visualization Analysis & Design

## *Color (Ch 10) II*

**Tamara Munzner**

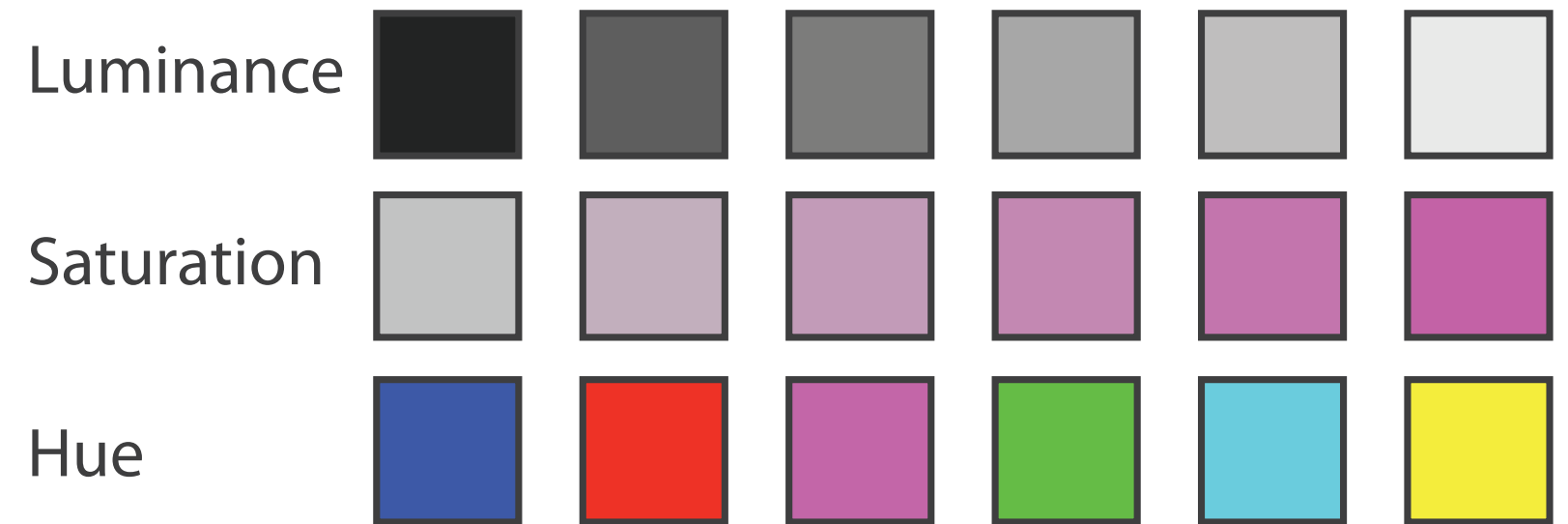
Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# Decomposing color

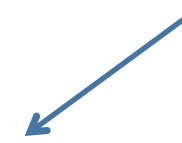
- decompose into three channels
  - ordered can show magnitude
    - **luminance**: how bright (B/W)
    - **saturation**: how colourful
  - categorical can show identity
    - **hue**: what color



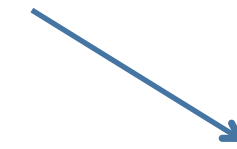
# Color Deficiency

# Luminance

- need luminance for edge detection
  - fine-grained detail only visible through luminance contrast
  - legible text requires luminance contrast!



Luminance information



Saturation/hue information

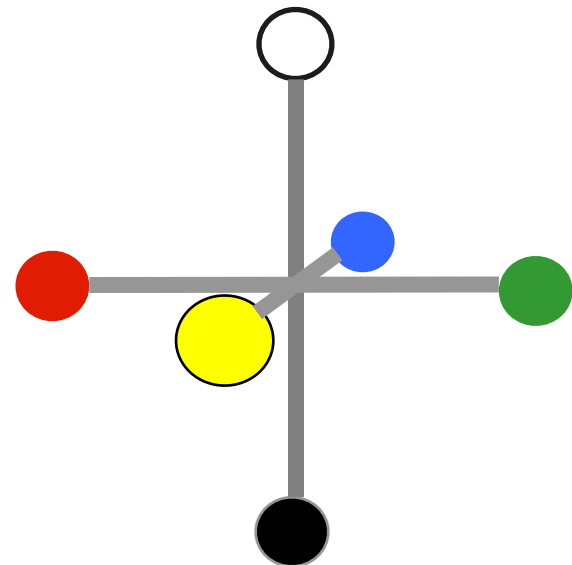


*[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]*



# Opponent color and color deficiency

- perceptual processing before optic nerve
  - one achromatic luminance channel ( $L^*$ )
    - edge detection through luminance contrast
  - 2 chroma channels
    - red-green ( $a^*$ ) & yellow-blue axis ( $b^*$ )



Luminance information



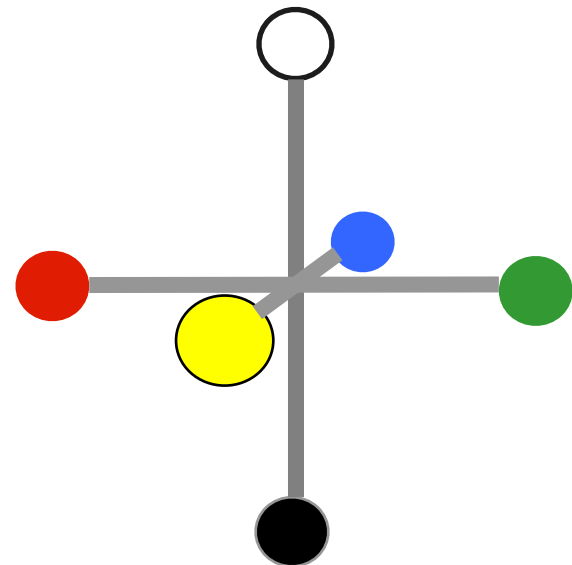
Chroma information



[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]

# Opponent color and color deficiency

- perceptual processing before optic nerve
  - one achromatic luminance channel ( $L^*$ )
    - edge detection through luminance contrast
  - 2 chroma channels
    - red-green ( $a^*$ ) & yellow-blue axis ( $b^*$ )
- “colorblind”: degraded acuity, one axis
  - 8% of men are red/green color deficient
  - blue/yellow is rare



Luminance information



Chroma information



[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]

# Designing for color deficiency: Check with simulator



**Normal vision**



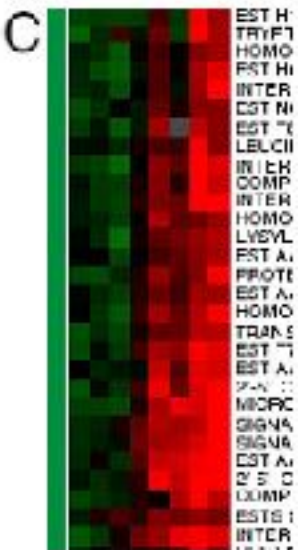
**Deuteranope  
*green-weak***



**Protanope  
*red-weak***



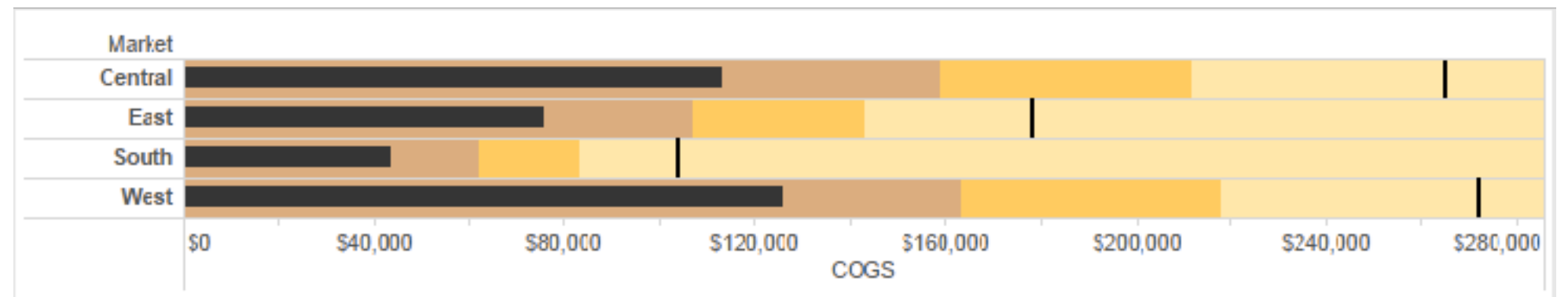
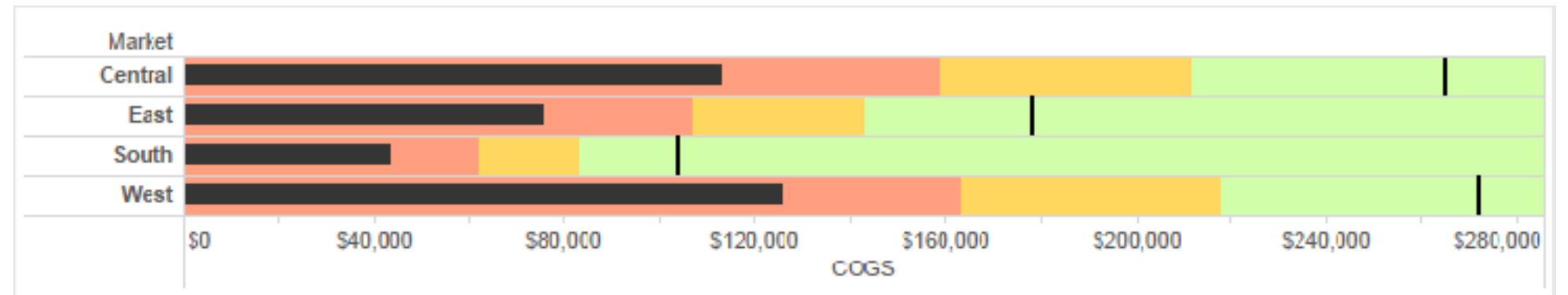
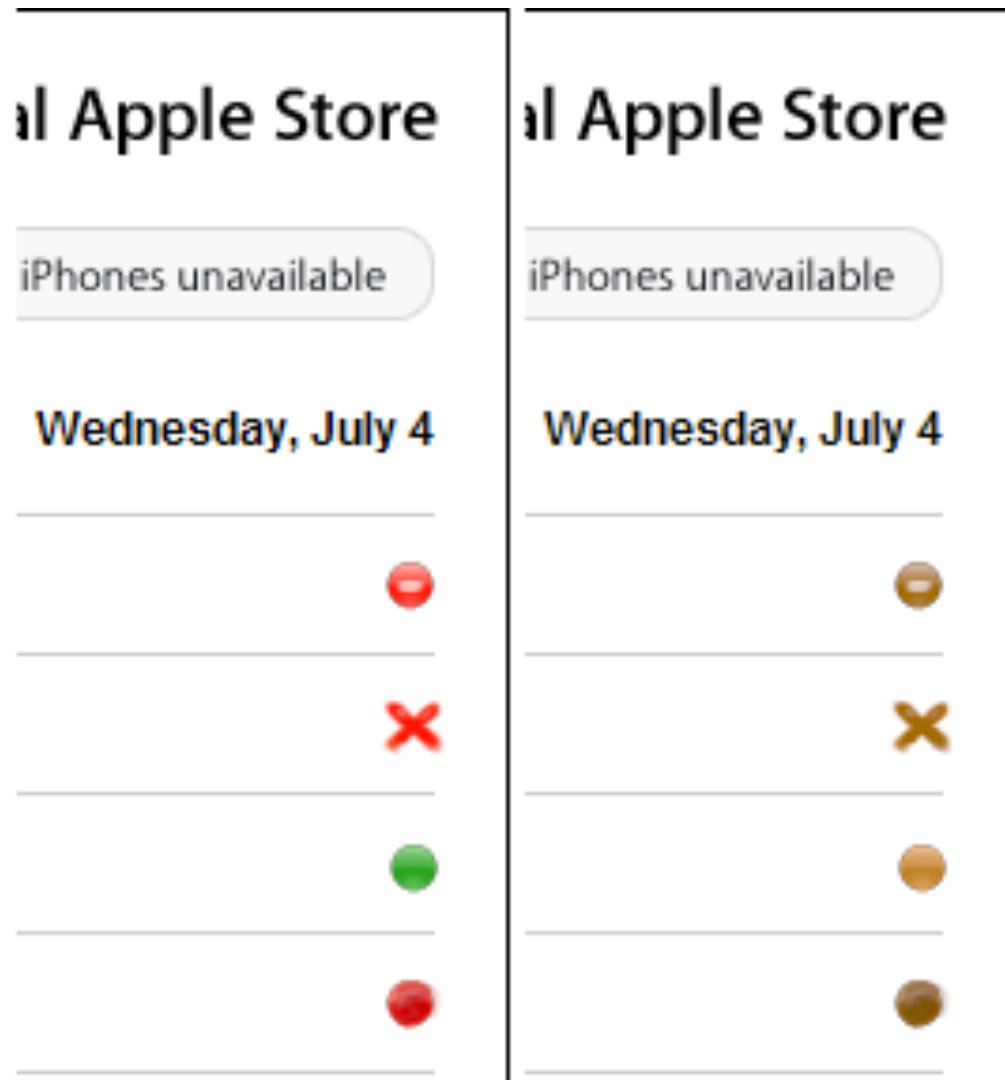
**Tritanope  
*blue-weak***



<https://www.color-blindness.com/coblis-color-blindness-simulator/>

# Designing for color deficiency: Avoid encoding by hue alone

- redundantly encode
  - vary luminance
  - change shape

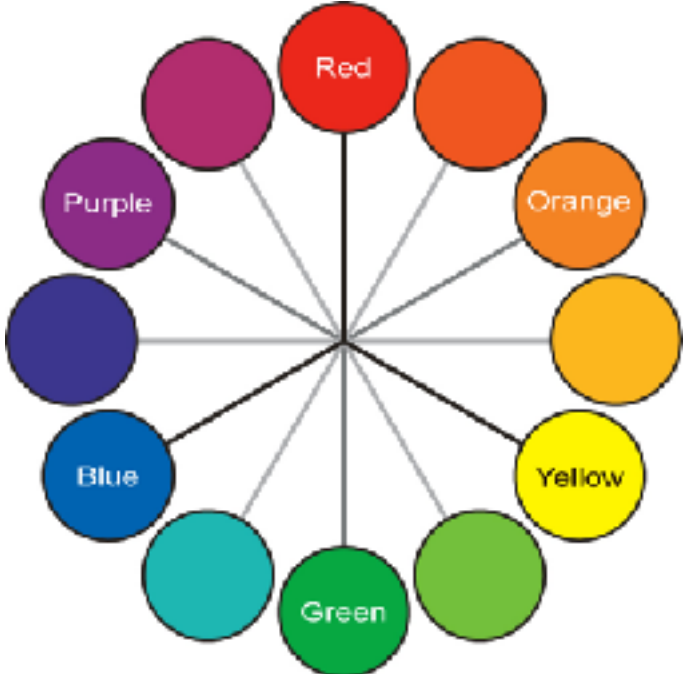


Deuteranope simulation

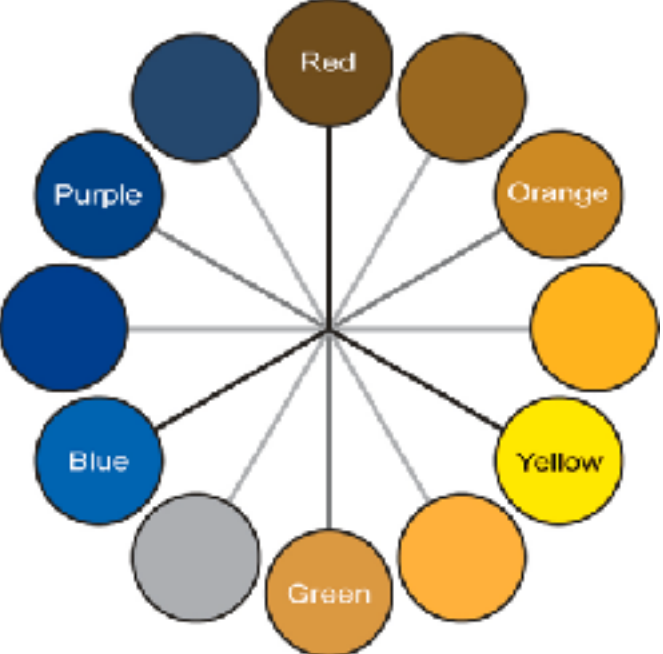
Change the shape

Vary luminance

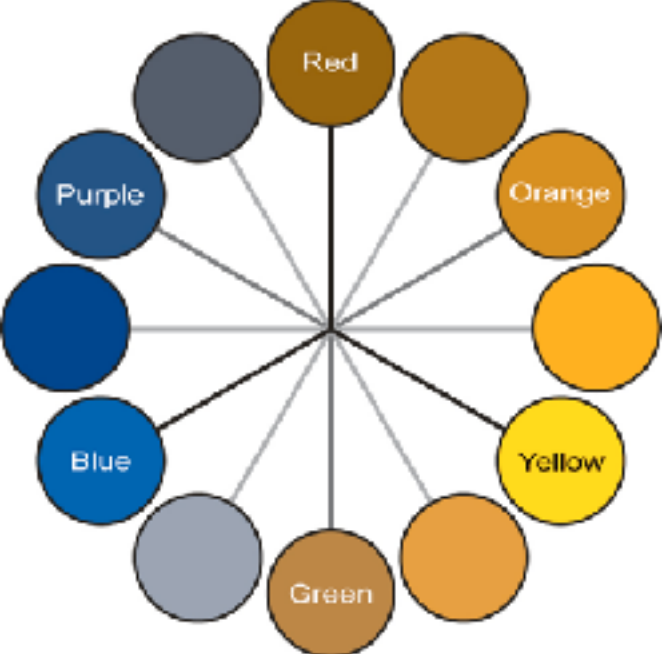
# Color deficiency: Reduces color to 2 dimensions



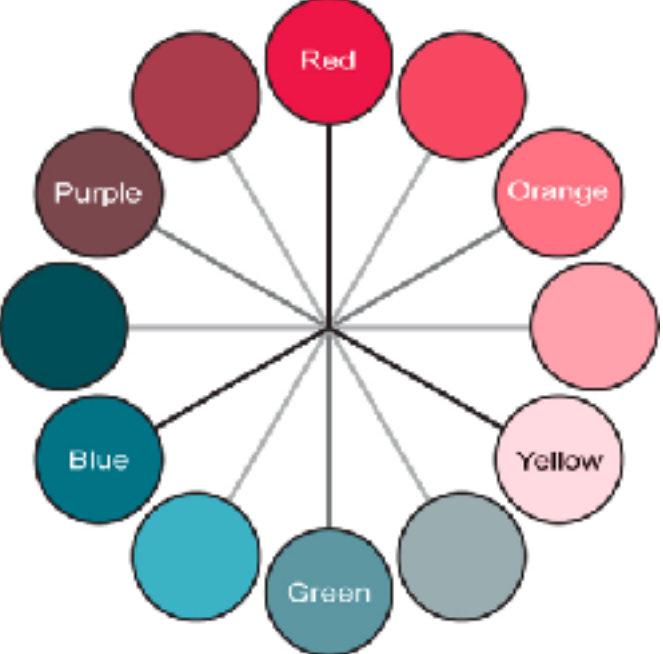
**Normal**



**Protanope**



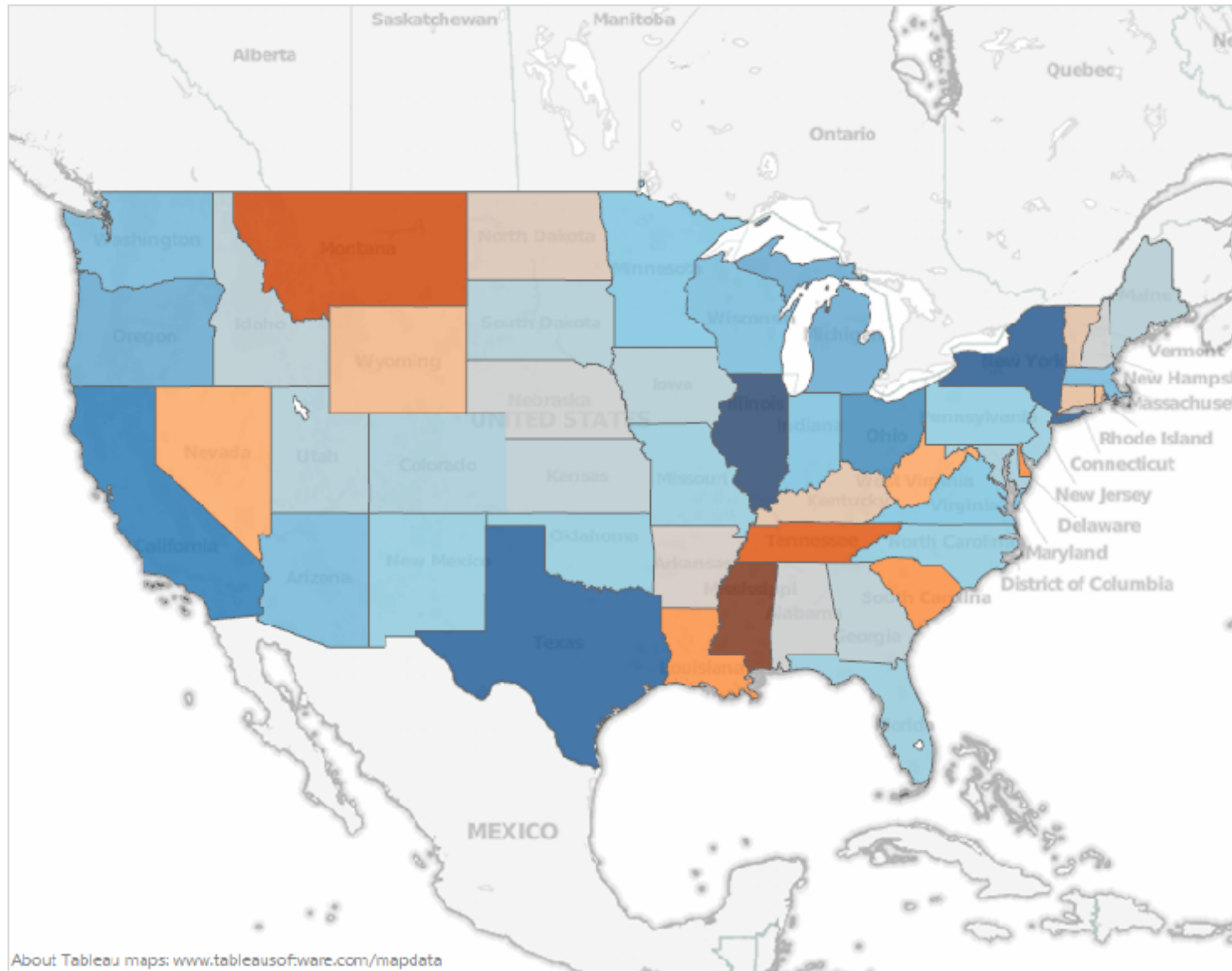
**Deuteranope**



**Tritanope**

[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]

# Designing for color deficiency: Blue-Orange is safe



[Seriously Colorful: Advanced Color Principles & Practices. Stone.Tableau Customer Conference 2014.]

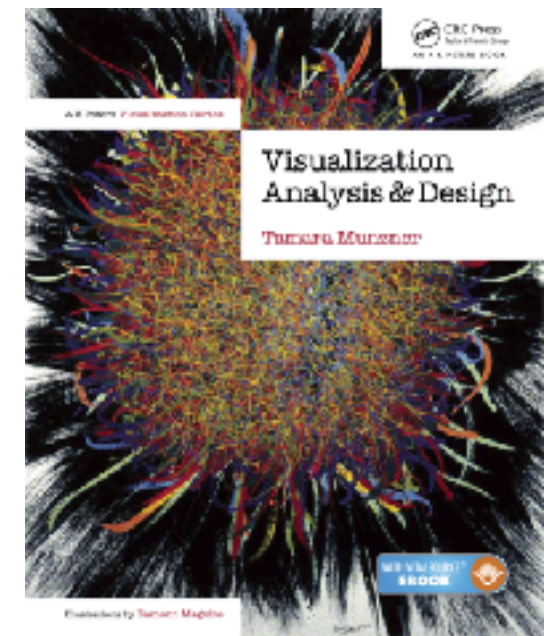
# Visualization Analysis & Design

## *Interactive Views (Ch 11/12)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# How to handle complexity: I previous strategy

→ *Derive*

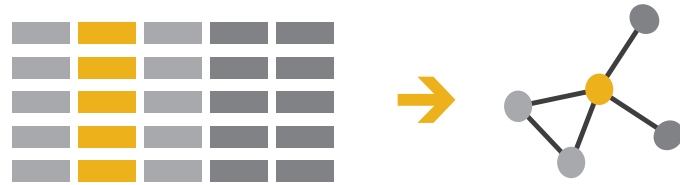


- derive new data to show within view



# How to handle complexity: 1 previous strategy + 2 more

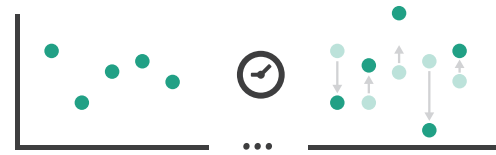
→ *Derive*



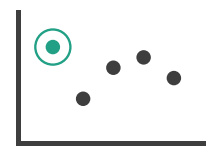
- derive new data to show within view
- change view over time
- facet across multiple views

Manipulate

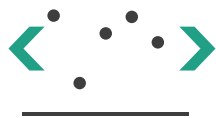
→ Change



→ Select

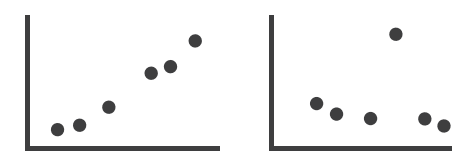


→ Navigate

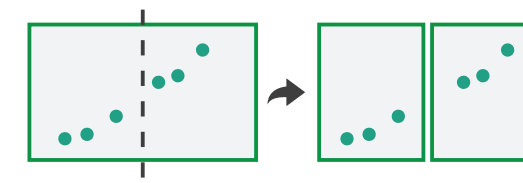


Facet

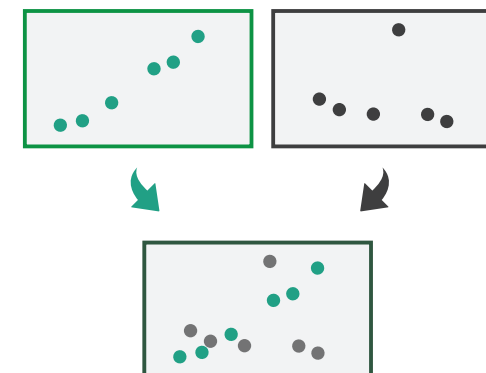
→ Juxtapose



→ Partition



→ Superimpose



# Manipulate View

# Manipulate

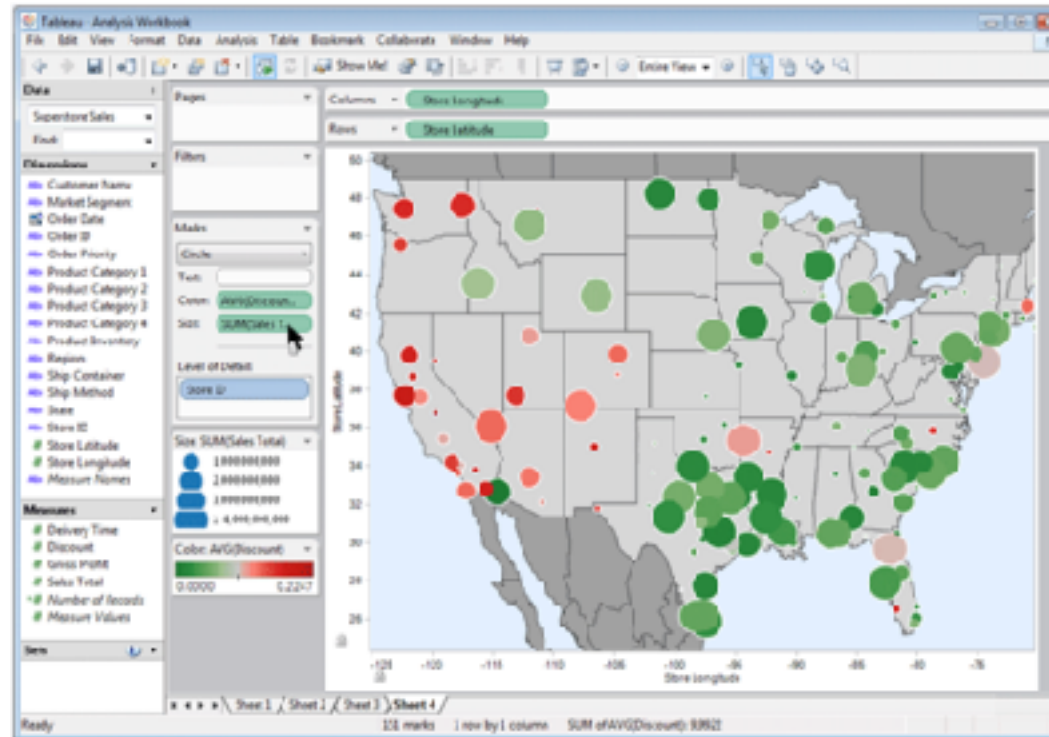
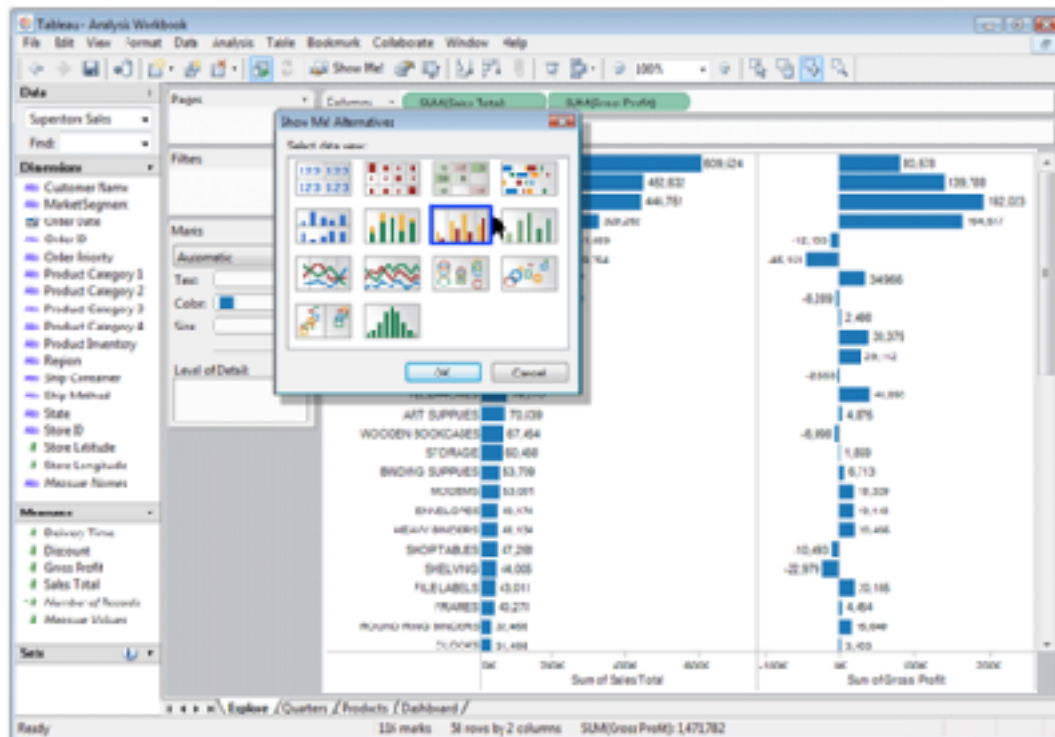
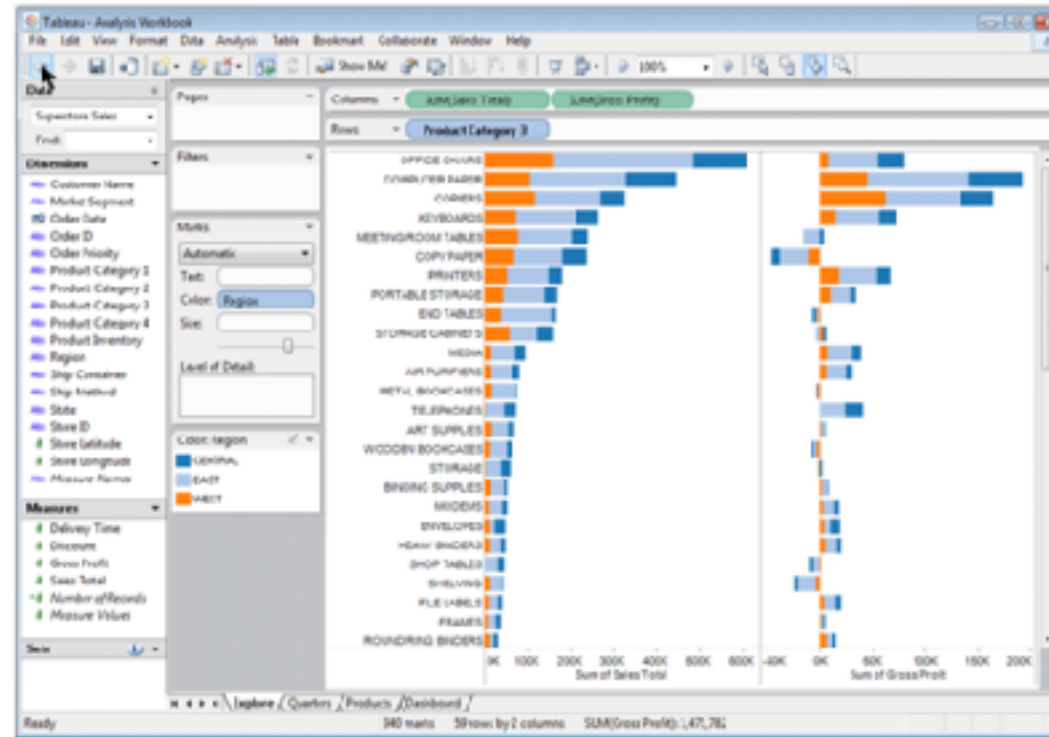
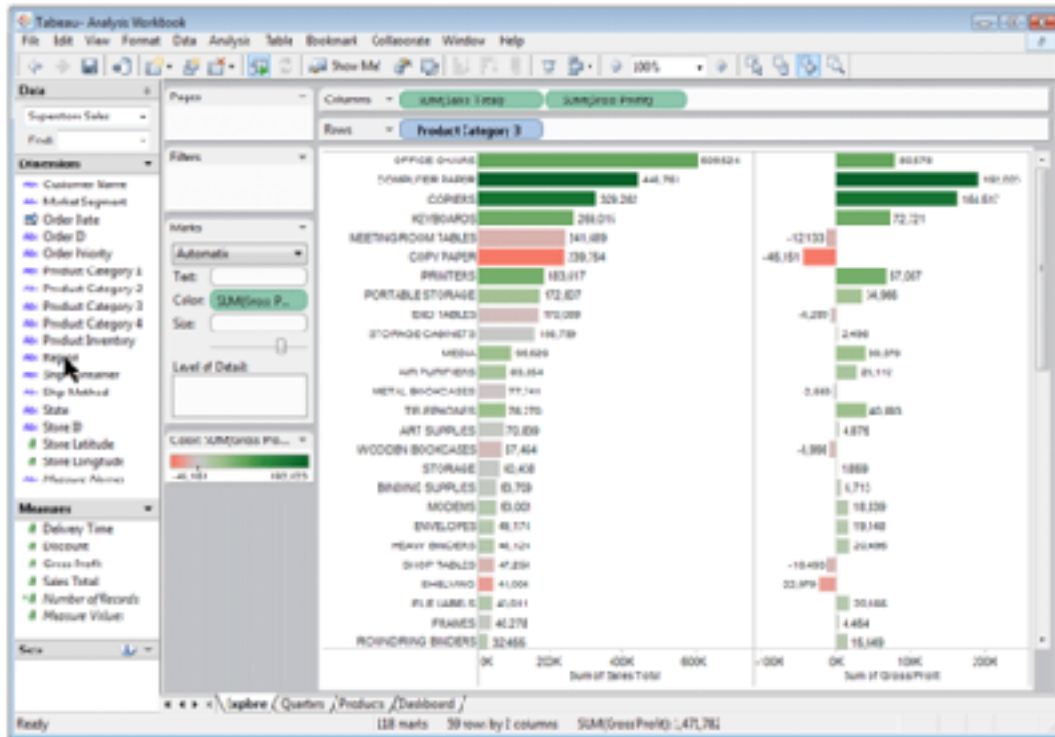
## → Change over Time



# Change over time

- change any of the other choices
  - encoding itself
  - parameters
  - arrange: rearrange, reorder
  - aggregation level, what is filtered...
  - interaction entails change
- powerful & flexible

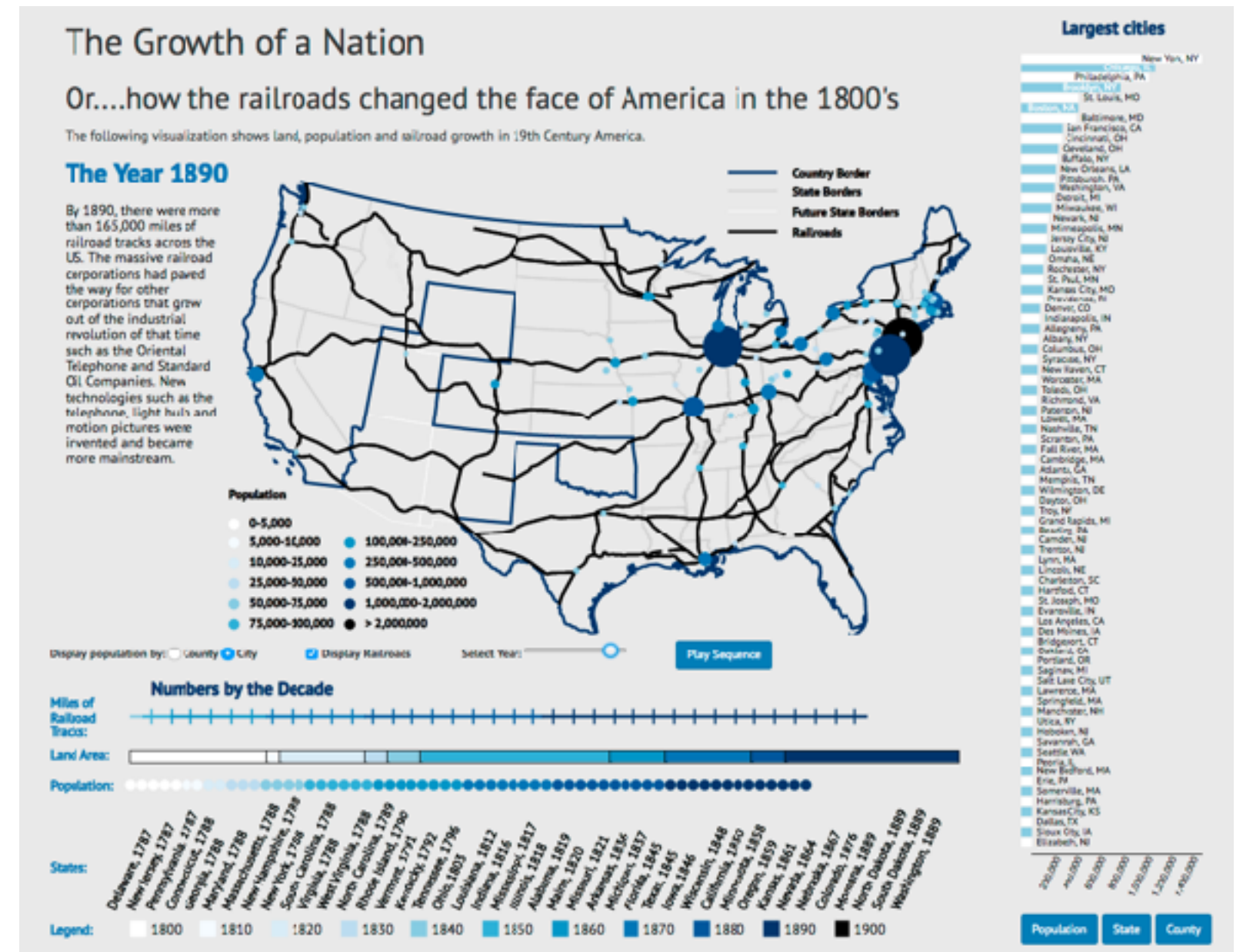
# Idiom: Re-encode



made with Tableau, <http://tableausoftware.com>

# Idiom: Change parameters

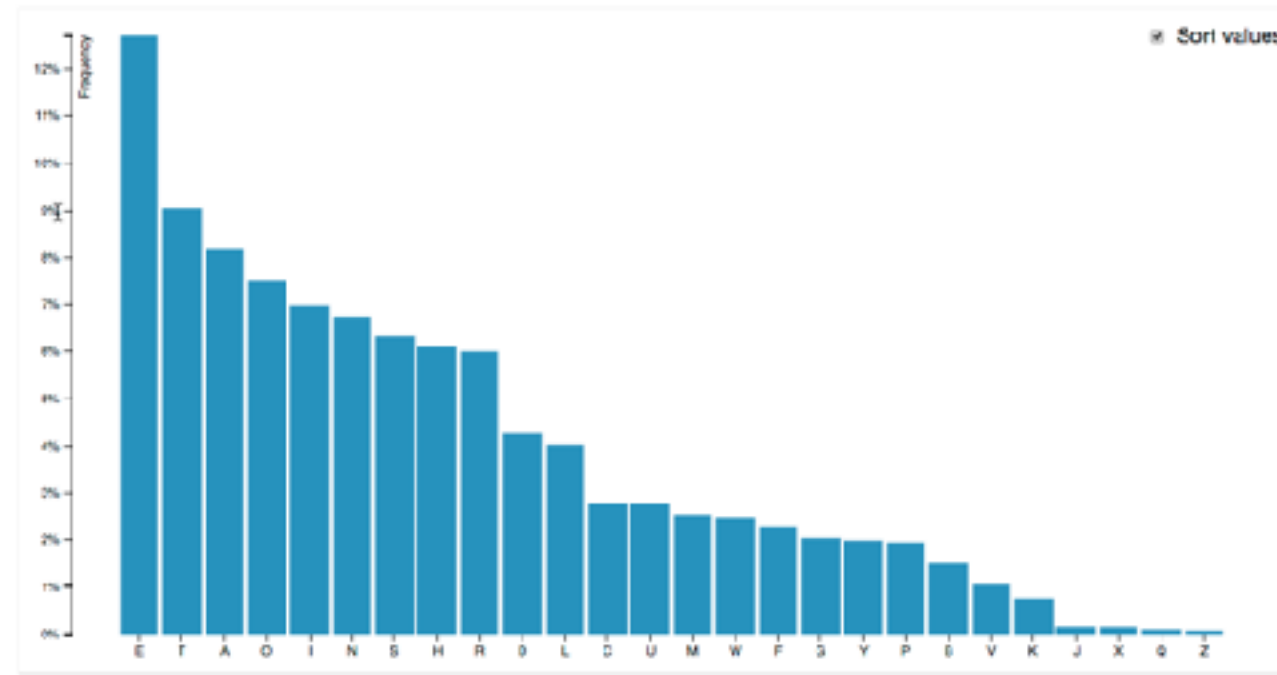
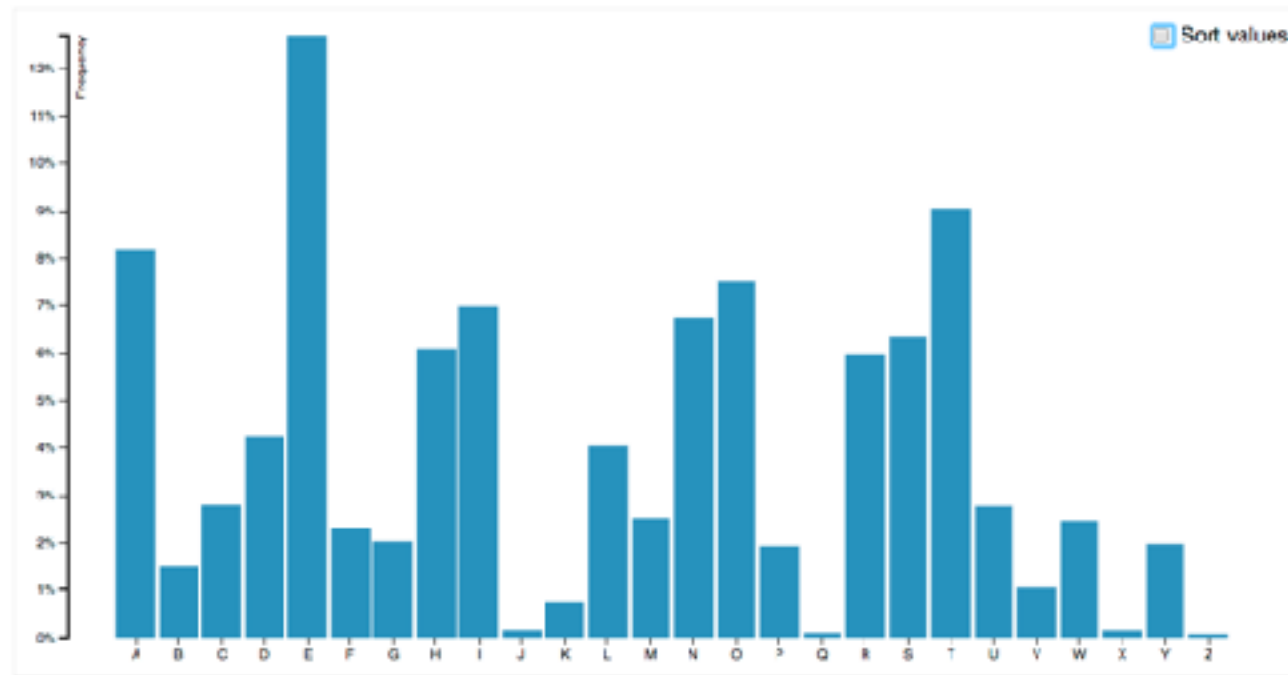
- widgets and controls
  - sliders, buttons, radio buttons, checkboxes, dropdowns/comboboxes
- pros
  - clear affordances, self-documenting (with labels)
- cons
  - uses screen space
- design choices
  - separated vs interleaved
    - controls & canvas



[Growth of a Nation](<http://laurenwood.github.io/>)  
made with D3

# Idiom: **Change order/arrangement**

- what: simple table
- how: data-driven reordering
- why: find extreme values, trends



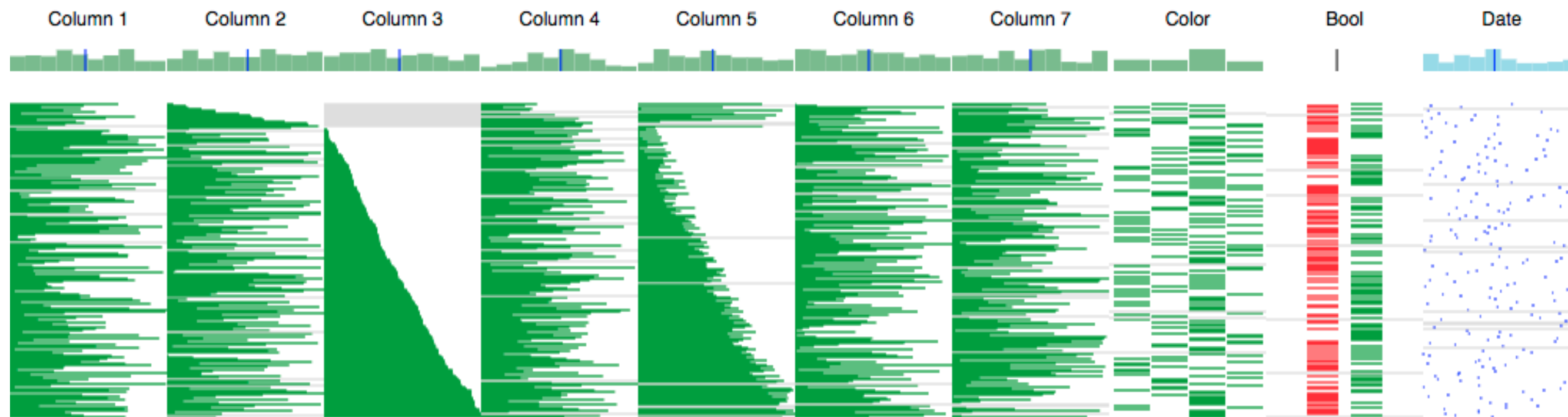
[Sortable Bar Chart] <https://observablehq.com/@d3/sortable-bar-chart>

made with D3

# Idiom: **Reorder**

## System: **DataStripes**

- what: table with many attributes
- how: data-driven reordering by selecting column
- why: find correlations between attributes



[\[http://carlmanaster.github.io/datastripes/\]](http://carlmanaster.github.io/datastripes/)

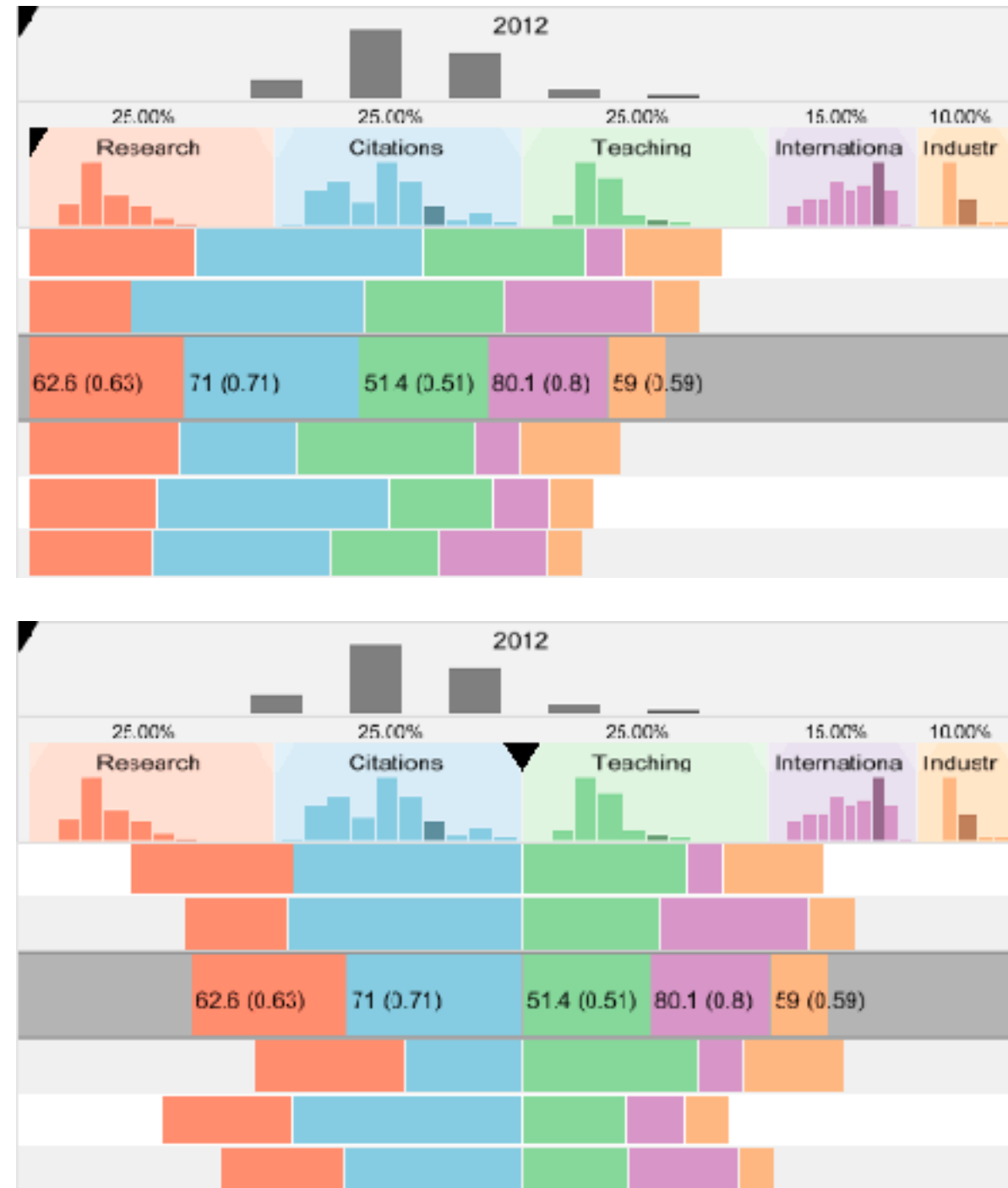
made with D3



# Idiom: **Change alignment**

# System: **LineUp**

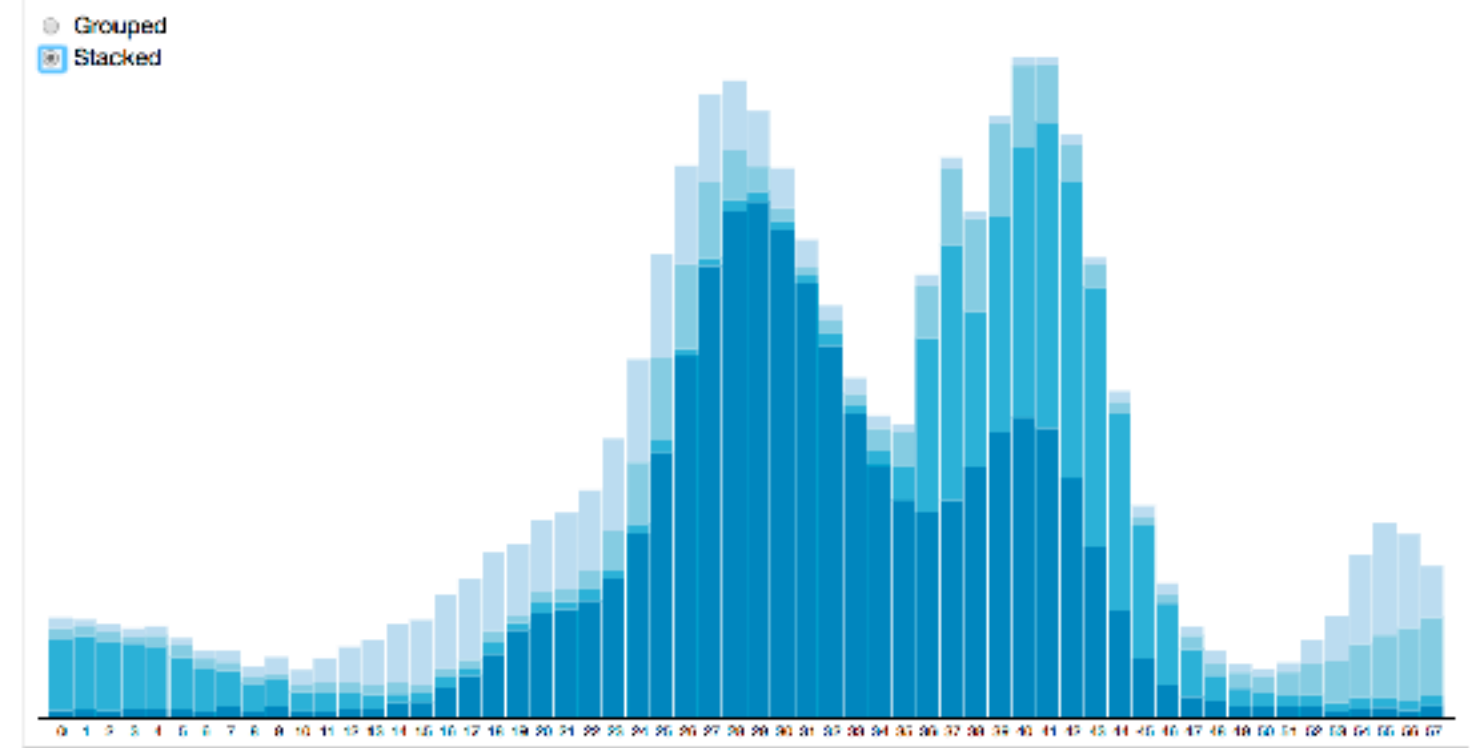
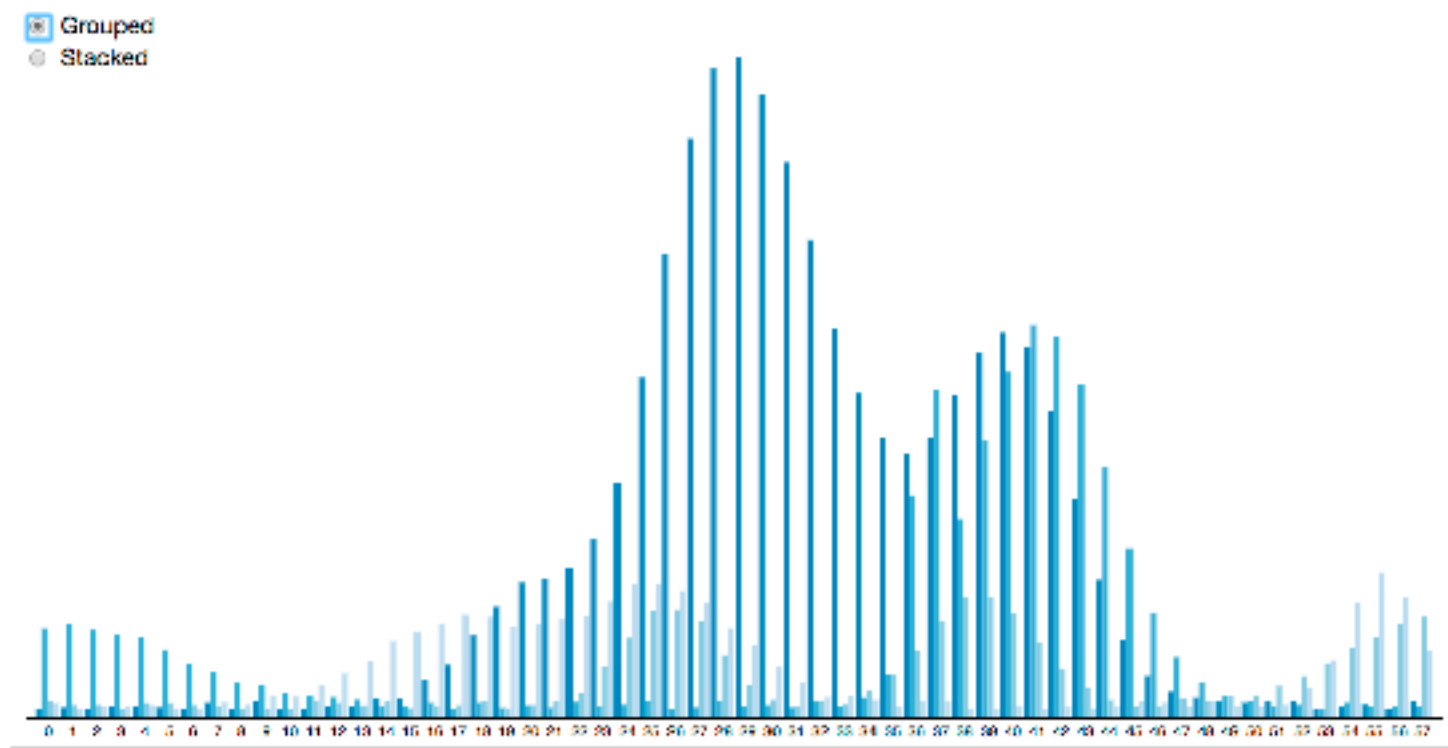
- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison



[LineUp: Visual Analysis of Multi-Attribute Rankings. Gratzl, Lex, Gehlenborg, Pfister, and Streit. *IEEE Trans. Visualization and Computer Graphics* (Proc. InfoVis 2013) 19:12 (2013), 2277–2286.]

# Idiom: **Animated transitions** - visual encoding change

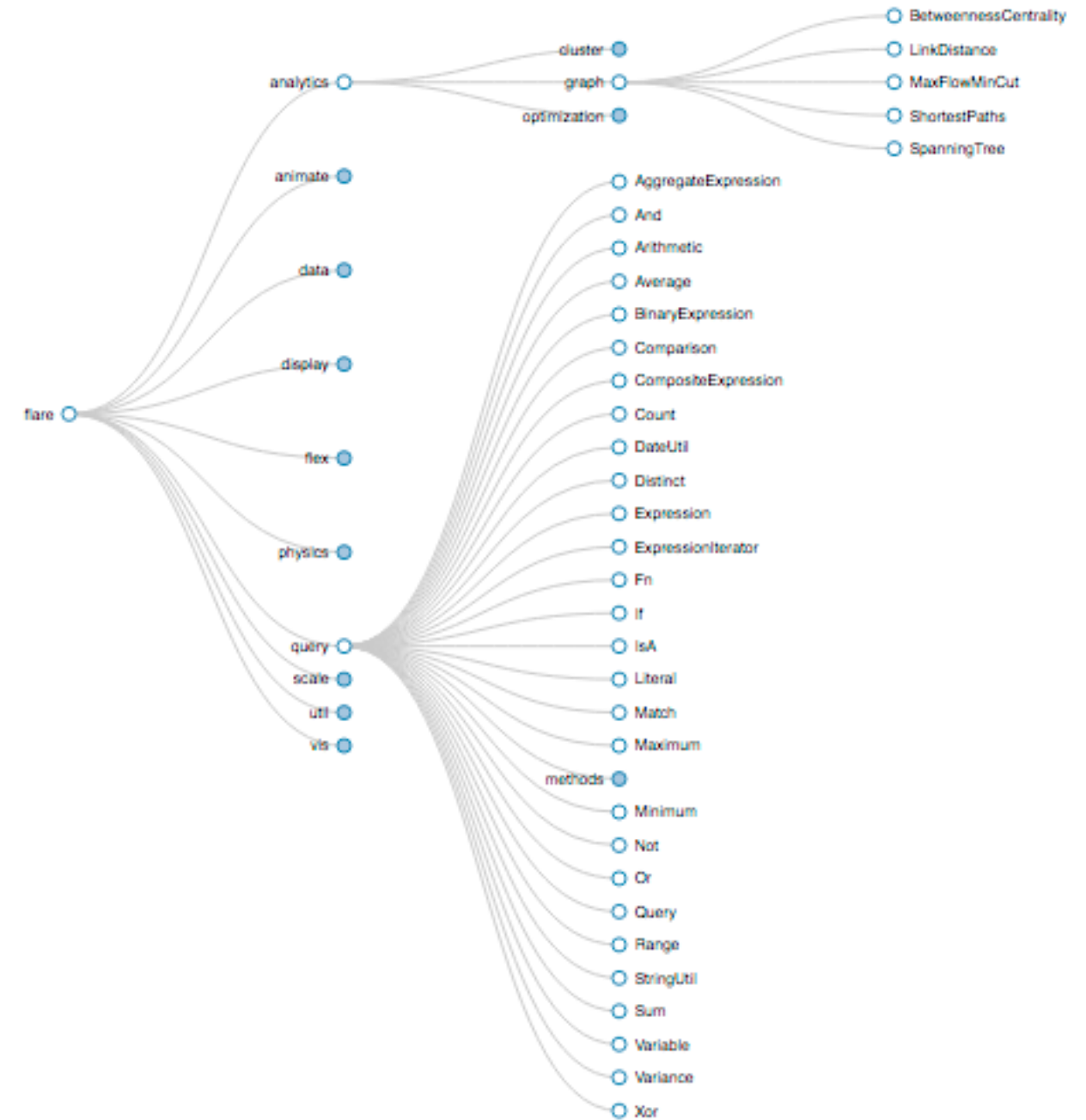
- smooth transition from one state to another
  - alternative to jump cuts, supports item tracking
    - best case for animation
  - staging to reduce cognitive load



[Stacked to Grouped Bars] <https://observablehq.com/@d3/stacked-to-grouped-bars>

# Idiom: **Animated transition** - tree detail

- animated transition
  - network drilldown/rollup

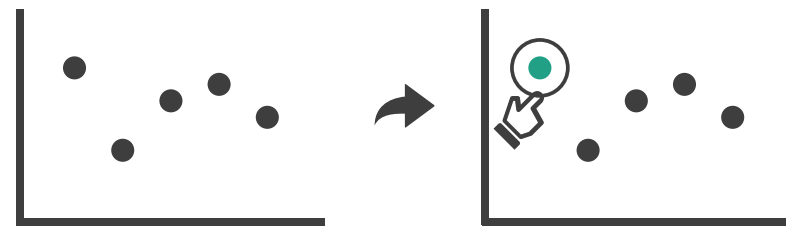


# Manipulate

## → Change over Time

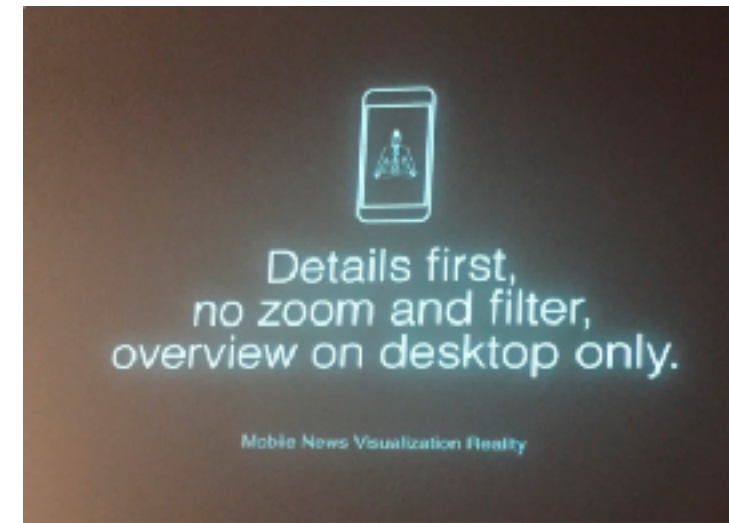


## → Select



# Interaction technology

- what do you design for?
  - mouse & keyboard on desktop?
    - large screens, hover, multiple clicks
  - touch interaction on mobile?
    - small screens, no hover, just tap
  - gestures from video / sensors?
    - ergonomic reality vs movie bombast
  - eye tracking?



Data visualization and the news - Gregor Aisch (37 min)  
[vimeo.com/182590214](https://vimeo.com/182590214)

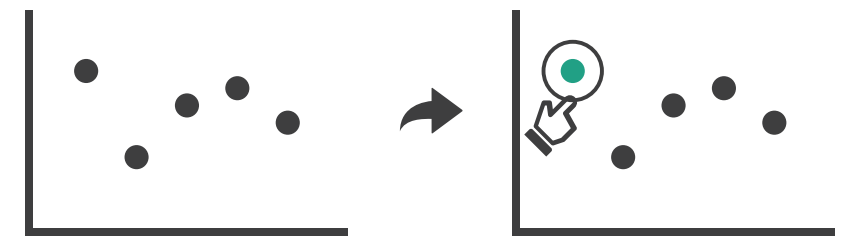


I Hate Tom Cruise - Alex Kauffmann (5 min)  
[www.youtube.com/watch?v=QXLfT9sFcbc](https://www.youtube.com/watch?v=QXLfT9sFcbc)

# Selection

- selection: basic operation for most interaction
- design choices
  - how many selection types?
    - interaction modalities
      - click/tap (heavyweight) vs hover (lightweight but not available on most touchscreens)
      - multiple click types (shift-click, option-click, ...)
      - proximity beyond click/hover (touching vs nearby vs distant)
    - application semantics
      - adding to selection set vs replacing selection
      - can selection be null?
        - ex: toggle so nothing selected if click on background
      - primary vs secondary (ex: source/target nodes in network)
      - group membership (add/delete items, name group, ...)

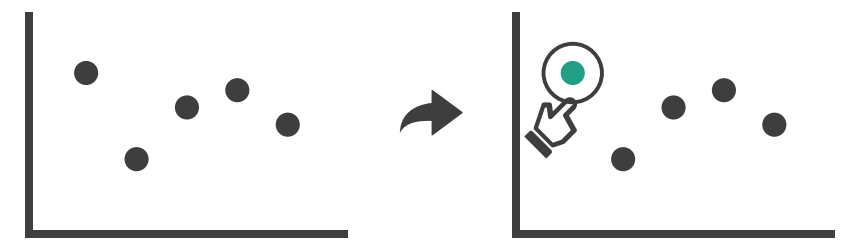
➔ Select



# Highlighting

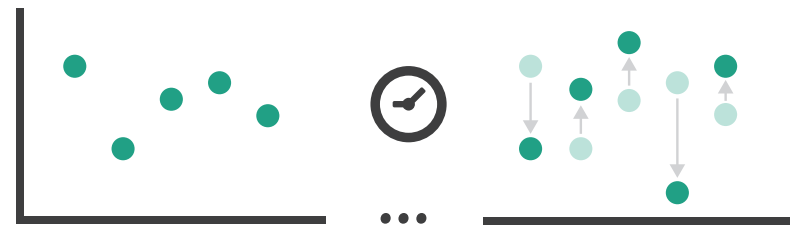
- highlight: change visual encoding for selection targets
  - visual feedback closely tied to but separable from selection (interaction)
- design choices: typical visual channels
  - change item color
    - but hides existing color coding
  - add outline mark
  - change size (ex: increase outline mark linewidth)
  - change shape (ex: from solid to dashed line for link mark)
- unusual channels: motion
  - motion: usually avoid for single view
    - with multiple views, could justify to draw attention to other views

➔ Select

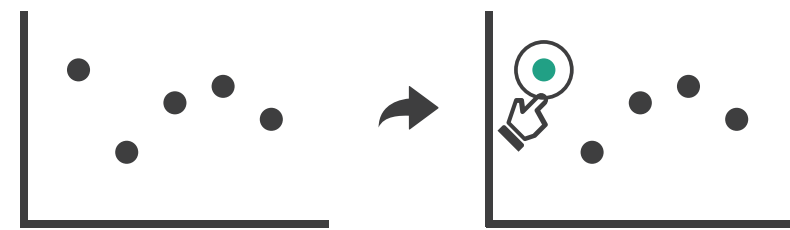


# Manipulate

## → Change over Time



## → Select



## → Navigate

### → Item Reduction

#### → Zoom

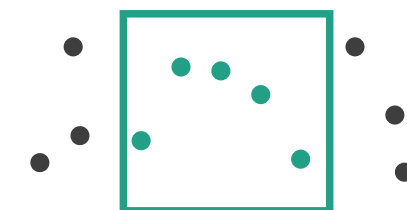
*Geometric* or *Semantic*



#### → Pan/Translate



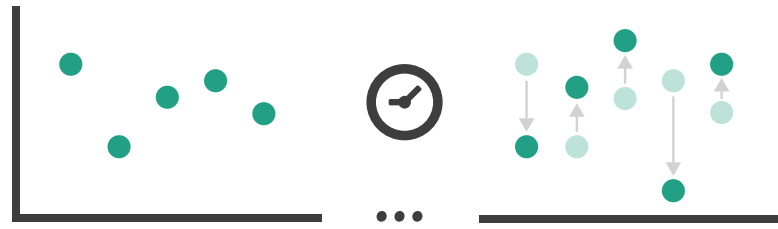
#### → Constrained





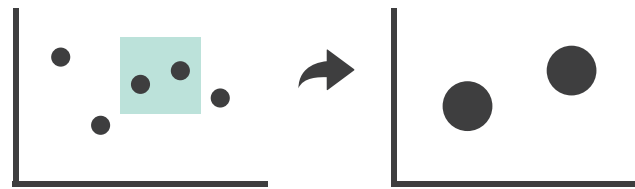
# Manipulate

## → Change over Time

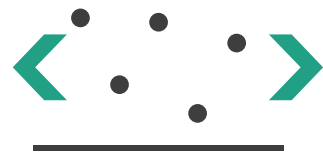


## → Navigate

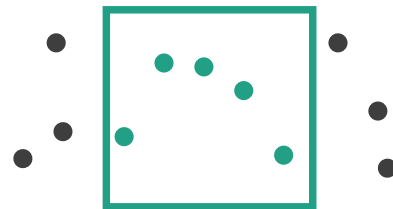
→ Zoom  
*Geometric*



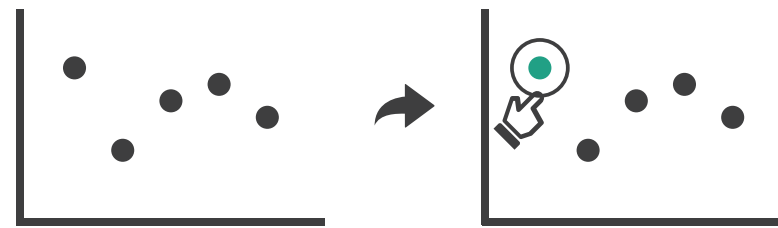
→ Pan/Translate



→ Constrained



## → Select

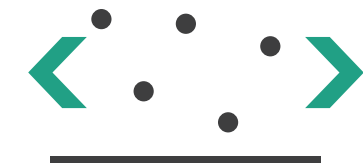


# Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
    - move up/down/sideways

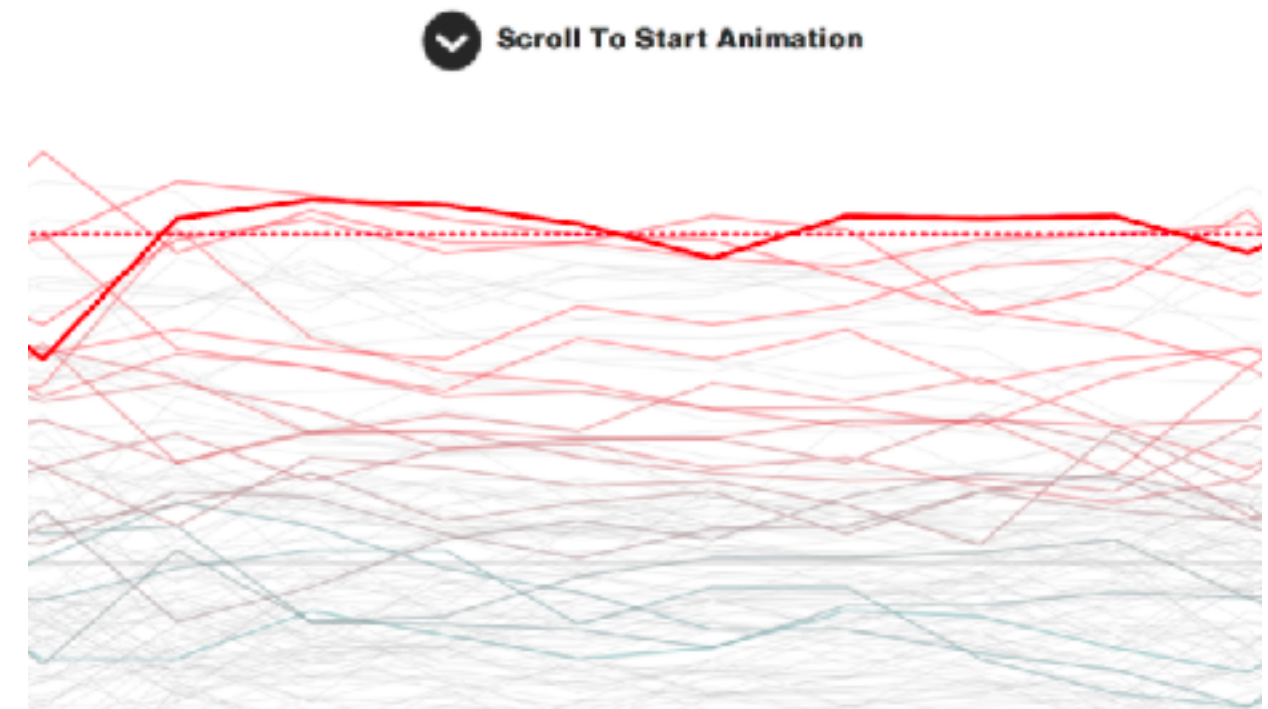
➔ Navigate

➔ *Pan/Translate*



# Idiom: Scrollytelling

- how: navigate page by scrolling (panning down)
- pros:
  - familiar & intuitive, from standard web browsing
  - linear (only up & down) vs possible overload of click-based interface choices
- cons:
  - full-screen mode may lack affordances
  - scrolljacking, no direct access
  - unexpected behaviour
  - continuous control for discrete steps



[How to Scroll, Bostock](<https://bost.ocks.org/mike/scroll/>)

<https://eagereyes.org/blog/2016/the-scrollytelling-scourge>

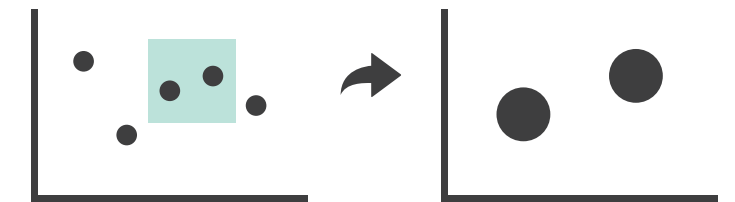
# Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
    - move up/down/sideways
  - rotate/spin
    - typically in 3D
  - zoom in/out
    - enlarge/shrink world == move camera closer/further
    - geometric zoom: standard, like moving physical object

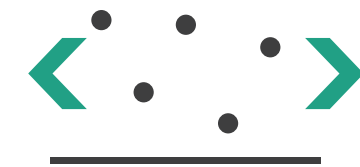
## ➔ Navigate

➔ Zoom

*Geometric*



➔ Pan/Translate



# Navigate: Unconstrained vs constrained

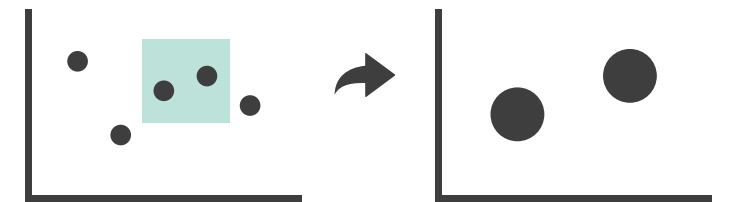
- unconstrained navigation
  - easy to implement for designer
  - hard to control for user
    - easy to overshoot/undershoot
- constrained navigation
  - typically uses animated transitions
  - trajectory automatically computed based on selection
    - just click; selection ends up framed nicely in final viewport

## ➔ Navigate

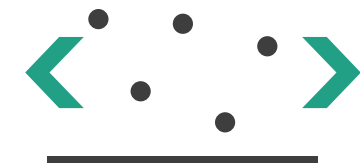
➔ Item Reduction

➔ Zoom

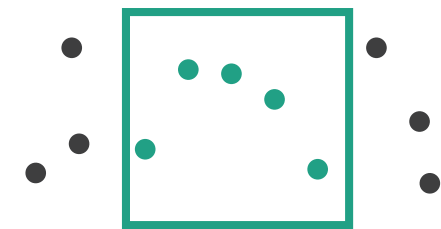
*Geometric* or *Semantic*



➔ Pan/Translate



➔ Constrained



# Idiom: **Animated transition + constrained navigation**

- example: geographic map
  - simple zoom, only viewport changes, shapes preserved

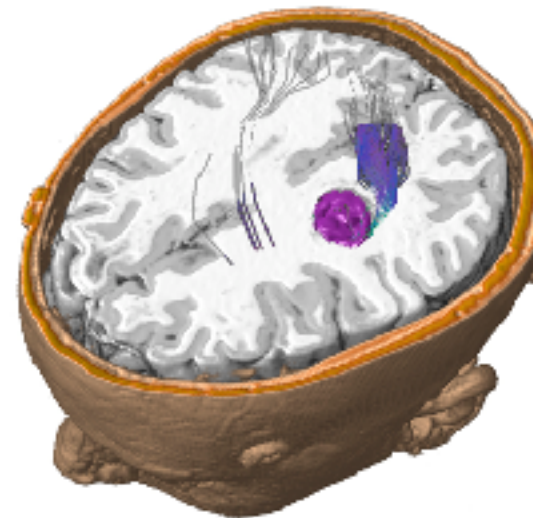
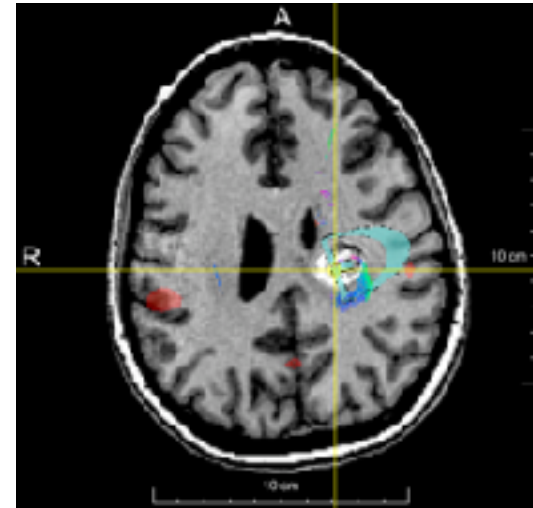
Zoom to Bounding Box



*[Zoom to Bounding Box] <https://observablehq.com/@d3/zoom-to-bounding-box>*

# Navigate: Reducing attributes

- continuation of camera metaphor
  - slice
    - show only items matching specific value for given attribute: slicing plane
    - axis aligned, or arbitrary alignment
  - cut
    - show only items on far side of plane from camera
  - project
    - change mathematics of image creation
      - orthographic
      - perspective
      - many others: Mercator, cabinet, ...

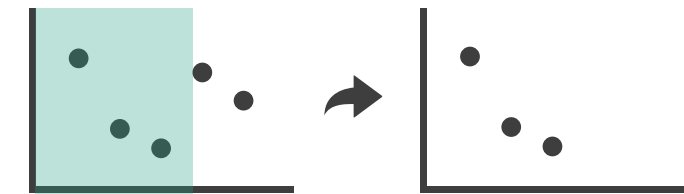


→ Attribute Reduction

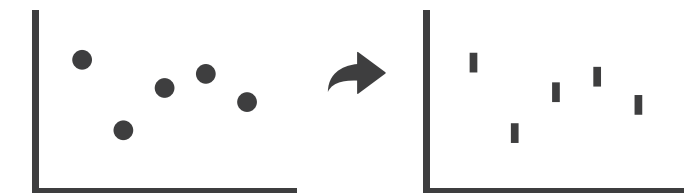
→ *Slice*



→ *Cut*



→ *Project*



# Interaction benefits

- interaction pros
  - major advantage of computer-based vs paper-based visualization
  - flexible, powerful, intuitive
    - exploratory data analysis: change as you go during analysis process
    - fluid task switching: different visual encodings support different tasks
  - animated transitions provide excellent support
    - empirical evidence that animated transitions help people stay oriented



# Interaction limitations

- interaction has a time cost
  - sometimes minor, sometimes significant
  - degenerates to human-powered search in worst case
- remembering previous state imposes cognitive load
- controls may take screen real estate
  - or invisible functionality may be difficult to discover (lack of affordances)
- users may not interact as planned by designer
  - NYTimes logs show ~90% don't interact beyond scrollytelling - Aisch, 2016

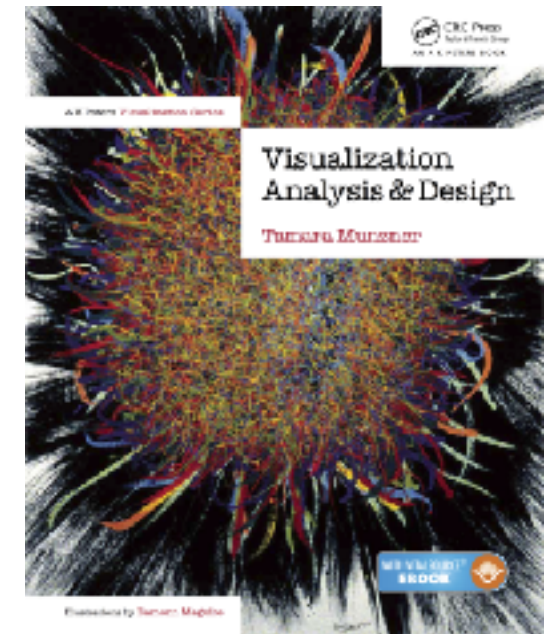
# Visualization Analysis & Design

## *Interactive Views (Ch 11/12) II*

**Tamara Munzner**

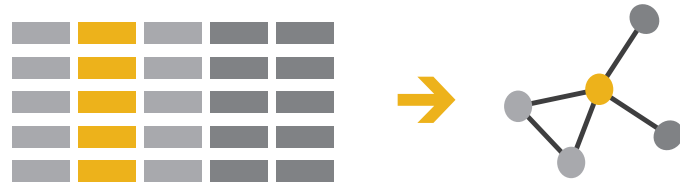
Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)



# How to handle complexity: 1 previous strategy + 2 more

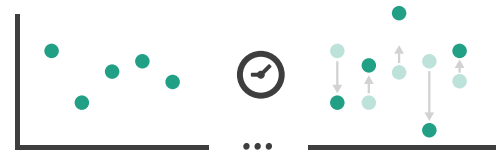
→ *Derive*



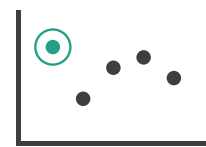
- derive new data to show within view
- change view over time
- facet across multiple views

## Manipulate

① Change



② Select

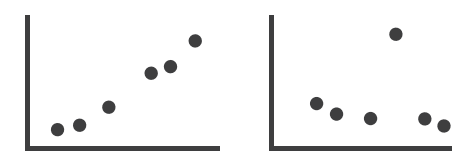


③ Navigate

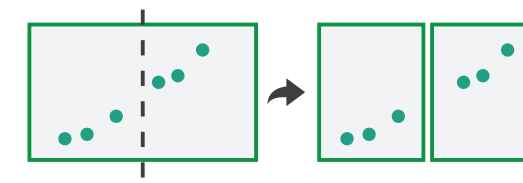


## Facet

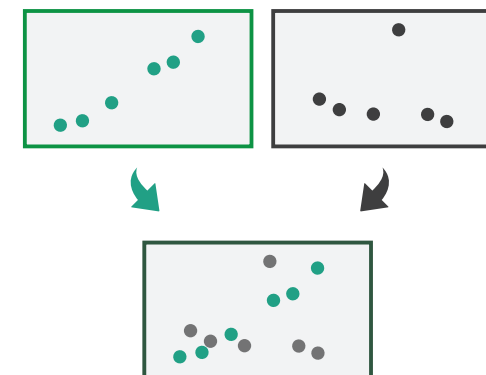
① Juxtapose



② Partition



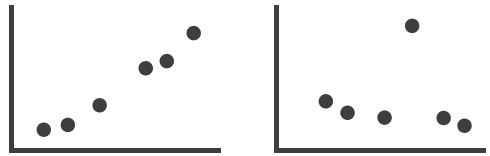
③ Superimpose



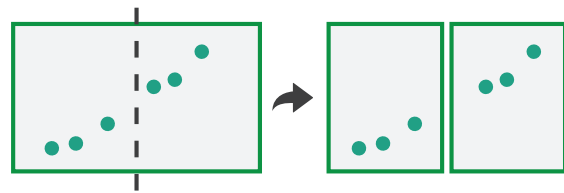
# Multiple Views

# Facet

## → Juxtapose



## → Partition

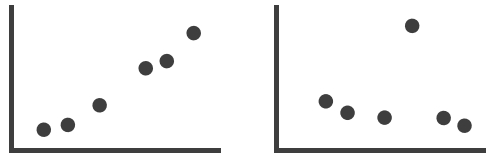


## → Superimpose

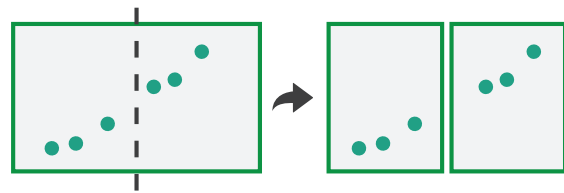


# Facet

## → Juxtapose



## → Partition



## → Superimpose



# Juxtapose and coordinate views

→ Share Encoding: Same/Different

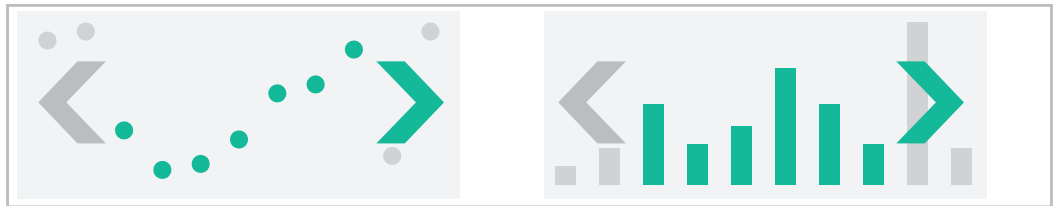
→ *Linked Highlighting*



→ Share Data: All/Subset/None



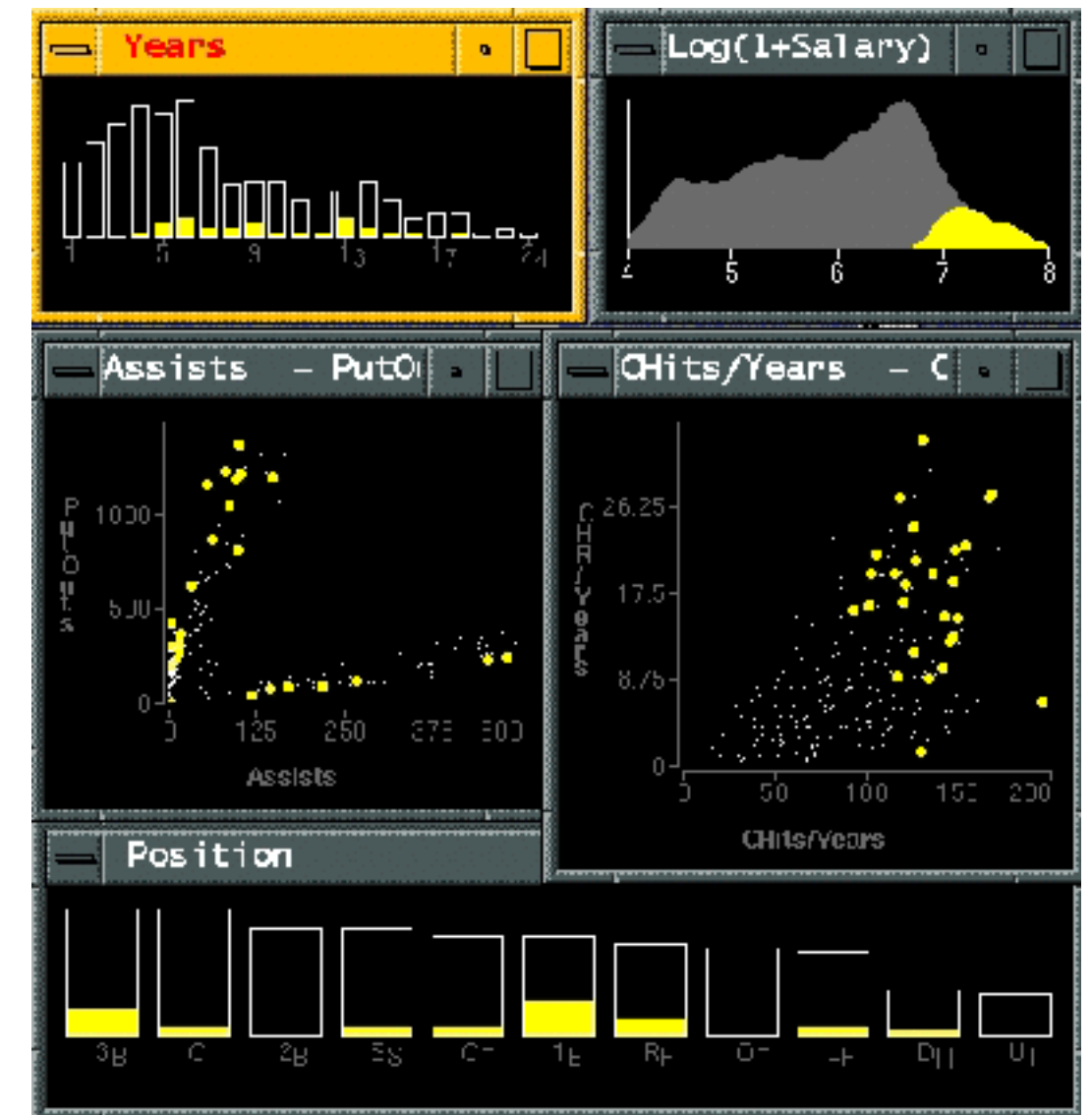
→ Share Navigation



# Idiom: **Linked highlighting**

System: **EDV**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom
- encoding: different
  - *multiform*
- data: all shared
  - all **items** shared
  - different **attributes** across the views
- aka: brushing and linking

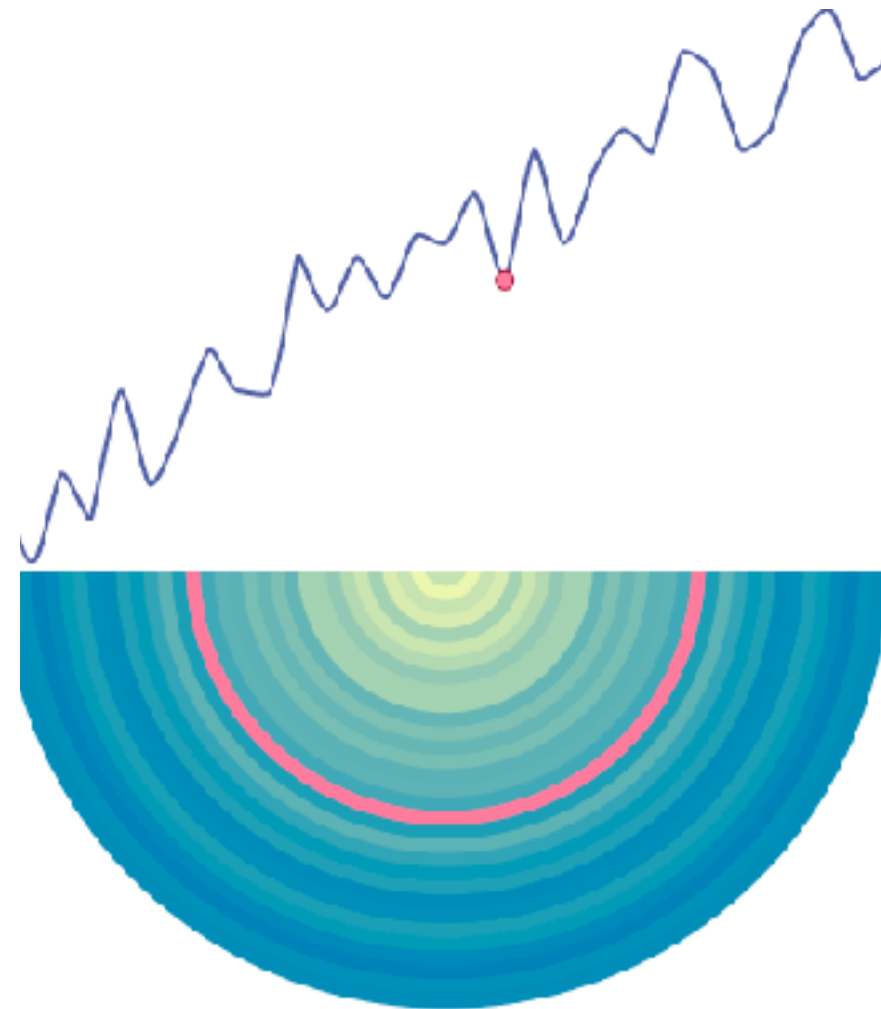


[Visual Exploration of Large Structured Datasets. Wills.  
Proc. New Techniques and Trends in Statistics (NTTS), pp. 237–246. IOS Press, 1995.]



# Linked views: Directionality

- unidirectional vs bidirectional linking
  - bidirectional almost always better!



<http://pbeshai.github.io/linked-highlighting-react-vega-redux/>

<https://medium.com/@pbesh/linked-highlighting-with-react-d3-js-and-reflux-16e9c0b2210b>

# Idiom: Overview-detail views

- encoding: same or different
  - ex: same (birds-eye map)
- data: subset shared
  - viewpoint differences: subset of data items
- navigation: shared
  - bidirectional linking
- other differences
  - (window size)

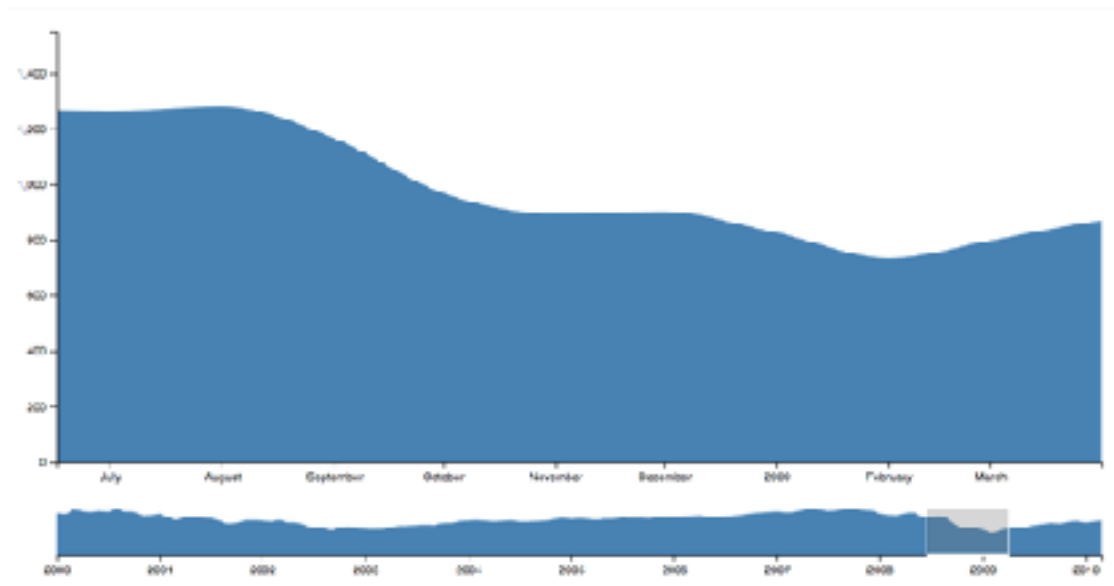
# System: Google Maps



[A Review of Overview+Detail, Zooming, and Focus+Context Interfaces.  
Cockburn, Karlson, and Bederson. *ACM Computing Surveys* 41:1 (2008), 1–31.]

# Idiom: Overview-detail navigation

- encoding: same or different
- data: subset shared
- navigation: shared
  - unidirectional linking
  - select in small overview, change extent in large detail view



<https://observablehq.com/@uwdata/interaction>

# Idiom: Tooltips

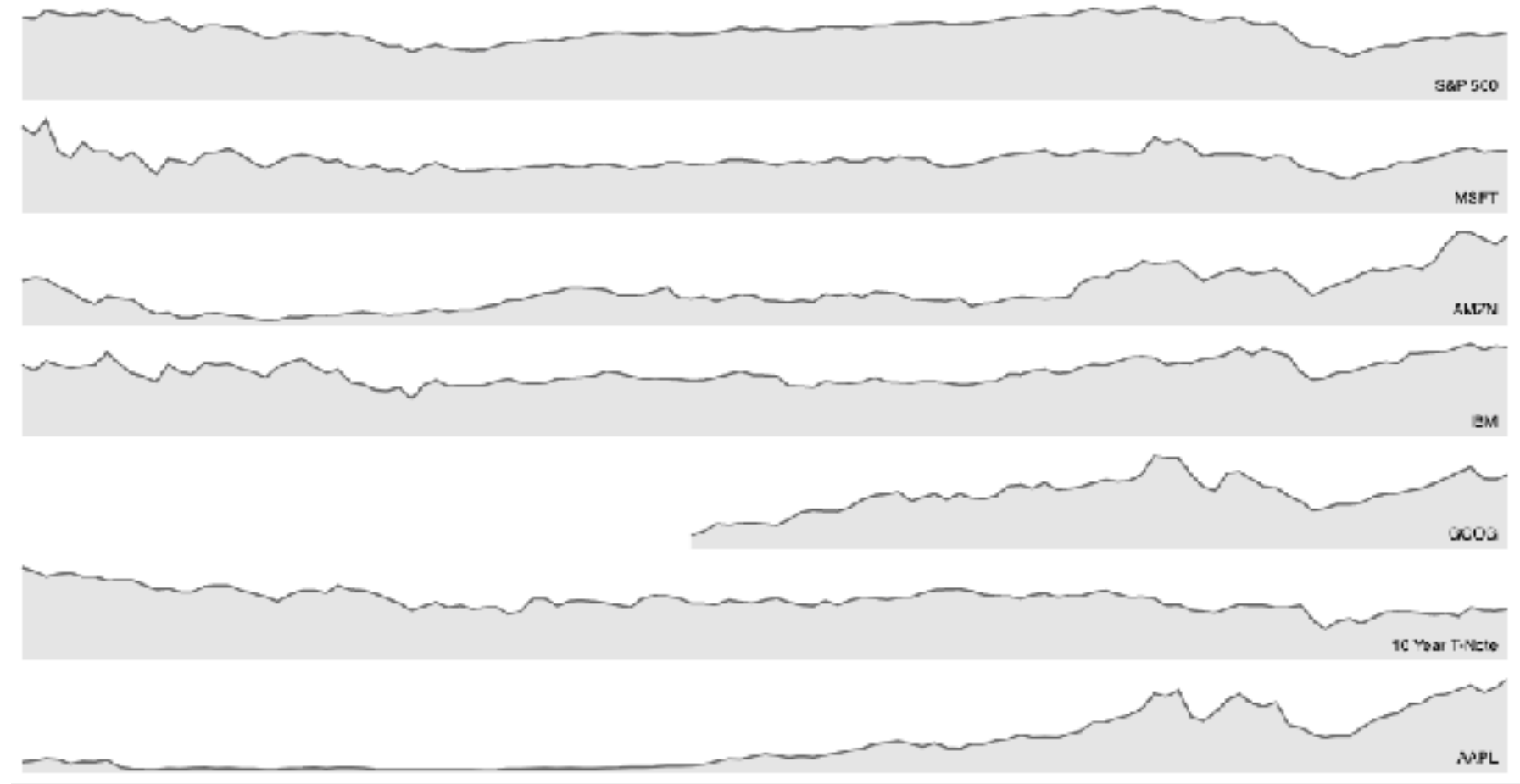
- popup information for selection
  - hover or click
  - specific case of detail view:  
provide useful additional detail on demand
  - beware: does not support overview!
    - always consider if there's a way to visually encode directly to provide overview
    - “If you make a rollover or tooltip, assume nobody will see it. If it's important, make it explicit.”
      - Gregor Aisch, NYTimes



[\[https://www.highcharts.com/demo/dynamic-master-detail\]](https://www.highcharts.com/demo/dynamic-master-detail)

# Idiom: **Small multiples**

- encoding: same
  - ex: line charts
- data: none shared
  - different slices of dataset
    - items or attributes
    - ex: stock prices for different companies



# Interactive small multiples

- linked highlighting:  
analogous item/attribute  
across views
  - same year highlighted across all charts if hover within any chart

## The Rise and Decline of Ask MetaFilter

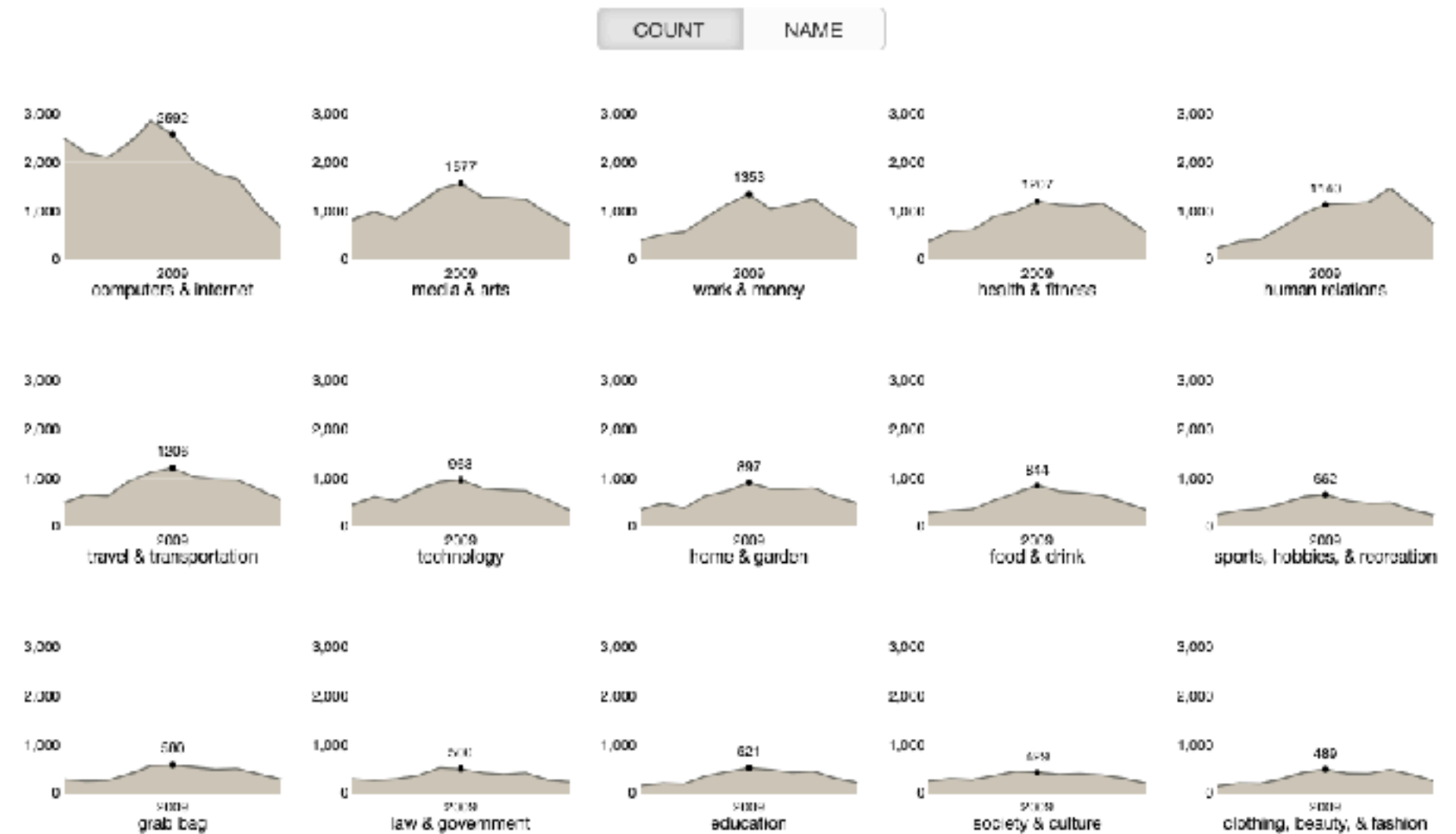
Metafilter's revenue has been on the decline, but has its content dried up as well?

Here we look at new posts on Ask Metafilter by category.

Categories like computers & internet have been dropping in use for a long time, most likely due to competition like Stack Overflow.

Other smaller categories have had consistent use patterns until more recently.

Disclaimer: 2014 is included, even though the year is not over yet.



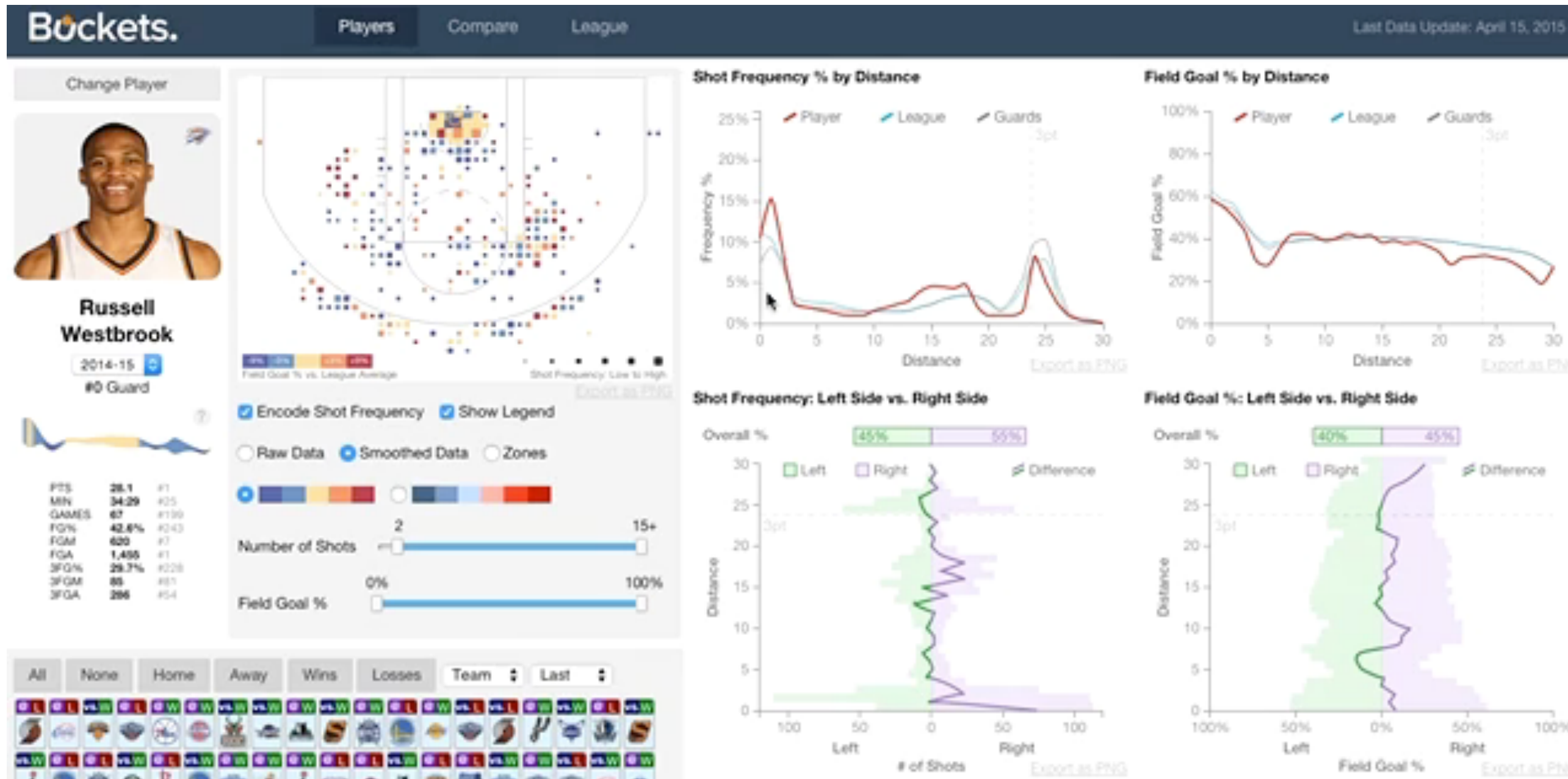
[<https://bl.ocks.org/ColinEberhardt/3c780088c363d1515403f50a87a87121>]

[<https://blog.scottlogic.com/2017/04/05/interactive-responsive-small-multiples.html>]

[[http://projects.flowingdata.com/tut/linked\\_small\\_multiples\\_demo/](http://projects.flowingdata.com/tut/linked_small_multiples_demo/)]

# Example: Combining many interaction idioms

# System: **Buckets**

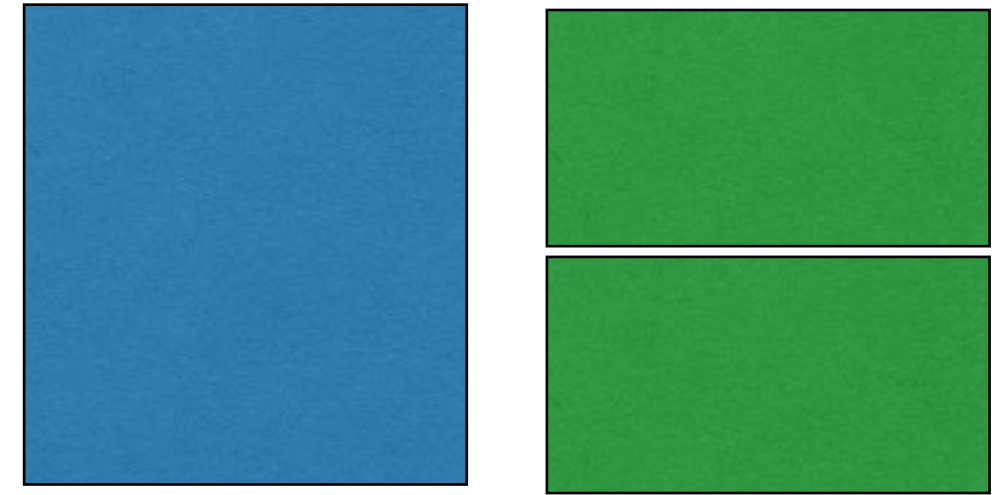


- multiform
- multidirectional linked highlighting of small multiples
- tooltips

<http://buckets.peterbeshai.com/>

# Juxtapose views: tradeoffs

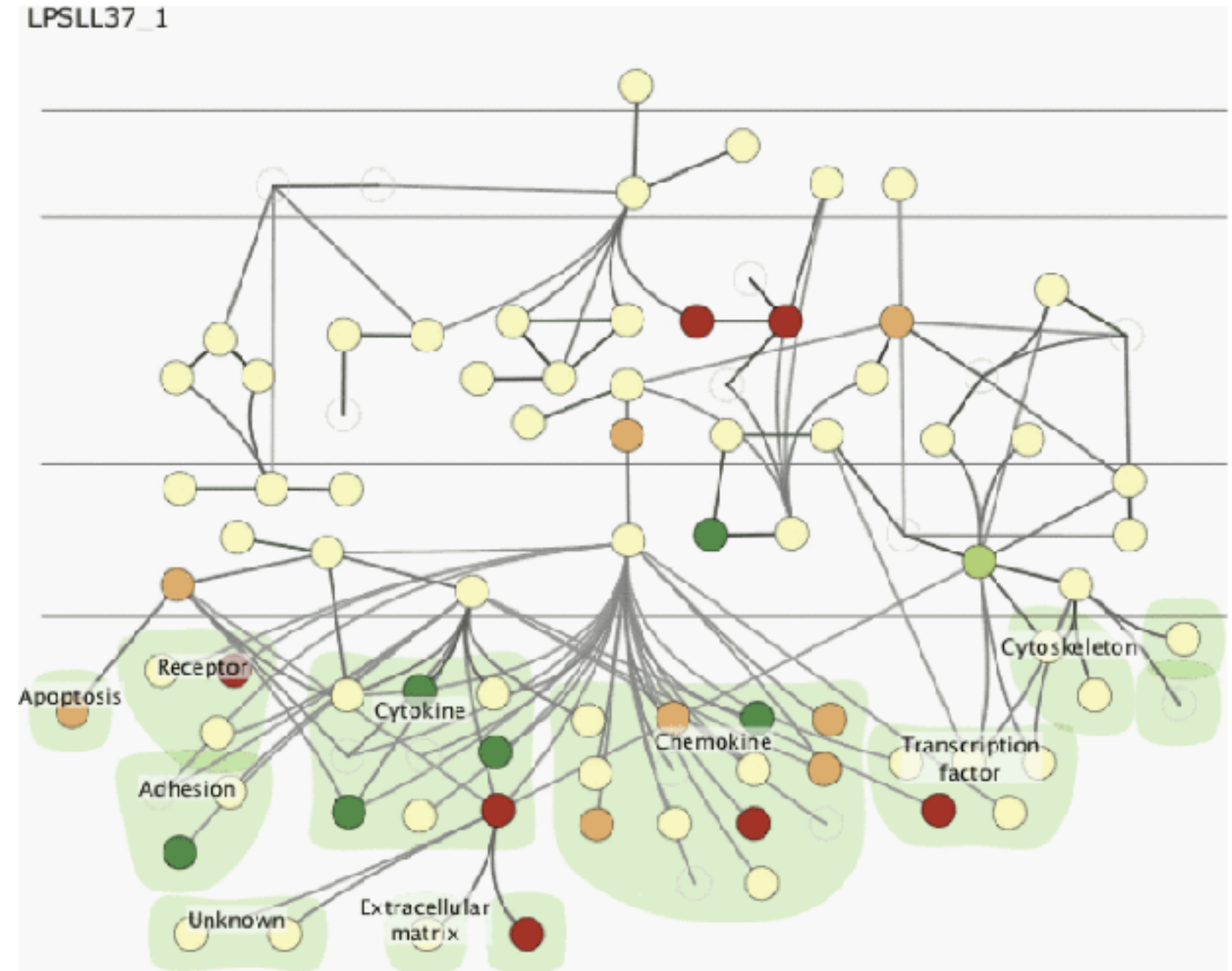
- juxtapose costs
  - display area
    - 2 views side by side: each has only half the area of one view
- juxtapose benefits
  - cognitive load: eyes vs memory
    - lower cognitive load: move eyes between 2 views
    - higher cognitive load: compare single changing view to memory of previous state





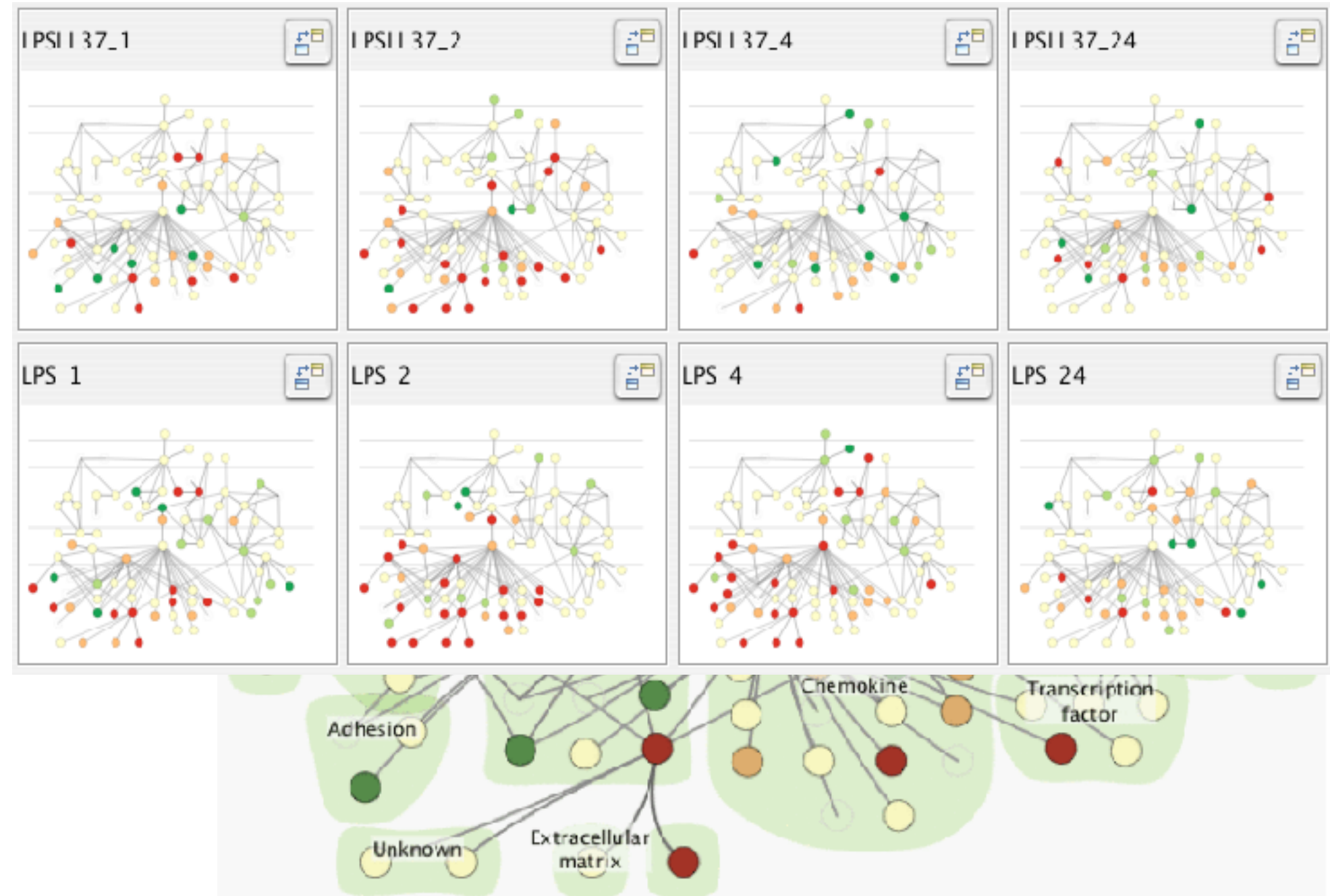
# Juxtapose vs animate

- animate: hard to follow if many scattered changes or many frames
  - vs easy special case: animated transitions

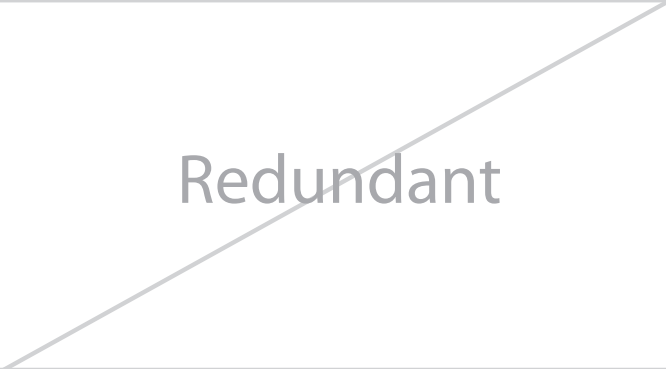
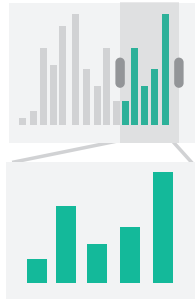
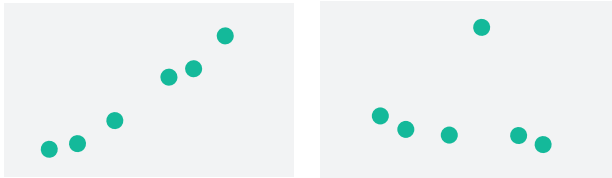





# Juxtapose vs animate

- animate: hard to follow if many scattered changes or many frames
  - vs easy special case: animated transitions
- juxtapose: easier to compare across small multiples
  - different conditions (color), same gene (layout)



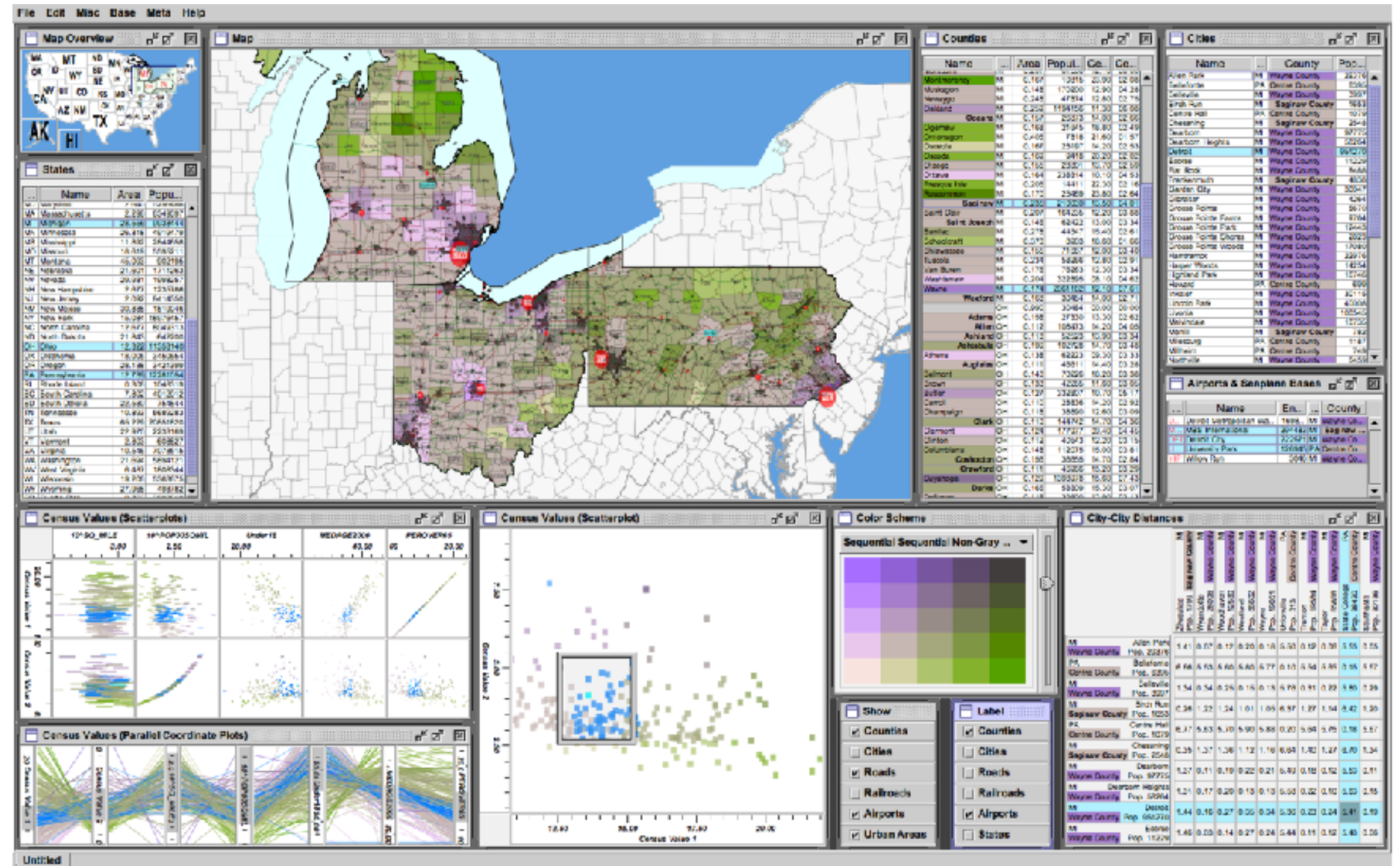
# View coordination: Design choices

		Data		
		All	Subset	None
Encoding	Same	<p>Redundant</p> 	 <p>Overview/ Detail</p>	 <p>Small Multiples</p>
	Different	 <p>Multiform</p>	 <p>Multiform, Overview/ Detail</p>	<p>No Linkage</p> 

# Idiom: Reorderable lists

# System: Improvise

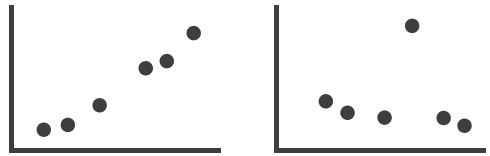
- list views
  - easy lookup
  - useful when linked to other views
- how many views is ok vs too complex?
  - open research question



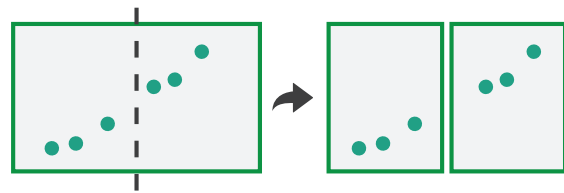
[Building Highly-Coordinated Visualizations In Improvise. Weaver. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 159–166, 2004.]

# Facet

## → Juxtapose



## → Partition



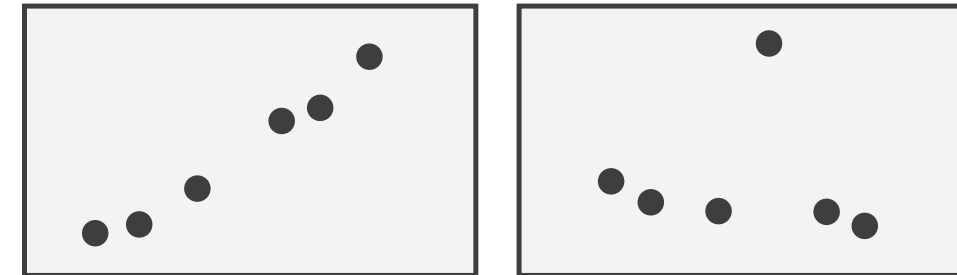
## → Superimpose



# Partition into views

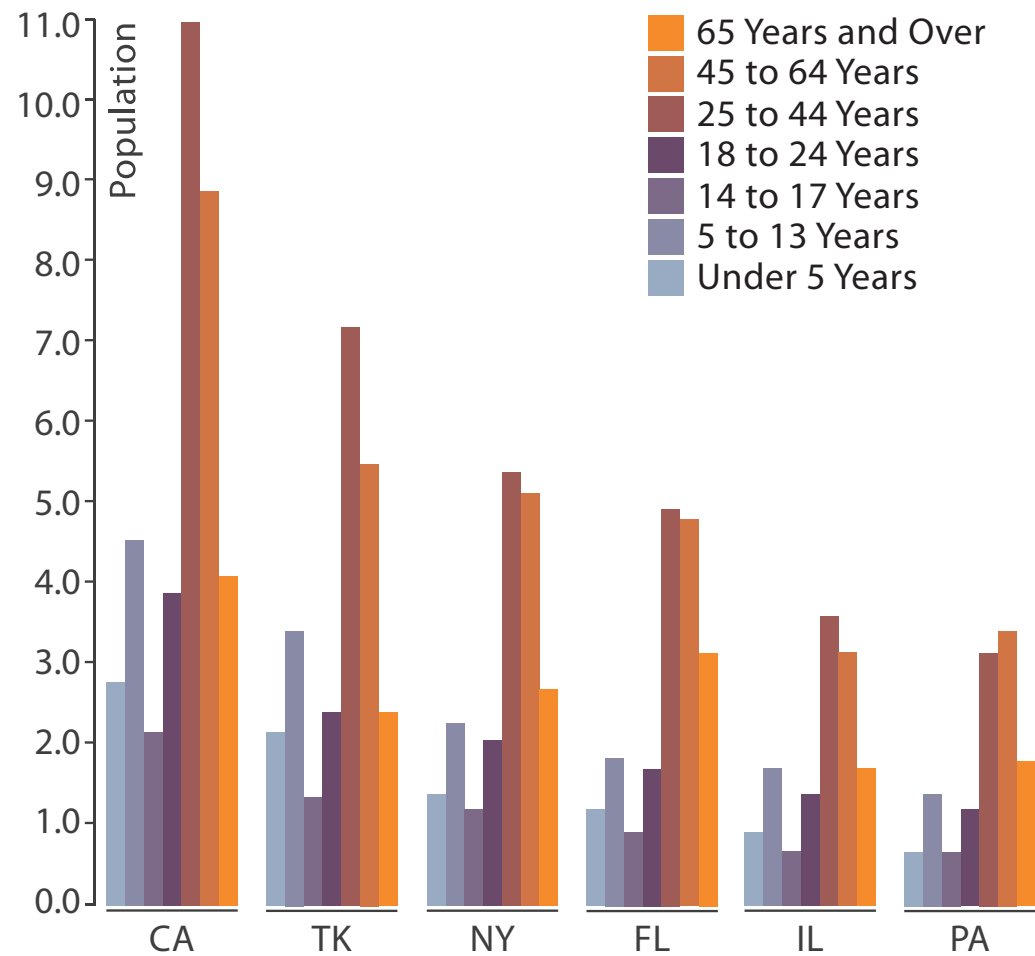
- how to divide data between views
  - split into regions by attributes
  - encodes association between items using spatial proximity
  - order of splits has major implications for what patterns are visible

## ➔ Partition into Side-by-Side Views



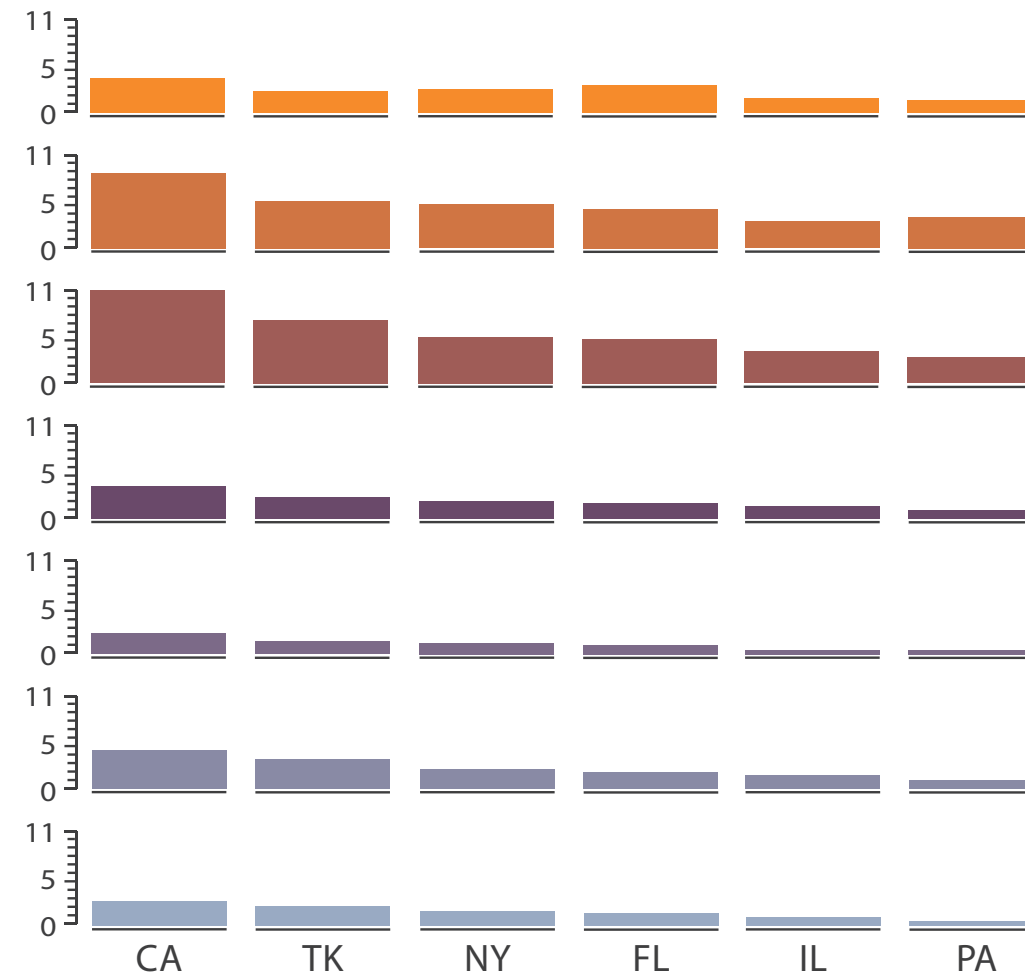
# Partitioning: Grouped vs small-multiple bars

- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages



[<https://observablehq.com/@d3/grouped-bar-chart>]

- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states

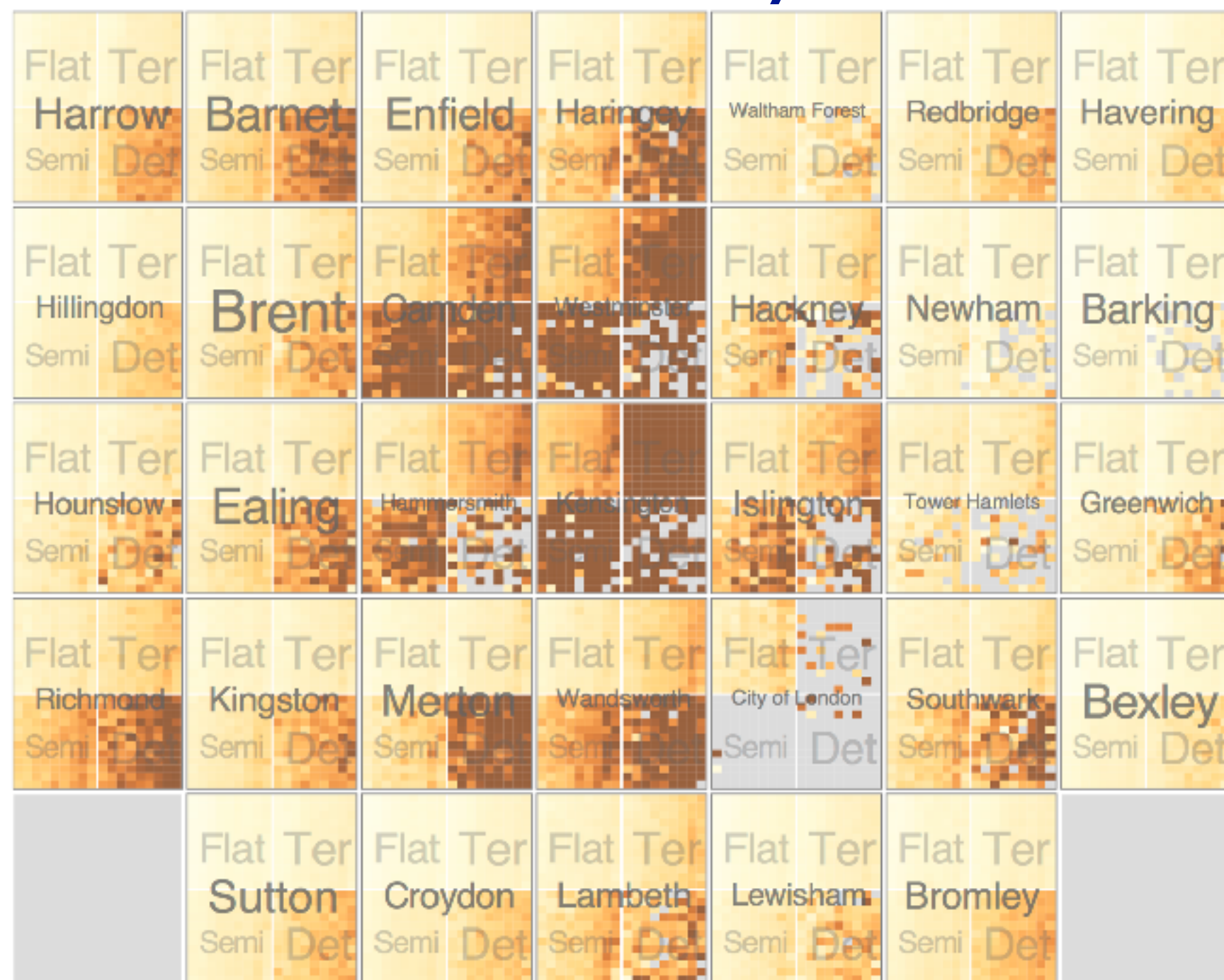


[<https://bl.ocks.org/mbostock/4679202>]

# Partitioning: Recursive subdivision

System: **HIVE**

- split by neighborhood
- then by type
  - flat, terrace, semi-detached, detached
- then time
  - years as rows
  - months as columns
- color by price
- neighborhood patterns
  - where it's expensive
  - where you pay much more for detached type





# Partitioning: Recursive subdivision

System: **HIVE**

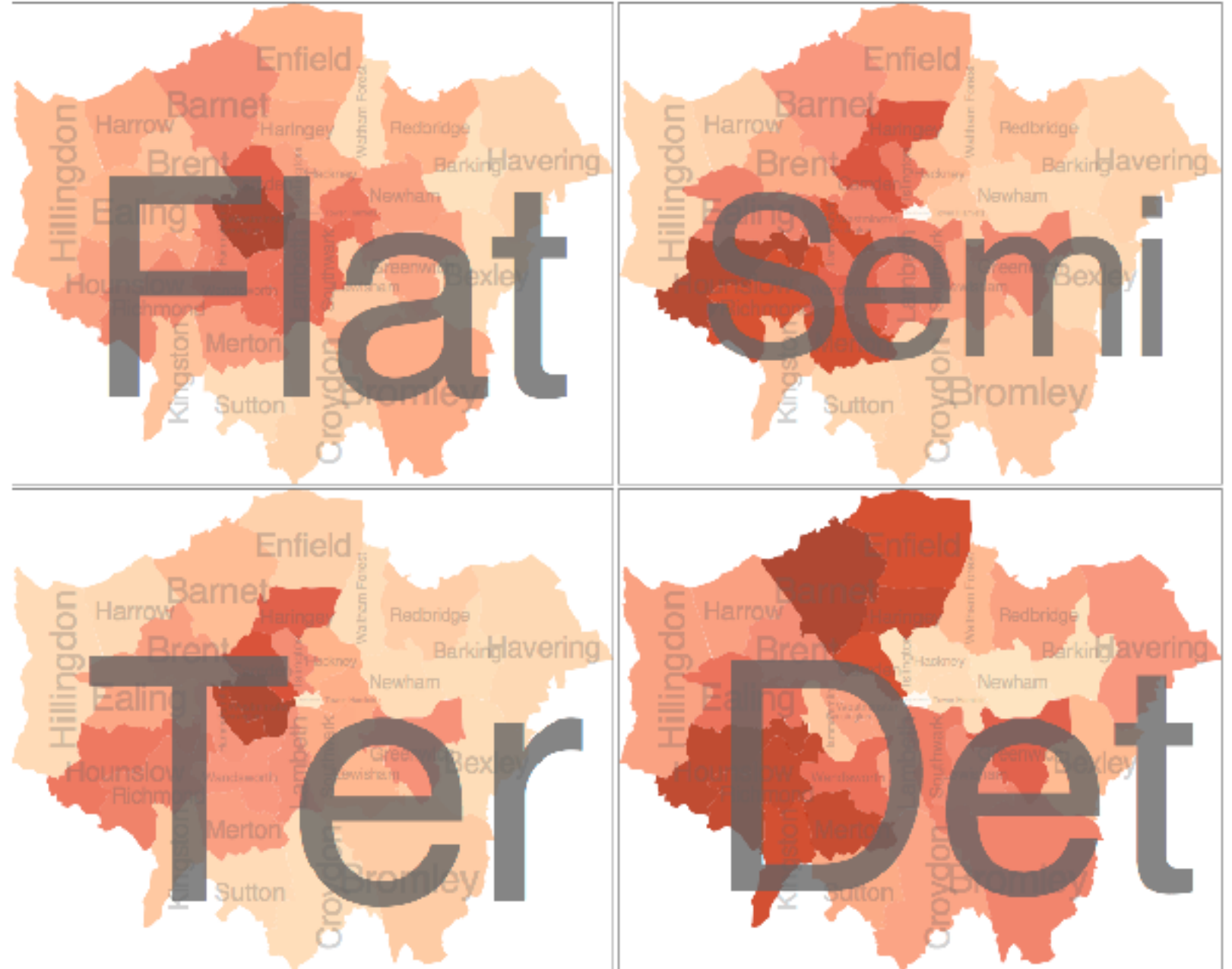
- switch order of splits
  - type then neighborhood
- switch color
  - by price variation
- type patterns
  - within specific type, which neighborhoods inconsistent



# Partitioning: Recursive subdivision

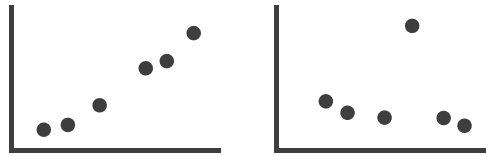
- different encoding for second-level regions  
– choropleth maps

System: **HIVE**

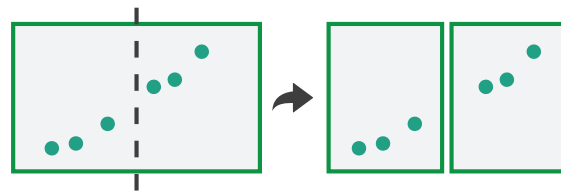


# Facet

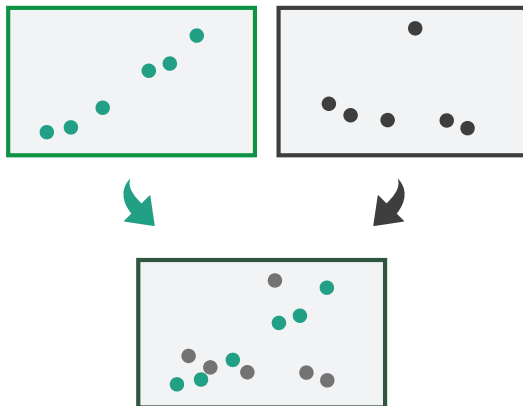
## → Juxtapose



## → Partition



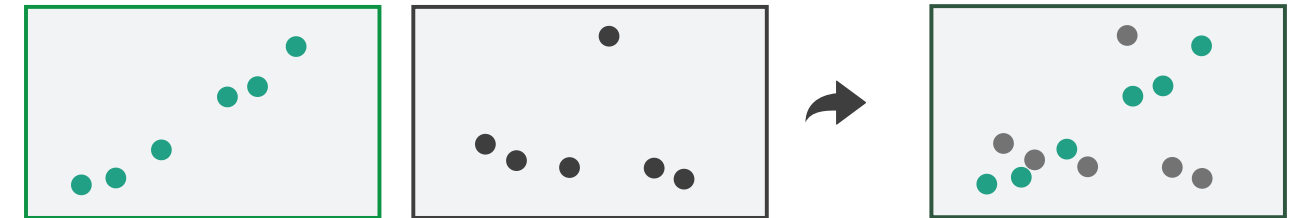
## → Superimpose



# Superimpose layers

- layer: set of objects spread out over region
  - each set is visually distinguishable group
  - extent: whole view
- design choices
  - how many layers, how to distinguish?
    - encode with different, nonoverlapping channels
    - two layers achievable, three with careful design
  - small static set, or dynamic from many possible?

## ➔ Superimpose Layers



# Static visual layering

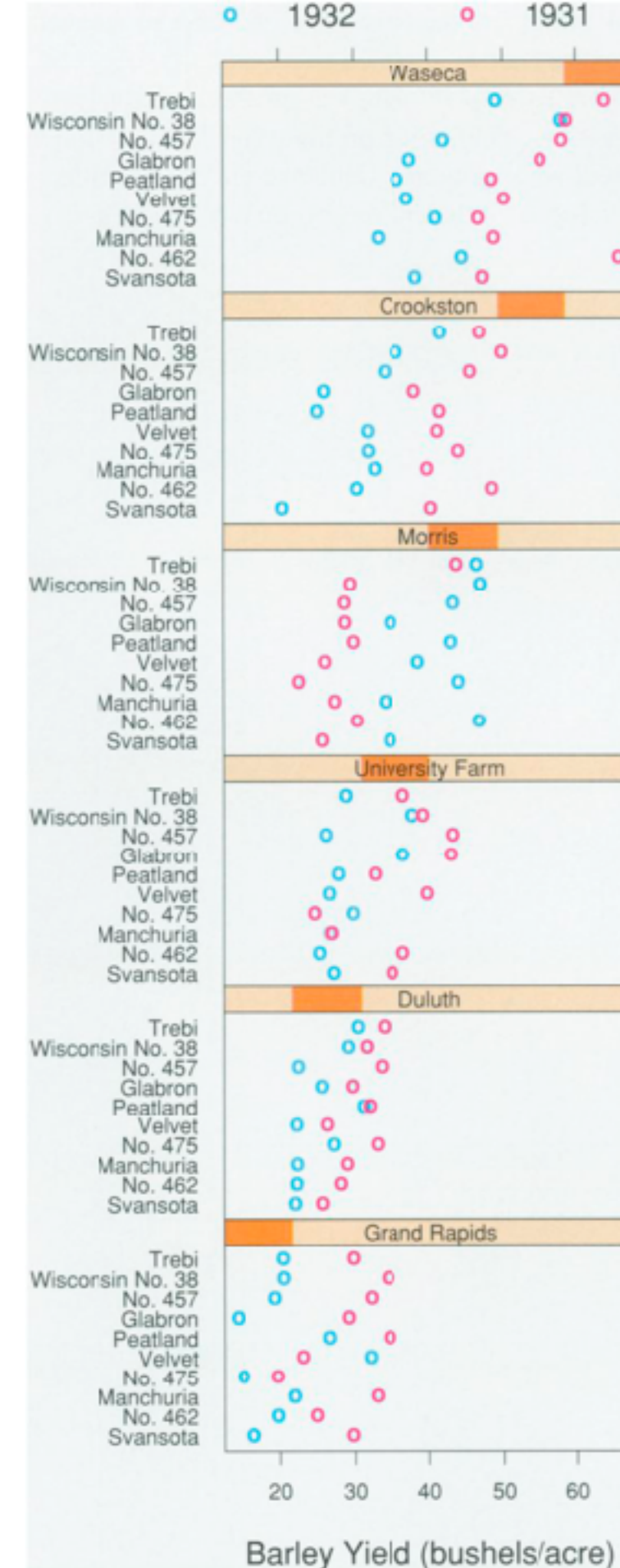
- foreground layer: roads
  - hue, size distinguishing main from minor
  - high luminance contrast from background
- background layer: regions
  - desaturated colors for water, parks, land areas
- user can selectively focus attention



[Get it right in black and white. Stone. 2010.  
<http://www.stonesc.com/wordpress/2010/03/get-it-right-in-black-and-white>]

# Idiom: Trellis plots

- superimpose within same frame
  - color code by year
- partitioning
  - split by site, rows are barley varieties
- main-effects ordering
  - derive value of median for group
  - order rows within view by variety median
  - order views themselves by site median



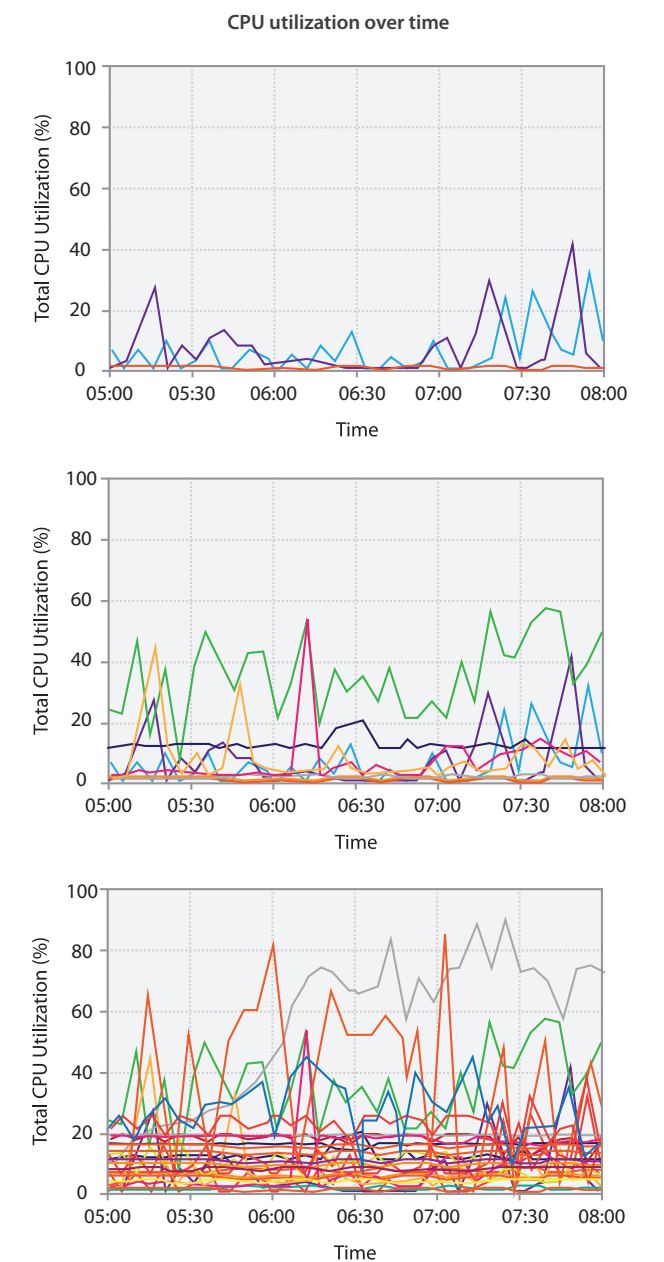
[The Visual Design and Control of Trellis Display. Becker, Cleveland, & Shyu.  
*Journal of Computational and Graphical Statistics* 5(2):123-155 1996.]

# Superimposing limits (static)

- few layers, more lines
  - up to a few dozen lines
  - but not hundreds
- superimpose vs juxtapose: empirical study
  - same size: all multiples, vs single superimposed
    - superimposed: local tasks
    - juxtaposed: global tasks, esp. for many charts



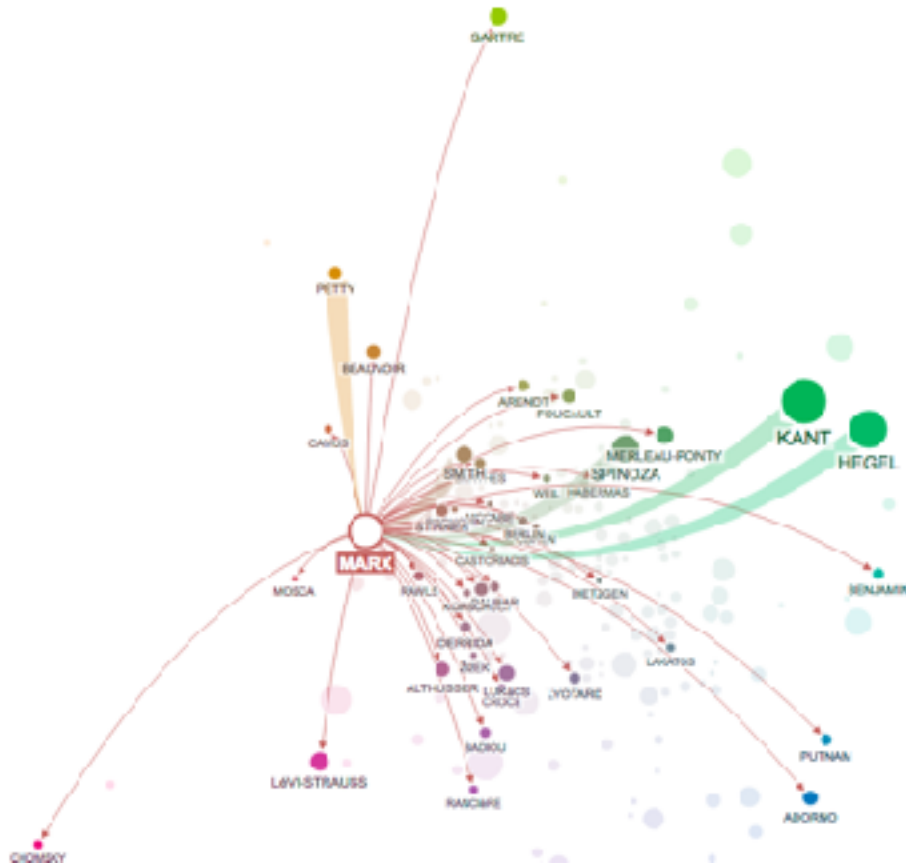
[Graphical Perception of Multiple Time Series. Javed, McDonnell, and Elmqvist. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE InfoVis 2010) 16:6 (2010), 927–934.]



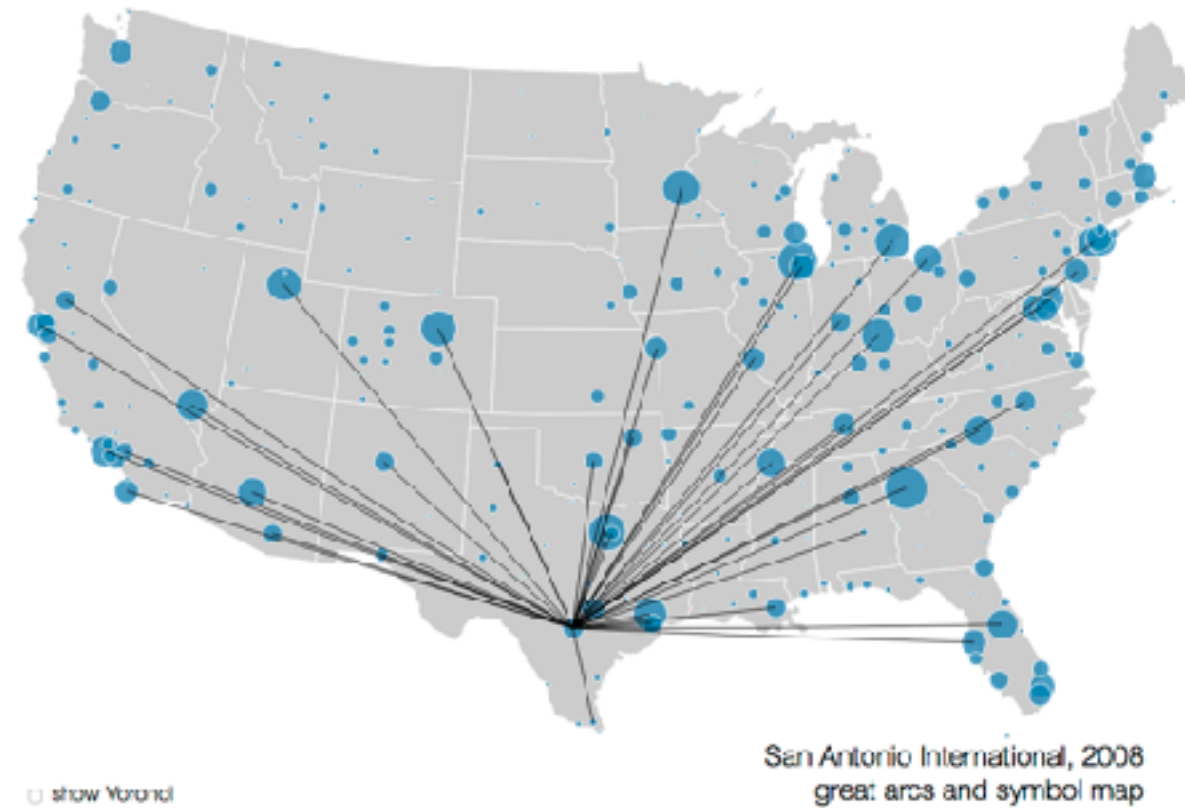
# Dynamic visual layering

- interactive, based on selection
- one-hop neighbour highlighting

click (heavyweight)



hover (fast)



<https://mariandoerk.de/edgemaps/demo/>

<http://mbostock.github.io/d3/talk/20111116/airports.html>



# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



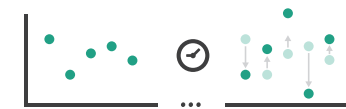
→ Motion

*Direction, Rate, Frequency, ...*

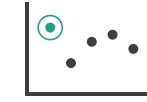


## Manipulate

### → Change



### → Select



### → Navigate

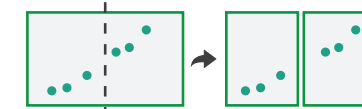


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

### → Filter



### → Aggregate



### → Embed

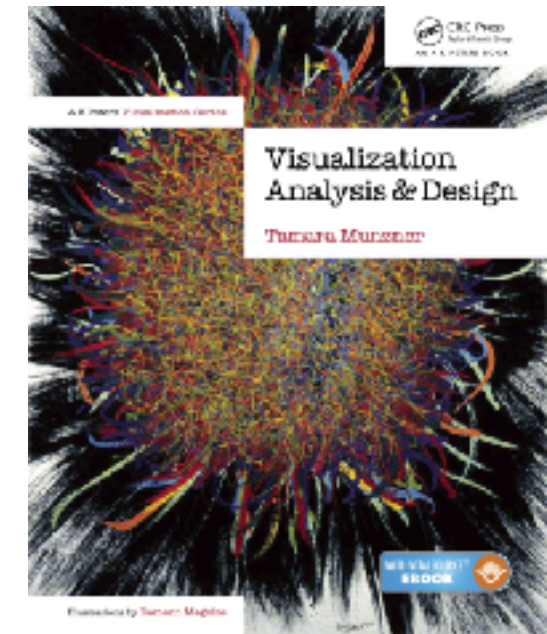


What?

Why?

How?

# Visualization Analysis & Design



## *Reduce: Aggregation & Filtering (Ch 13)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)

# How to handle complexity: 3 previous strategies

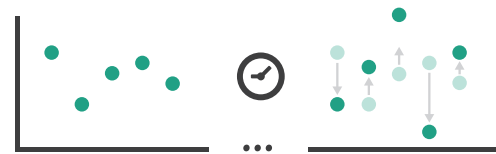
→ *Derive*



- derive new data to show within view
- change view over time
- facet across multiple views

Manipulate

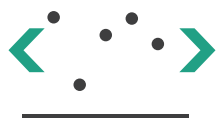
① Change



② Select

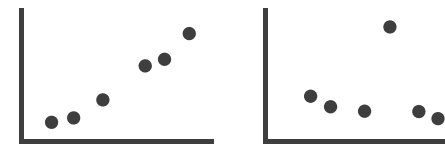


③ Navigate

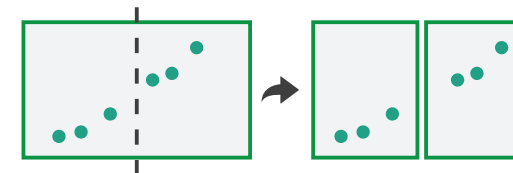


Facet

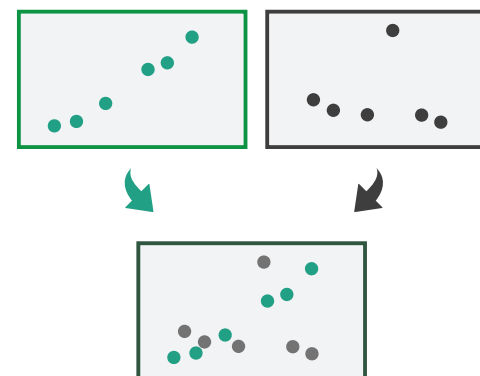
① Juxtapose



② Partition



③ Superimpose



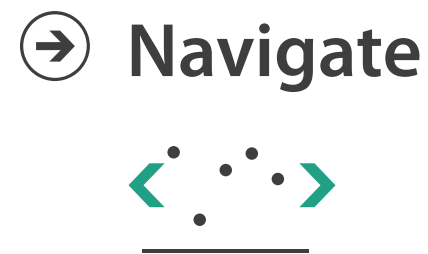
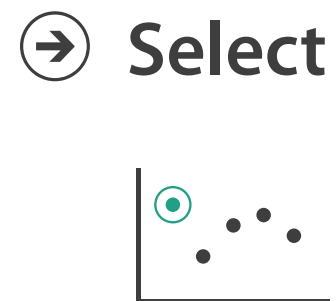
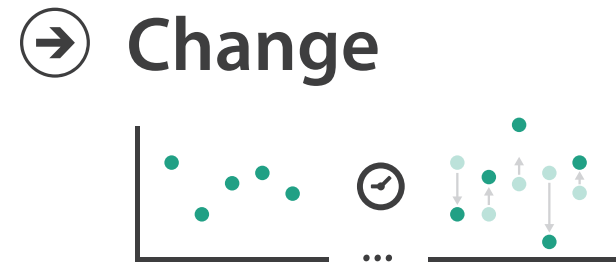
# How to handle complexity: 3 previous strategies + 1 more

→ *Derive*

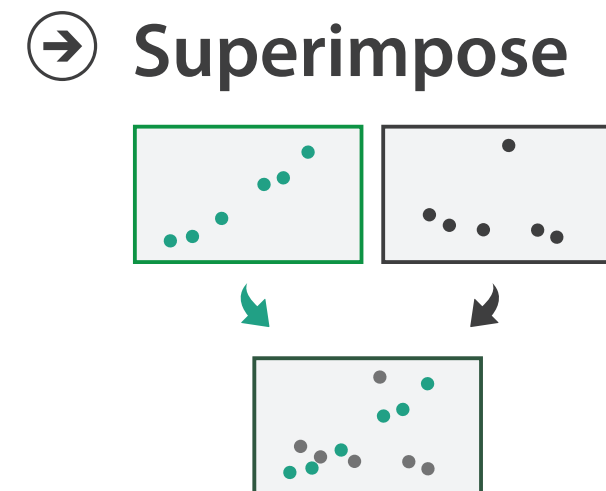
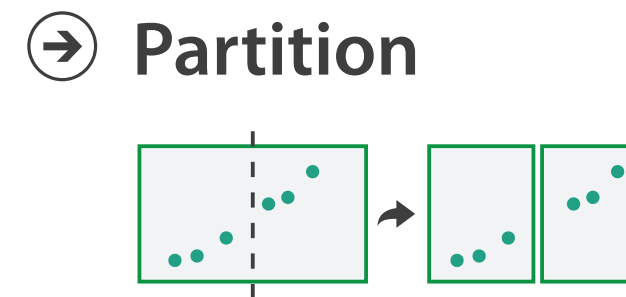


- derive new data to show within view
- change view over time
- facet across multiple views
- reduce items/attributes within single view

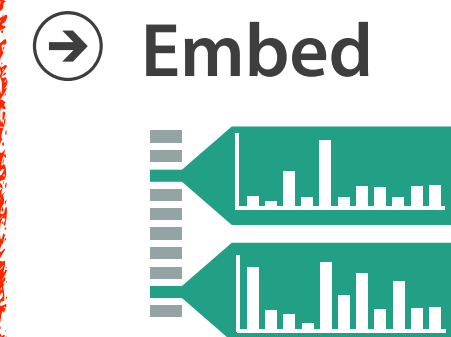
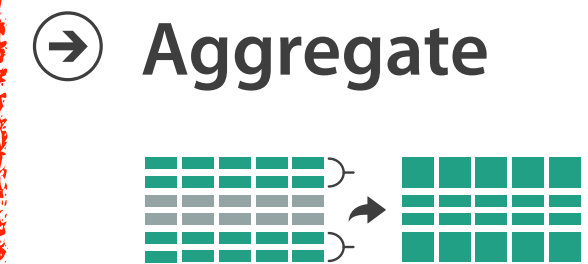
Manipulate



Facet



Reduce



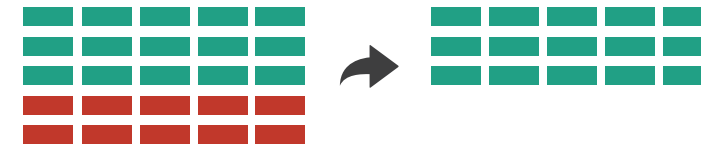
# Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind

## Reducing Items and Attributes

### → Filter

→ Items



→ Attributes



# Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind
- aggregation
  - pro: inform about whole set
  - con: difficult to avoid losing signal
- not mutually exclusive
  - combine filter, aggregate
  - combine reduce, change, facet

## Reducing Items and Attributes

### ➔ Filter

➔ Items



➔ Attributes

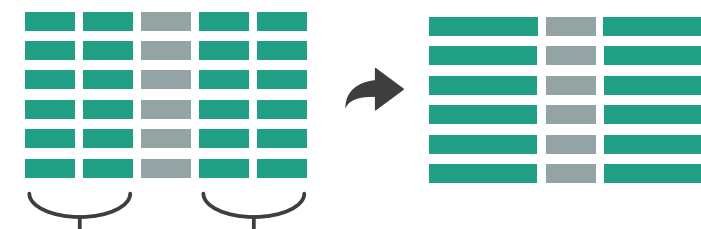


### ➔ Aggregate

➔ Items



➔ Attributes



# Filter

- eliminate some elements
  - either items or attributes
- according to what?
  - any possible function that partitions dataset into two sets
    - attribute values bigger/smaller than  $x$
    - noise/signal
- filters vs queries
  - query: start with nothing, add in elements
  - filters: start with everything, remove elements
  - best approach depends on dataset size

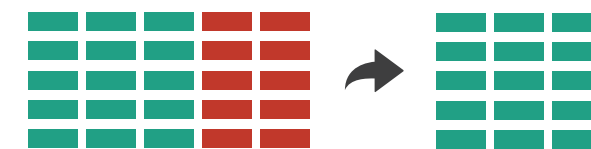
## Reducing Items and Attributes

### ② Filter

→ Items

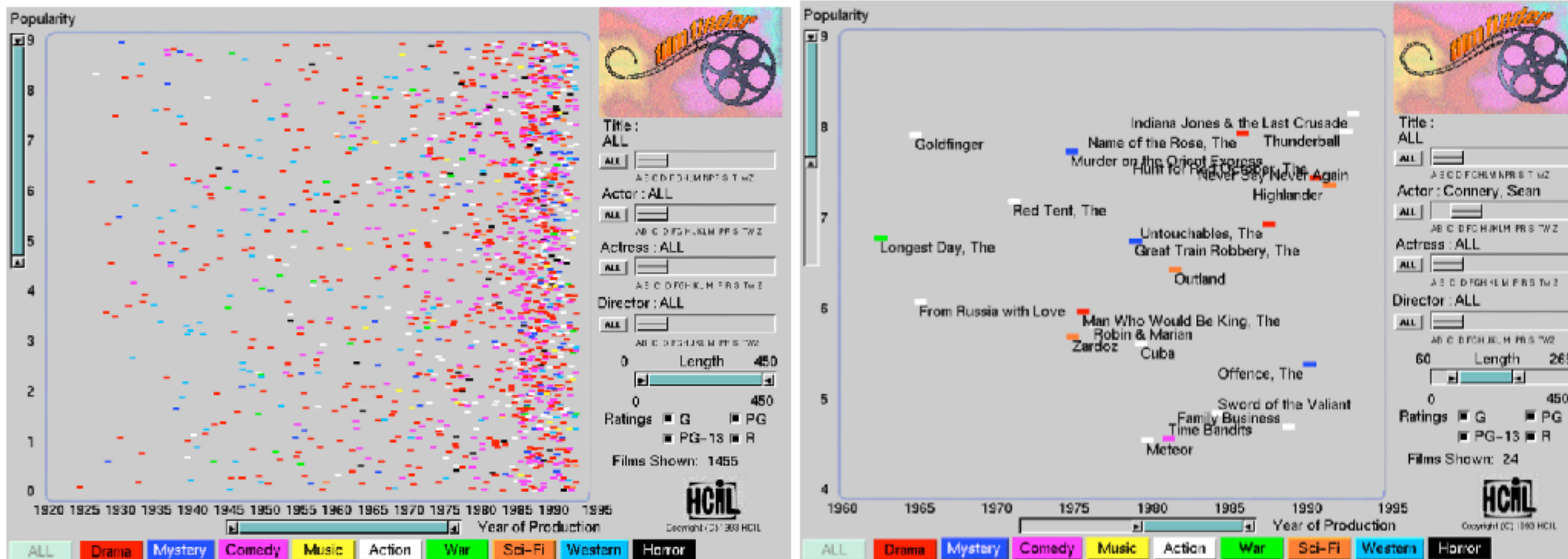


→ Attributes



# Idiom: **FilmFinder**

- dynamic queries/filters for items
  - tightly coupled interaction and visual encoding idioms, so user can immediately see results of action

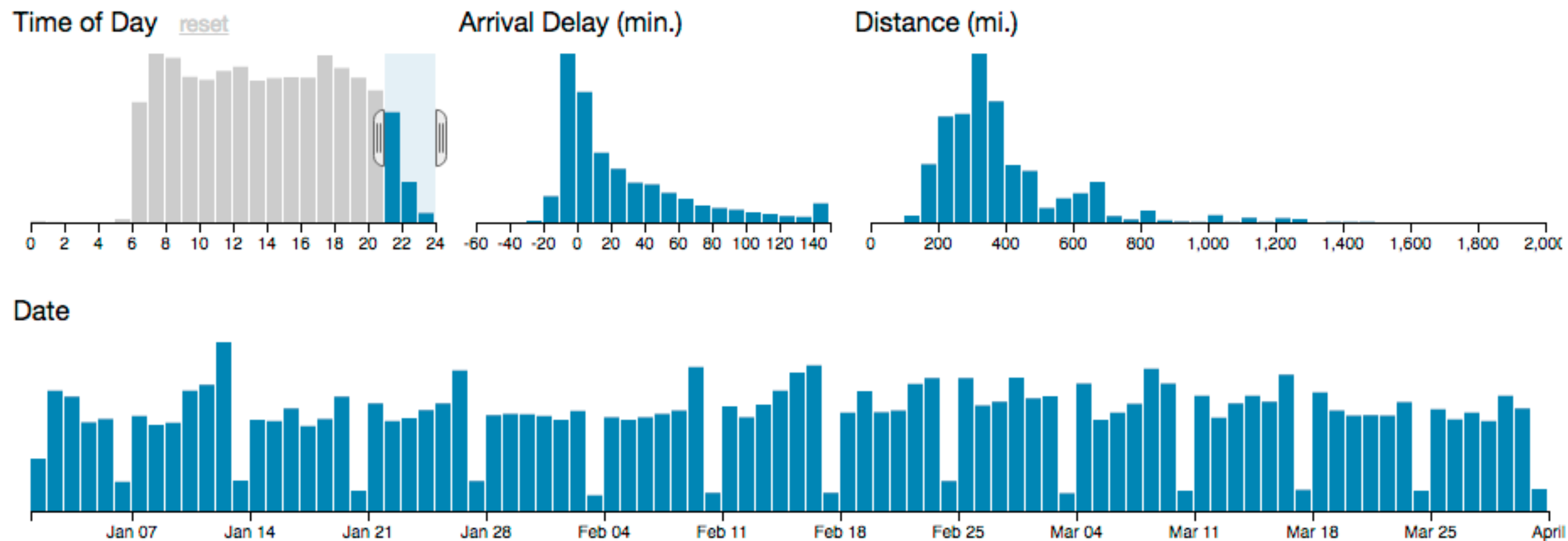




# Idiom: **cross filtering**

# System: **Crossfilter**

- item filtering
- coordinated views/controls combined
  - all scented histogram bisliders update when any ranges change



<http://square.github.io/crossfilter/>

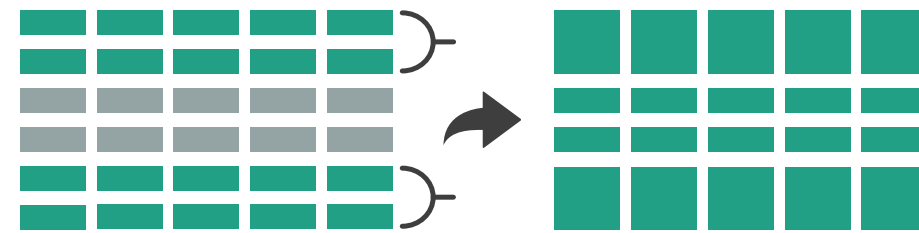
<https://observablehq.com/@uwdata/interaction>

# Aggregate

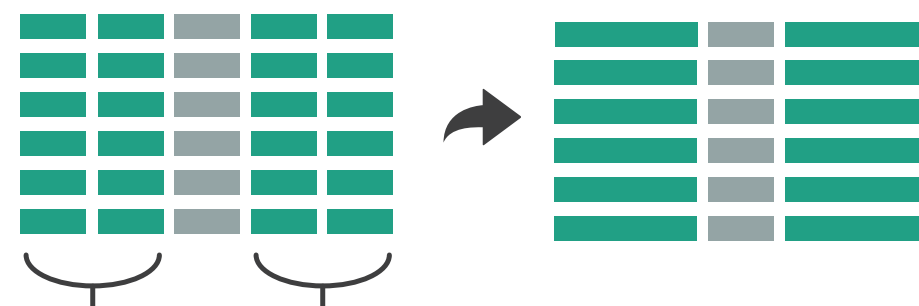
- a group of elements is represented by a smaller number of derived elements

## ➔ Aggregate

➔ Items

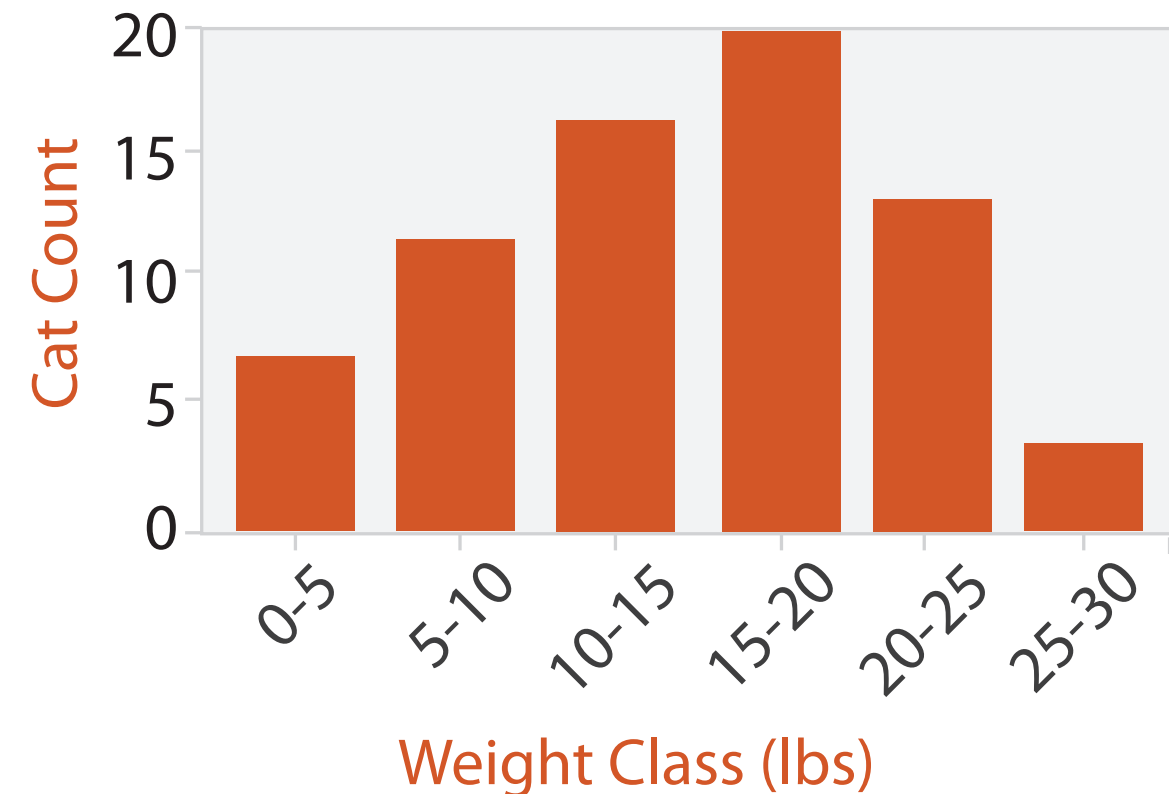


➔ Attributes



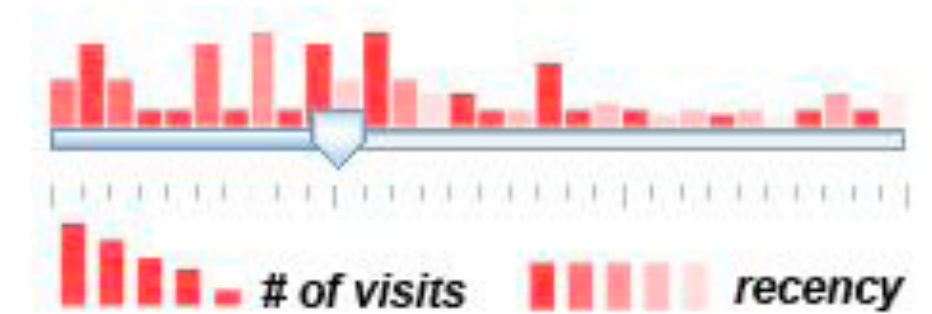
# Idiom: **histogram**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly



# Idiom: **scented widgets**

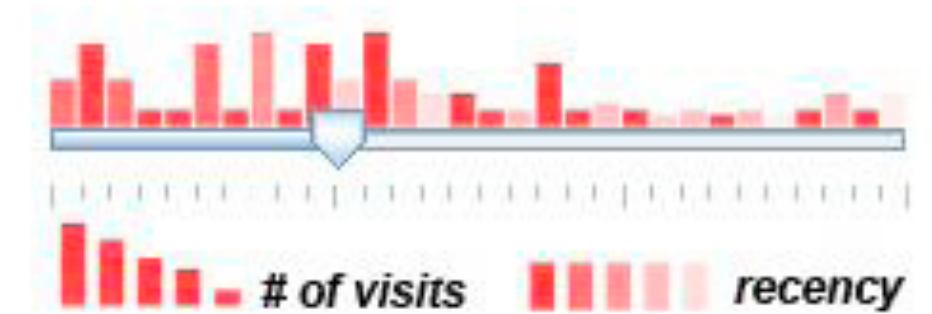
- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider



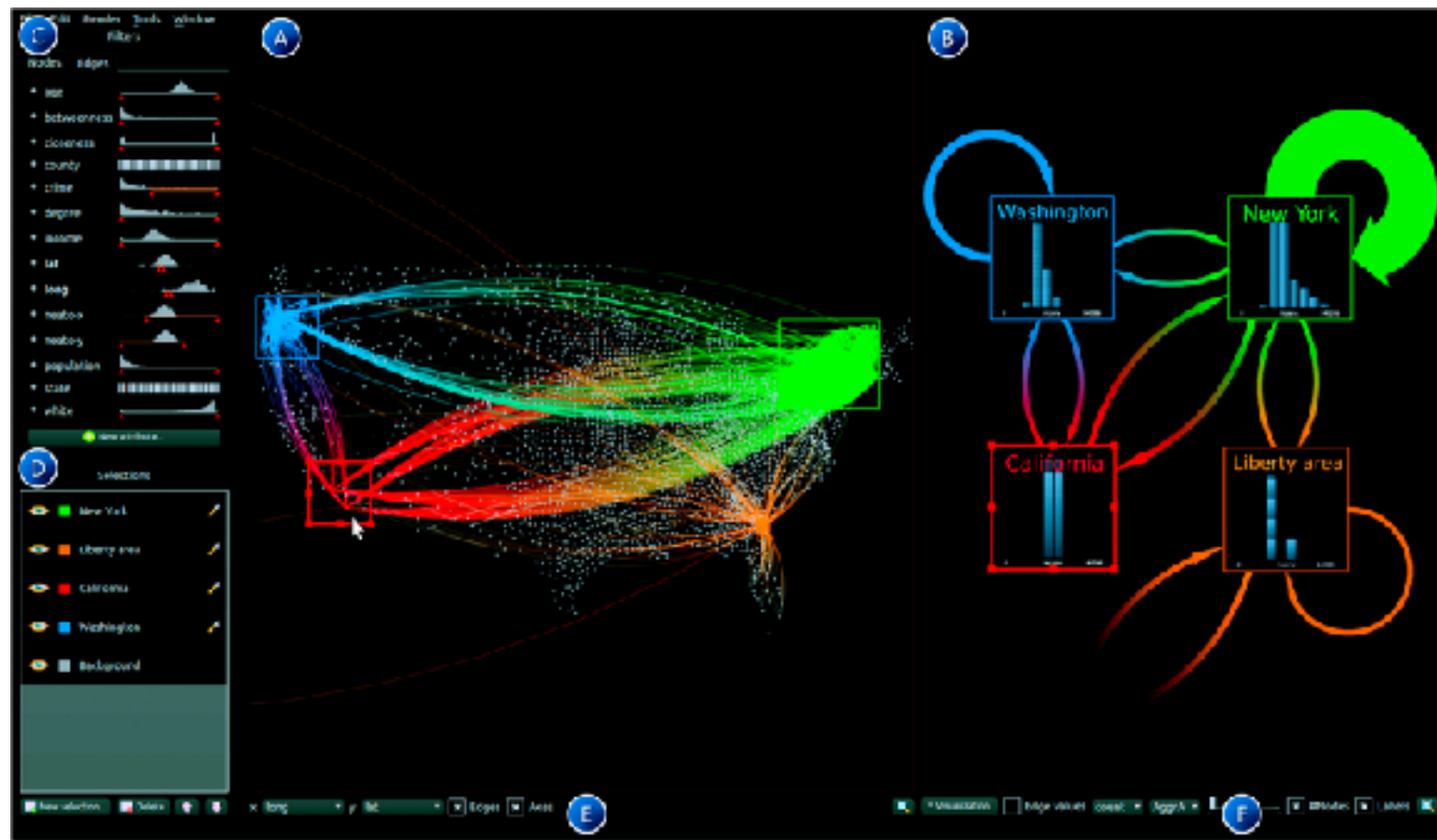
[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]

# Idiom: scented widgets

- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider



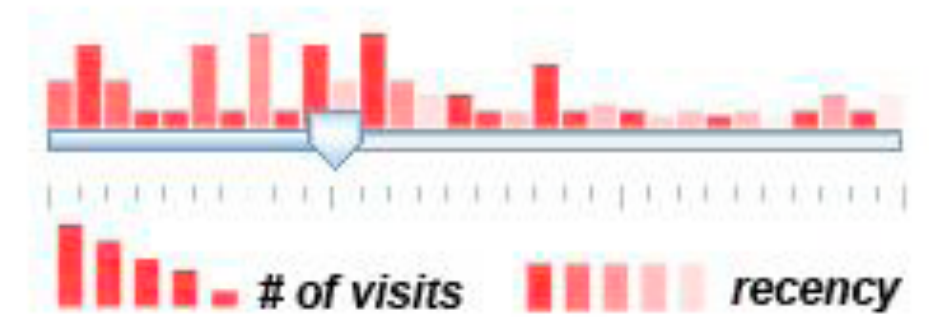
[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]



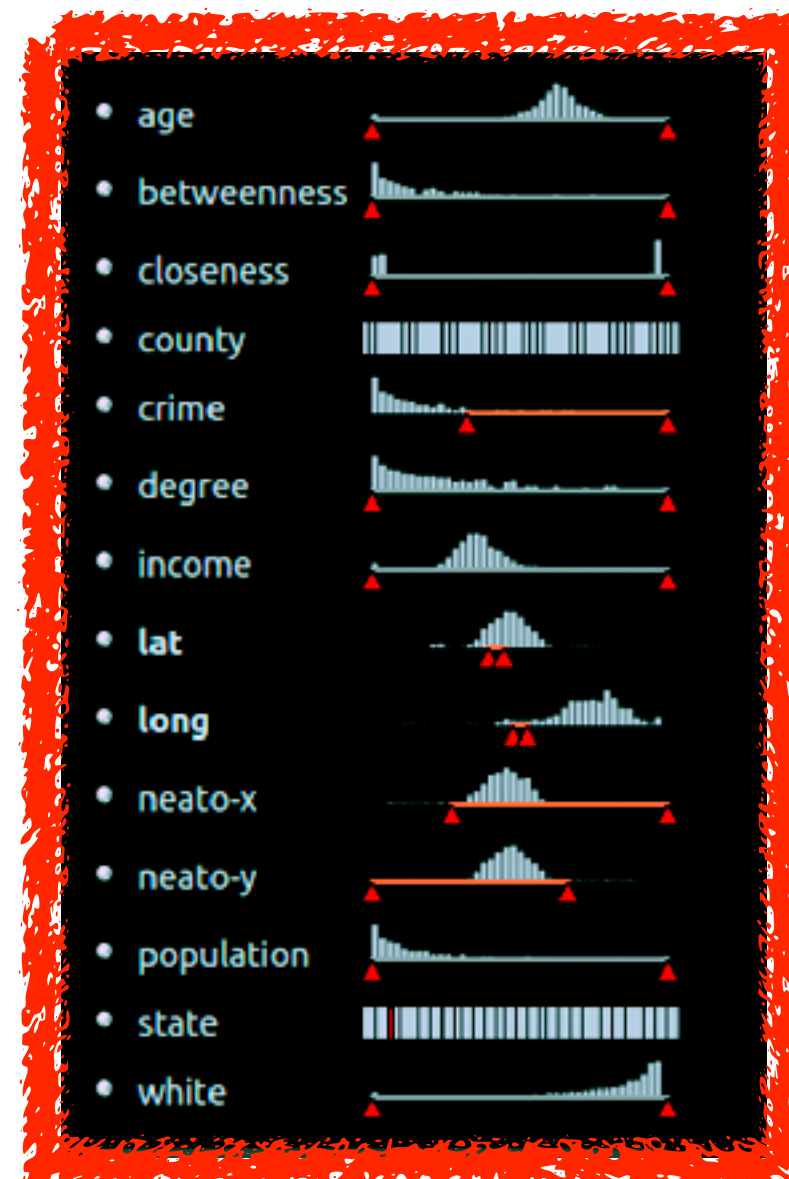
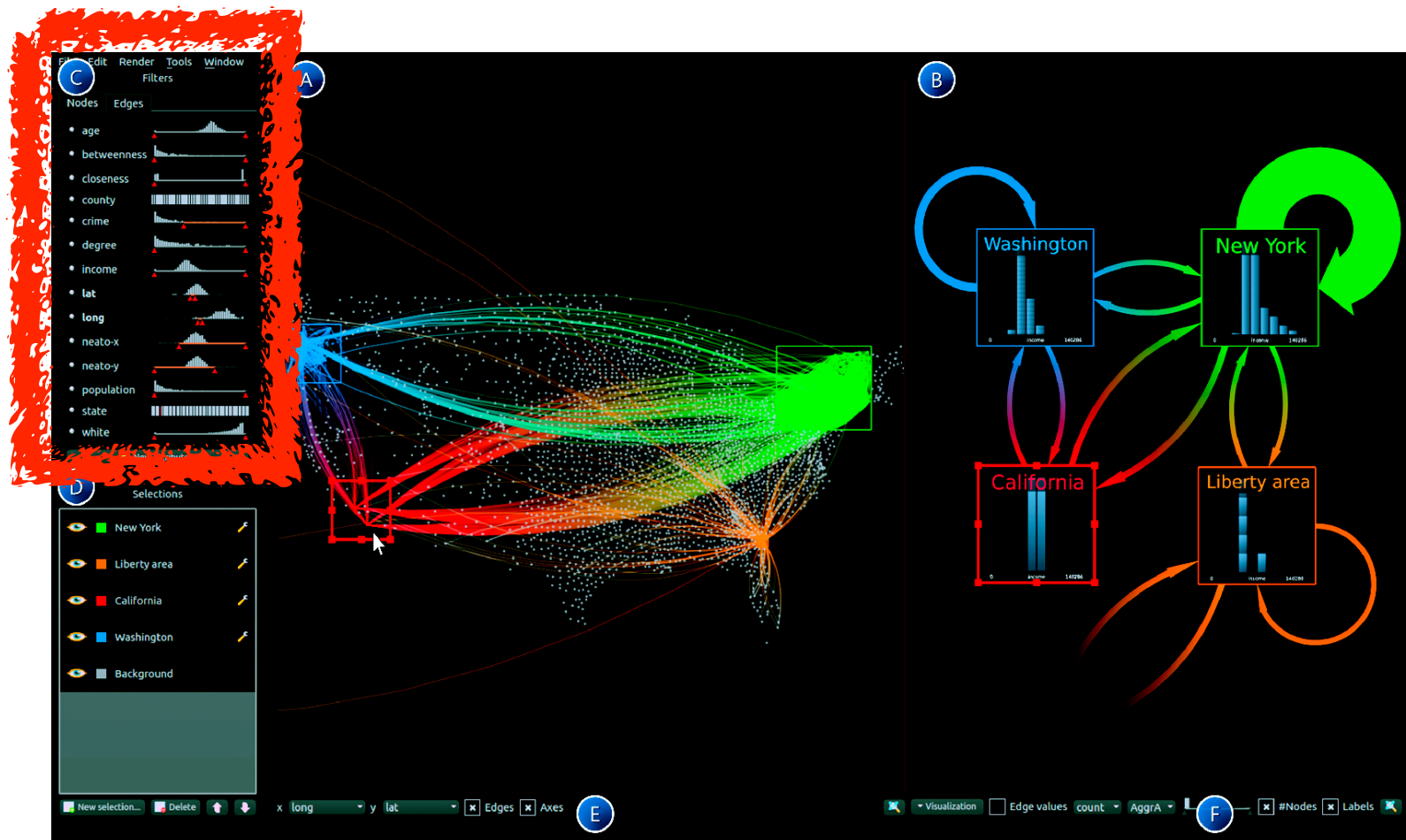
[Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. van den Elzen, van Wijk, IEEE TVCG 20(12): 2014 (Proc. InfoVis 2014).]

# Idiom: scented widgets

- augmented widgets show *information scent*
  - better cues for *information foraging*: show whether value in drilling down further vs looking elsewhere
- concise use of space: histogram on slider



[Scented Widgets: Improving Navigation Cues with Embedded Visualizations. Willett, Heer, and Agrawala. IEEE TVCG (Proc. InfoVis 2007) 13:6 (2007), 1129–1136.]



[Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. van den Elzen, van Wijk, IEEE TVCG 20(12): 2014 (Proc. InfoVis 2014).]

# Scented histogram bisliders: detailed

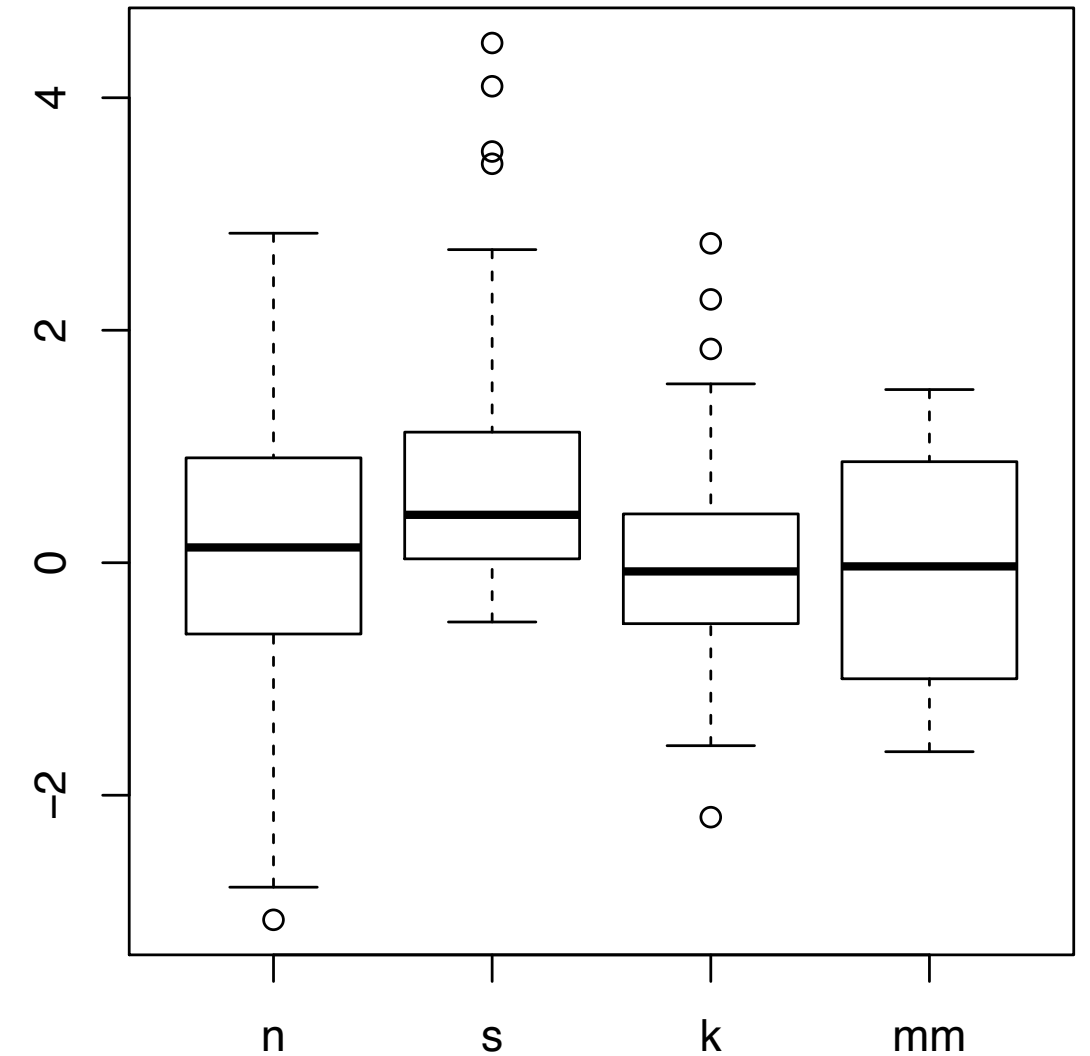
The screenshot displays the ICLIC interface with various components:

- Top Bar:** Includes menu items (File, View, Help), a search bar (a), a slider (b), a mode selector (c), a scale slider (d), selection tools (e), and a search field (f).
- Grid of Histograms:** A 4x3 grid of histograms for attributes: Date and Time (Original), Make, Model, Comments, Country, County, Date taken, Date taken (day of week), Latitude, Locality, Longitude, Neighbourhood, Region, and Type. A 'h' icon is next to the 'Make' histogram.
- Color Map:** A vertical bar (g) showing a color distribution for the selected attribute, with a value of 2.
- Country Selection:** A row of color-coded boxes for Germany, United Kingdom, France, Belgium, United States, Norway, Spain, and Other.
- Image Gallery:** A grid of image thumbnails categorized by 'Event' (Vacation: 443, Sports: 719) and 'Object' (Airplane: 2K, Car: 7K). A 'k' icon is next to the first image, and an 'i' icon is next to the 'Country' label.
- Bottom Bar:** Shows 'images in collection' (92K) and 'visible images' (11K) with progress bars. A 'm' icon is at the bottom right.
- Status:** 'visible: 11199, selected: 2561' is shown at the bottom left.

[ICLIC: Interactive categorization of large image collections. van der Corput and van Wijk. Proc. PacificVis 2016.]

# Idiom: **boxplot**

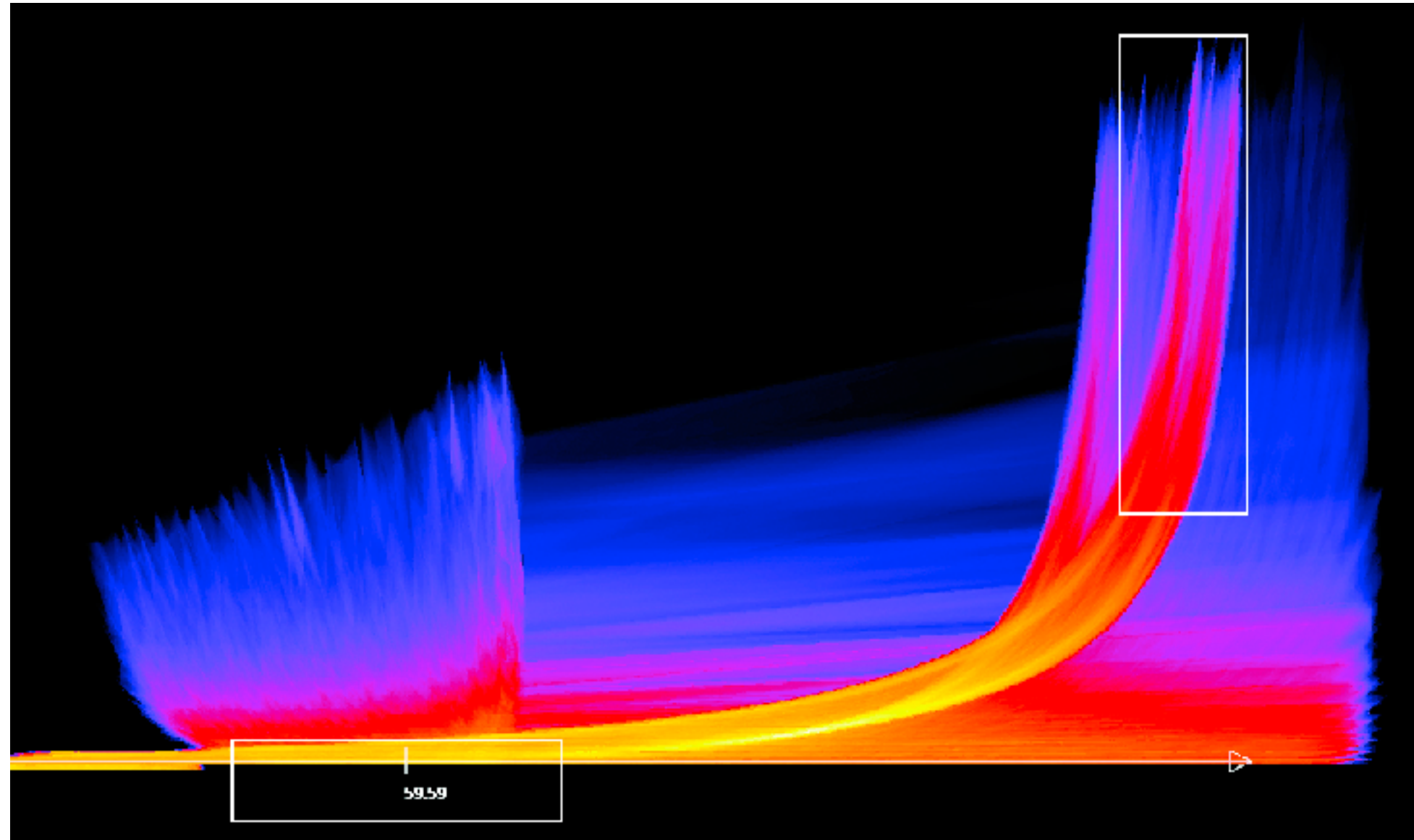
- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown
- scalability
  - unlimited number of items!





# Idiom: Continuous scatterplot

- static item aggregation
- data: table
- derived data: table
  - key attribs x,y for pixels
  - quant attrib: overplot density
- dense space-filling 2D matrix
- color:  
sequential categorical hue +  
ordered luminance colormap
- scalability
  - no limits on overplotting:  
millions of items



[Continuous Scatterplots. Bachthaler and Weiskopf.  
IEEE TVCG (Proc. Vis 08) 14:6 (2008), 1428–1435. 2008.]

# Spatial aggregation

- MAUP: Modifiable Areal Unit Problem

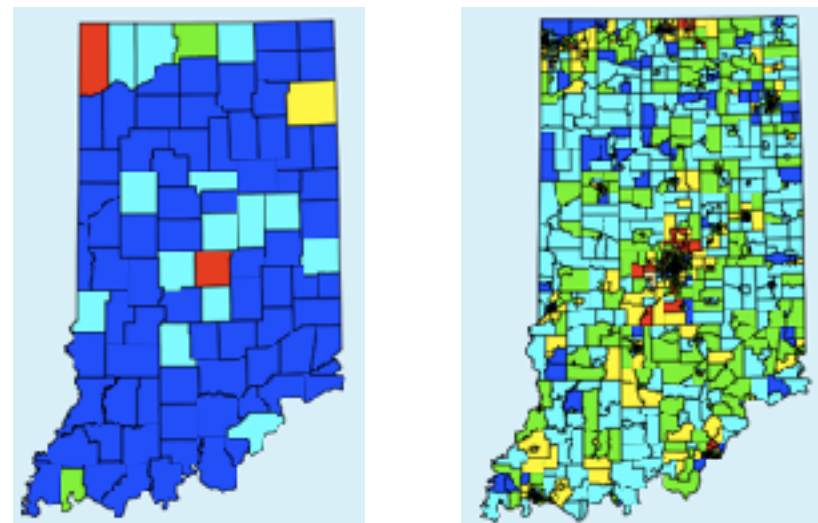
- changing boundaries of cartographic regions can yield dramatically different results

- zone effects



[[http://www.e-education.psu.edu/geog486/l4\\_p7.html](http://www.e-education.psu.edu/geog486/l4_p7.html), Fig 4.cg.6]

- scale effects

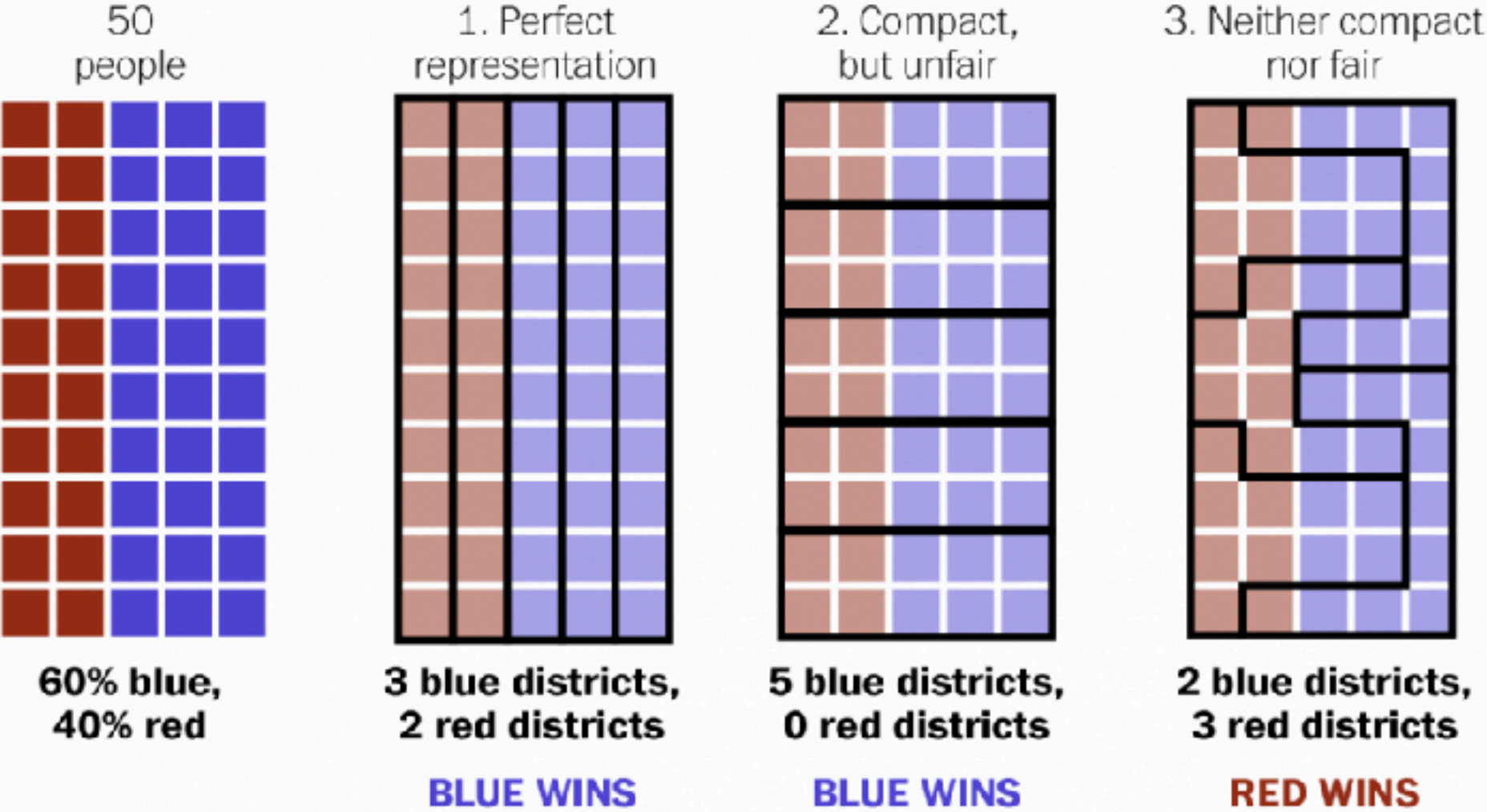


<https://blog.cartographica.com/blog/2011/5/19/the-modifiable-areal-unit-problem-in-gis.html>

# Gerrymandering: MAUP for political gain

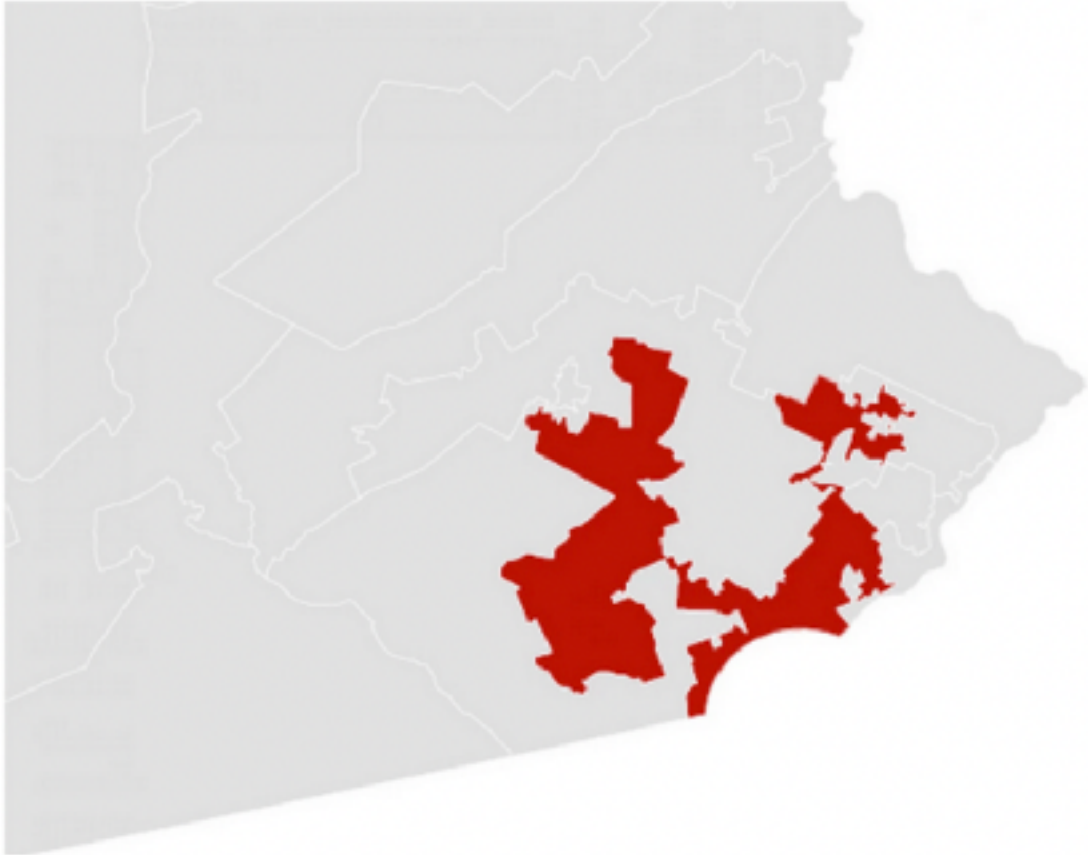
## Gerrymandering, explained

Three different ways to divide 50 people into five districts



WASHINGTONPOST.COM/WONKBLOG

Adapted from Stephen Nass



A real district in Pennsylvania:  
Democrats won 51% of the vote but only 5 out of 18 house seats

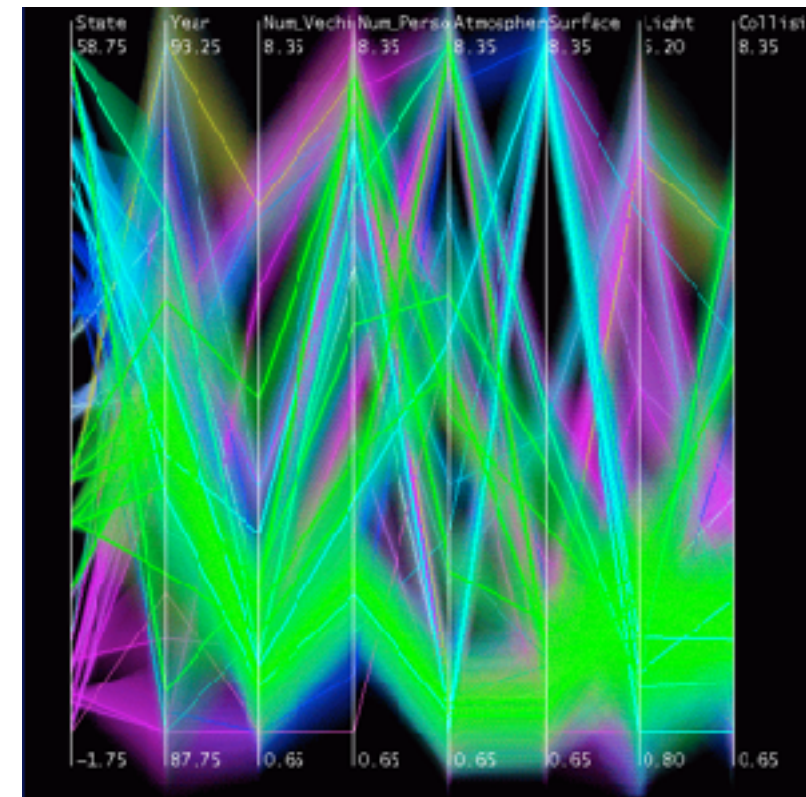
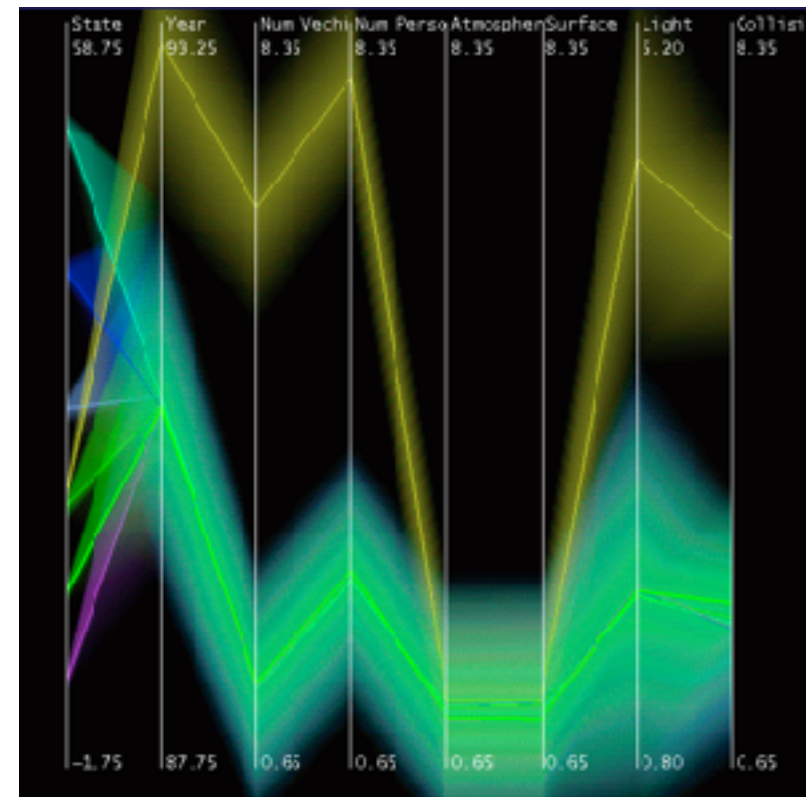
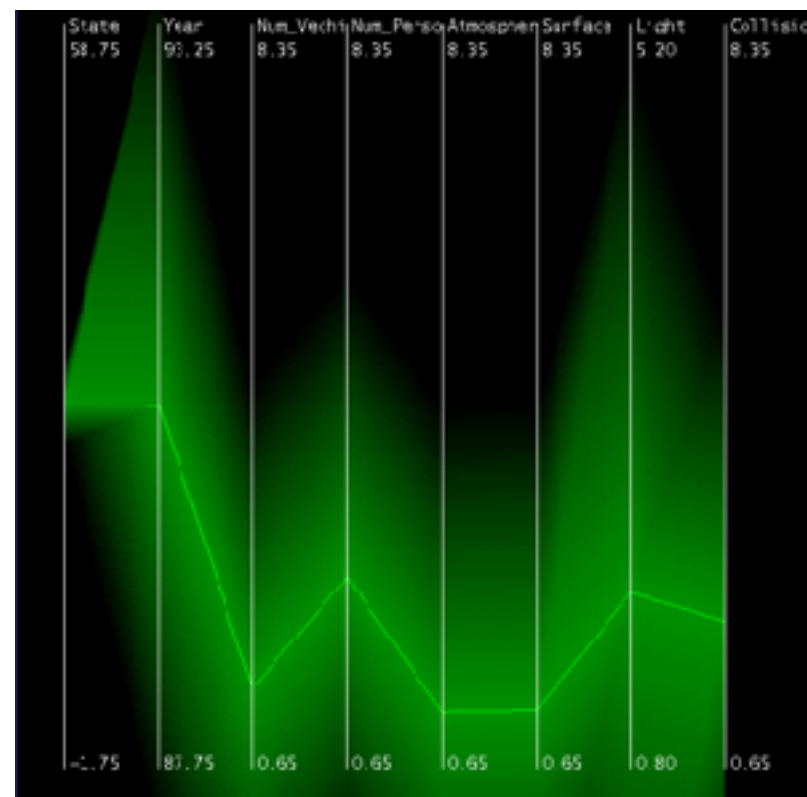
<https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/>

# Dynamic aggregation: Clustering

- **clustering: classification of items into similar bins**
  - based on similarity measure
  - hierarchical algorithms produce "similarity tree": cluster hierarchy
    - agglomerative clustering: start w/ each node as own cluster, then iteratively merge
- **cluster hierarchy: derived data used w/ many dynamic aggregation idioms**
  - cluster more homogeneous than whole dataset
    - statistical measures & distribution more meaningful

# Idiom: Hierarchical parallel coordinates

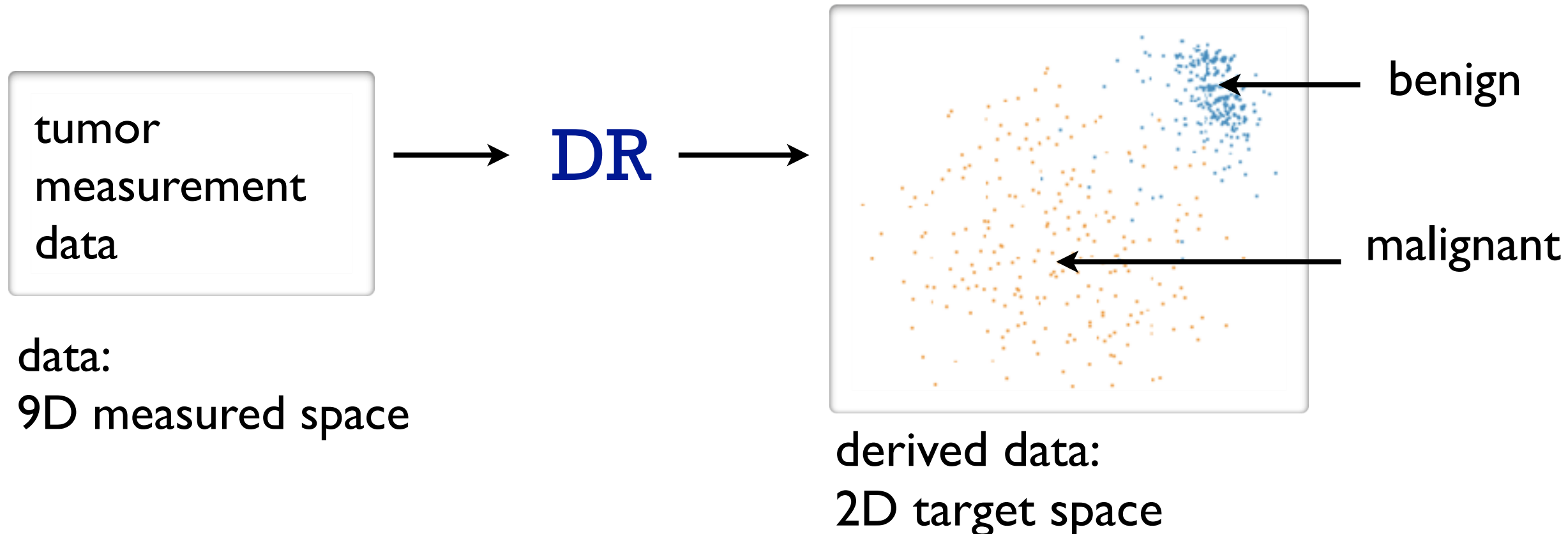
- dynamic item aggregation
- derived data: **cluster hierarchy**
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy



[Hierarchical Parallel Coordinates for Exploration of Large Datasets. Fua, Ward, and Rundensteiner. Proc. IEEE Visualization Conference (Vis '99), pp. 43– 50, 1999.]

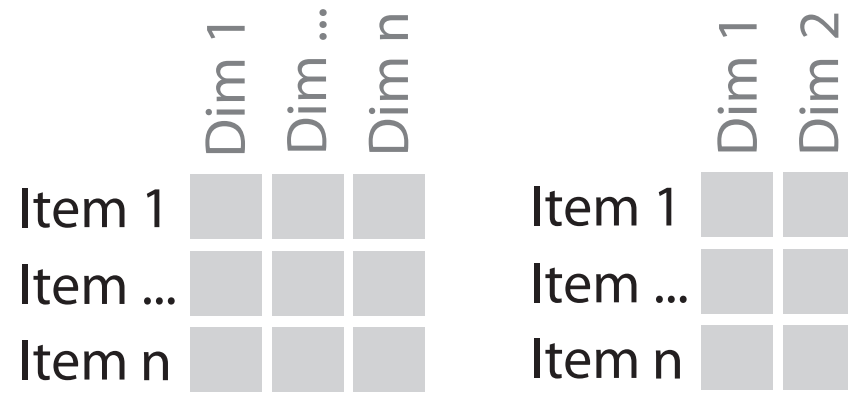
# Attribute aggregation: Dimensionality reduction

- attribute aggregation
  - derive low-dimensional target space from high-dimensional measured space
    - capture most of variance with minimal error
  - use when you can't directly measure what you care about
    - true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    - latent factors, hidden variables



# Idiom: Dimensionality reduction for documents

## Task 1



**In** HD data → **Out** 2D data

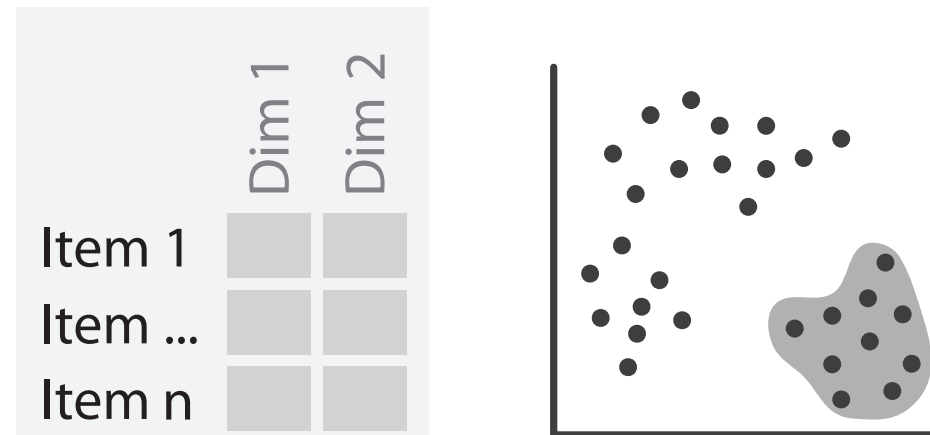
### What?

- In High-dimensional data
- Out 2D data

### Why?

- Produce
- Derive

## Task 2



**In** 2D data → **Out** Scatterplot Clusters & points

### What?

- In 2D data
- Out Scatterplot
- Out Clusters & points

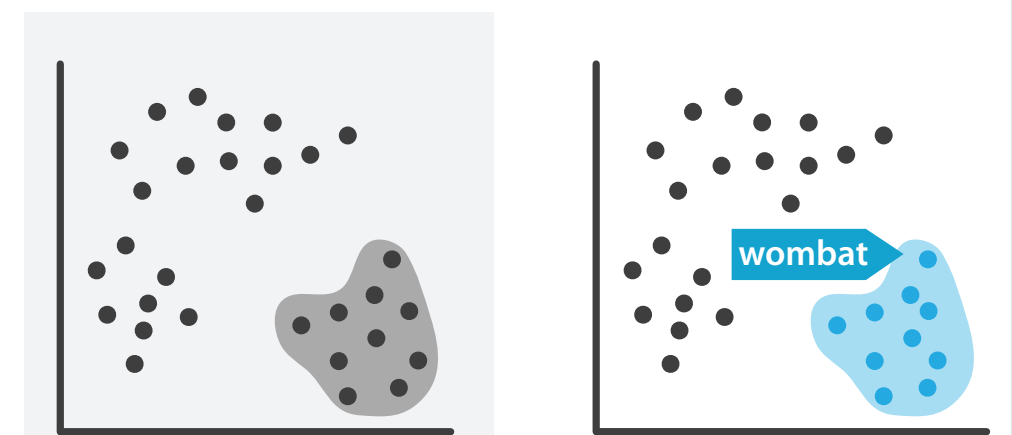
### Why?

- Discover
- Explore
- Identify

### How?

- Encode
- Navigate
- Select

## Task 3



**In** Scatterplot Clusters & points → **Out** Labels for clusters

### What?

- In Scatterplot
- In Clusters & points
- Out Labels for clusters

### Why?

- Produce
- Annotate

# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



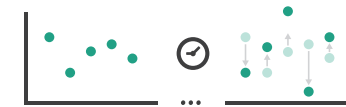
→ Motion

*Direction, Rate, Frequency, ...*

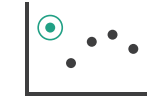


## Manipulate

### → Change



### → Select



### → Navigate

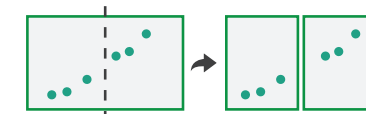


## Facet

### → Juxtapose



### → Partition



### → Superimpose

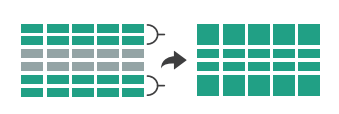


## Reduce

### → Filter



### → Aggregate



### → Embed



What?

Why?

How?



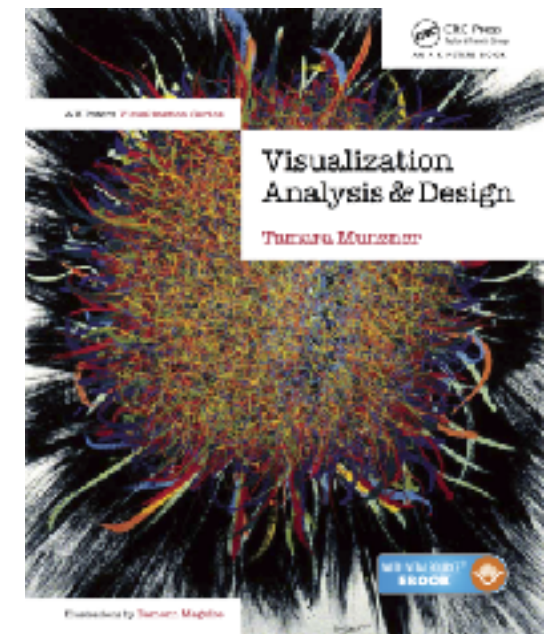
# Visualization Analysis & Design

## *Rules of Thumb (Ch 6)*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

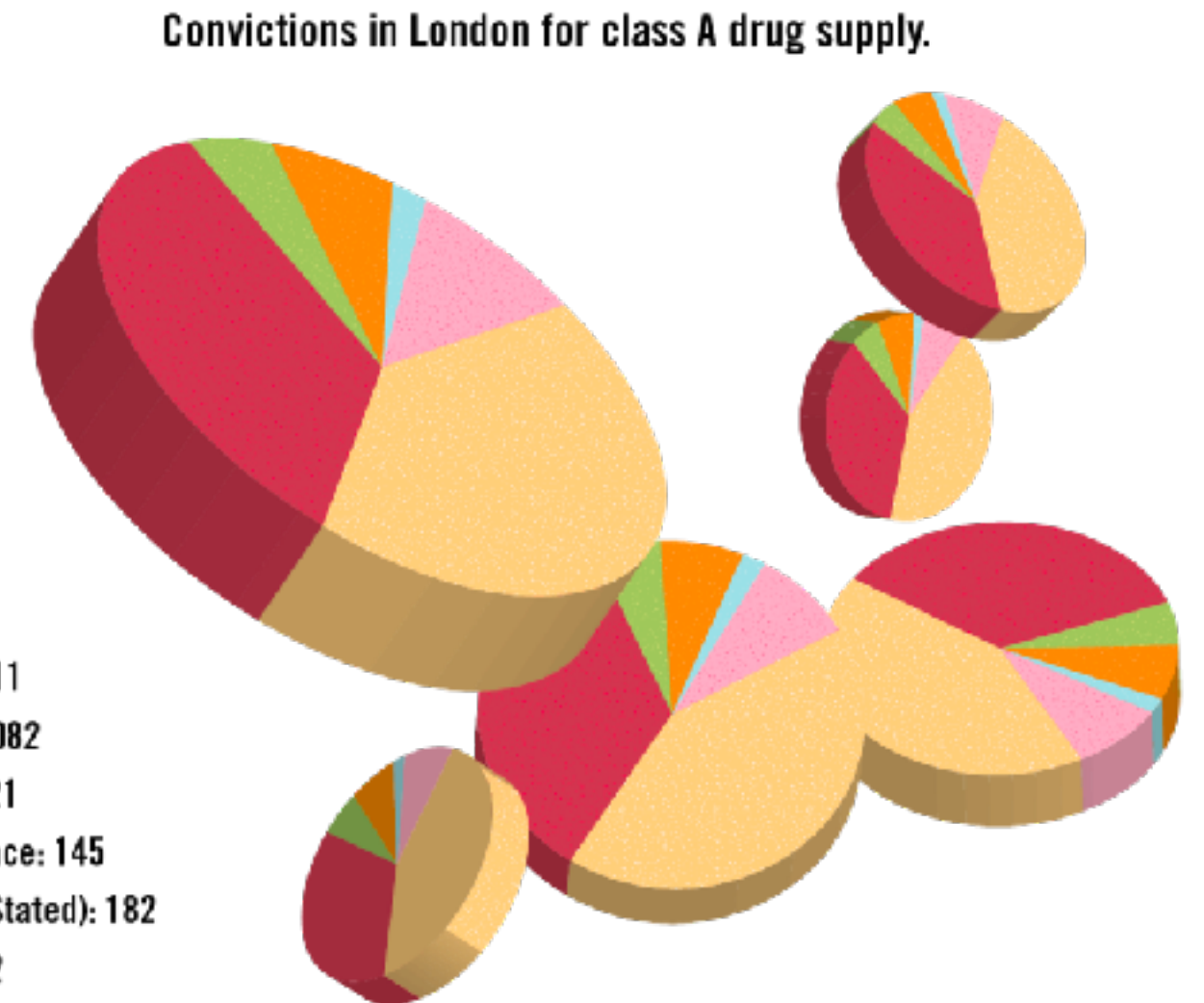
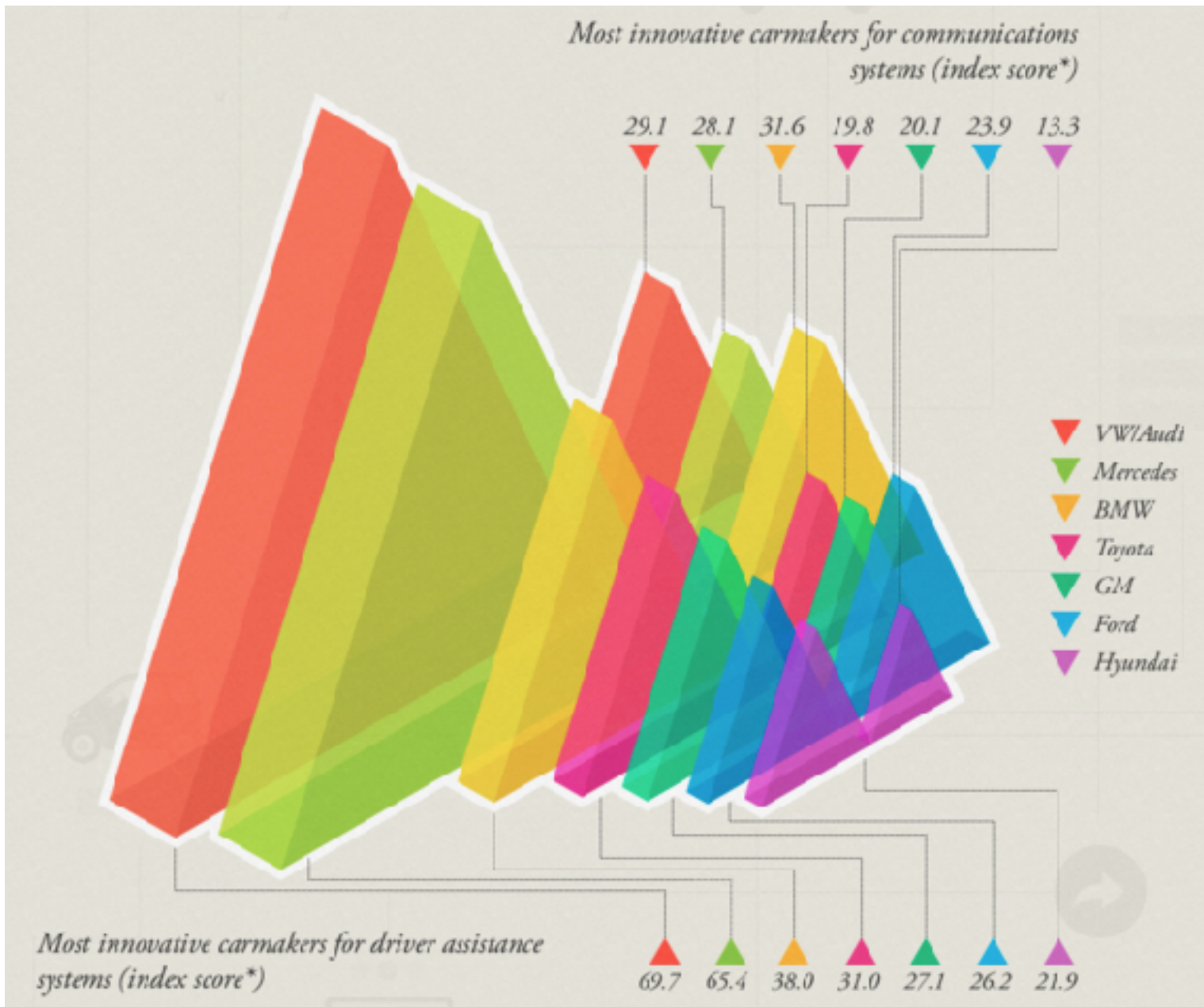
[@tamaramunzner](#)



# Rules of Thumb

- Guidelines and considerations, not absolute rules
  - when to use 3D? when to use 2D?
  - when to use eyes instead of memory?
  - when does immersion help?
  - when to use overviews?
  - how long is too long?
  - which comes first, form or function?

# Unjustified 3D all too common, in the news and elsewhere



<http://viz.wtf/post/137826497077/eye-popping-3d-triangles>

<http://viz.wtf/post/139002022202/designer-drugs-ht-ducqn>

# Depth vs power of the plane

- high-ranked spatial position channels: **planar** spatial position
  - not depth!

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale



Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



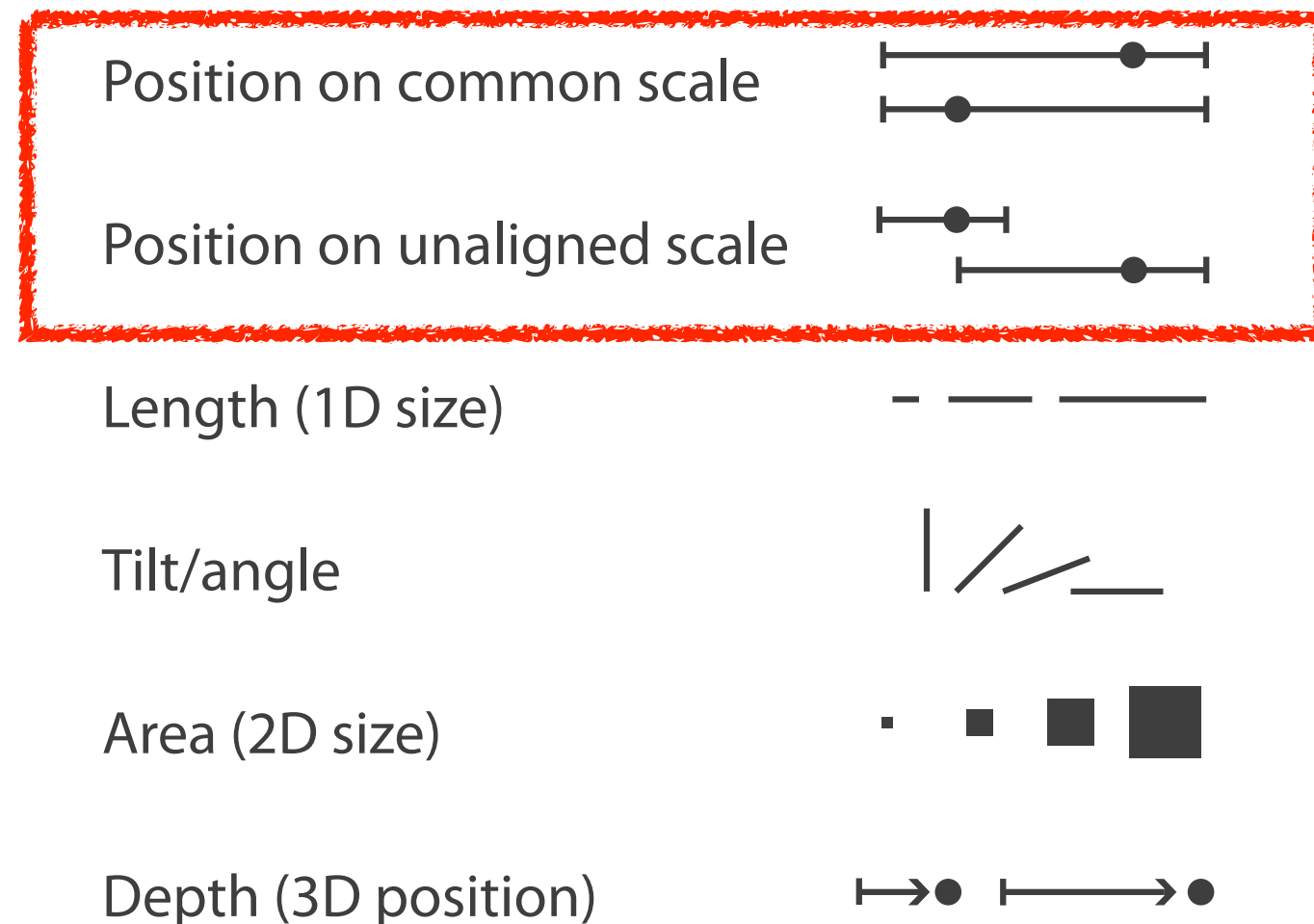
Depth (3D position)



# Depth vs power of the plane

- high-ranked spatial position channels: **planar** spatial position
  - not depth!

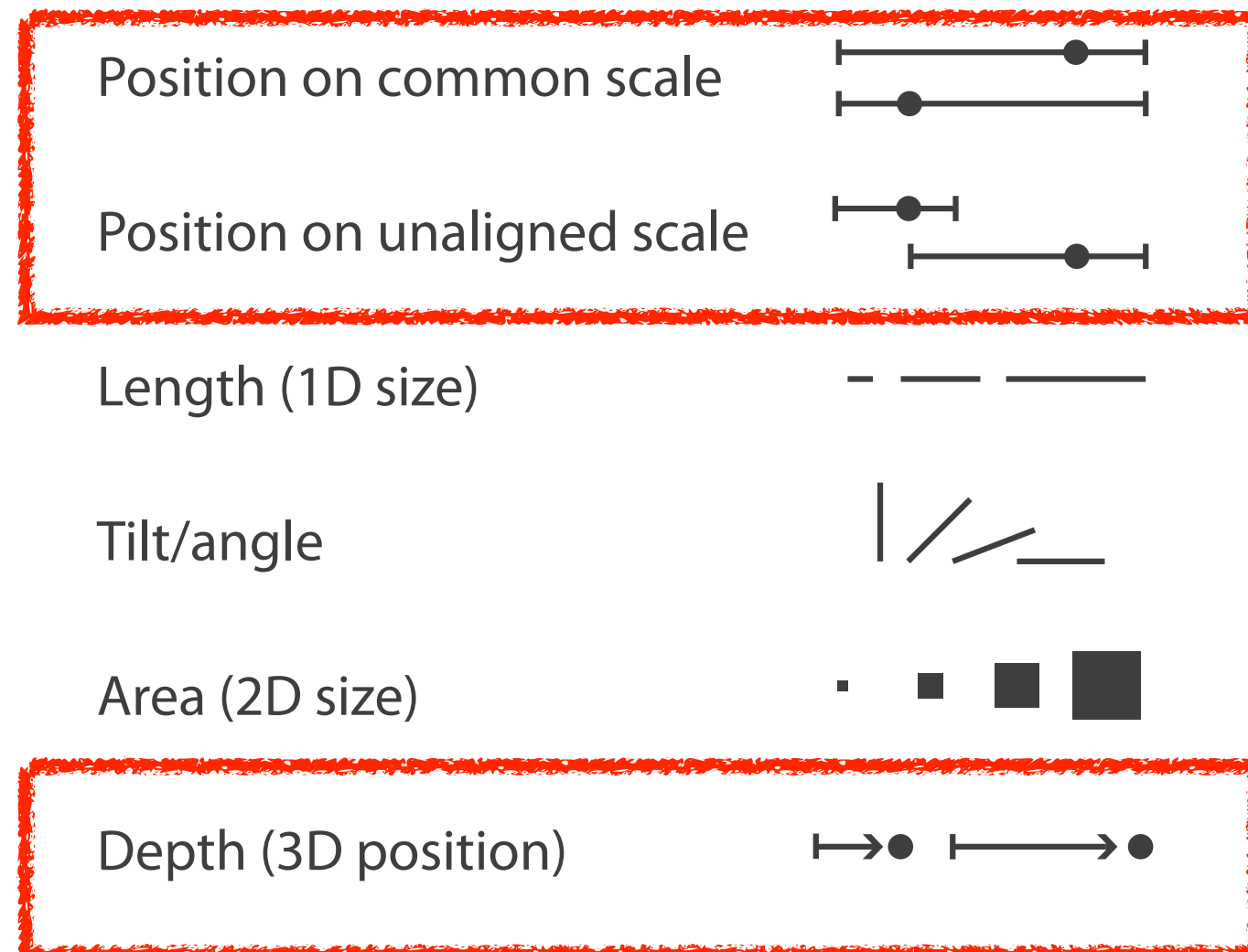
## → Magnitude Channels: Ordered Attributes



# Depth vs power of the plane

- high-ranked spatial position channels: **planar** spatial position
  - not depth!

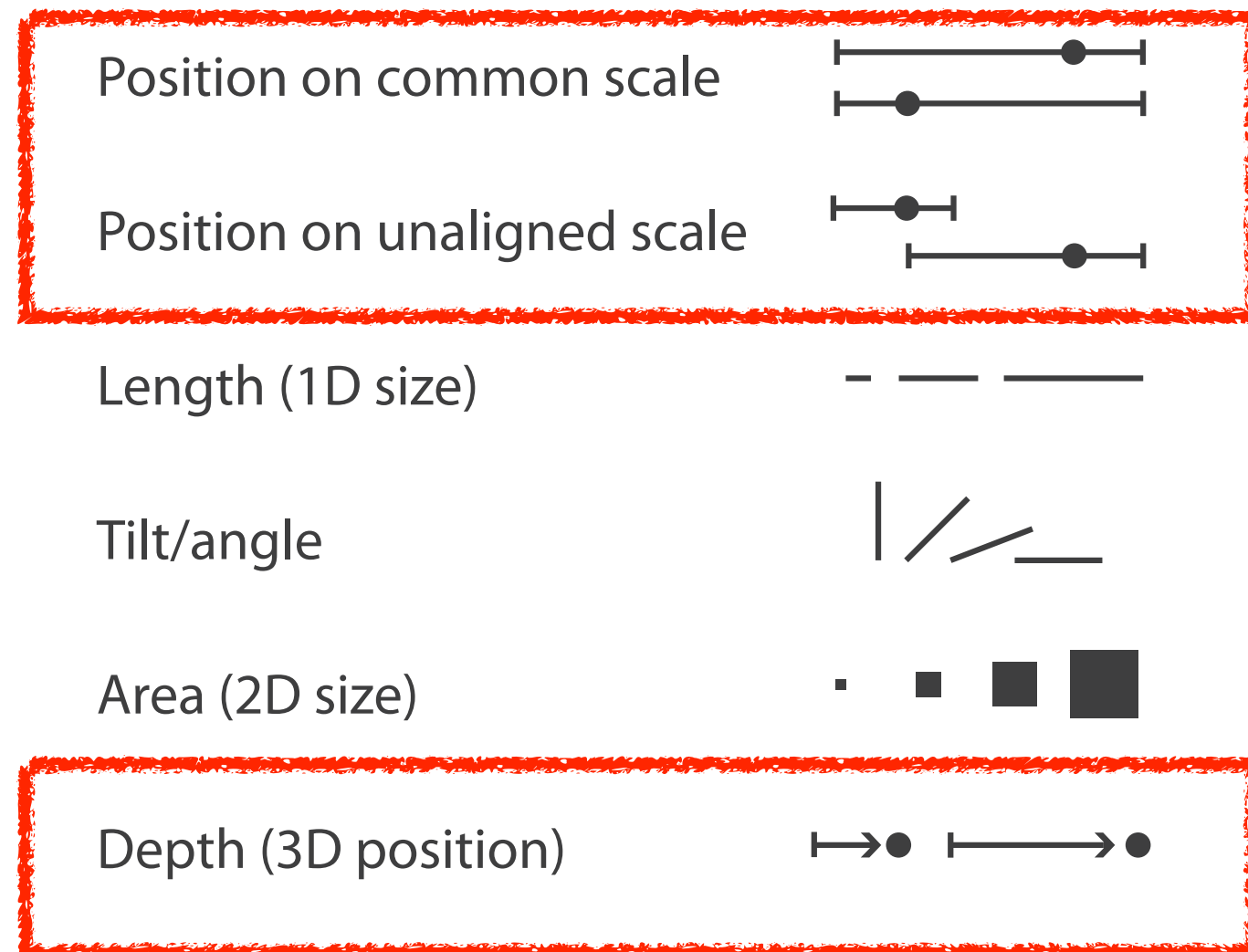
## → Magnitude Channels: Ordered Attributes



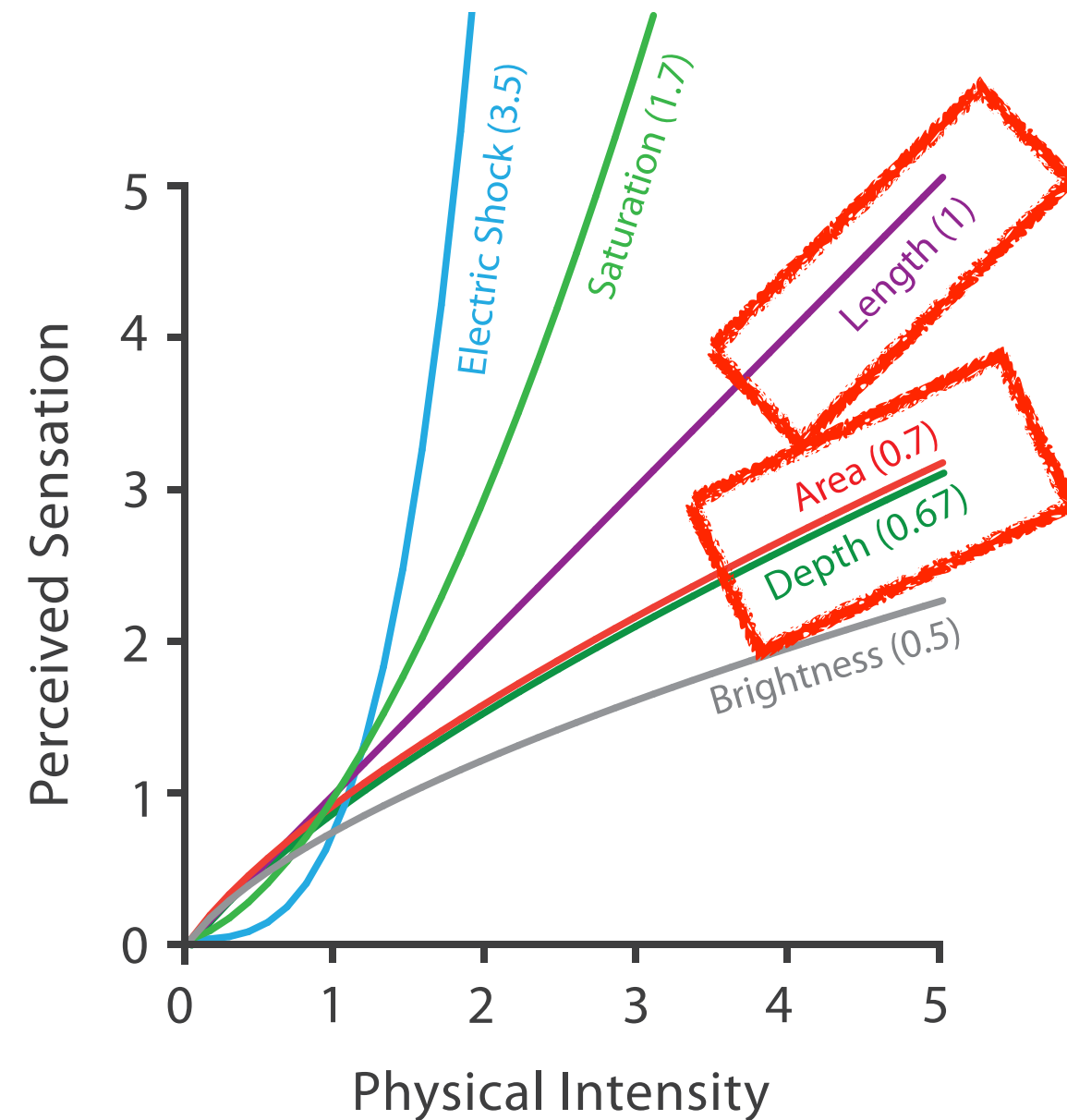
# Depth vs power of the plane

- high-ranked spatial position channels: **planar** spatial position
  - not depth!

## ➔ Magnitude Channels: Ordered Attributes

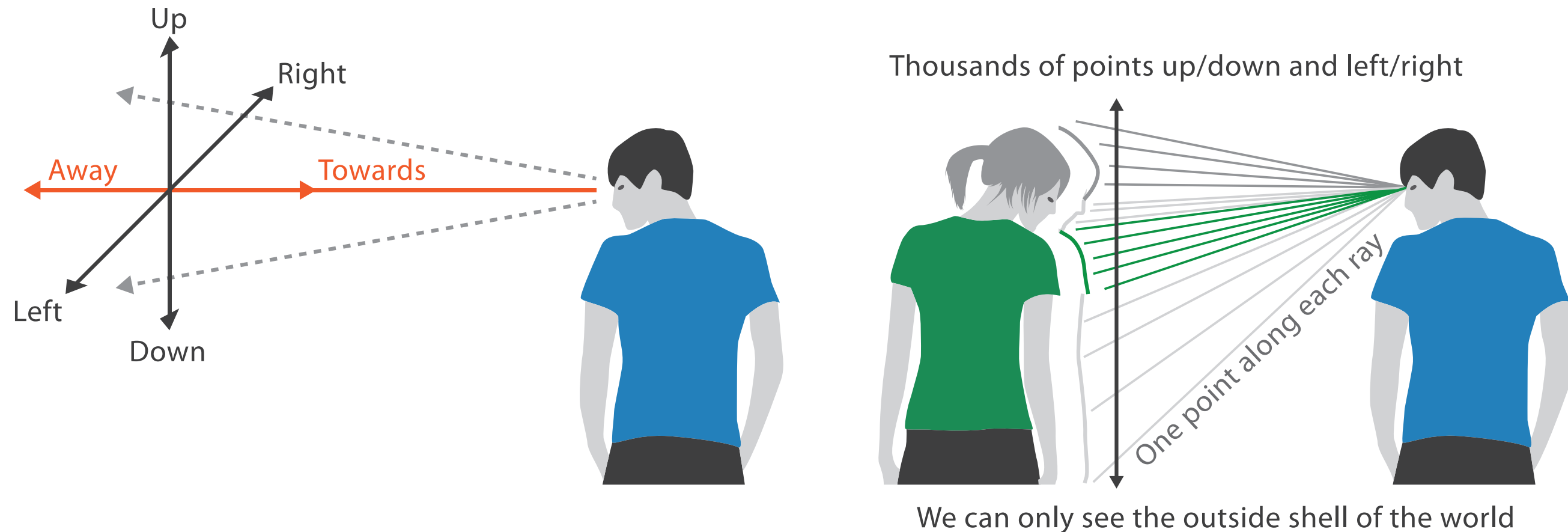


Steven's Psychophysical Power Law:  $S = I^N$



# No unjustified 3D: Danger of depth

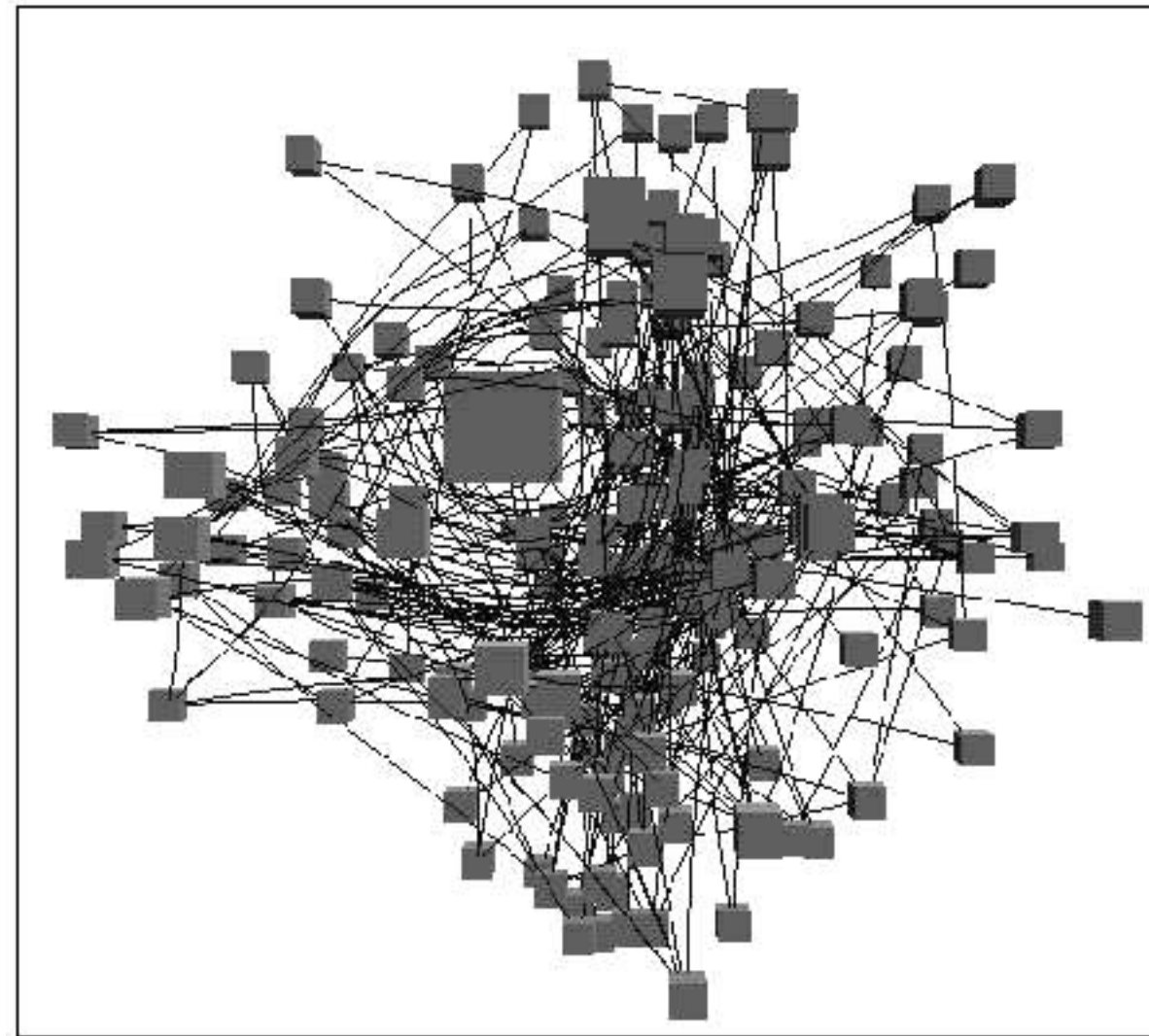
- we don't really live in 3D: we **see** in 2.05D
  - acquire more info on image plane quickly from eye movements
  - acquire more info for depth slower, from head/body motion





# Occlusion hides information

- occlusion
- interaction can resolve, but at cost of time and cognitive load



*[Distortion Viewing Techniques for 3D Data. Carpendale et al. InfoVis I 1996.]*

# Perspective distortion loses information

- perspective distortion
  - interferes with all size channel encodings
  - power of the plane is lost!



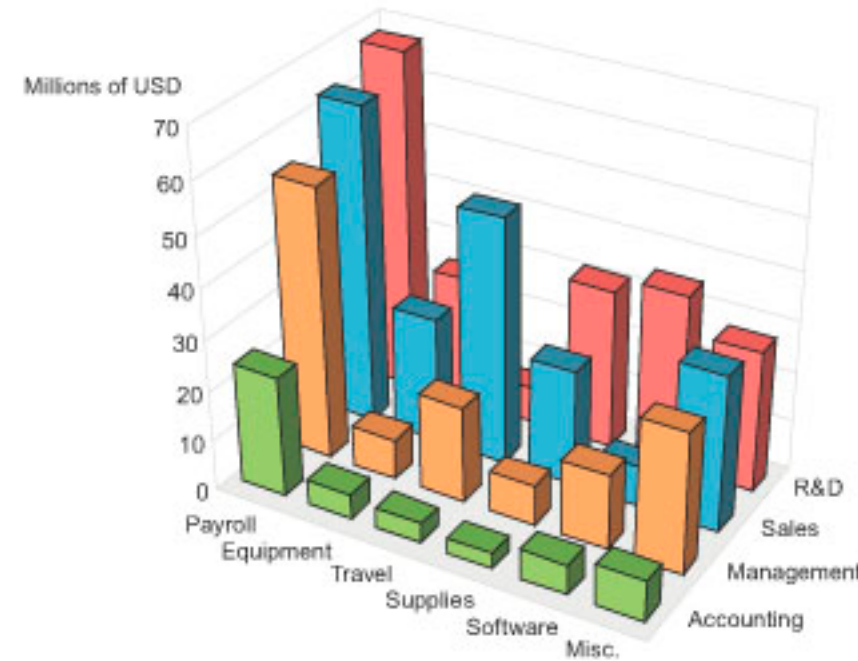
*[Visualizing the Results of Multimedia Web Search Engines. Mukherjea, Hirata, and Hara. InfoVis 96]*

# 3D vs 2D bar charts

- 3D bars:
  - very difficult to justify!
  - perspective distortion
  - occlusion
- faceting into 2D almost always better choice

Question 7: Which graph makes it easier to determine R&D's travel expense?

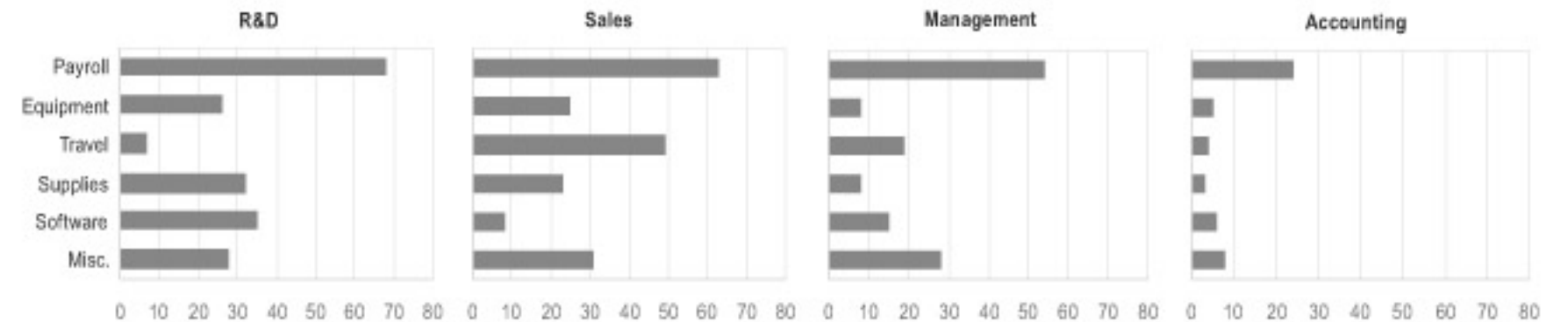
2006 Expenses by Department



3-D Bar Graph (left)

2-D Bar Graphs (below)

2006 Expenses by Department in Millions of USD

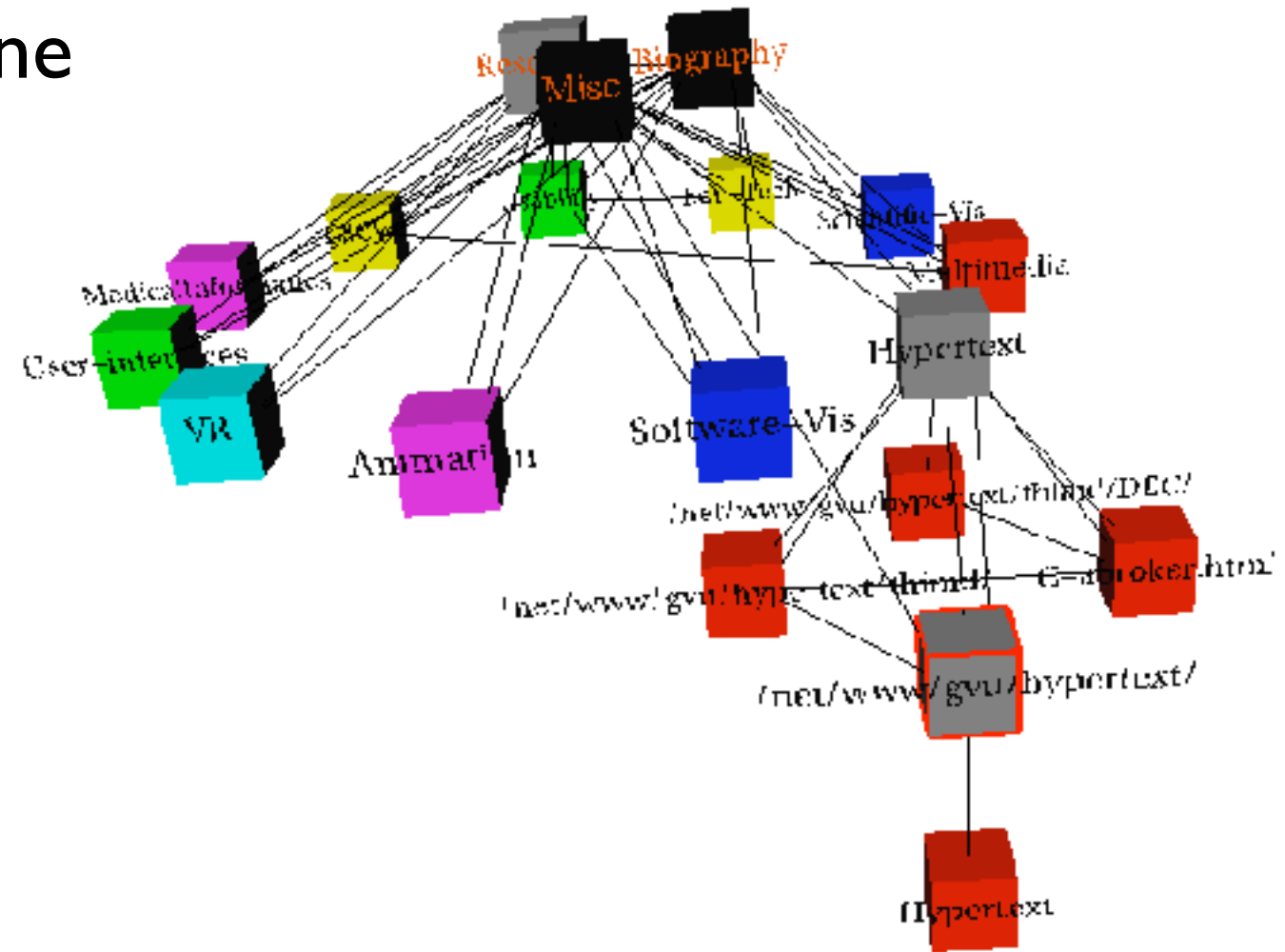


# Tilted text isn't legible

- text legibility
  - far worse when tilted from image plane

- further reading

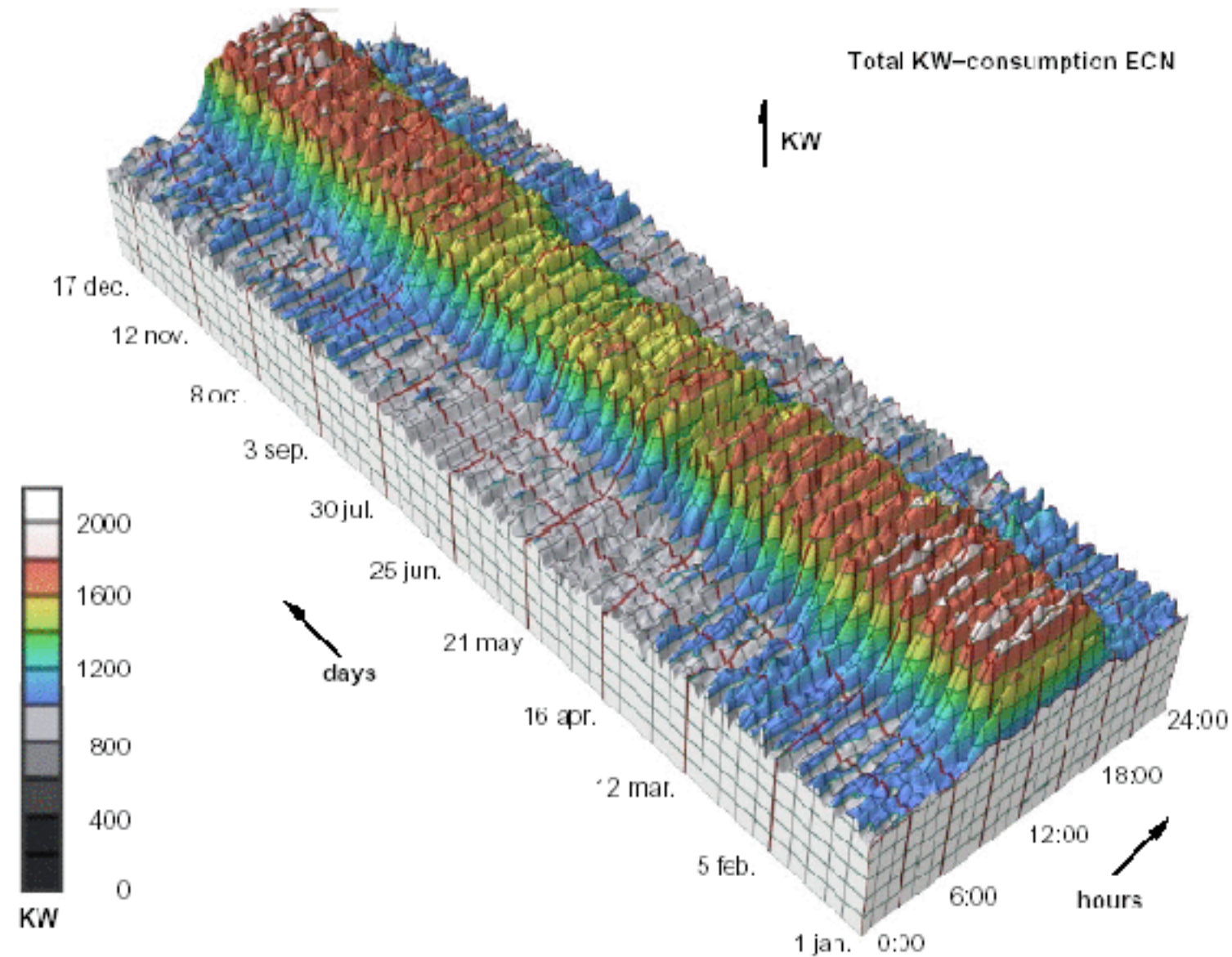
Exploring and Reducing the Effects of Orientation on Text Readability in Volumetric Displays.  
Grossman et al. CHI 2007



[Visualizing the World-Wide Web with the Navigational View Builder. Mukherjea and Foley. Computer Networks and ISDN Systems, 1995.]

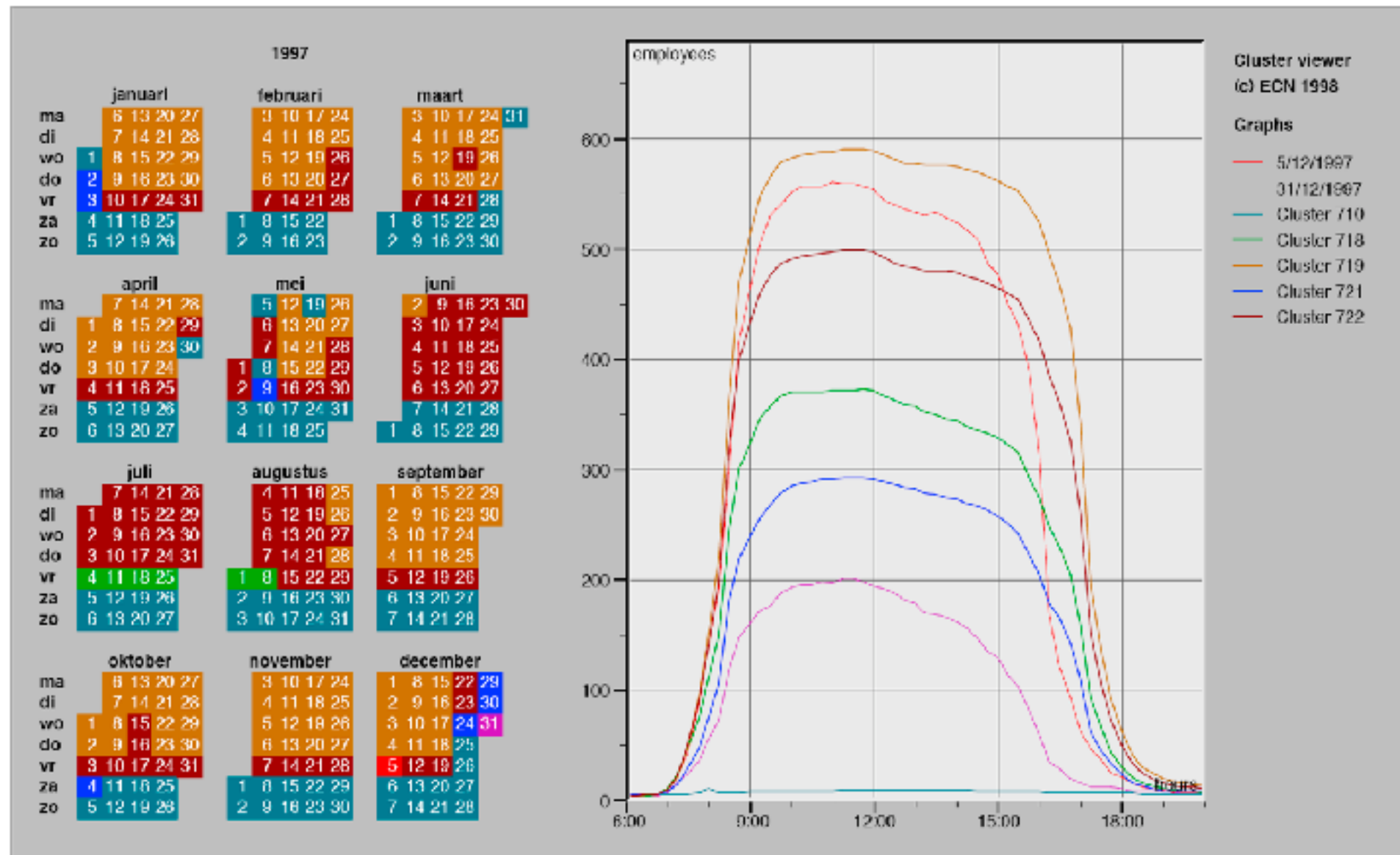
# No unjustified 3D example: Time-series data

- extruded curves: detailed comparisons impossible



# No unjustified 3D example: Transform for new data abstraction

- derived data: cluster hierarchy
- juxtapose multiple views: calendar, superimposed 2D curves



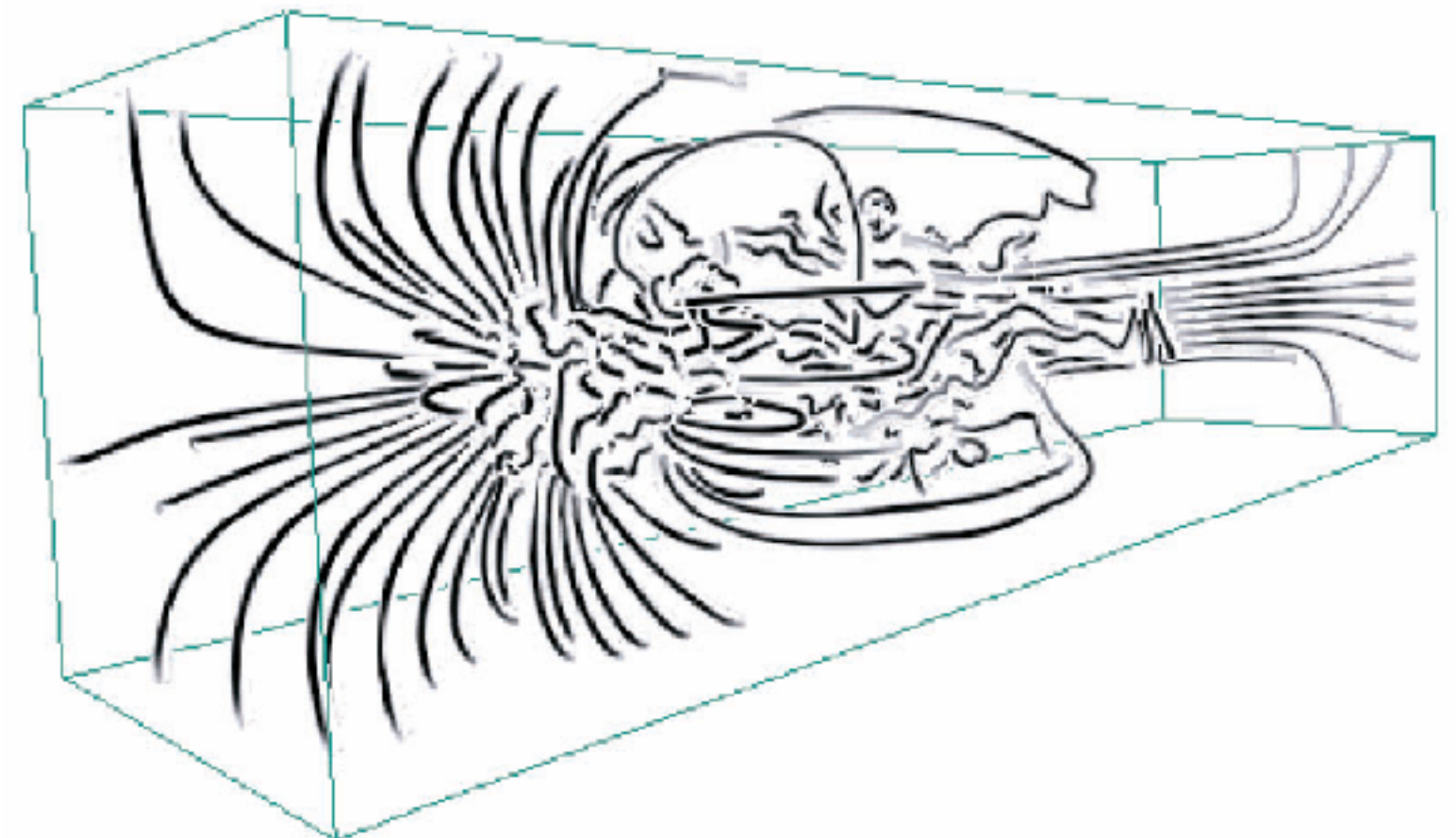
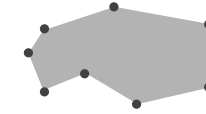
# Justified 3D: shape perception

- benefits outweigh costs when task is shape perception for 3D spatial data
  - interactive navigation supports synthesis across many viewpoints

 Targets

⊙ Spatial Data

→ Shape



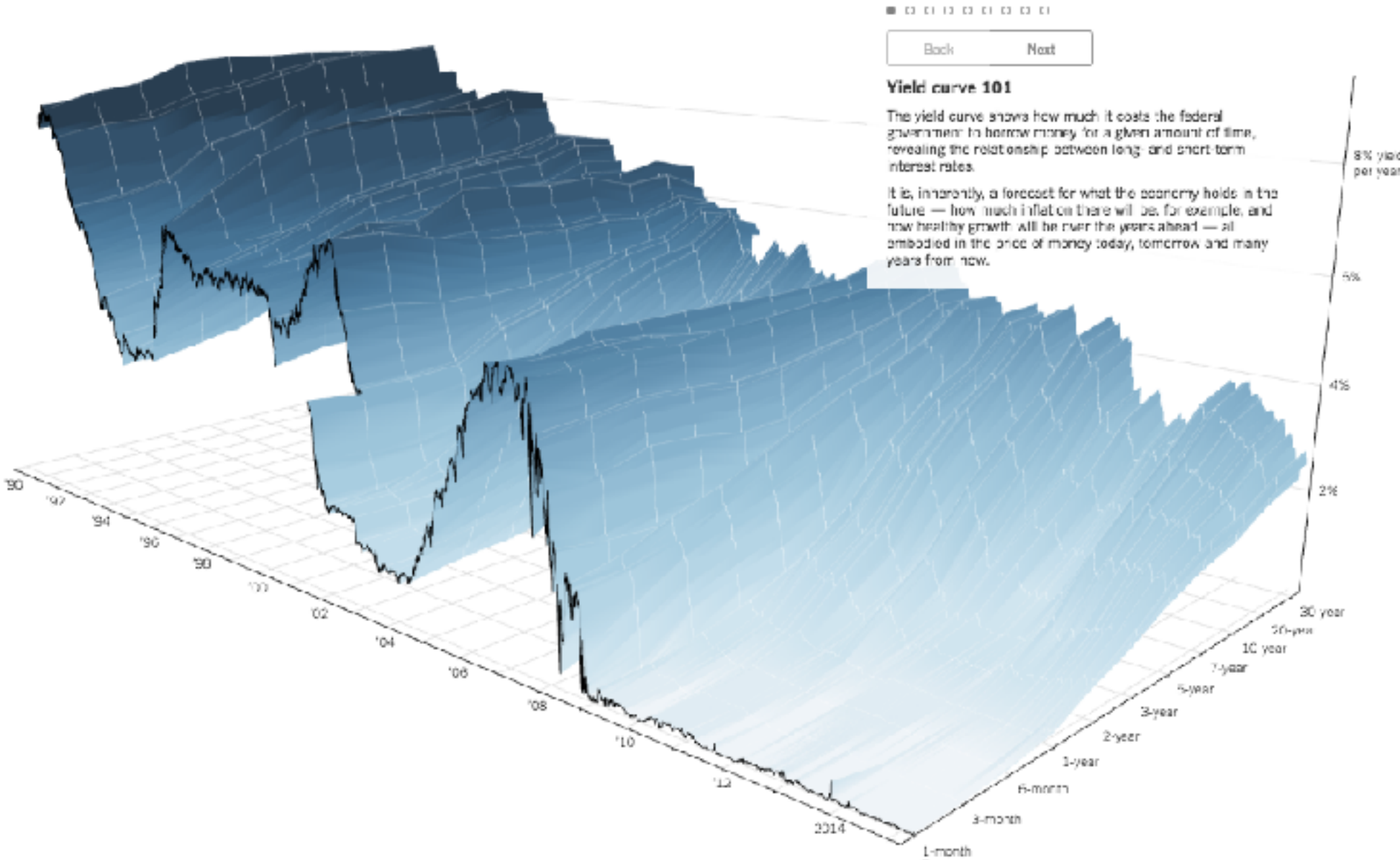
[Image-Based Streamline Generation and Rendering. Li and Shen.  
IEEE Trans. Visualization and Computer Graphics (TVCG) 13:3 (2007), 630–640.]

# Justified 3D: Economic growth curve

- constrained navigation steps through carefully designed viewpoints

## A 3-D View of a Chart That Predicts The Economic Future: The Yield Curve

By GREGOR ALSCH and AMANDA COX MARCH 18, 2015

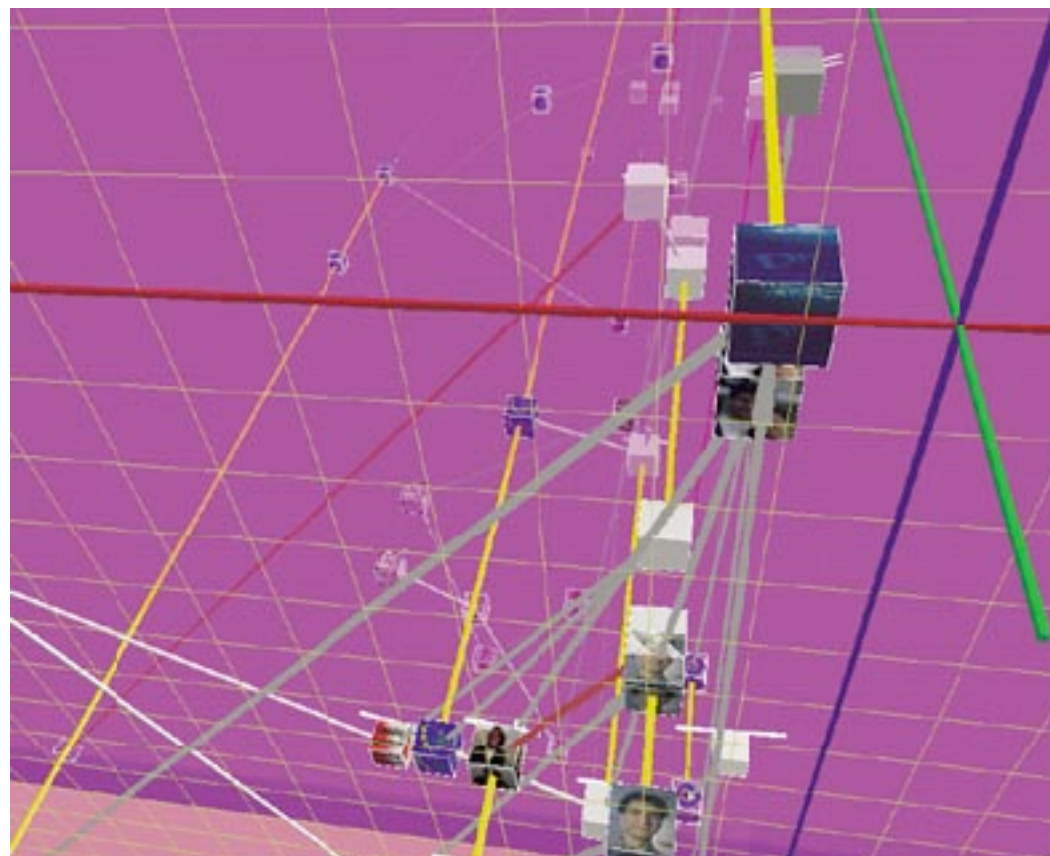


<http://www.nytimes.com/interactive/2015/03/19/upshot/3d-yield-curve-economic-growth.html>



# No unjustified 3D

- 3D legitimate for true 3D spatial data
- 3D needs very careful justification for abstract data
  - enthusiasm in 1990s, but now skepticism
  - be especially careful with 3D for point clouds or networks



*[WEBPATH-a three dimensional Web history. Frecon and Smith. Proc. InfoVis 1999]*

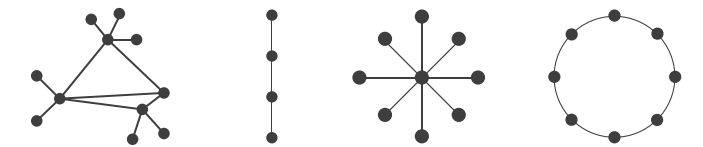
# No unjustified 2D

- consider whether network data requires 2D spatial layout
  - especially if reading text is central to task!
  - arranging as network means lower information density and harder label lookup compared to text lists
- benefits outweigh costs when topological structure/context important for task
  - be especially careful for search results, document collections, ontologies



## → Network Data

### → Topology



### → Paths



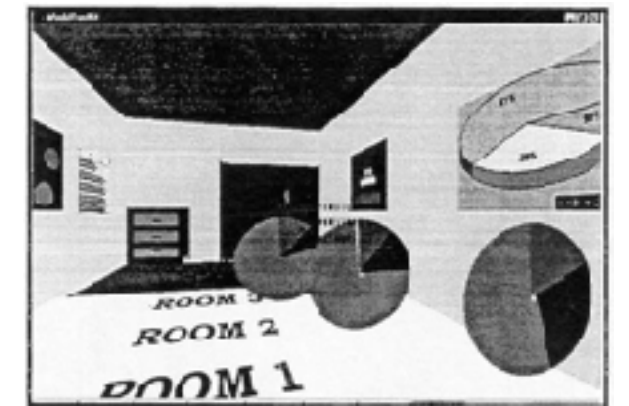
# Eyes beat memory

- principle: external cognition vs. internal memory
  - easy to compare by moving eyes between side-by-side views
  - harder to compare visible item to memory of what you saw
- implications for animation
  - great for choreographed storytelling
  - great for transitions between two states
  - poor for many states with changes everywhere
    - consider small multiples instead

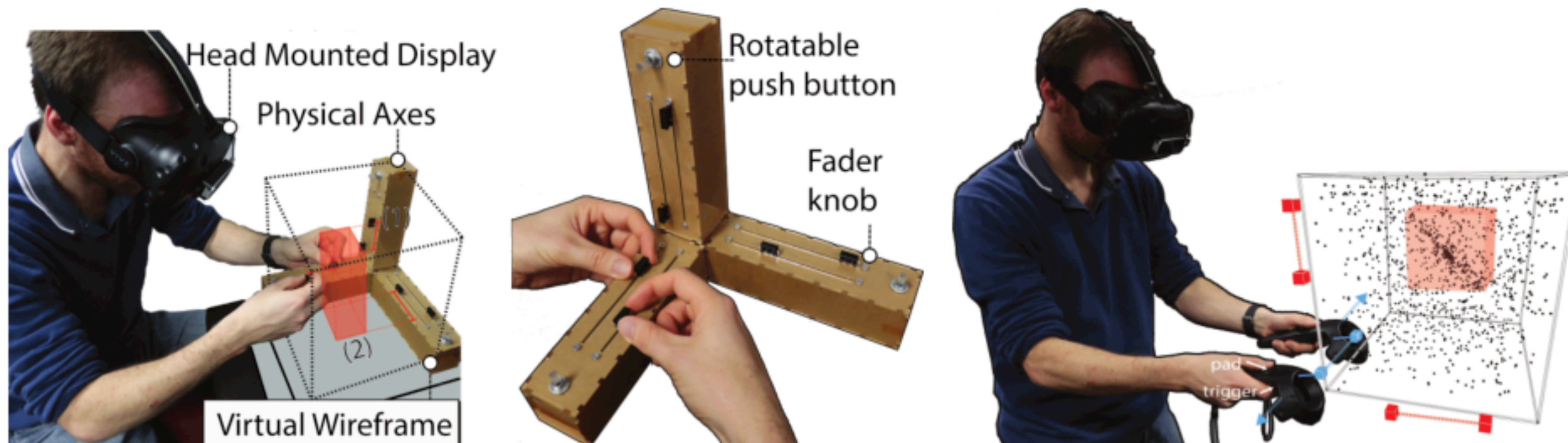


# Resolution beats immersion

- immersion typically not helpful for abstract data
  - do not need sense of presence or stereoscopic 3D
  - desktop also better for workflow integration
- resolution much more important: pixels are the scarcest resource
- first wave: virtual reality for abstract data difficult to justify
- second wave: AR/MR (augmented/mixed reality) has more promise



[Development of an information visualization tool using virtual reality. Kirner and Martins. Proc. Symp. Applied Computing 2000]



[A Design Space for Spatio-Data Coordination: Tangible Interaction Devices for Immersive Information Visualisation. Cordeil, Bach, Li, Elliott, and Dwyer. Proc. PacificVis 2017 Notes.]

# Overview first, zoom and filter, details on demand

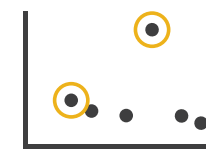
- influential mantra from Shneiderman

*[The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Shneiderman. Proc. IEEE Visual Languages, pp. 336–343, 1996.]*

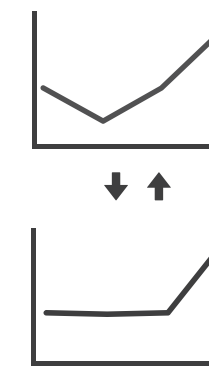
- **overview = summary**
  - microcosm of full vis design problem

→ Query

→ Identify



→ Compare



→ Summarise



# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control



# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)
- **scalability considerations**

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)
- **scalability considerations**
  - highlight selection without complete redraw of view (graphics frontbuffer)

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)
- **scalability considerations**
  - highlight selection without complete redraw of view (graphics frontbuffer)
  - show hourglass for multi-second operations (check for cancel/undo)

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)
- **scalability considerations**
  - highlight selection without complete redraw of view (graphics frontbuffer)
  - show hourglass for multi-second operations (check for cancel/undo)
  - show progress bar for long operations (process in background thread)

# Rule of thumb: **Responsiveness is required**

- *visual feedback: three rough categories*
  - *0.1 seconds: perceptual processing*
    - subsecond response for mouseover highlighting - ballistic motion
  - *1 second: immediate response*
    - fast response after mouseclick, button press - Fitts' Law limits on motor control
  - *10 seconds: brief tasks*
    - bounded response after dialog box - mental model of heavyweight operation (file load)
- **scalability considerations**
  - highlight selection without complete redraw of view (graphics frontbuffer)
  - show hourglass for multi-second operations (check for cancel/undo)
  - show progress bar for long operations (process in background thread)
  - rendering speed when item count is large (guaranteed frame rate)

# Function first, form next

- dangerous to start with aesthetics
  - usually impossible to add function retroactively

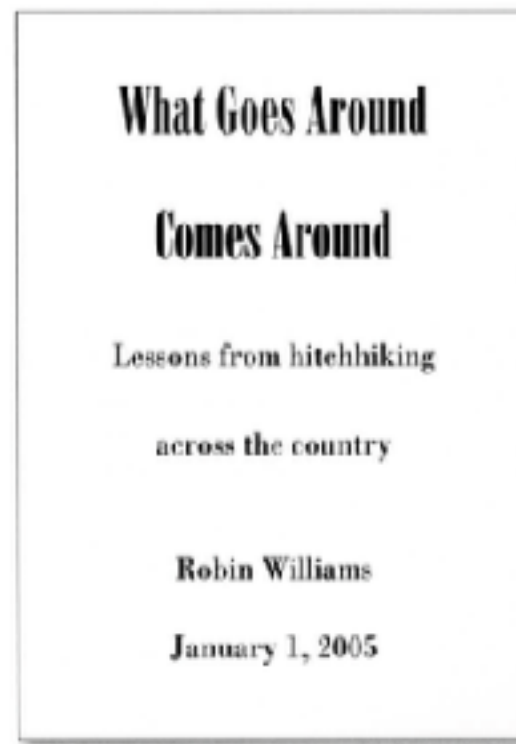
# Function first, form next

- dangerous to start with aesthetics
  - usually impossible to add function retroactively
- start with focus on functionality
  - possible to improve aesthetics later on, as refinement
  - if no expertise in-house, find good graphic designer to work with
  - aesthetics do matter! another level of function
    - visual hierarchy, alignment, flow
    - Gestalt principles in action



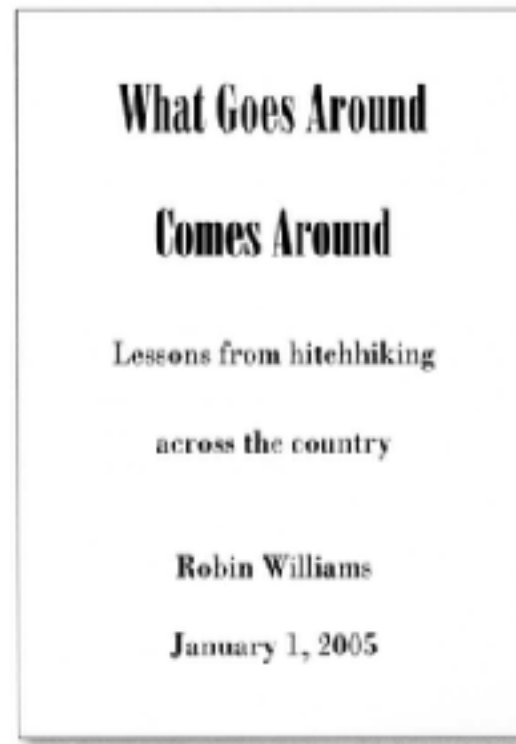
# Form: Basic graphic design ideas

# Form: Basic graphic design ideas



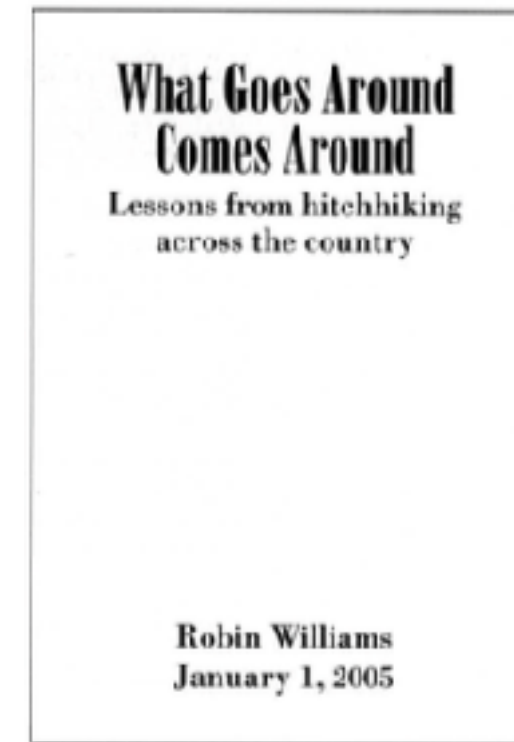
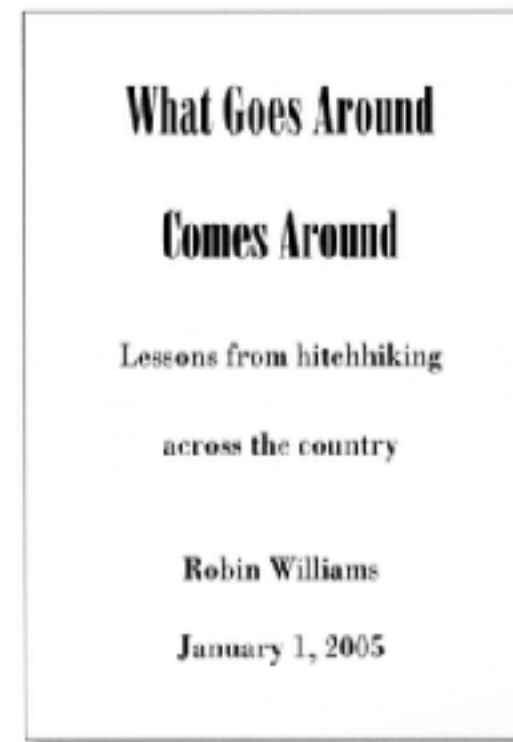
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated



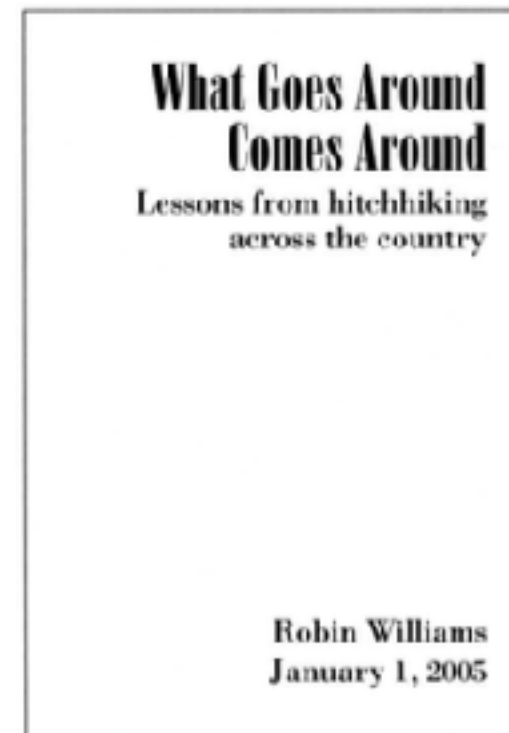
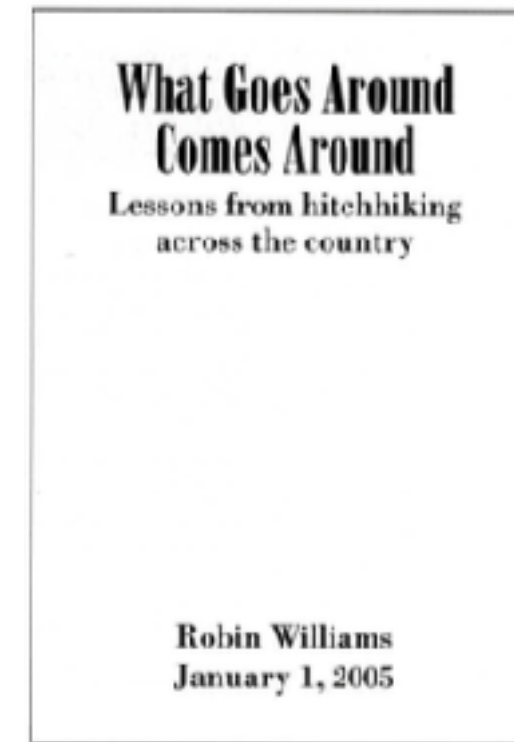
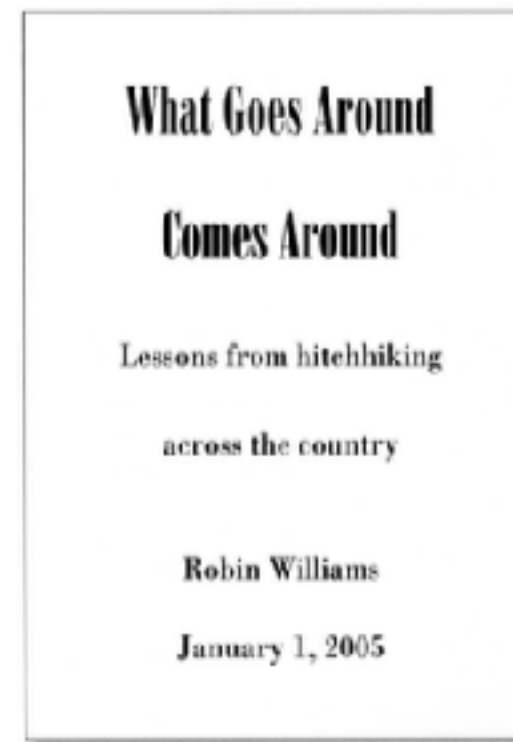
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering



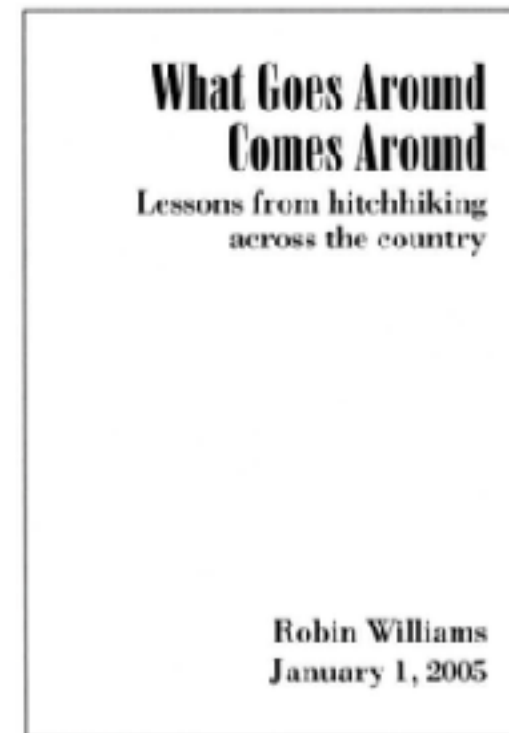
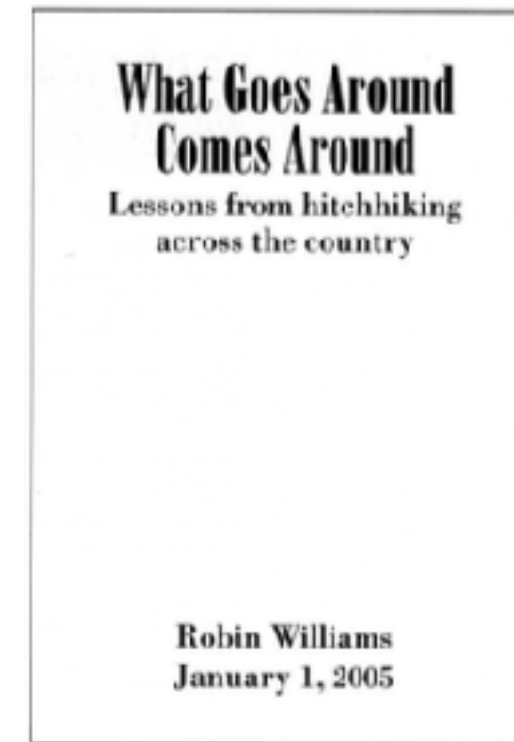
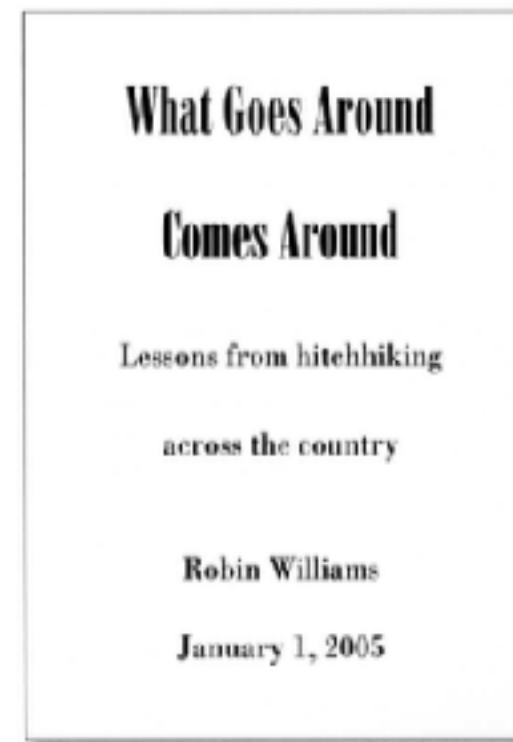
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering



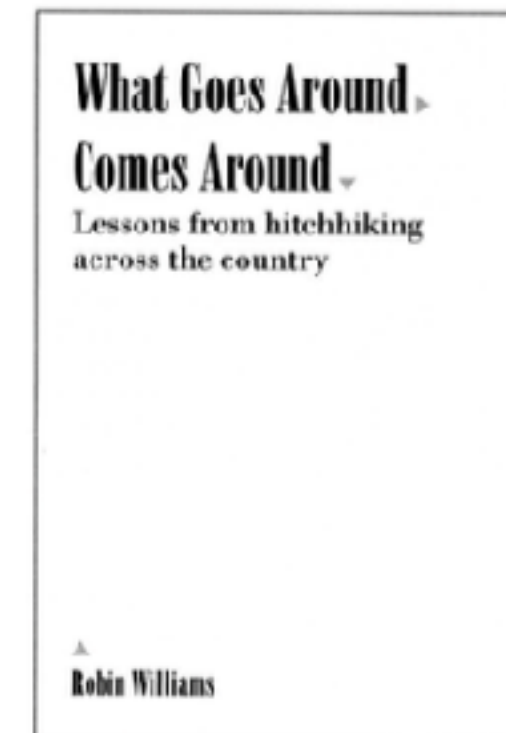
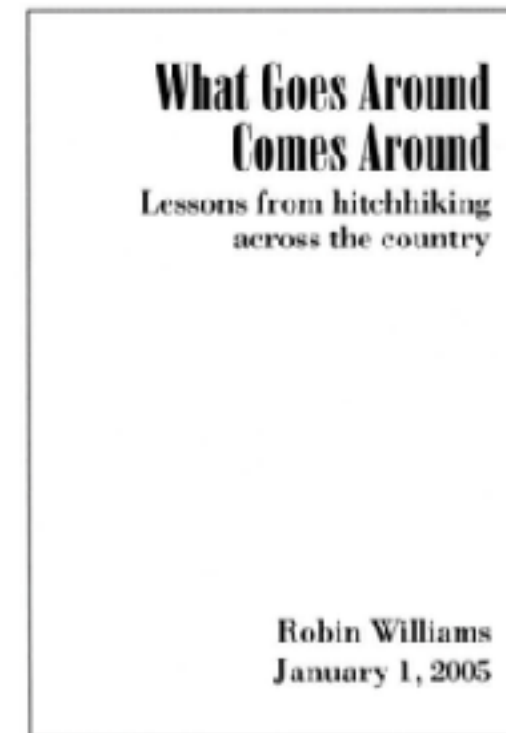
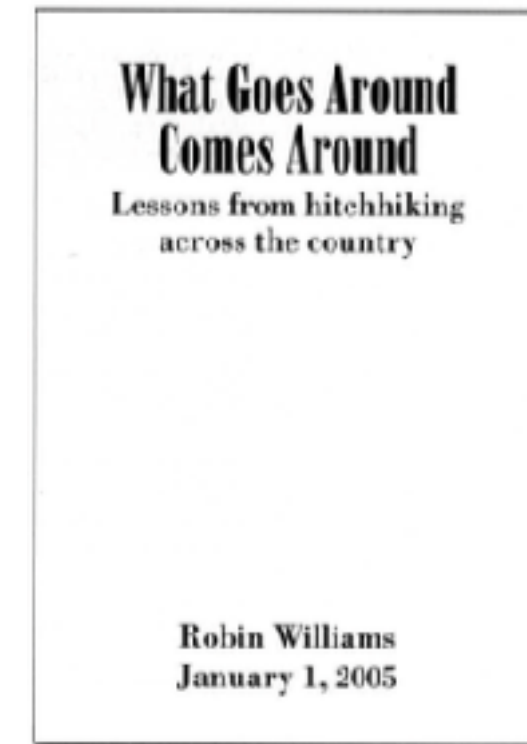
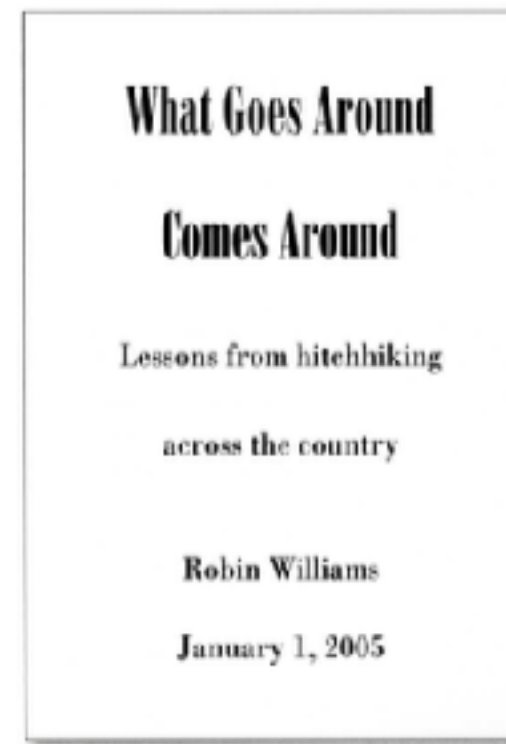
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering
- repetition
  - do unify by pushing existing consistencies



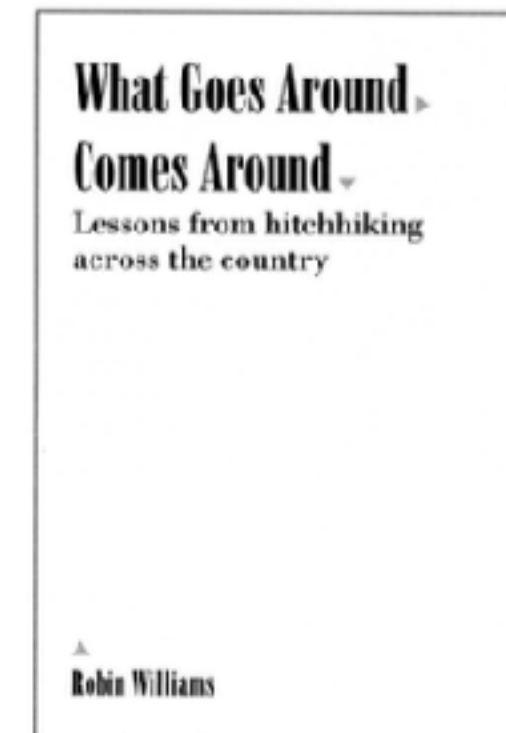
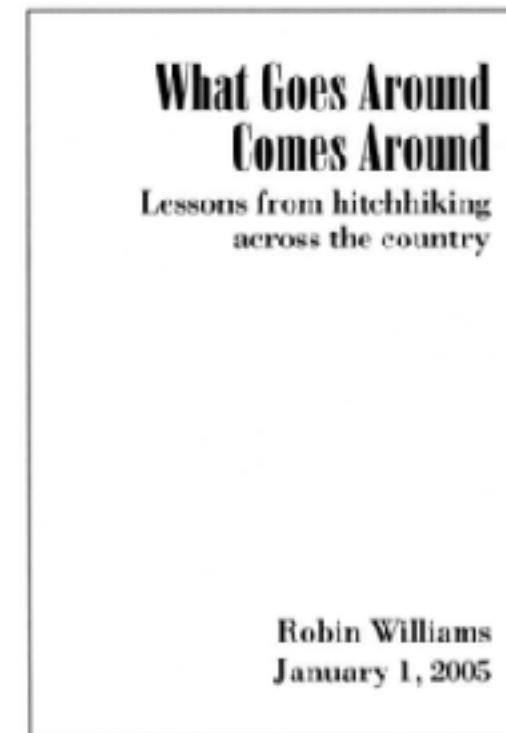
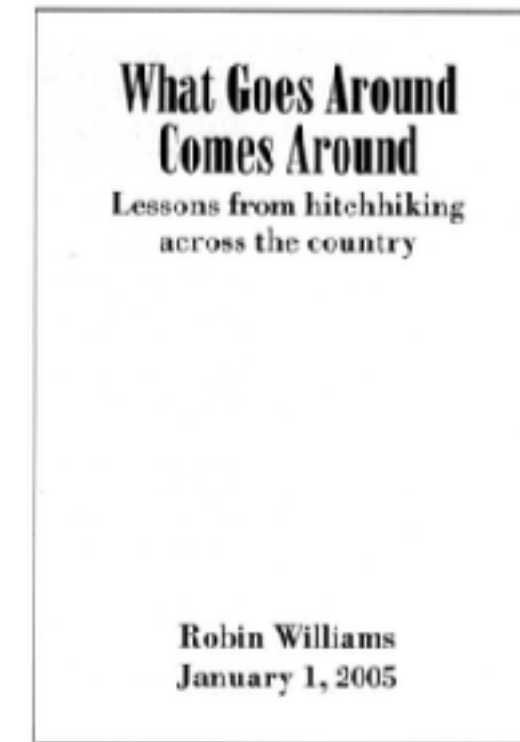
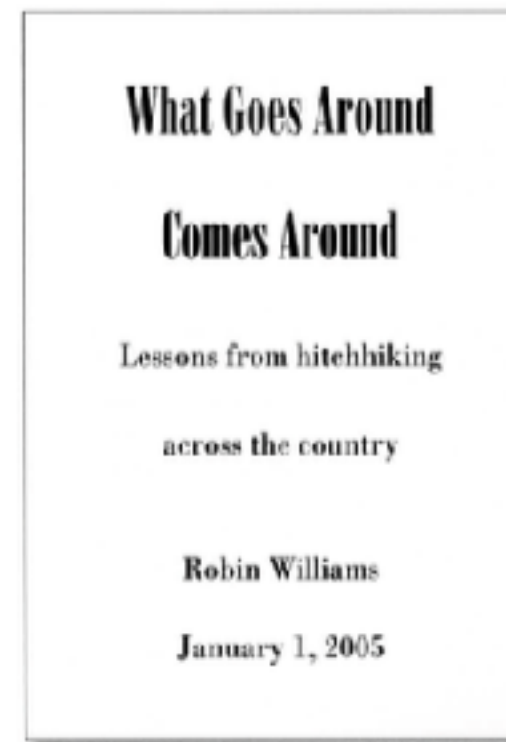
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering
- repetition
  - do unify by pushing existing consistencies



# Form: Basic graphic design ideas

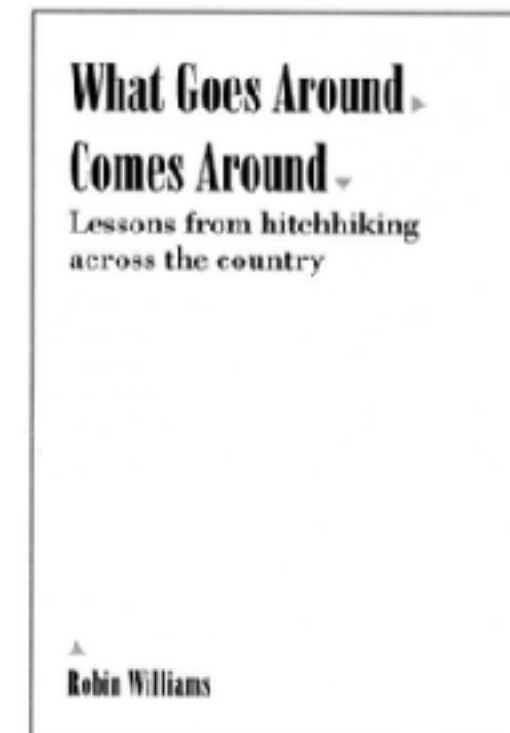
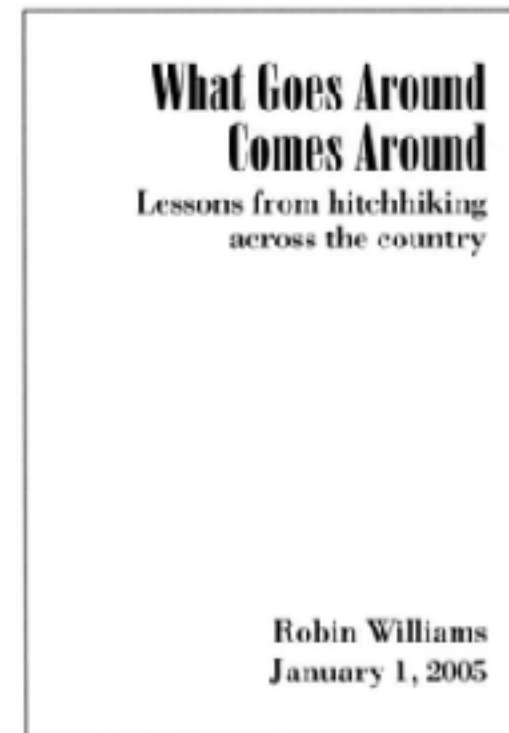
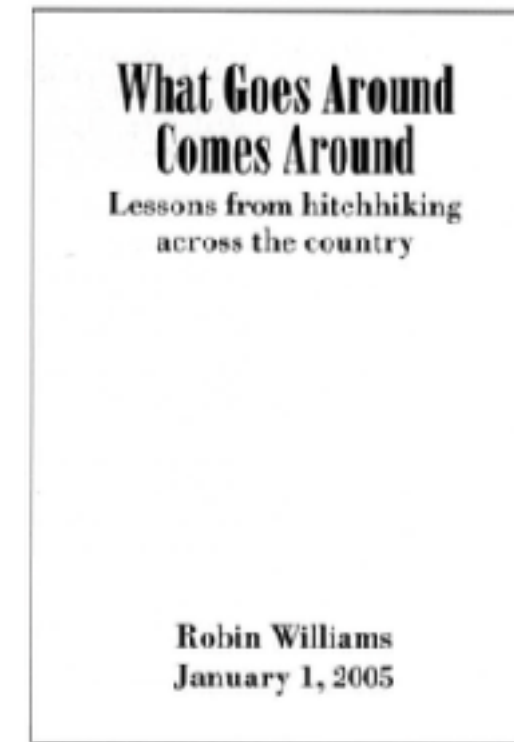
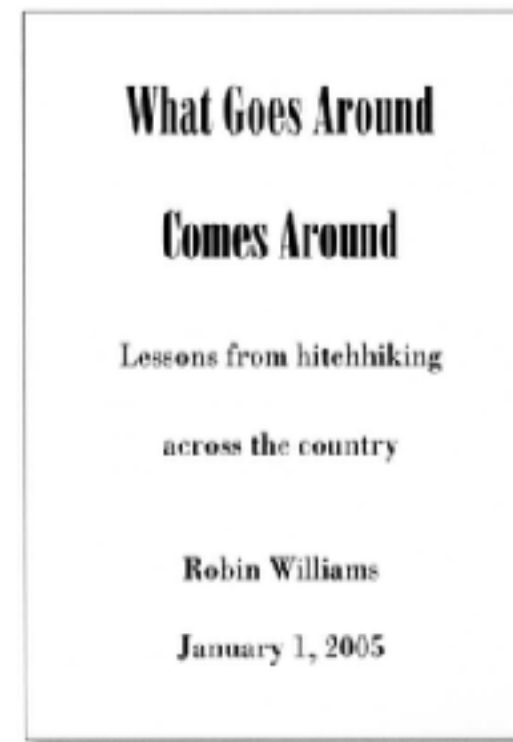
- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering
- repetition
  - do unify by pushing existing consistencies
- contrast
  - if not identical, then very different
  - avoid not quite the same





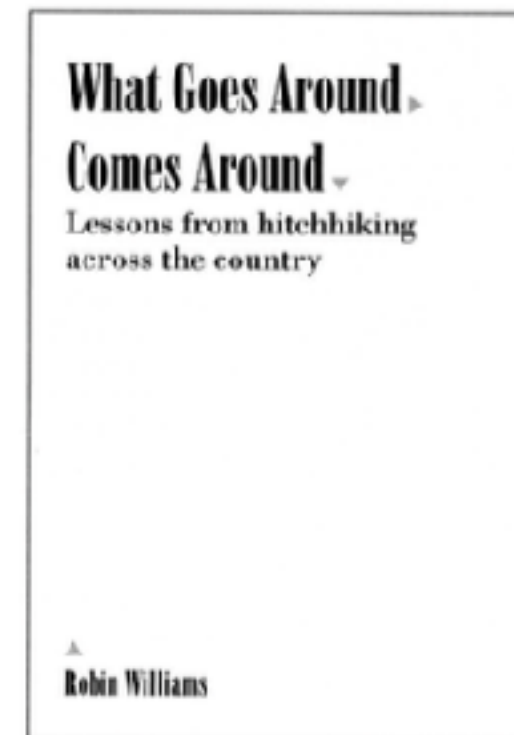
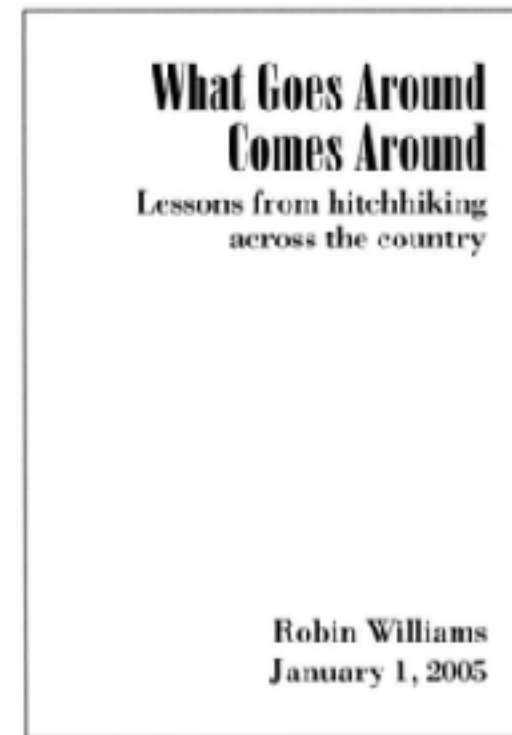
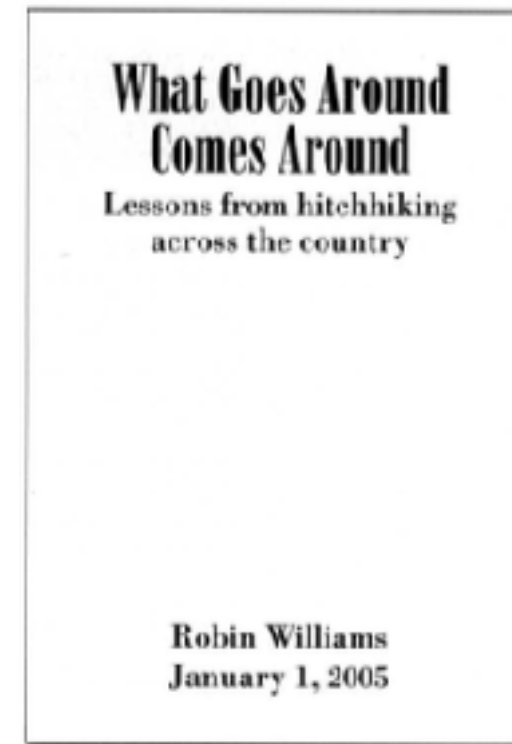
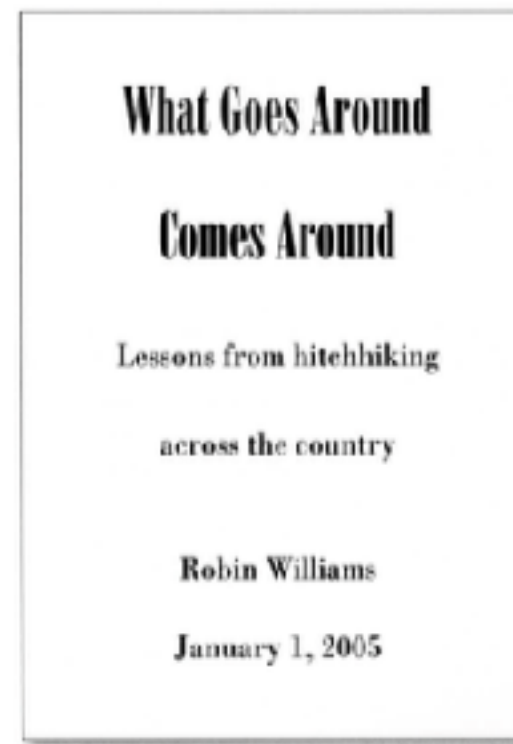
# Form: Basic graphic design ideas

- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering
- repetition
  - do unify by pushing existing consistencies
- contrast
  - if not identical, then very different
  - avoid not quite the same



# Form: Basic graphic design ideas

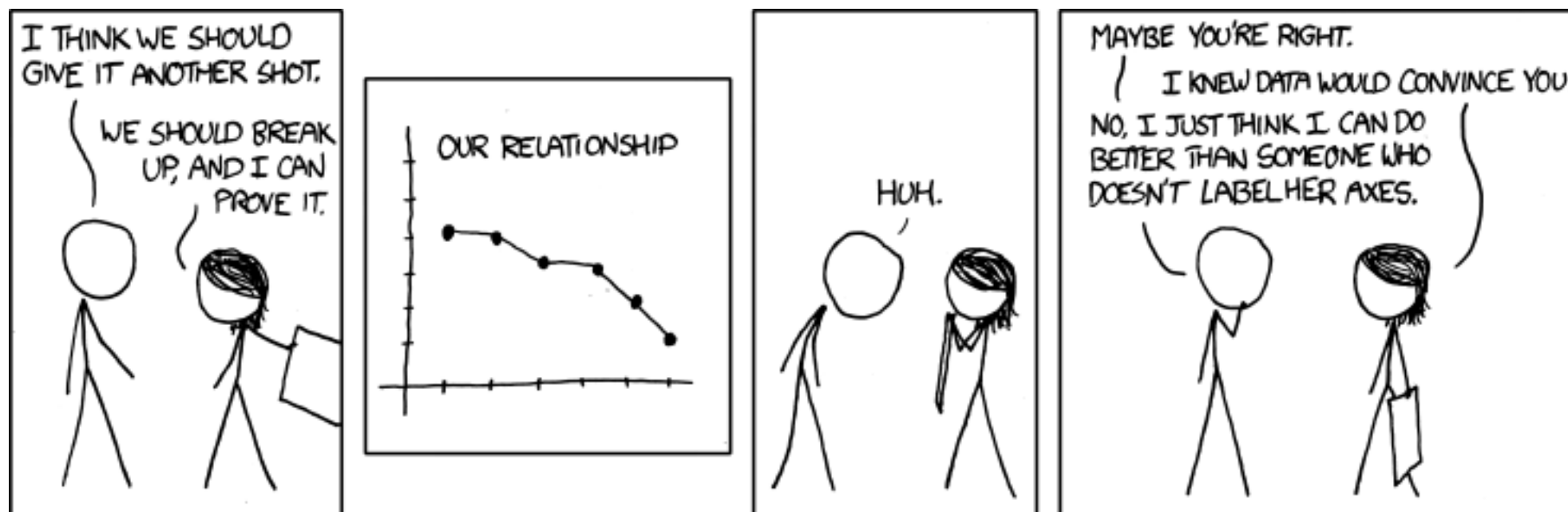
- proximity
  - do group related items together
  - avoid equal whitespace between unrelated
- alignment
  - do find/make strong line, stick to it
  - avoid automatic centering
- repetition
  - do unify by pushing existing consistencies
- contrast
  - if not identical, then very different
  - avoid not quite the same



- *The Non-Designer's Design Book, 4th ed. Robin Williams, Peachpit Press, 2015.*
  - fast read, very practical to work through whole thing

# Best practices: Labelling

- make visualizations as self-documenting as possible
  - meaningful & useful title, labels, legends
  - axes and panes/subwindows should have labels
    - and axes should have good mix/max boundary tick marks
  - everything that's plotted should have a legend
    - and own header/labels if not redundant with main title
  - use reasonable numerical format
    - avoid scientific notation in most cases



# Rules of Thumb Summary

- **No unjustified 3D**
  - Power of the plane
  - Disparity of depth
  - Occlusion hides information
  - Perspective distortion dangers
  - Tilted text isn't legible
- **No unjustified 2D**
- **Eyes beat memory**
- **Resolution over immersion**
- **Overview first, zoom and filter, details on demand**
- **Responsiveness is required**
- **Function first, form next**

# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



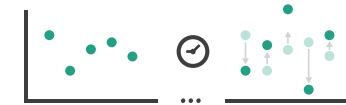
→ Motion

*Direction, Rate, Frequency, ...*

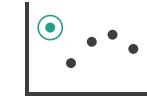


## Manipulate

### → Change



### → Select



### → Navigate



## Facet

### → Juxtapose



### → Partition



### → Superimpose

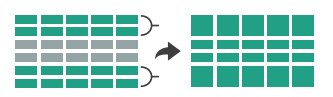


## Reduce

### → Filter



### → Aggregate



### → Embed



What?

Why?

How?

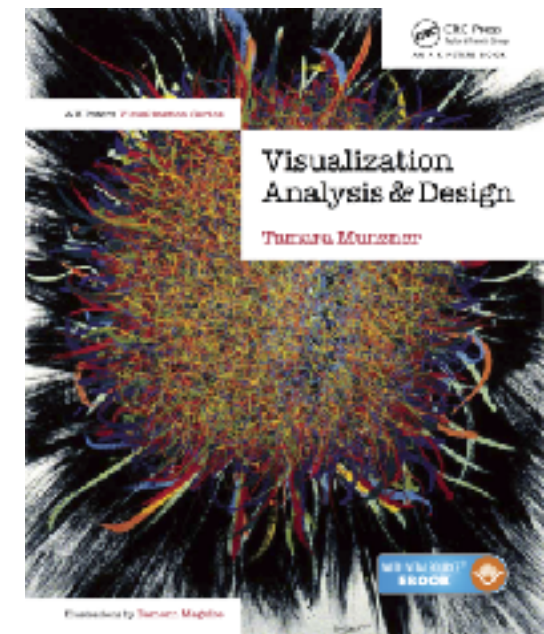
# Visualization Analysis & Design

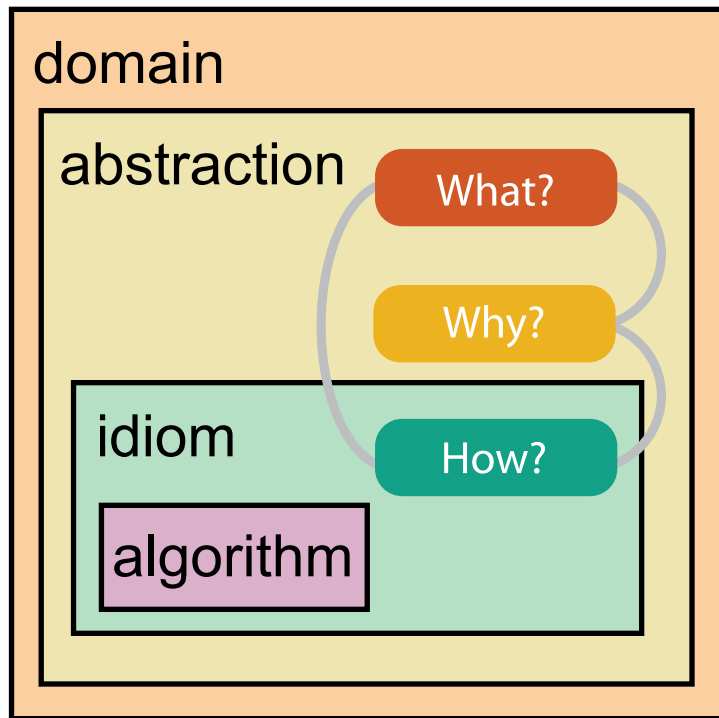
## *Wrapup*

**Tamara Munzner**

Department of Computer Science  
University of British Columbia

[@tamaramunzner](#)





# What?

## Datasets

### → Data Types

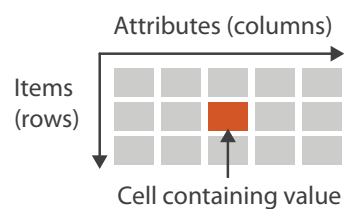
→ Items → Attributes → Links → Positions → Grids

### → Data and Dataset Types

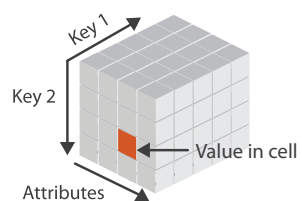
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

### → Dataset Types

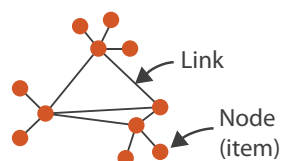
#### → Tables



#### → Multidimensional Table



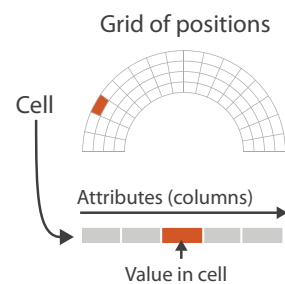
#### → Networks



#### → Trees



#### → Fields (Continuous)



#### → Geometry (Spatial)



## Attributes

### → Attribute Types

→ Categorical



→ Ordered

→ Ordinal



→ Quantitative



### → Ordering Direction

→ Sequential



→ Diverging



→ Cyclic



## domain

abstraction

What?

Why?

idiom

How?

algorithm



# What?

## Datasets

## Attributes

### Why?

#### Actions

#### Targets

#### → 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

#### → All Data

##### → Trends



##### → Outliers



##### → Features



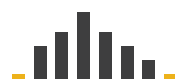
#### → Attributes

##### → One

###### → Distribution



###### → Extremes

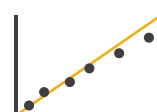


##### → Many

###### → Dependency



###### → Correlation

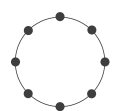
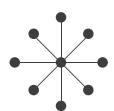


###### → Similarity



#### → Network Data

##### → Topology



##### → Paths



## domain

### abstraction

What?

Why?

How?

### idiom

### algorithm

# What?

Datasets

Attributes

## Why?

Actions

Targets

## How?

Encode

Manipulate

Facet

Reduce

Arrange

Express

Separate

Map

from **categorical** and **ordered** attributes

Color

Hue

Saturation

Luminance

Size, Angle, Curvature, ...

Shape

Motion

Direction, Rate, Frequency, ...

Change

Select

Navigate

Juxtapose

Partition

Superimpose

Filter

Aggregate

Embed

What?

Why?

How?

domain

abstraction

What?

Why?

idiom

How?

algorithm

# More information

- book

- <http://www.cs.ubc.ca/~tmm/vadbook>

- 20% promo code for book+ebook combo: HVN17

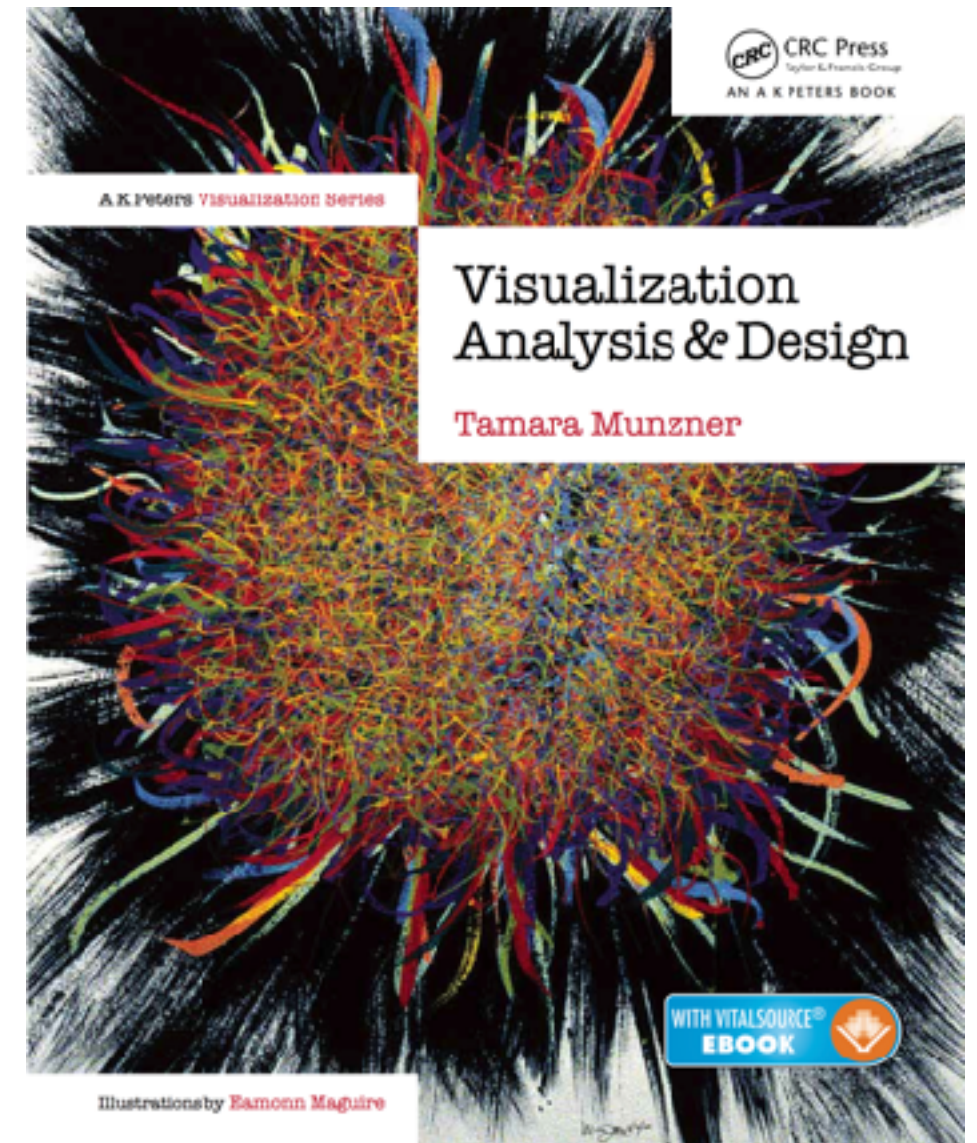
- <http://www.crcpress.com/product/isbn/9781466508910>

- illustration acknowledgement: Eamonn Maguire

- full courses, papers, videos, software, talks

- <http://www.cs.ubc.ca/group/infovis>

- <http://www.cs.ubc.ca/~tmm>



Visualization Analysis and Design. Munzner. CRC Press, AK Peters Visualization Series, 2014.