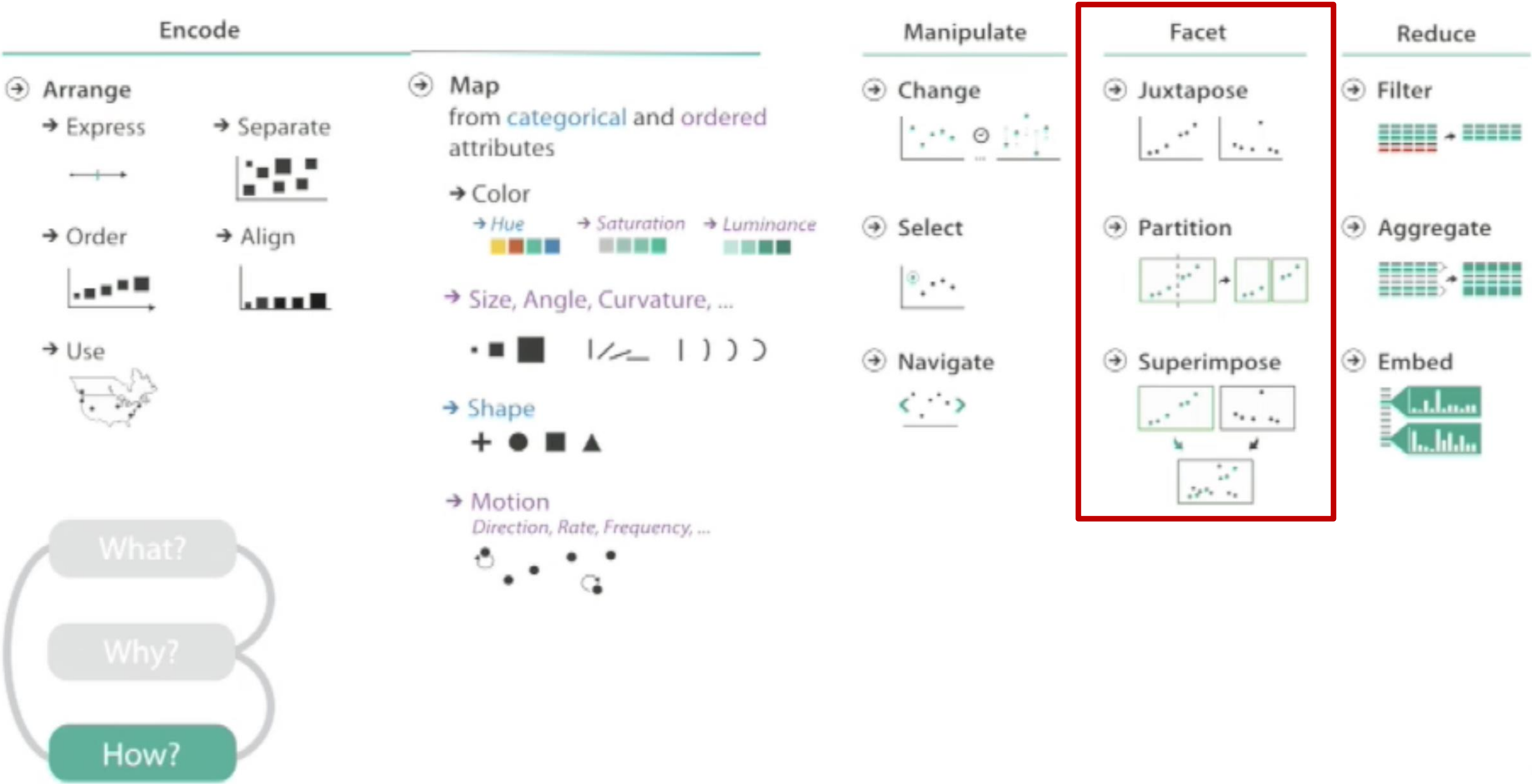
Handle Complexity:
Facet (Juxtapose, Partition, Superimpose)

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?

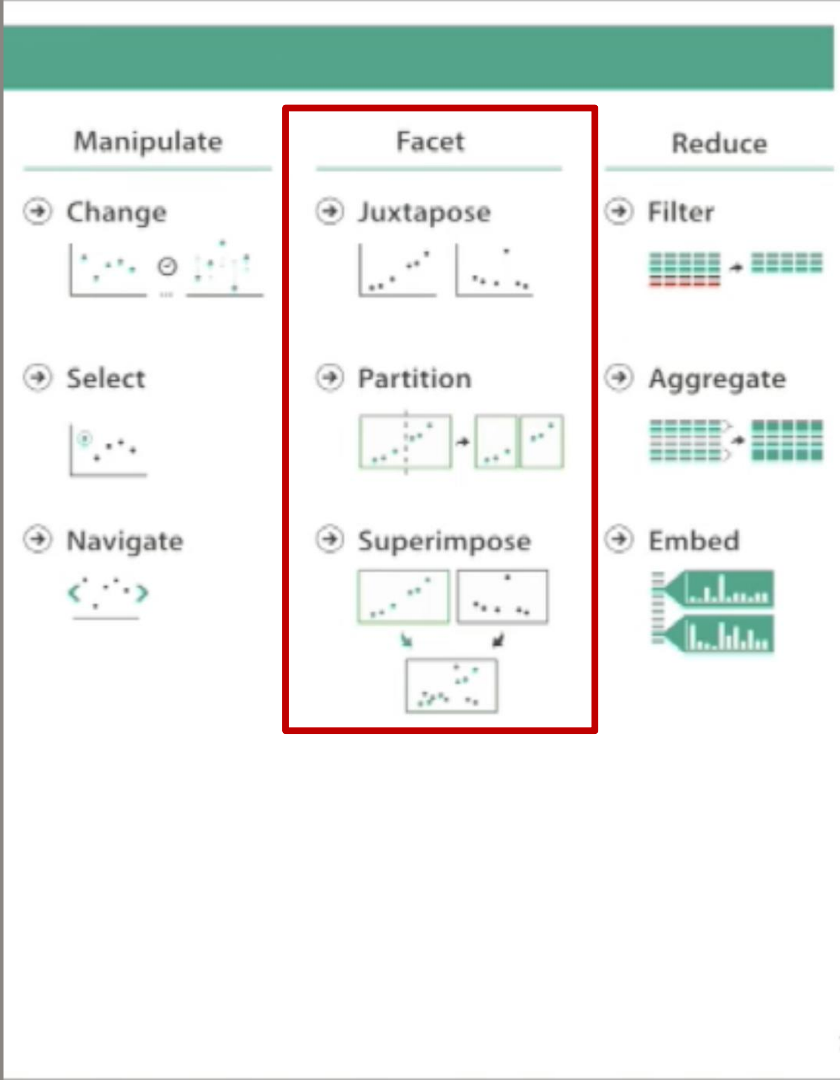
Why?

How?

Handle Complexity

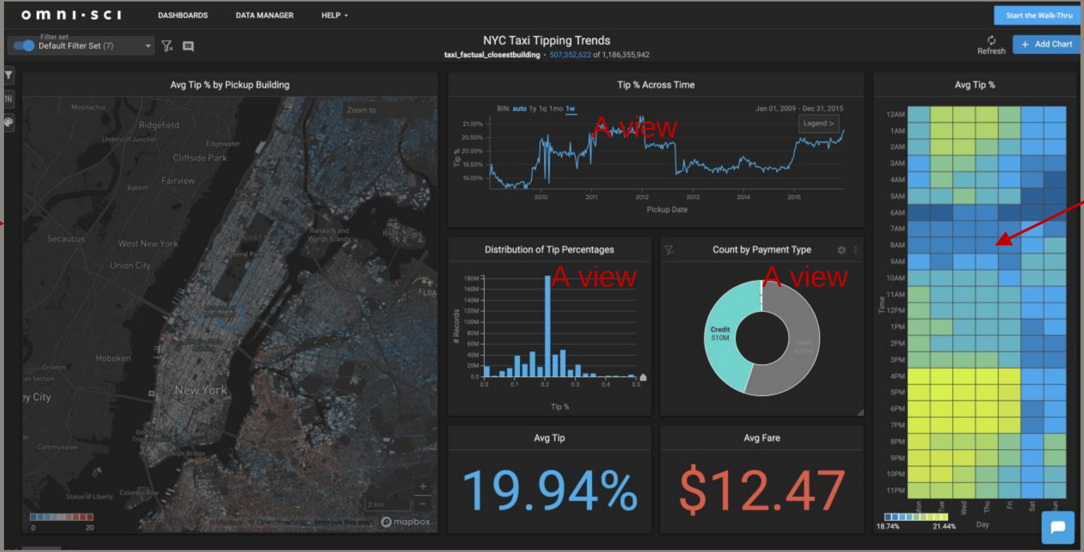
- If what we have before does not work
  - If the data or tasks are too complicated, do not insist on **one static view** to solve all problems

- Change view (what you see) over time
- **Facet across multiple view**
- Reduce item/attribute within single view



Facet

- One example: Juxtapose
  - Get different insight from different views



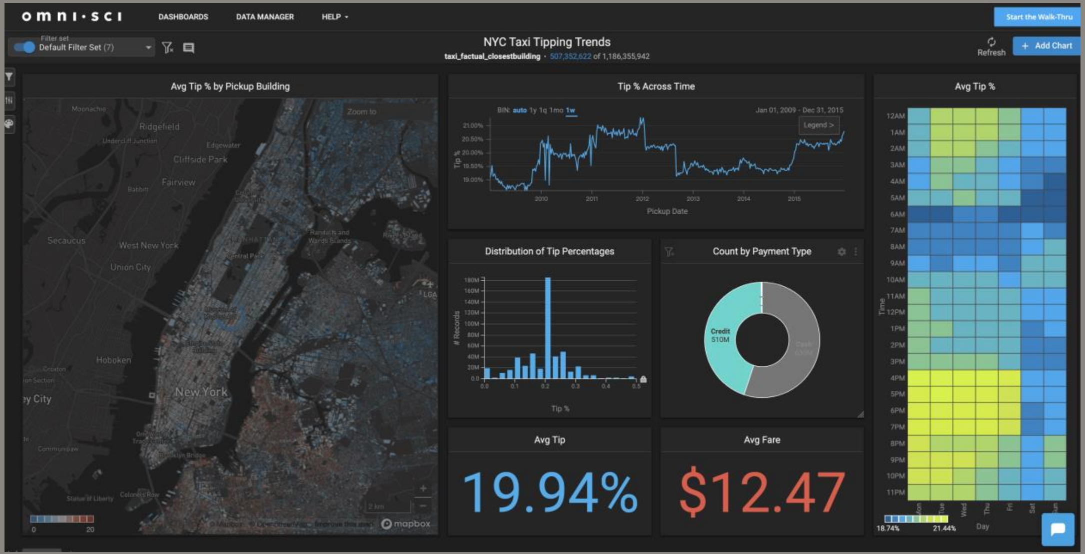
A view

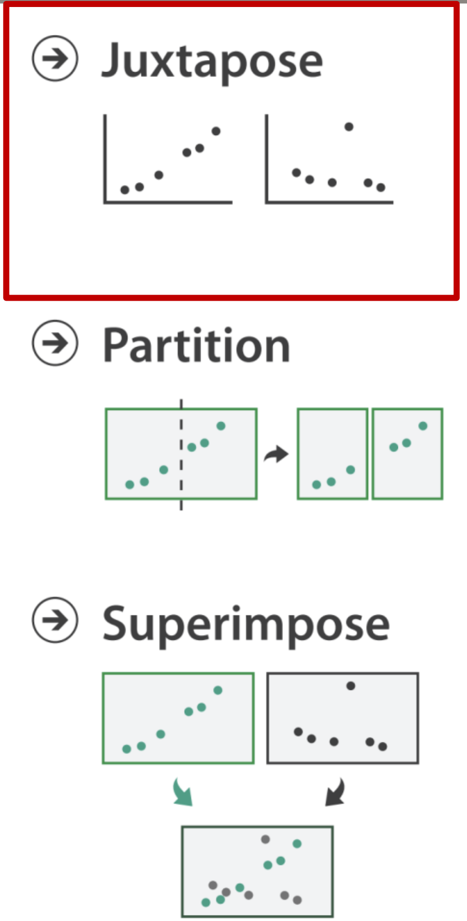
A view

https://www.omnisci.com/demos/taxis

Facet

- Juxtapose, Partition, Superimpose
  - Different ways to layout multiple views





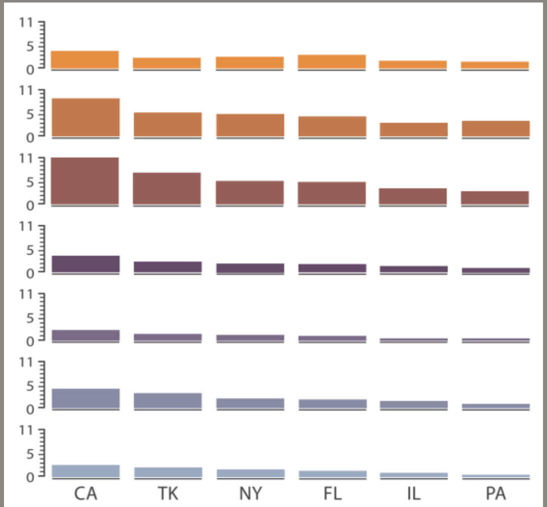
Juxtapose

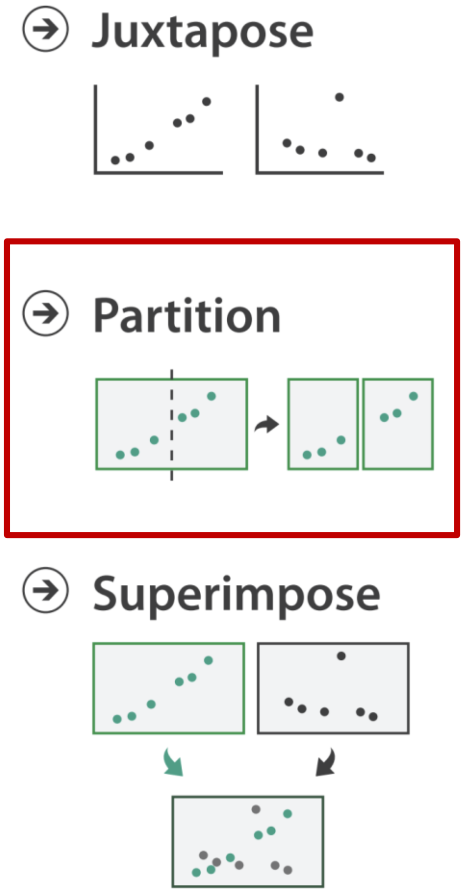
Partition

Superimpose

Facet

- Juxtapose, Partition, Superimpose
  - Different ways to layout multiple views





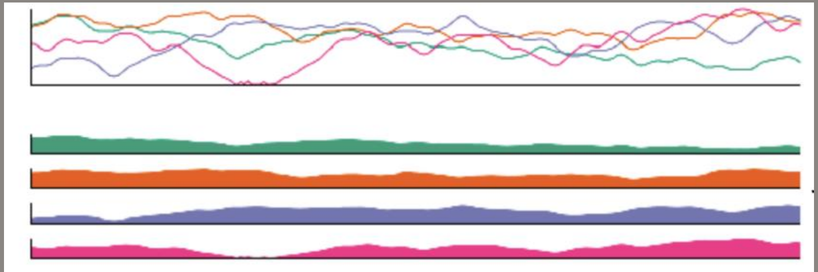
Juxtapose

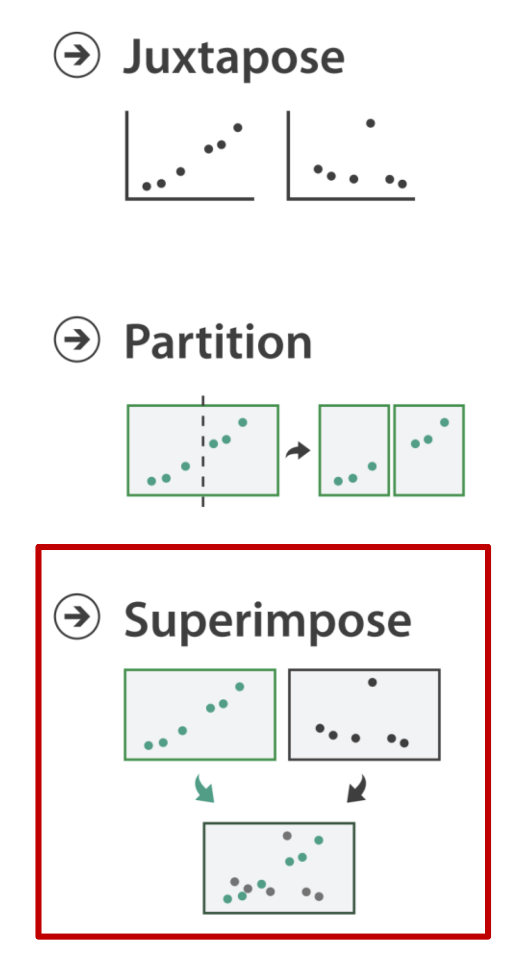
Partition

Superimpose

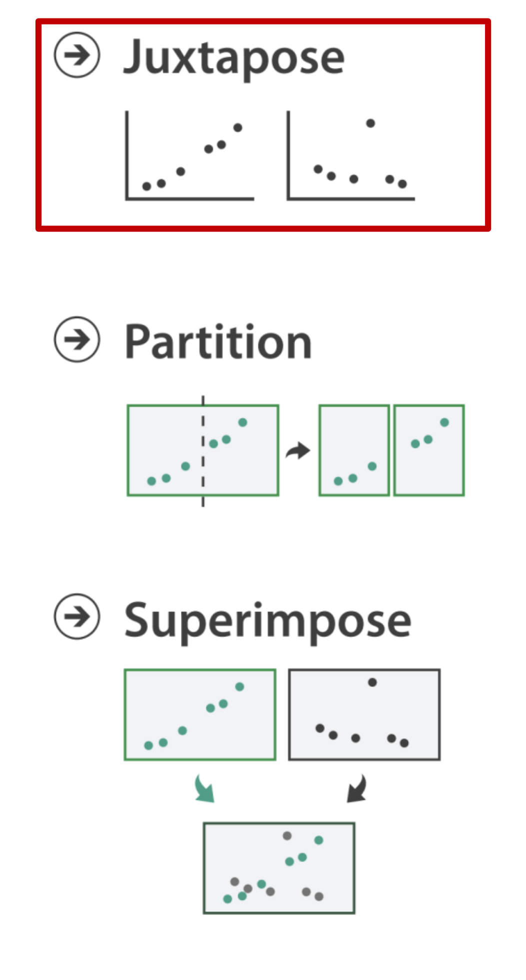
Facet

- Juxtapose, Partition, Superimpose
  - Different ways to layout multiple views



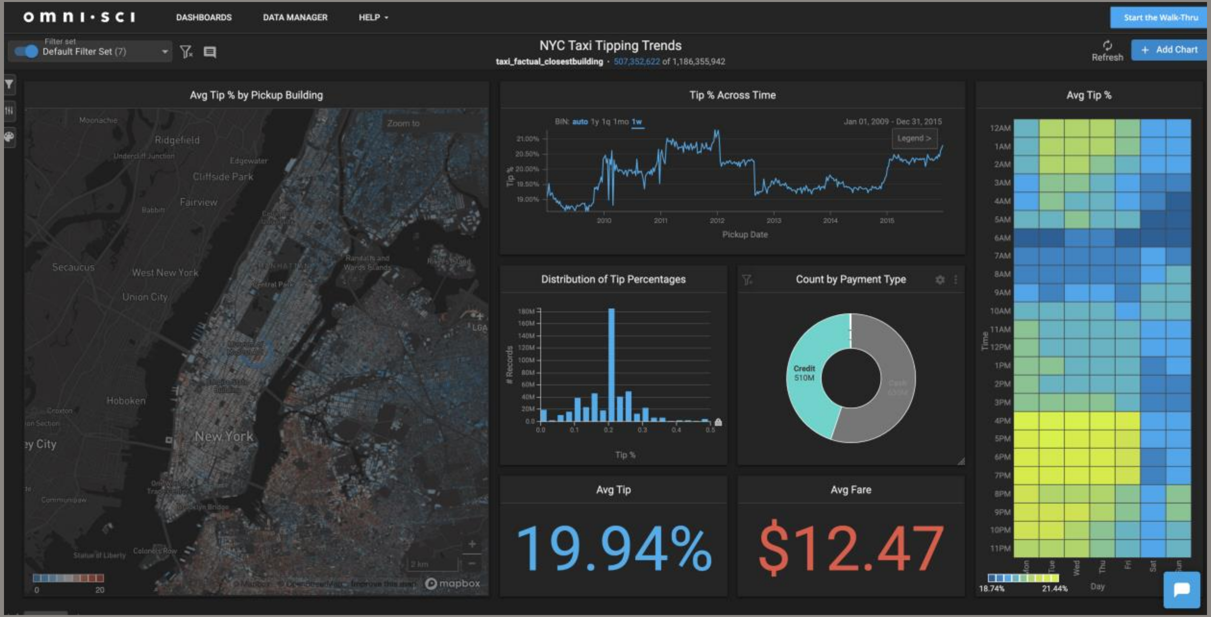


Juxtapos



Juxtapose

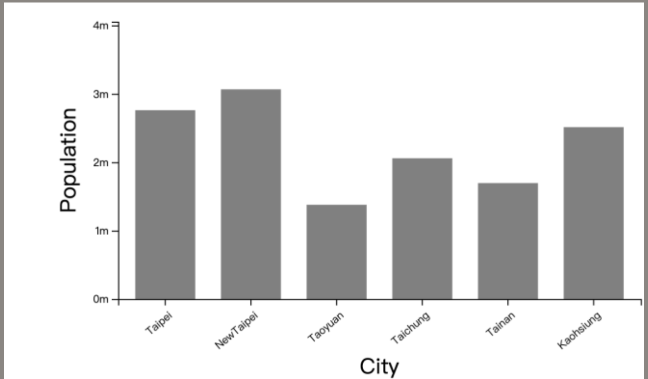
- Show multiple views on the screen at the same time



Why Juxtapose View?

- Benefits: eye vs. memory
  - Lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - Easy to compare
  - Usually, eye beats memory
- Cost?
  - Display area, multiple views spend more display area

animation

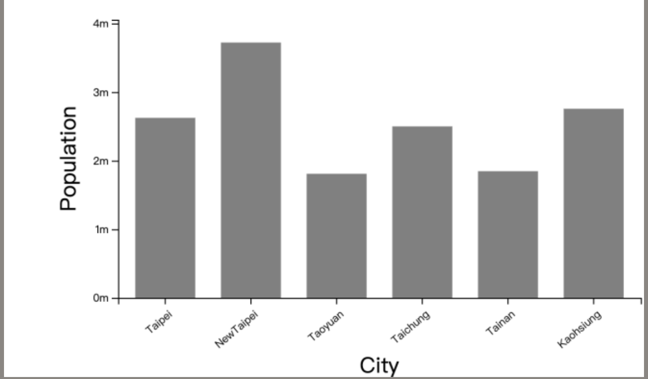


1990

Why Juxtapose View?

- Benefits: eye vs. memory
  - Lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - Easy to compare
  - Usually, eye beats memory
- Cost?
  - Display area, multiple views spend more display area

animation

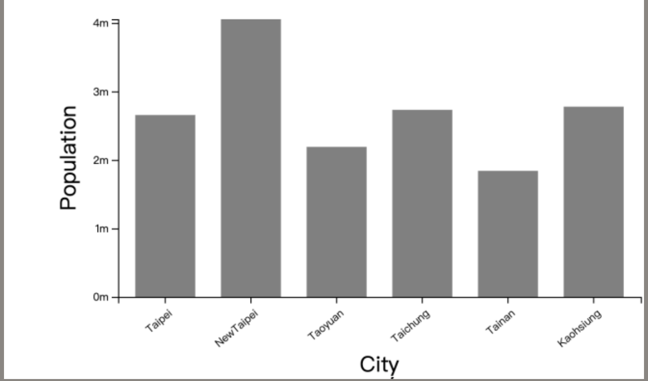


2000

Why Juxtapose View?

- Benefits: eye vs. memory
  - Lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - Easy to compare
  - Usually, eye beats memory
- Cost?
  - Display area, multiple views spend more display area

animation

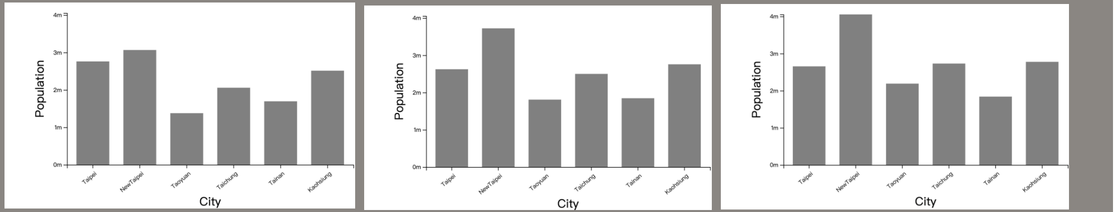


2010

Why Juxtapose View?

- Benefits: eye vs. memory
  - Lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - Easy to compare
  - Usually, eye beats memory
- Cost?
  - Display area, multiple views spend more display area

Juxtapose

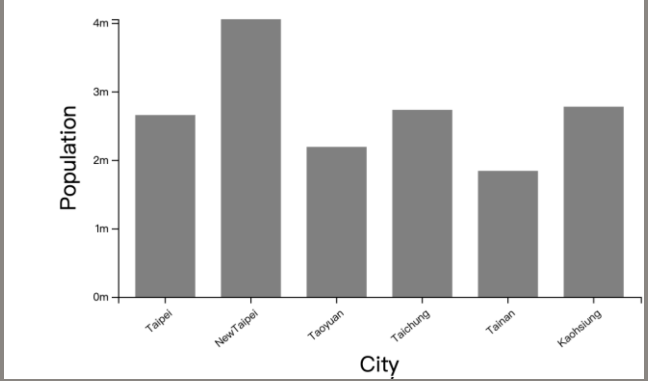


1990

2000

2010

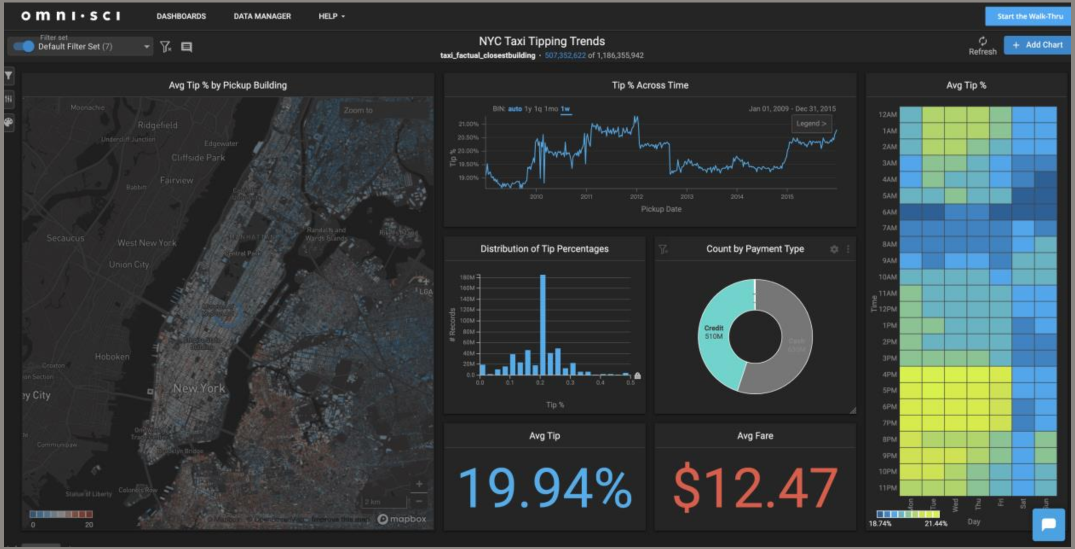
animation



2010

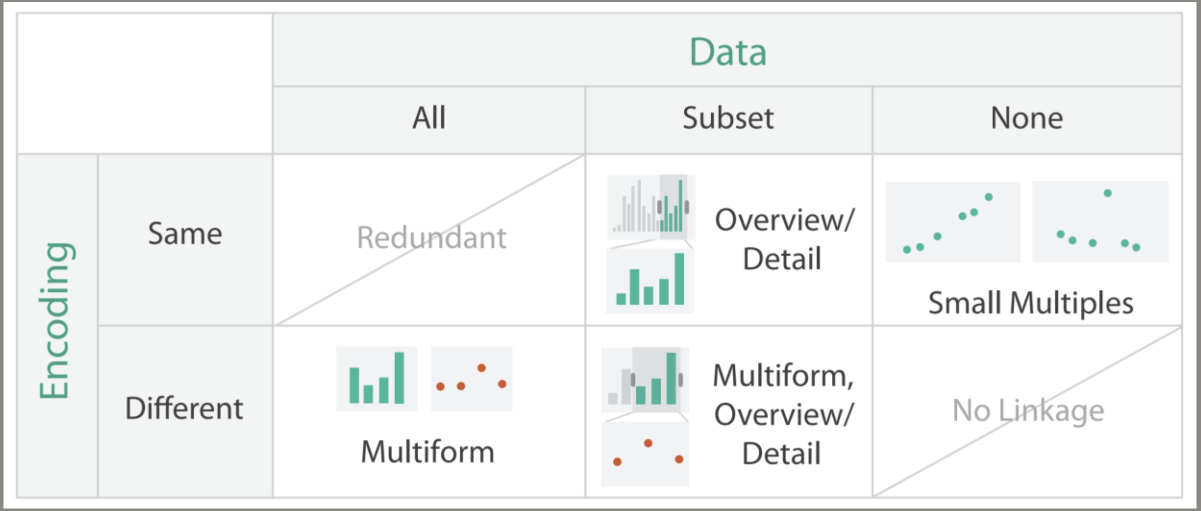
Juxtapose

- Without linking/coordinating views, it would be boring



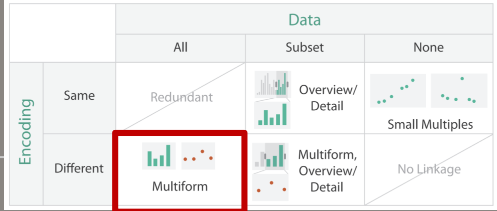
Juxtapose

- We can have different design choice (between views) of juxtapose
  - Data: all/subset/none
  - Encoding: same/different



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

We are going to introduce different idioms of juxtapose

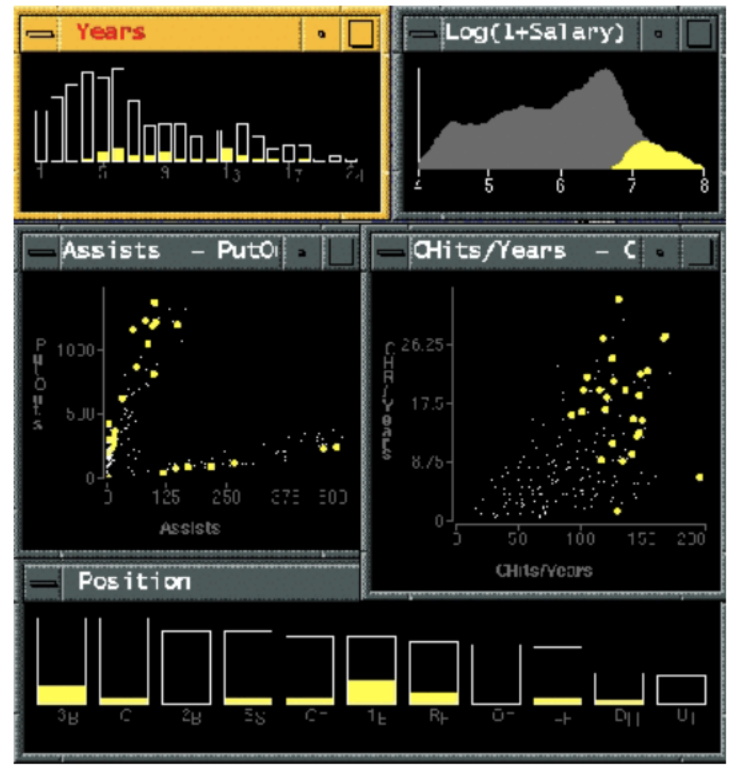


		Data		
		All	Subset	None
Encoding	Same	Redundant	Overview/ Detail	Small Multiples
	Different	Multiform	Multiform, Overview/ Detail	No Linkage

Idiom: Link Highlighting

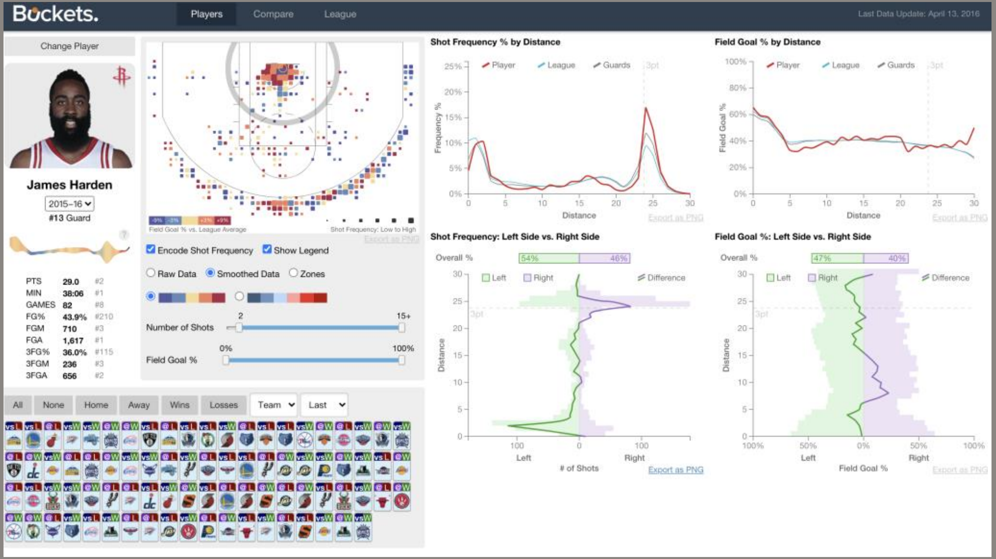
- See how regions contiguous in one view are distributed within another
  - Powerful and pervasive interaction idiom

- Encoding: different
  - Multiform
- Data:
  - all items shared
  - Different attribute across views
- brushing and linking



Idiom: Link Highlighting

- Multidirectional linking (more useful than unidirectional linking)

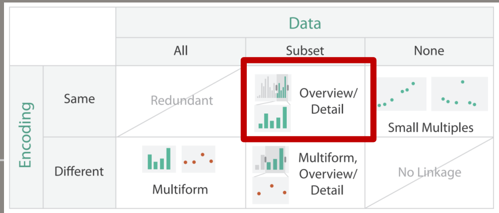


PTS	29.0	#2
MIN	38:06	#1
GAMES	82	#8
FG%	43.9%	#210
FGM	710	#3
FGA	1,617	#1
3FG%	36.0%	#115
3FGM	236	#3
3FGA	656	#2

☑ Encode Shot Frequency ☑ Show Legend

○ Raw Data ● Smoothed Data ○ Zones

https://buckets.peterbeshai.com/app/#/playerView/201935_2015

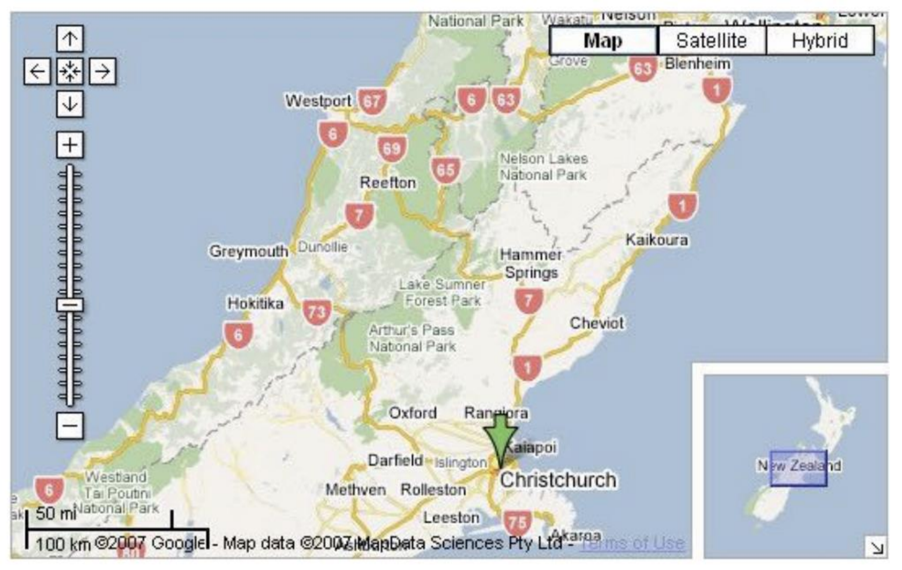


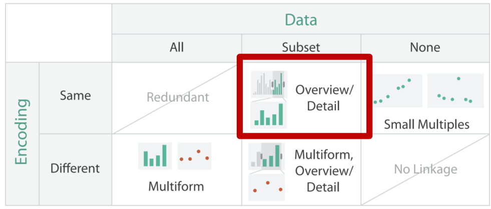
		Data		
		All	Subset	None
Encoding	Same	Redundant	Overview/ Detail	Small Multiples
	Different	Multiform	Multiform, Overview/ Detail	No Linkage

Idiom: Overview-detail views

- Encoding: same
- Data: subset shared
- Navigation: bi-directional

- Differences
  - Viewpoint, size

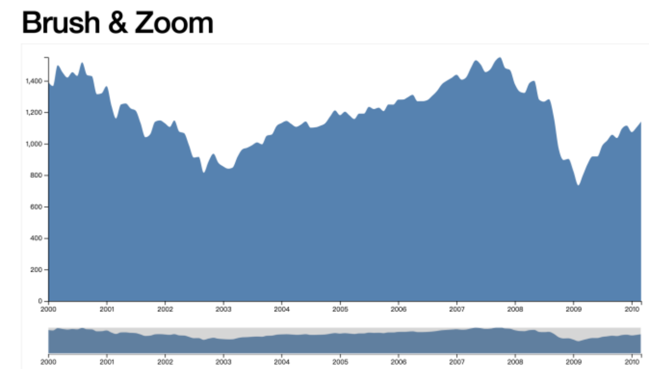




		Data		
		All	Subset	None
Encoding	Same	Redundant	Overview/ Detail	Small Multiples
	Different	Multiform	Multiform, Overview/ Detail	No Linkage

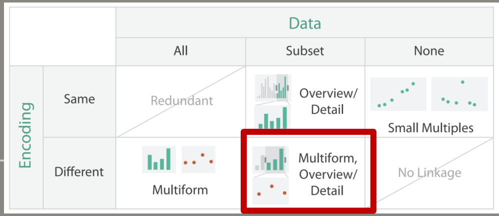
Idiom overview-detail navigation

- Encoding: same
- Data: subset shared
- Navigation: shared
  - Unidirectional linking
  - Select in small overview
  - Change extent in large detail view

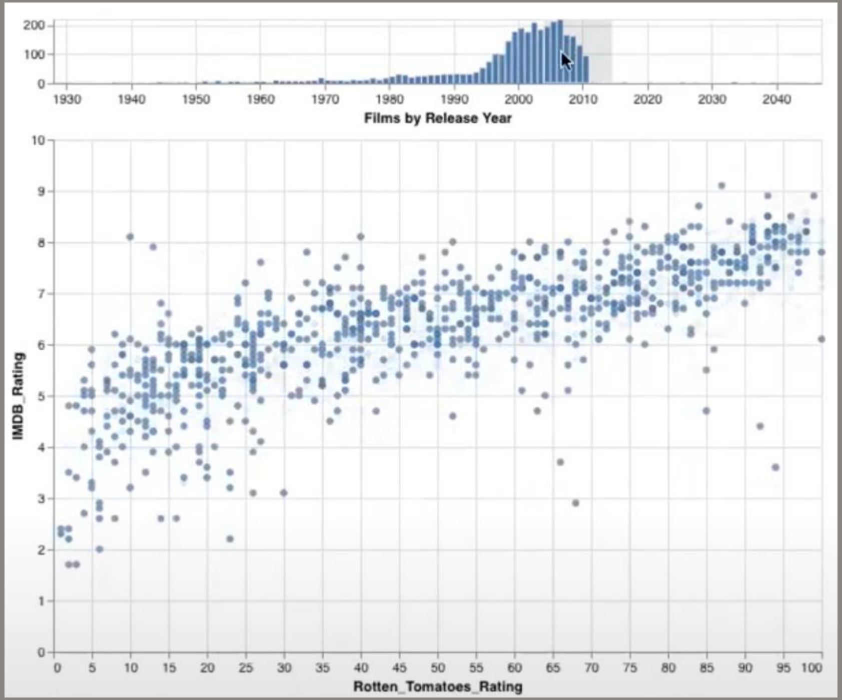

Brush & Zoom

https://observablehq.com/@d3/focus-context

Idiom: overview-detail navigation

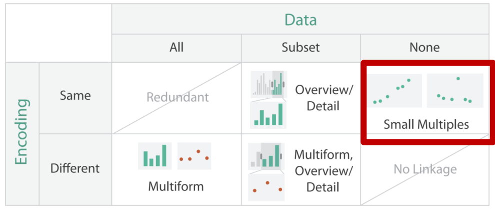


		Data		
		All	Subset	None
Encoding	Same	Redundant	Overview/ Detail	Small Multiples
	Different	Multiform	Multiform, Overview/ Detail	No Linkage



https://observablehq.com/@uwdata/interaction?collection=@uwdata/visualization-curriculum

Widget for selection and analysis

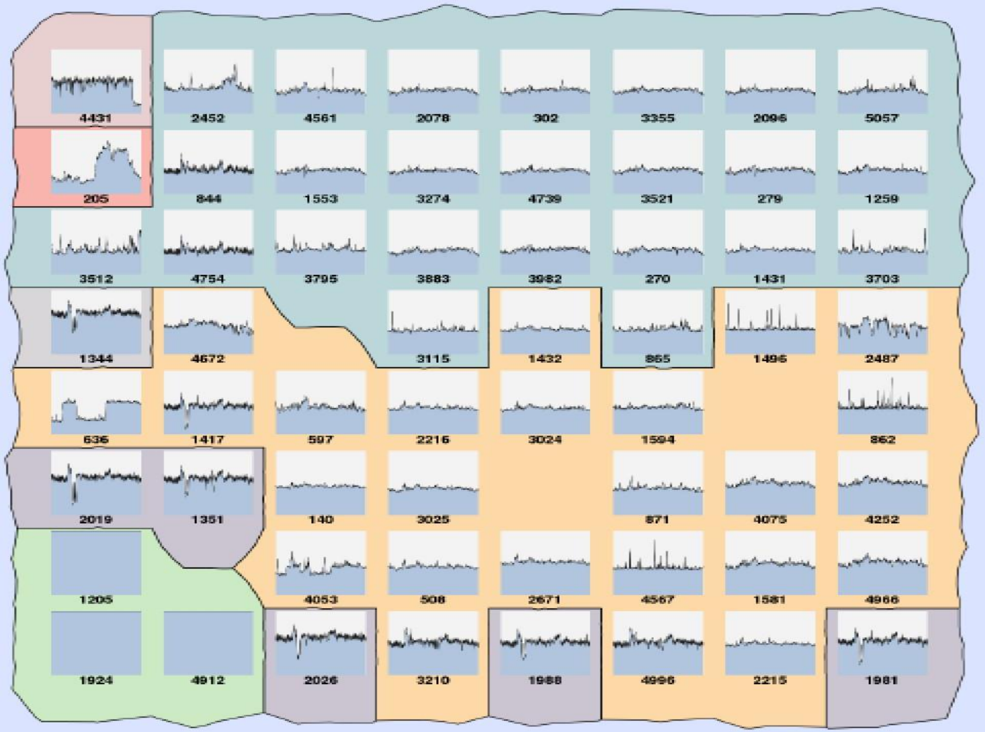


		Data		
		All	Subset	None
Encoding	Same	Redundant	Overview/ Detail	Small Multiples
	Different	Multiform	Multiform, Overview/ Detail	No Linkage

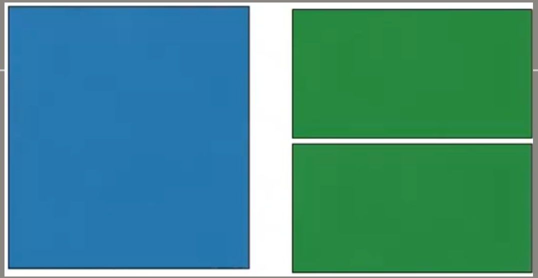
Idiom: Small Multiples

- Encoding: same
- Data: none shared (different data partition)
  - Different stock price over time

- Make different partition of data simultaneous visible
- Often aligned into a list or matrix
- Often use as an alternative of animation
- Small screen real estate is a weakness



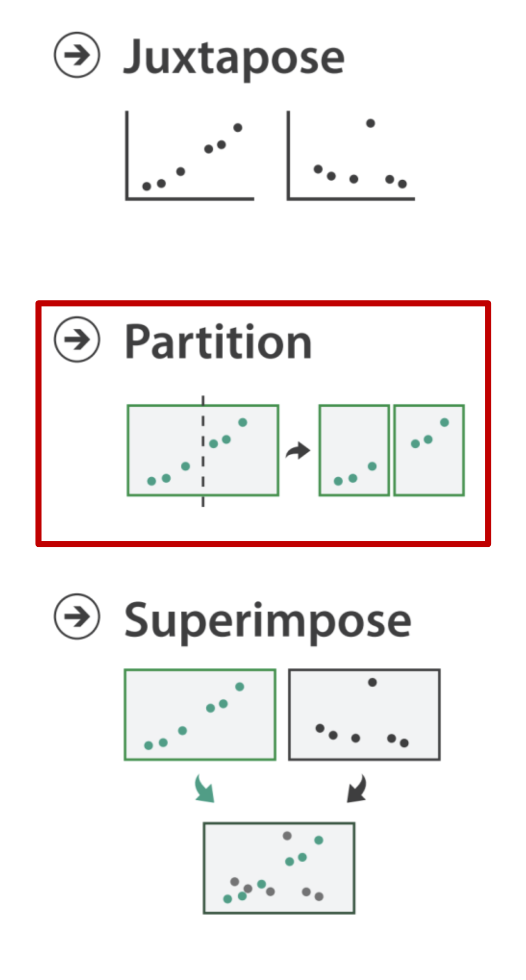
Juxtapose View: Tradeoffs



- Juxtapose costs
  - Display area
  - 2 views side by side: each has only half of the area of one view
- Juxtapose benefits
  - Cognitive load: eye (juxtapose) vs memory (animation)
    - Lower cognitive load: move eye between 2 views
    - Higher cognitive load: compare single changing view to memory of previous state

S09-01

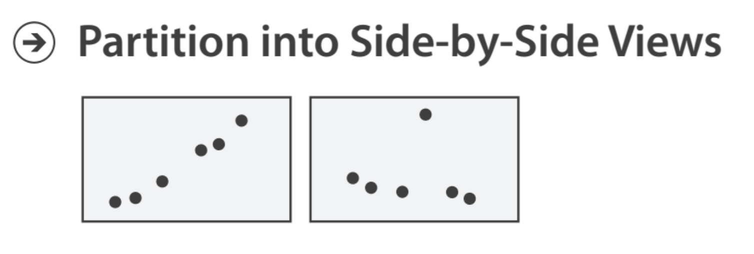
Partition



Partition (data) into Views

- How to divide data between views
  - Split into regions by attributes
    - Only one attribute? You may not need multiple views
  - Encodes association between items using spatial proximity
    - Close <-> easy to compare
  - **Order** of splits has major implications for what patterns are visible
    - Hierarchically partition

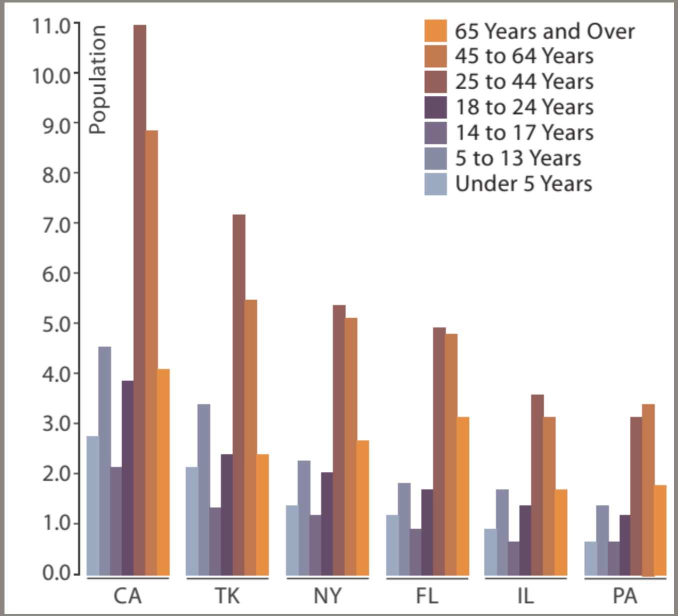
- Even if you visualize the same dataset, different ways to partition may give you different insights



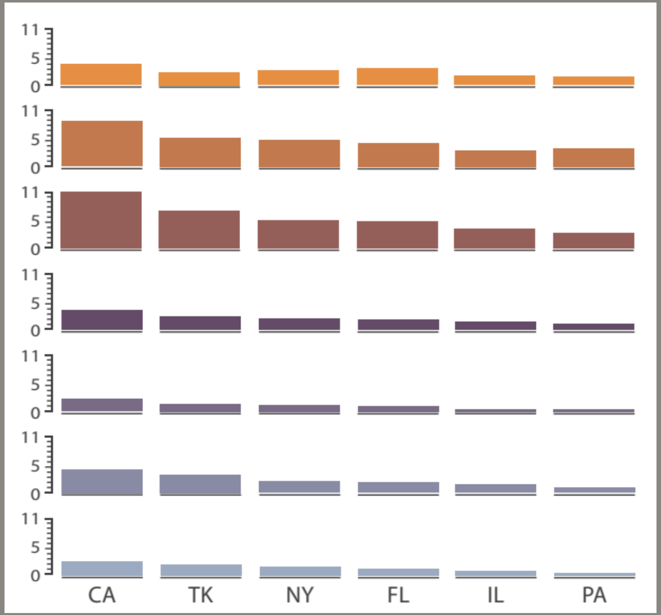
Partition into Side-by-Side Views

Partition: List Alignment

- Split by state into regions
  - Easy within state, hard across ages



- Split by age into regions
  - Easy within age, harder within states



S09-02

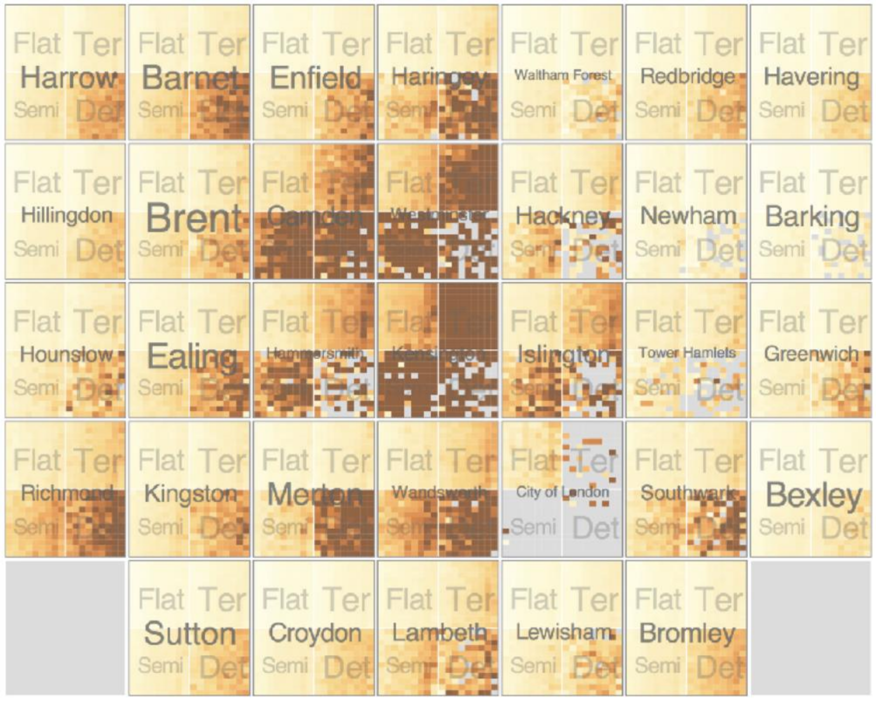
Partitioning: Recursive Subdivision

- Split by neighborhood,
- then by type,
- then by time (row: years, column: months)

- Color: price

- Easy to know
  - Where it is expensive
  - Compared to the other 3 house types, where you pay much more for detached type

Terraced houses
Semi-detached houses
Block of flats
Detached house



Each big rectangle is a region in London

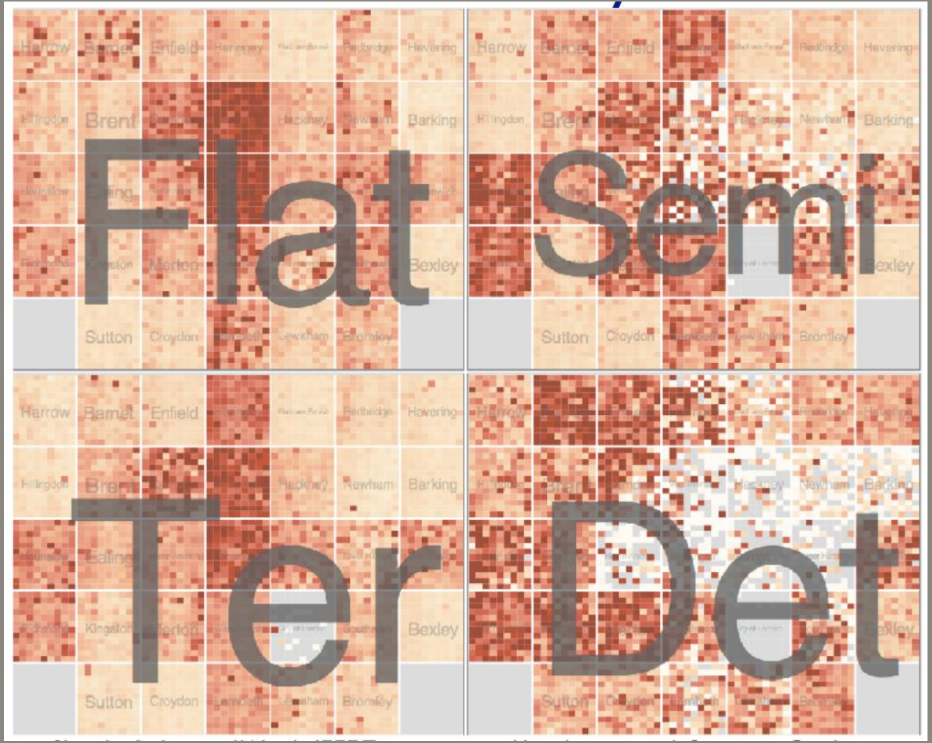
In UK, they have four type of hours (flat, attached terrance semidetached, detached)

Partitioning: Recursive Subdivision

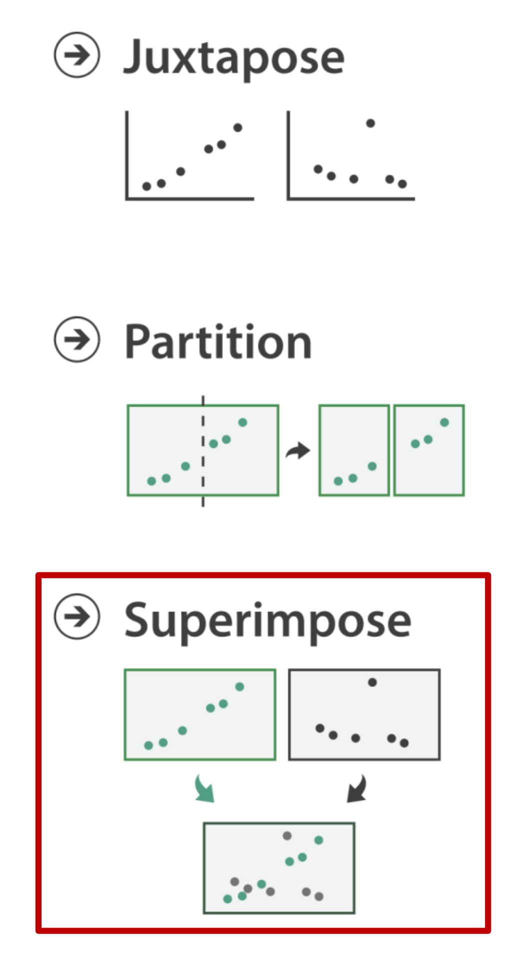
- Split by house types
- then neighborhood
- then by time (row: years, column: months)

- Color: price

- Easy to know
  - Within specific type, which neighborhoods is more expensive

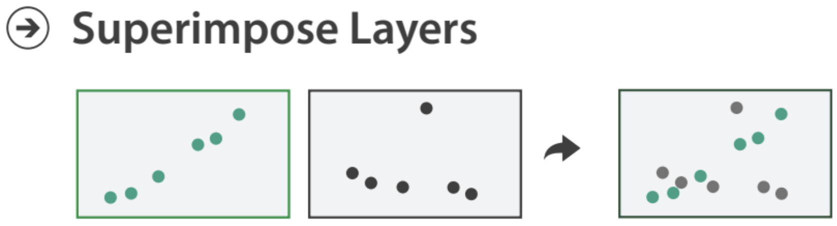


Superimpose



Superimpose Layers

- Layer: set of objects spread out over region
  - Each set is visually distinguishable group

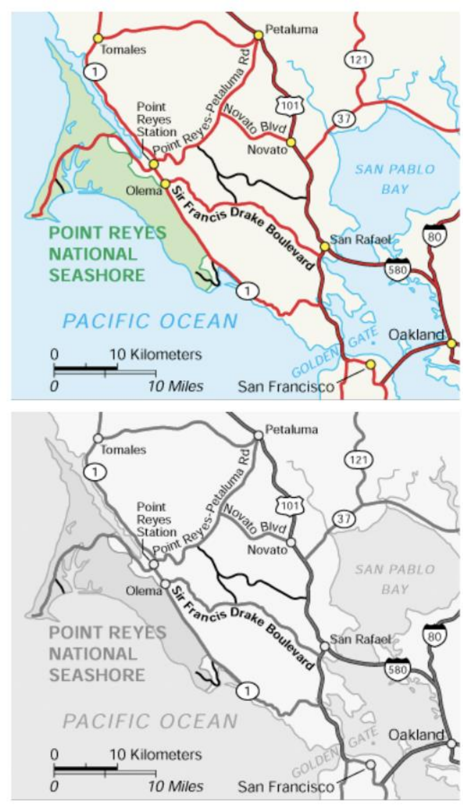


- Design choices
  - How many layers, how to distinguish?
    - Encode with different, nonoverlapping channels
    - Two layers achievable, three with careful design
  - Small static set, or dynamic from many possible?

Static Visual Layering

- Foreground layer: road
- Background layer: regions
  - Hue or saturation to separate

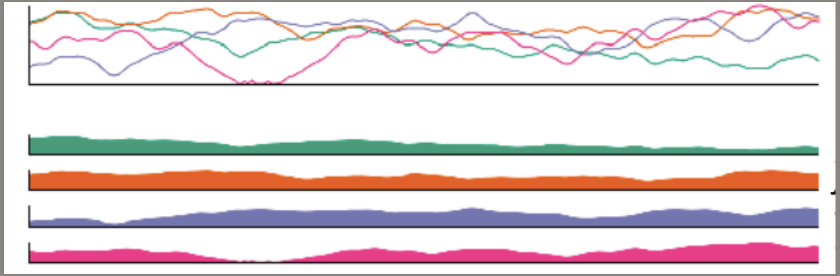
- User can selectively focus attention

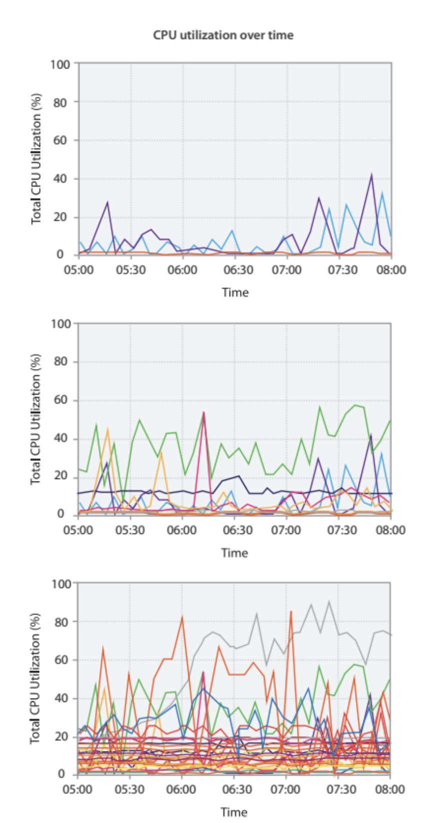


Superimposing Limits

- Few layers, but many lines
  - Up to a few dozen
  - But not hundreds

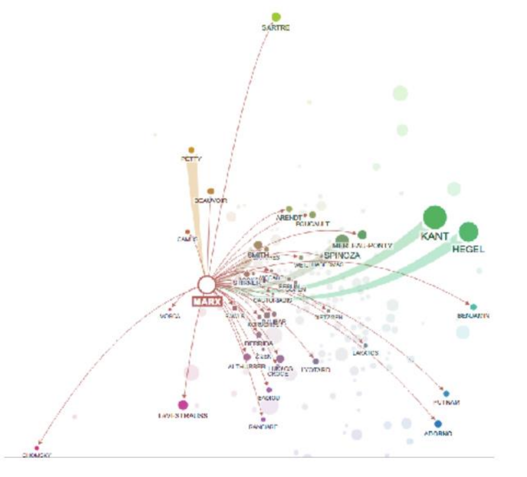
- Superimpose vs juxtapose: empirical study
  - Superimposed for local, multiple for global
  - Tasks:
    - Local: maximum
    - Global: slope, discrimination



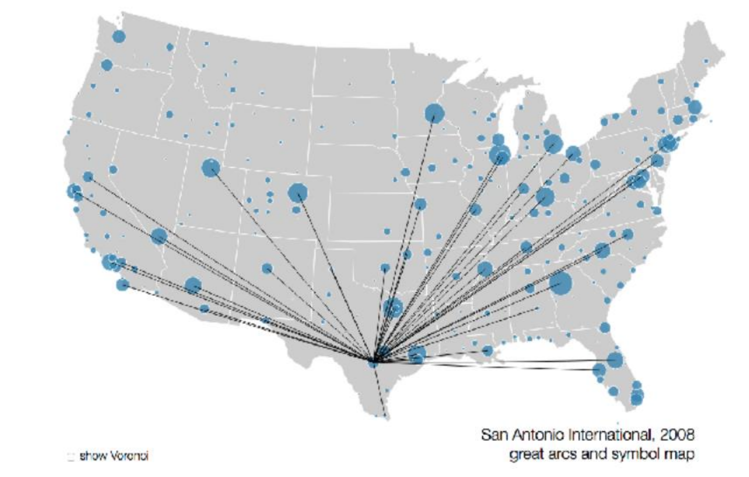


Dynamic Visual Layers

- Interactive based on selection
- One-hop neighbor highlighting demo

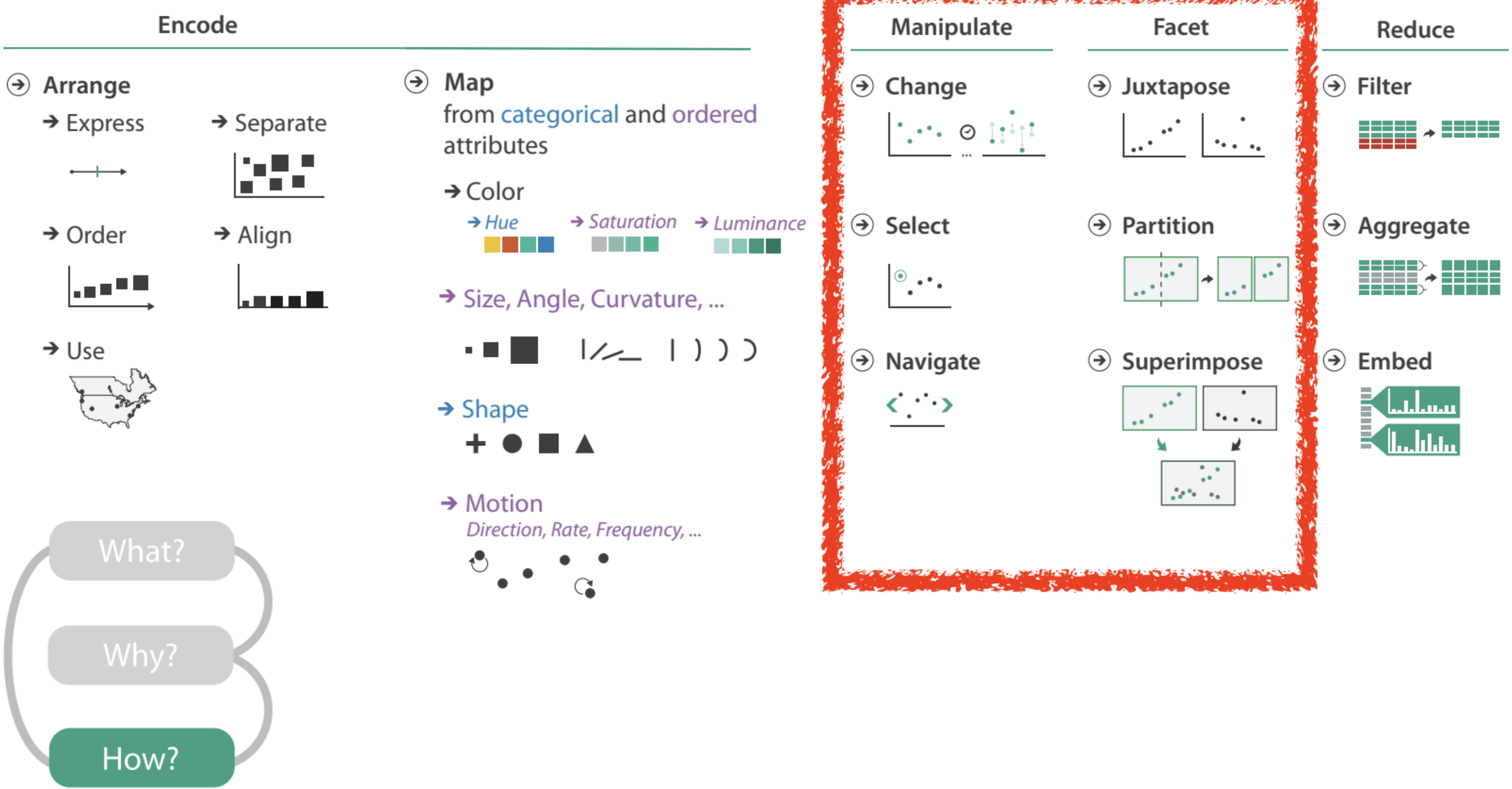


https://mariandoerk.de/edgemaps/demo/#phils;map;;/en/immanuel_kant;



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?