



CS-GY 6313 B: Information Visualization

10/24/2024

Logistics

- Assignment 3 due next week
- Final project group + topic due today
 - NYU email to log in
 - https://docs.google.com/spreadsheets/d/1mJ_CGBU1KNzqhw1M9mWNgF0ffl2GqPBT0HaCDUHUPKM/edit?usp=sharing



Interaction & Animation

Managing Complexity in Viz

- Large and complex datasets can lead to cluttered, unhelpful visualizations
- It is often not possible, and usually not desirable, to show *all* the data
- How do we decide what to show and how to show it?
 - *Derive* more meaningful attributes
 - Enable the user to change – *manipulate* – the view
 - *Facet* the data into multiple views and/or layers
 - *Reduce* the number of items and/or attributes to be shown

Interaction benefits and limitations

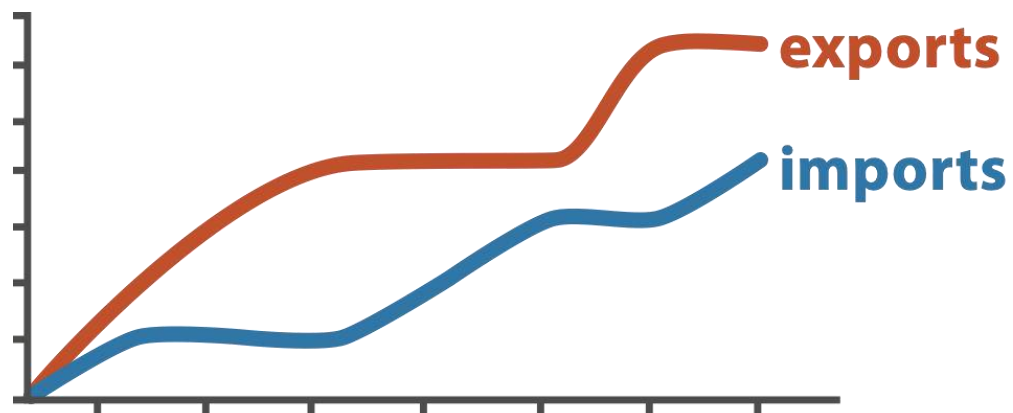
- Pros:
 - Major advantage of computer-based vs paper-based visualization
 - Flexible, powerful, intuitive
 - Exploratory data analysis: change as you go during analysis process
 - Fluid task switching: different visual encodings support different tasks
 - Animated transitions provide excellent support
 - Empirical evidence that animated transitions help people stay oriented
- Cons:
 - Has a time cost
 - Sometimes small, 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



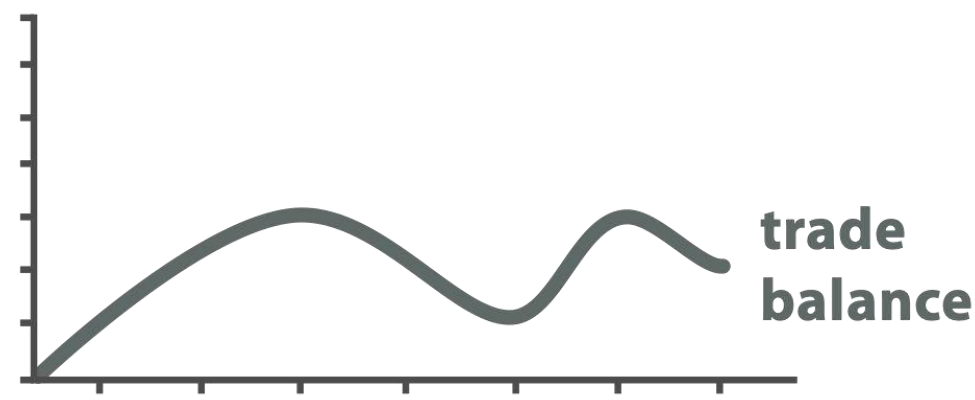
Derive

Derive new data

- Compute (derive) new data from existing data
 - Derived data should be more useful than existing data!
- Combines multiple attributes into one
 - E.g. add, subtract, average, minimum, maximum



Original Data

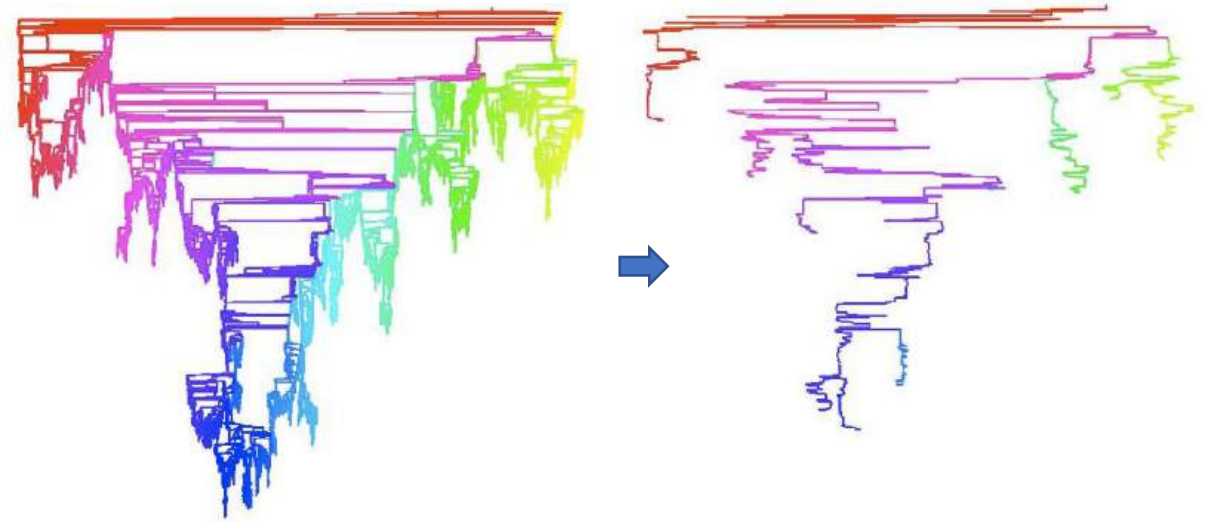


$$\text{trade balance} = \text{exports} - \text{imports}$$

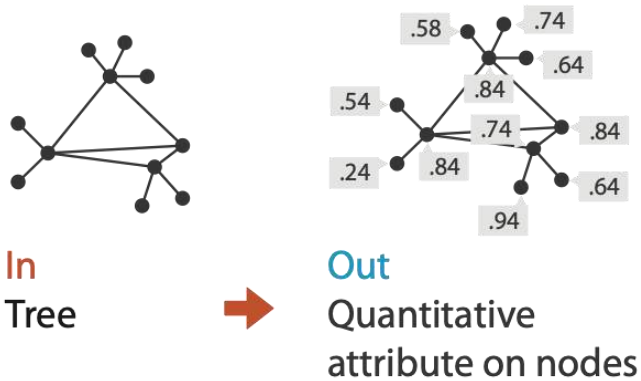
Derived Data

Derive new data

- Strahler measure
 - Centrality metric for networks
 - Derived, quantitative metric



Task 1



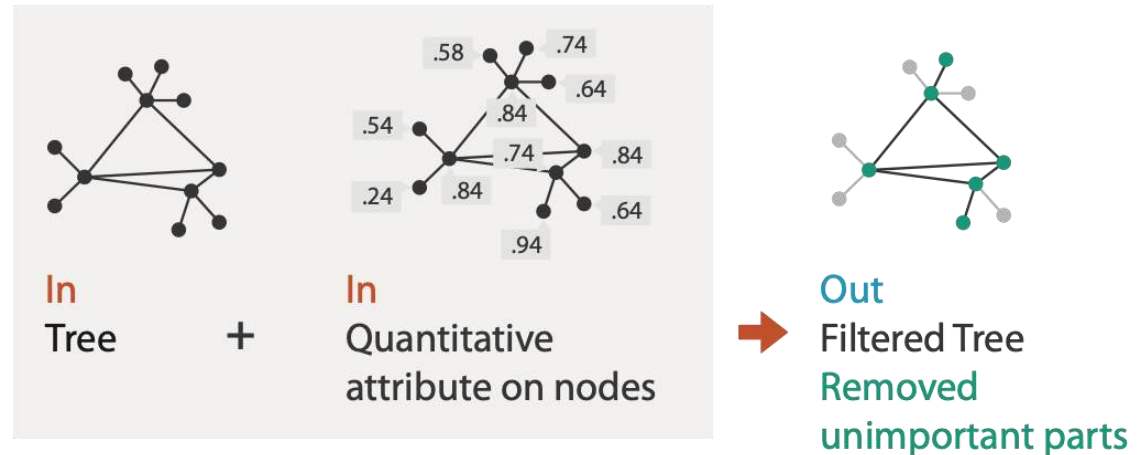
What?

- In Tree
- Out Quantitative attribute on nodes

Why?

- Derive

Task 2



What?

- In Tree
- In Quantitative attribute on nodes
- Out Filtered Tree

Why?

- Summarize
- Topology

How?

- Reduce
- Filter



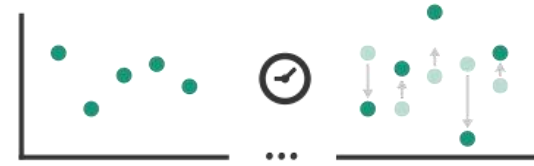
Manipulate

Manipulate the view

- Multiple options for how to manipulate:
 - Change the viz itself
 - Select
 - Navigate

Manipulate

➔ Change



➔ Select



➔ Navigate

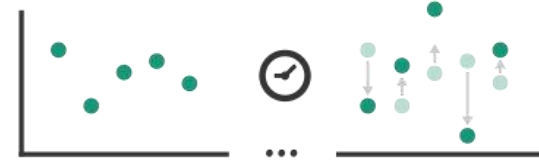


Manipulate the view

- Multiple options for how to manipulate:
 - Change the viz itself
 - Select
 - Navigate

Manipulate

➔ Change



➔ Select



➔ Navigate

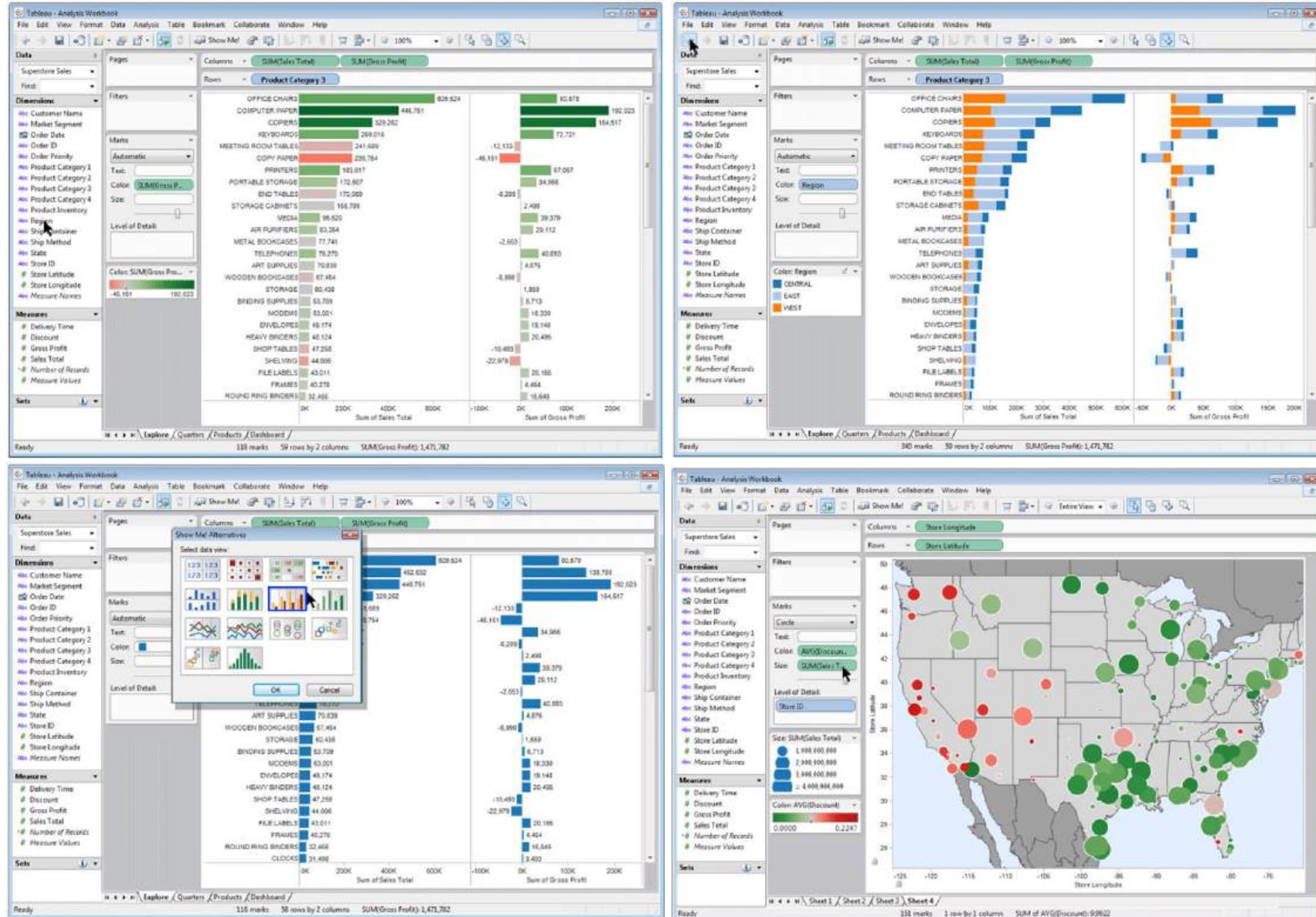


Change the viz itself

- Change any of the following:
 - Encoding itself
 - Viz parameters
 - Arrangement (rearrange, reorder)
 - Aggregation level, what is filtered...
- Interaction produces change
- Very flexible

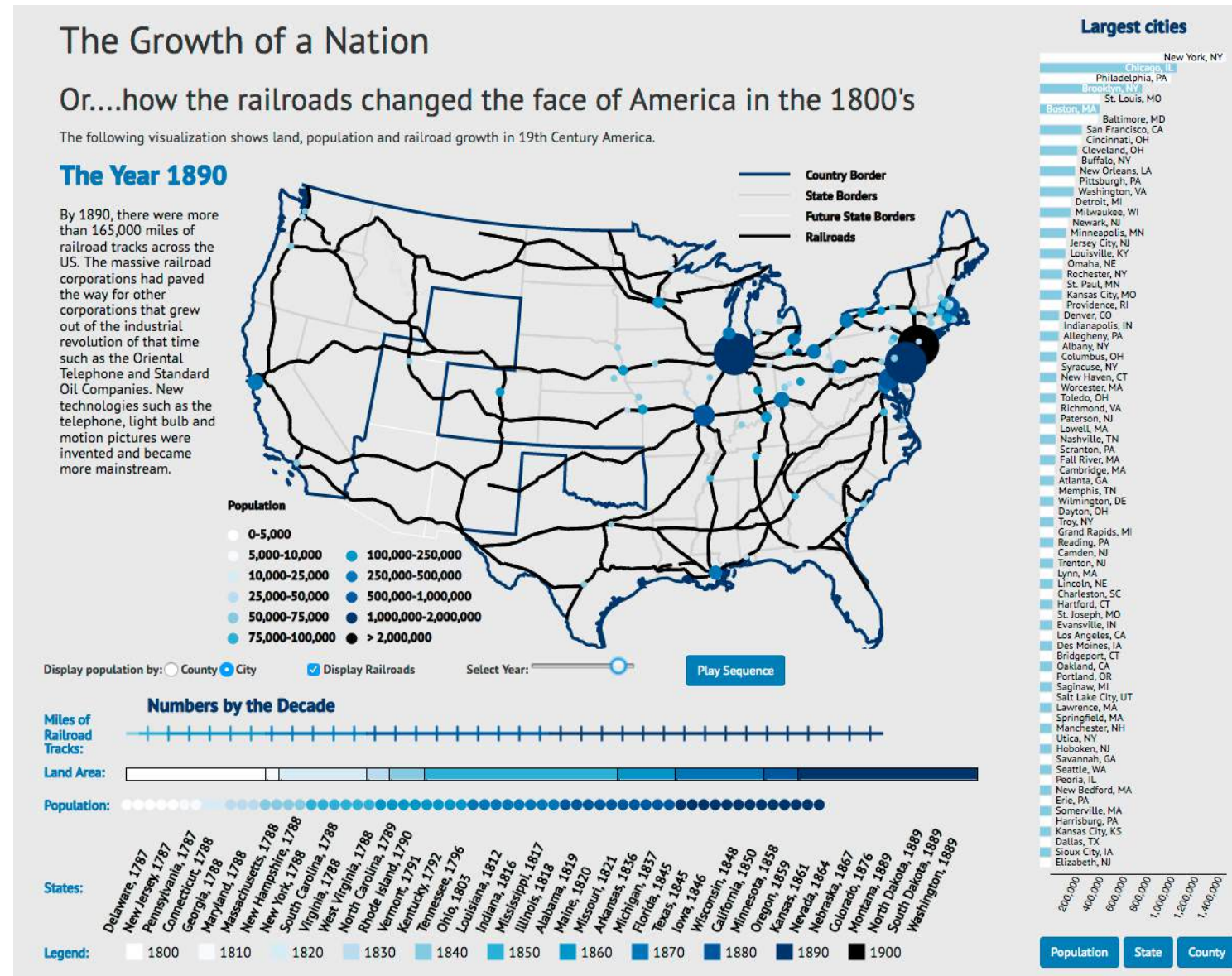
Change the viz itself: Re-encode

- Same data, different encodings



Change the viz itself: Change parameters

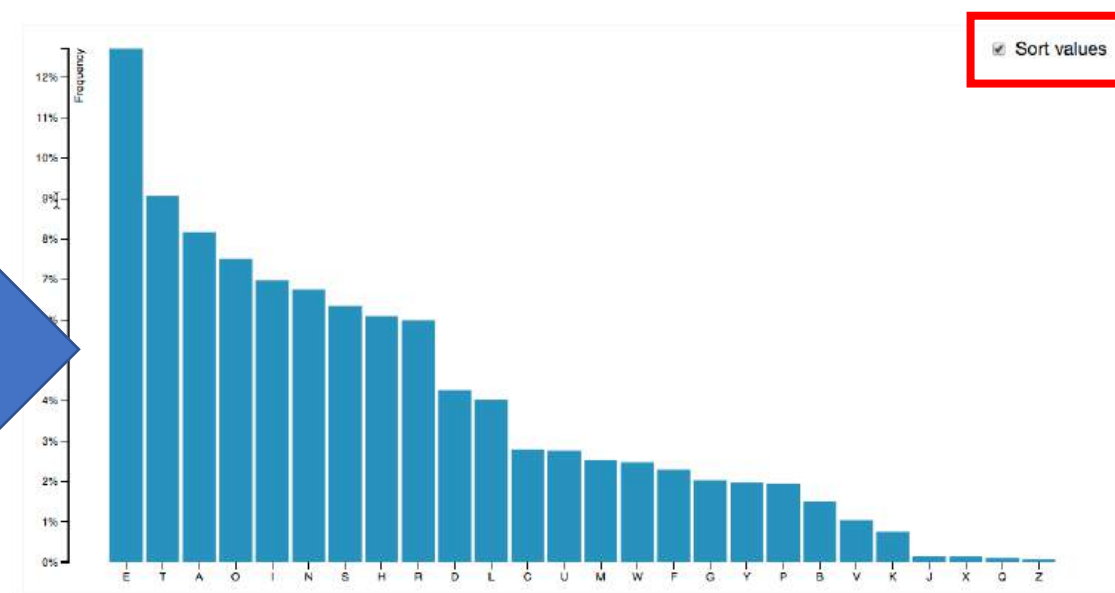
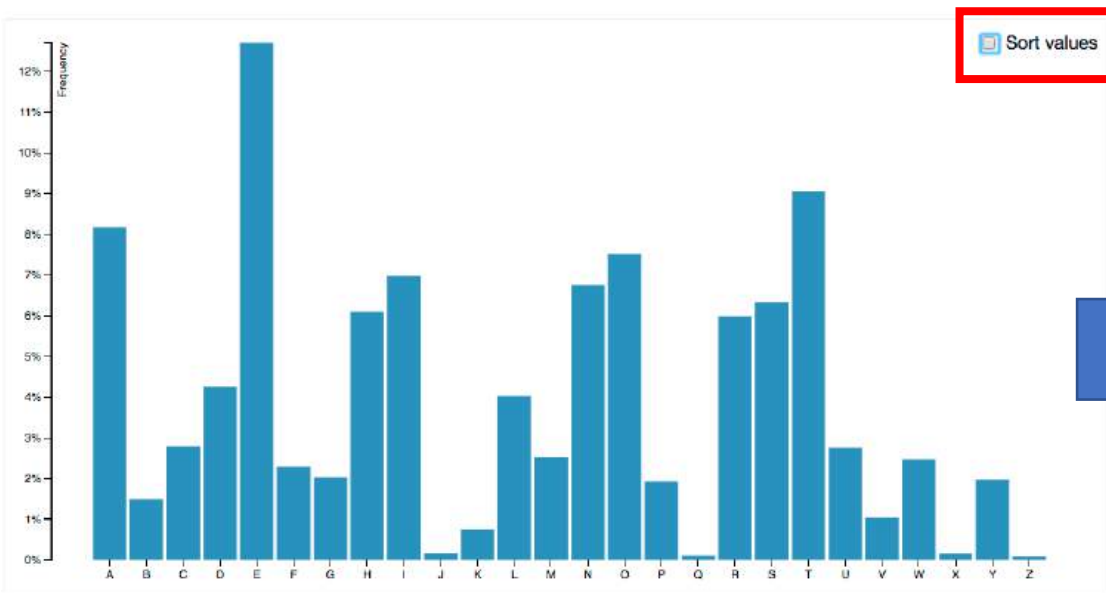
- Widgets and controls
 - Sliders, buttons, radio buttons, checkboxes, dropdowns
- Pros:
 - Clear affordances (Obvious what the user can do)
 - Self-documenting w. labels
- Cons:
 - Uses screen space
- Design choices:
 - Separated vs interleaved



<http://laurenwood.github.io/>

Change the viz itself: Rearrange/reorder

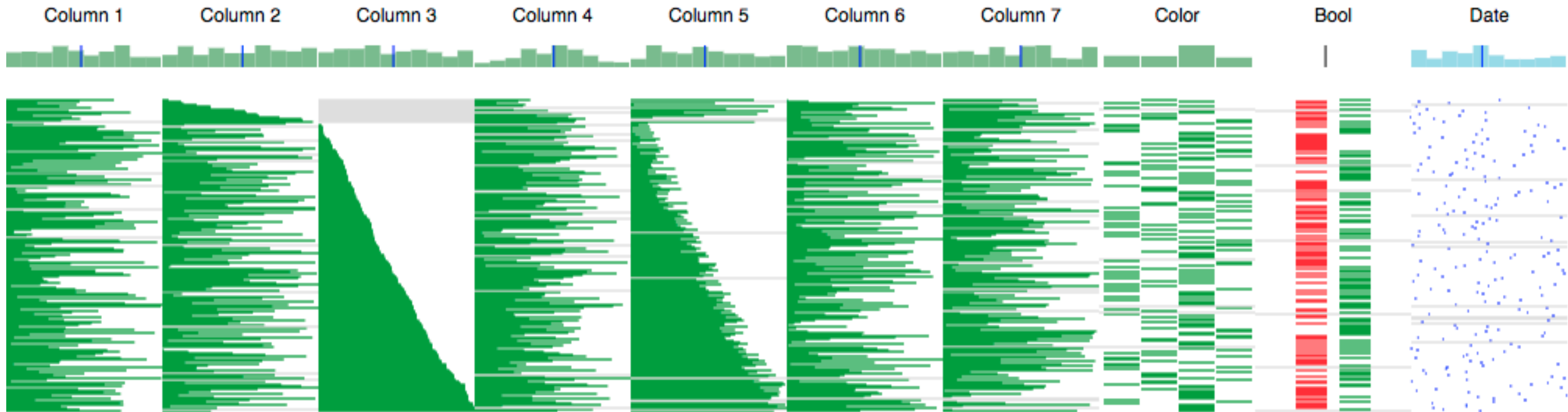
- AKA “sort”
- Useful for identifying:
 - Extreme values
 - Trends
 - Correlations between attributes



<https://observablehq.com/@d3/sortable-bar-chart>

Change the viz itself: Rearrange/reorder

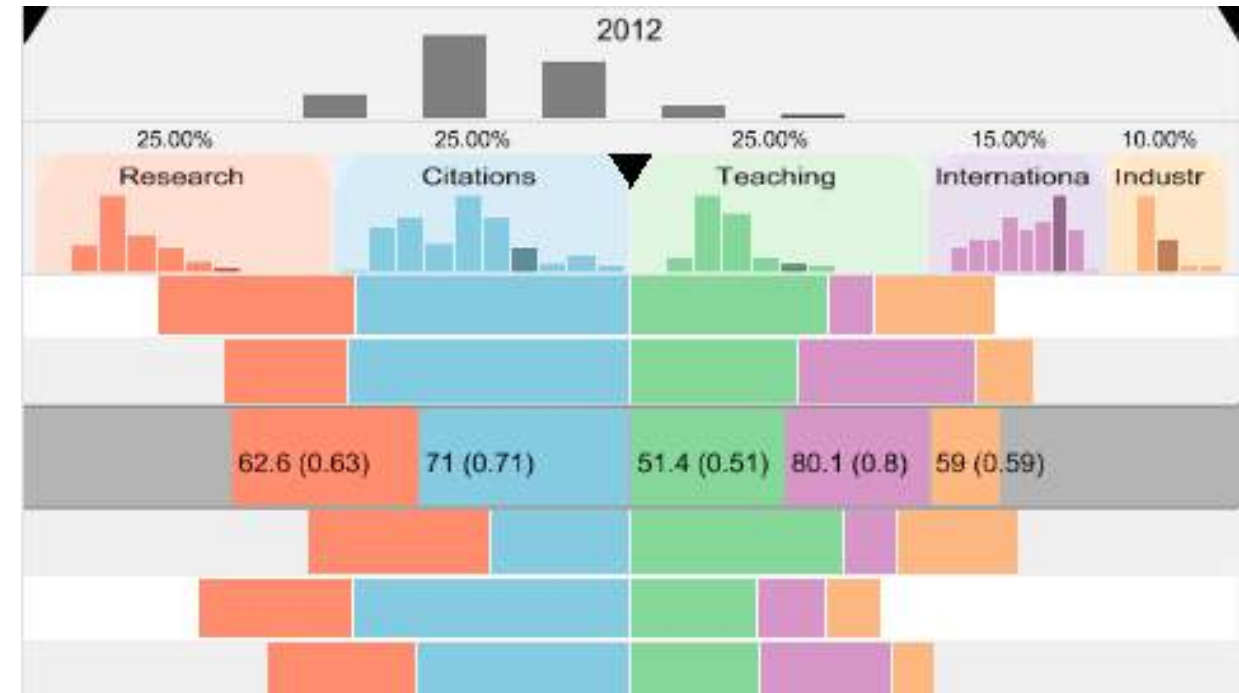
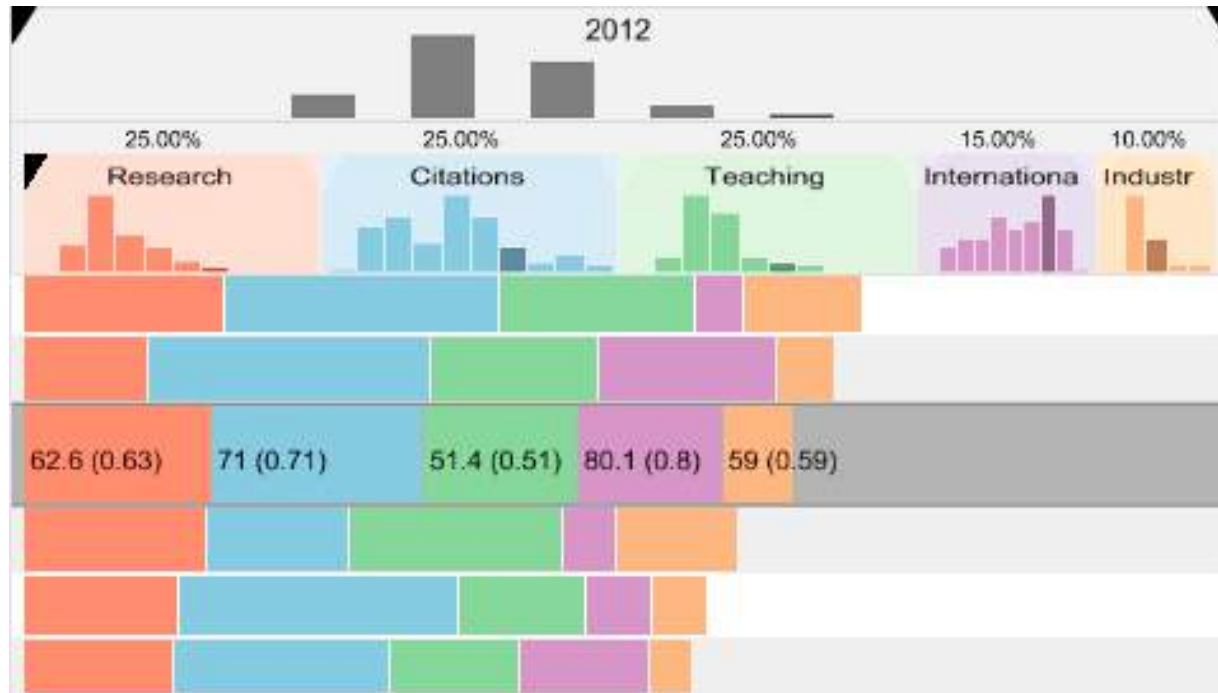
- AKA “sort”
- Useful for identifying:
 - Extreme values
 - Trends
 - Correlations between attributes



<http://carlmanaster.github.io/datastripes/>

Change the viz itself: Change alignment

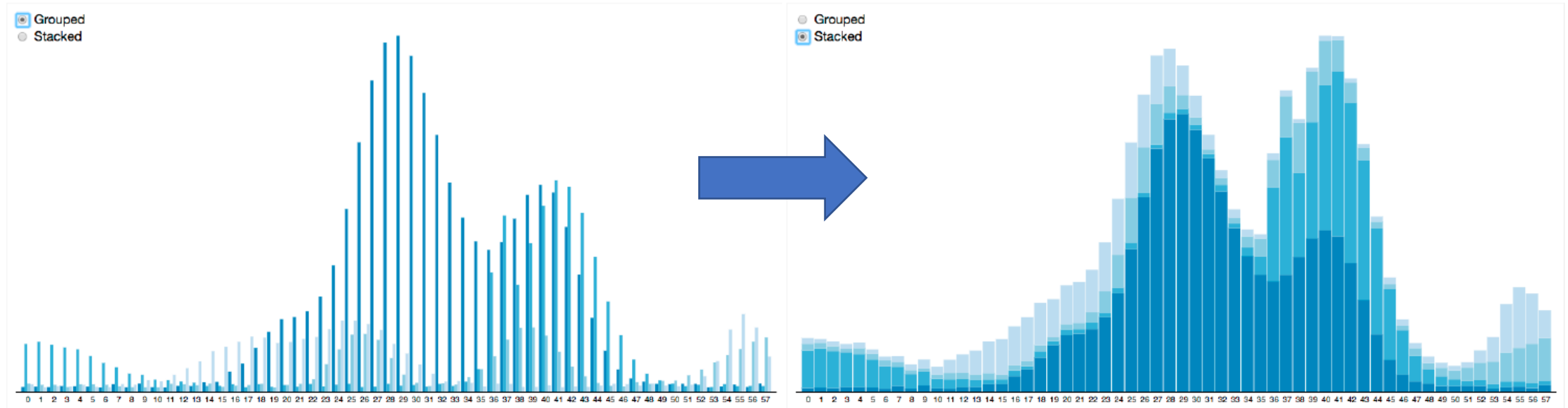
- Comparing aligned lengths is easy
- Allow the user to specify which category is edge-aligned



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

Change the viz itself: Animated transitions

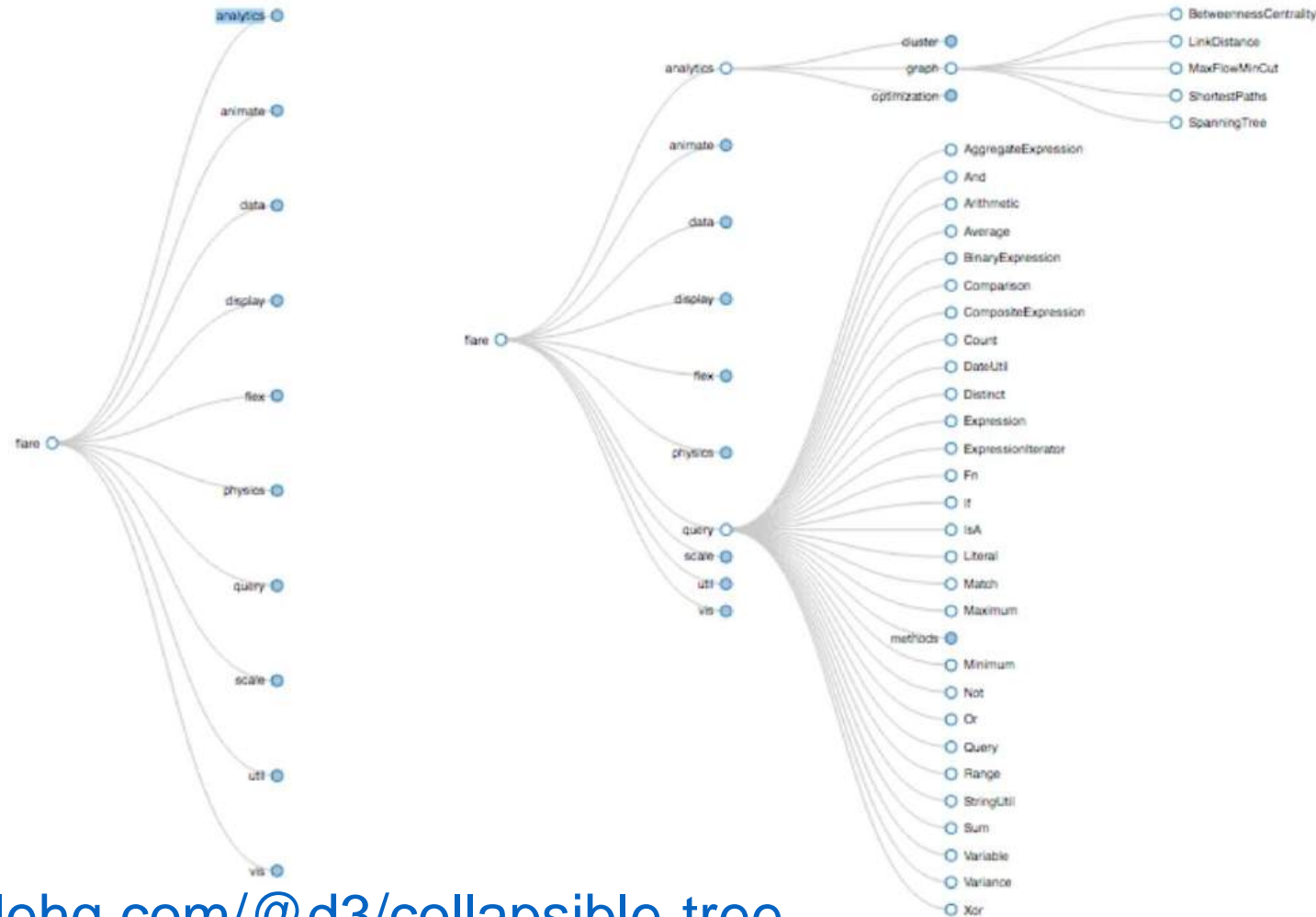
- Sudden changes in the viz can be confusing
- Animations can make it easier to track changes



<https://observablehq.com/@d3/stacked-to-grouped-bars>

Change the viz itself: Animated transitions

- Sudden changes in the viz can be confusing
- Animations can make it easier to track changes



<https://observablehq.com/@d3/collapsible-tree>

Manipulate the view

- Multiple options for how to manipulate:
 - Change the viz itself
 - Select
 - Navigate

Manipulate

➔ Change



➔ Select



➔ Navigate



Selection

- Consider the device being used
 - Mouse & keyboard:
 - Large screen, hover, multiple clicks
 - Mobile:
 - Small screen, no hover, just tap
- Gestures from sensors?
 - Mostly just in the movies...
- Eye tracking?
 - Still emerging



I Hate Tom Cruise - Alex Kauffmann

<https://www.youtube.com/watch?v=QXLfT9sFcbc>

Selection

- Selection is the basic operation for most interactions
- Design choices:
 - How many selection types?
 - Interaction modalities
 - Click/tap (heavyweight) vs hover (lightweight but device-dependent)
 - 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 the selection be null?
 - Primary vs secondary (e.g. source/target nodes in network)

Highlighting

- Visual indication of a selection
- Highlight: change the visual encoding for selected targets
 - The visual feedback is closely tied to but separable from selection
- Design choices: typical visual channels
 - Change item color
 - Hides existing color coding
 - Add outline mark
 - Change size (e.g. increase outline mark linewidth)
 - Change shape (e.g. solid → dashed line for link mark)
- Unusual channels: motion
 - Usually avoid for a single view
 - With multiple views, could justify to draw attention to other views (peripheral vision)

Manipulate the view

- Multiple options for how to manipulate:
 - Change the viz itself
 - Select
 - Navigate

Manipulate

➔ Change



➔ Select



➔ Navigate



Navigation

- Change viewpoint → changes which items are visible
- Camera metaphor
 - Zoom
 - Pan/translate
 - Constrained view
 - Tilt (usually only 3D)
 - Rotate/spin (usually only 3D)

➔ Navigate

➔ Item Reduction

➔ Zoom

Geometric or *Semantic*



Navigation: constrained vs unconstrained

➔ Navigate

- Unconstrained navigation – no “guard rails”
 - Easy to implement, hard to use
 - Easy to overshoot/undershoot the desired view
- Constrained navigation
 - Harder to implement, more user friendly
 - Usually uses animated transitions to make it clear
 - E.g. drag to select a subset of data, double click to fill the view with the selection
- E.g. <https://observablehq.com/@d3/zoom-to-bounding-box>

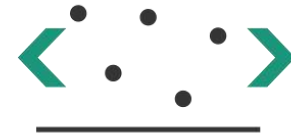
➔ Item Reduction

➔ Zoom

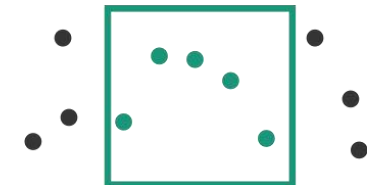
Geometric or *Semantic*



➔ Pan/Translate

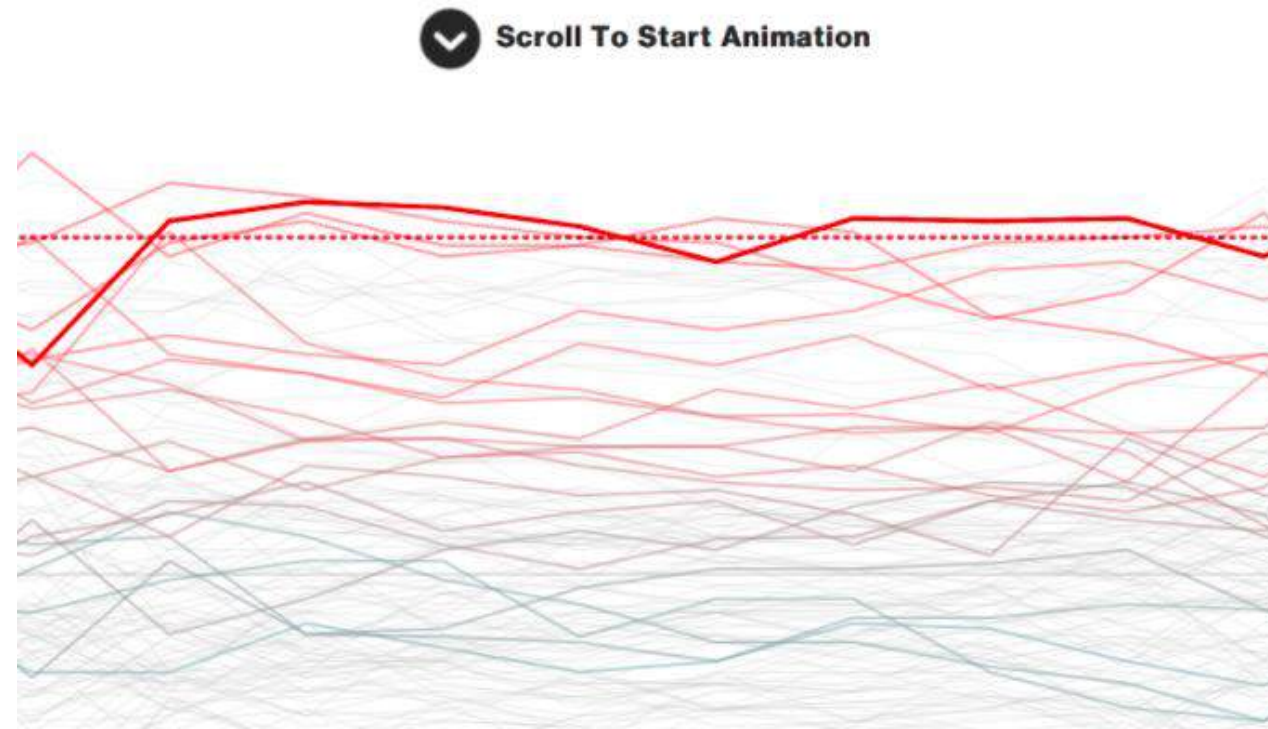


➔ Constrained



Navigation idiom: Scrollytelling

- How? Navigate page by scrolling
- Pros:
 - Familiar and intuitive (based on standard web browsing)
 - Linear (only up and down)
 - No overload of too many click-based buttons
- Cons:
 - Full-screen mode may lack affordances (no scrollbar visible)
 - Scrolljacking: user is forced to scroll, which constrains mouse usage
 - Unexpected behavior
 - Continuous control for discrete steps

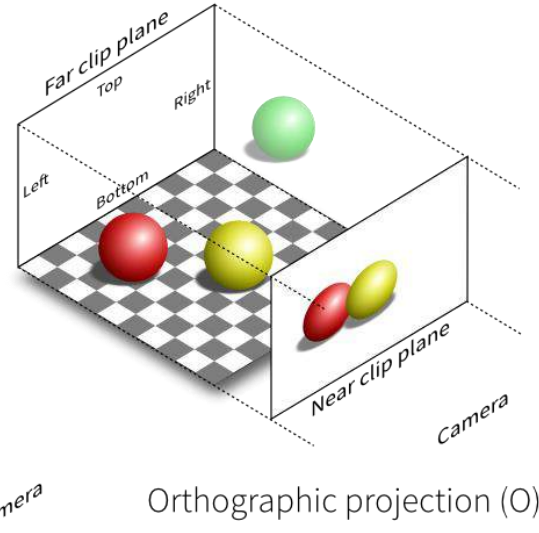
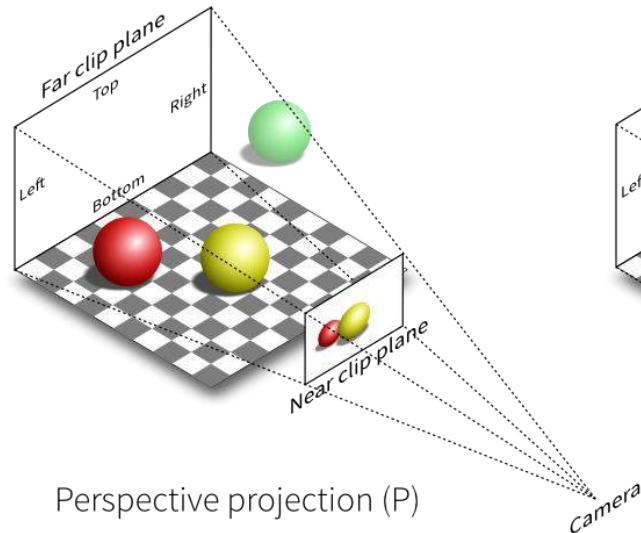


How to Scroll, Bostock: <https://bost.ocks.org/mike/scroll/>

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

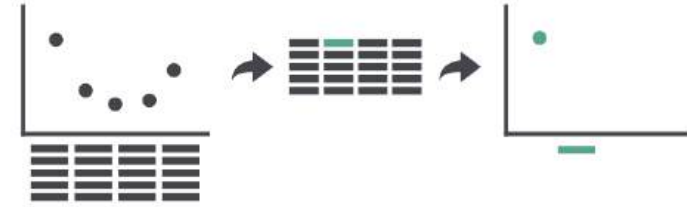
Navigation: reducing attributes

- Extension of the camera metaphor
- Slice
 - Show only items matching specific value for given attribute: slicing plane
 - Axis aligned, or arbitrary alignment
- Project
 - Change mathematics of image creation
 - Orthographic
 - Perspective
 - Many others



→ Attribute Reduction

→ Slice



→ Cut



→ Project





Multiple Views

Facet

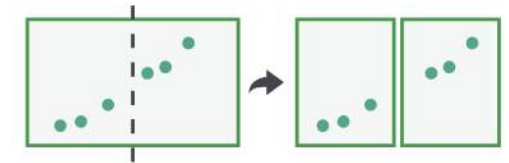
- Another way to handle complexity
- Juxtapose
 - Put multiple views of the same data next to each other
- Partition
 - Split one view into two (or more) views
- Superimpose
 - Store multiple views as different layers that can be displayed on top of one another

Facet

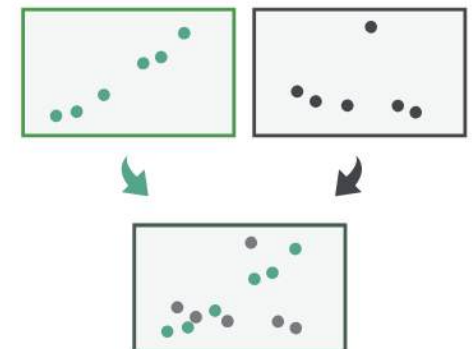
➔ Juxtapose



➔ Partition



➔ Superimpose



Facet

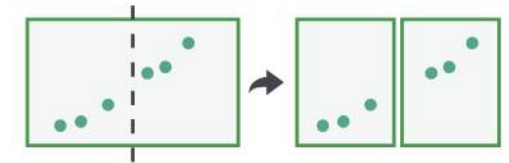
- Another way to handle complexity
- Juxtapose
 - Put multiple views of the same data next to each other
- Partition
 - Split one view into two (or more) views
- Superimpose
 - Store multiple views as different layers that can be displayed on top of one another

Facet

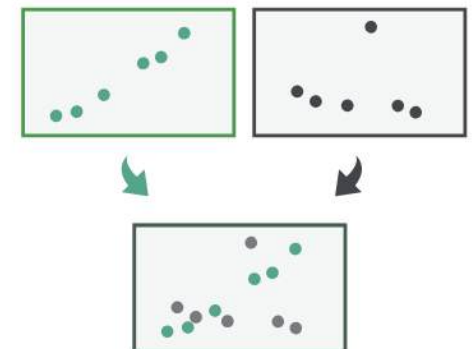
➔ Juxtapose



➔ Partition



➔ Superimpose



Juxtapose and coordinate views

- Main problem: How to link the views? → Share Encoding: Same/Different

- How to design the views so that manipulating one view does (or does not) change the other view

→ *Linked Highlighting*



- 4 main design questions:

- Do the views have the same or different visual encoding idioms?
 - Is selection/highlighting linked across multiple views?
 - Do the views show the same data, or different data?
 - Is navigation synchronized between the views?

→ Share Data: All/Subset/None

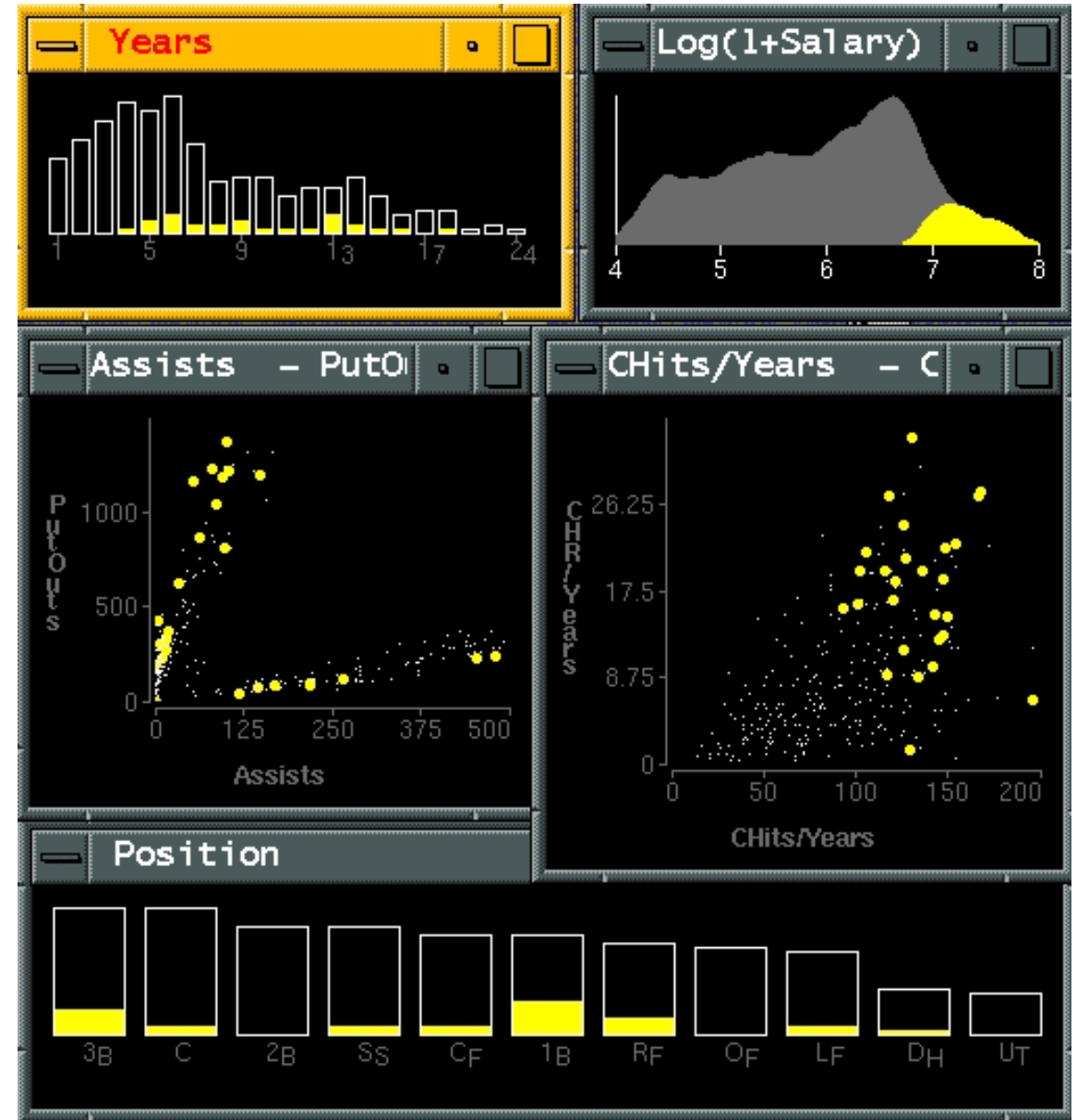


→ Share Navigation



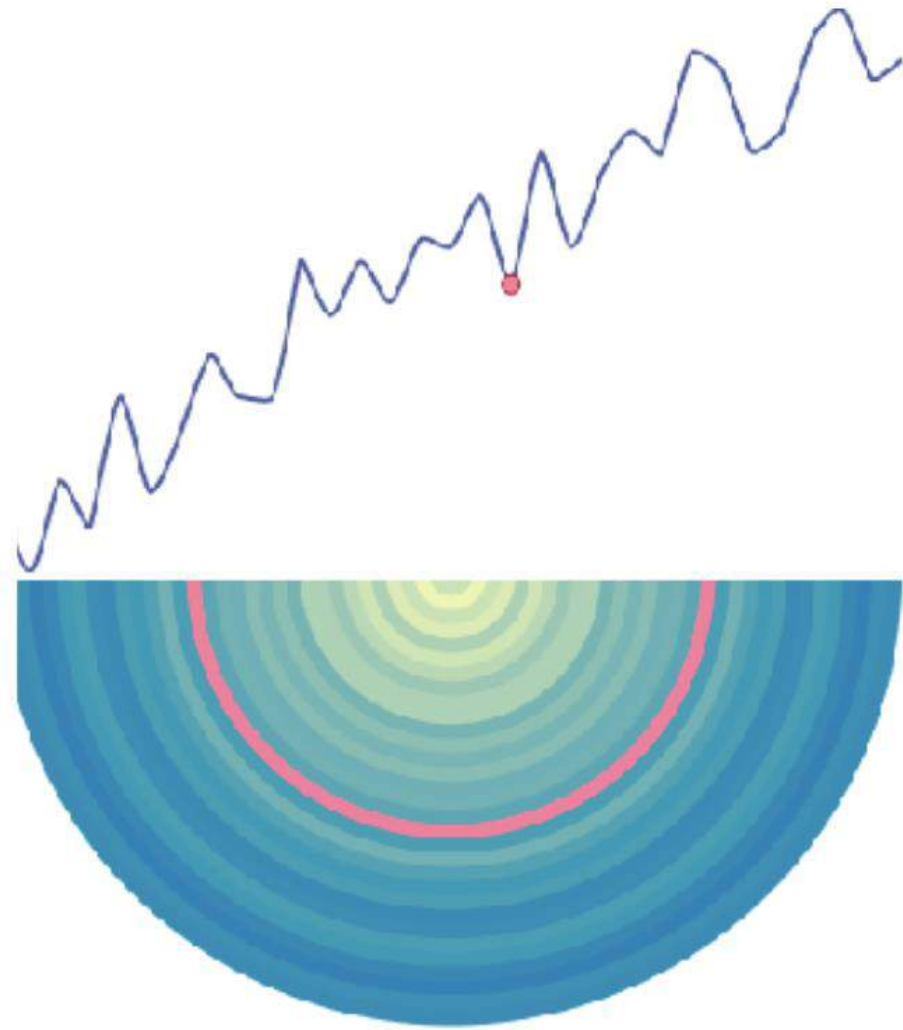
Juxtapose idiom: Linked highlighting

- See how regions contiguous in one view are distributed in another
 - Powerful and pervasive interaction idiom
- Encoding: different
 - Multiform
- Data: all shared
 - All **items** shared
 - Different **attributes** across the views
- AKA: brushing and linking



Linked views: directionality

- Unidirectional vs bidirectional linking
 - Bidirectional is almost always better!

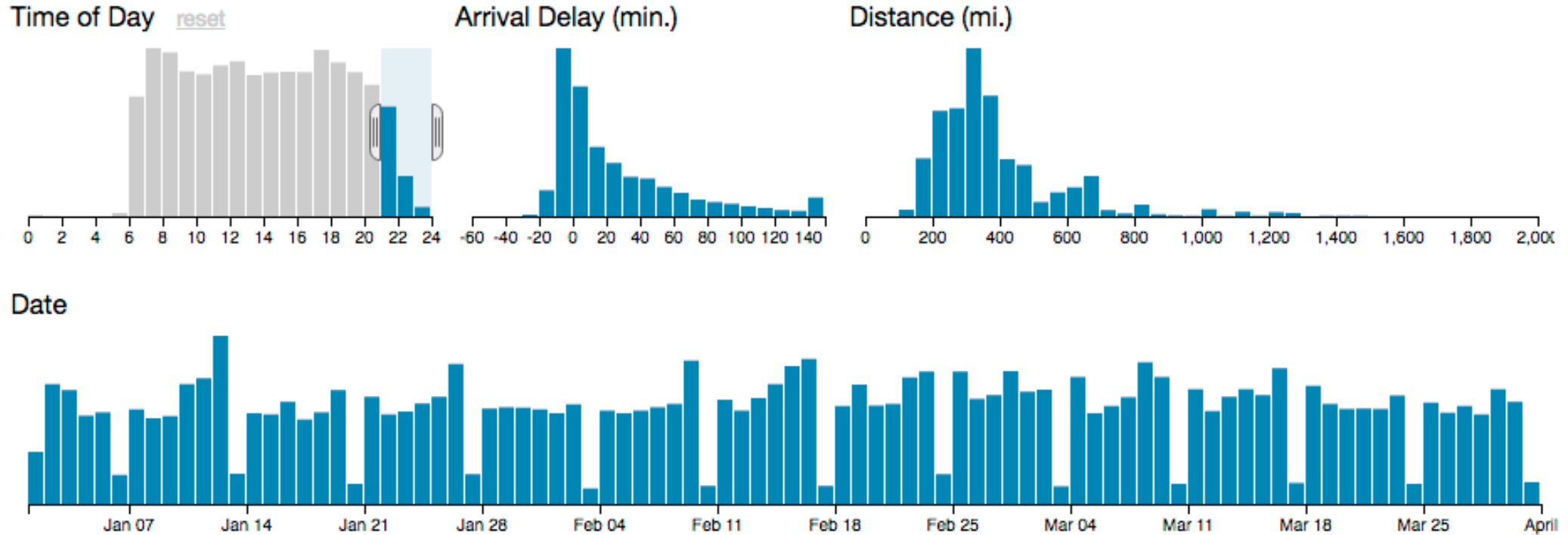


<http://pbeshai.github.io/linked-highlighting-react-vega-redux/>

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

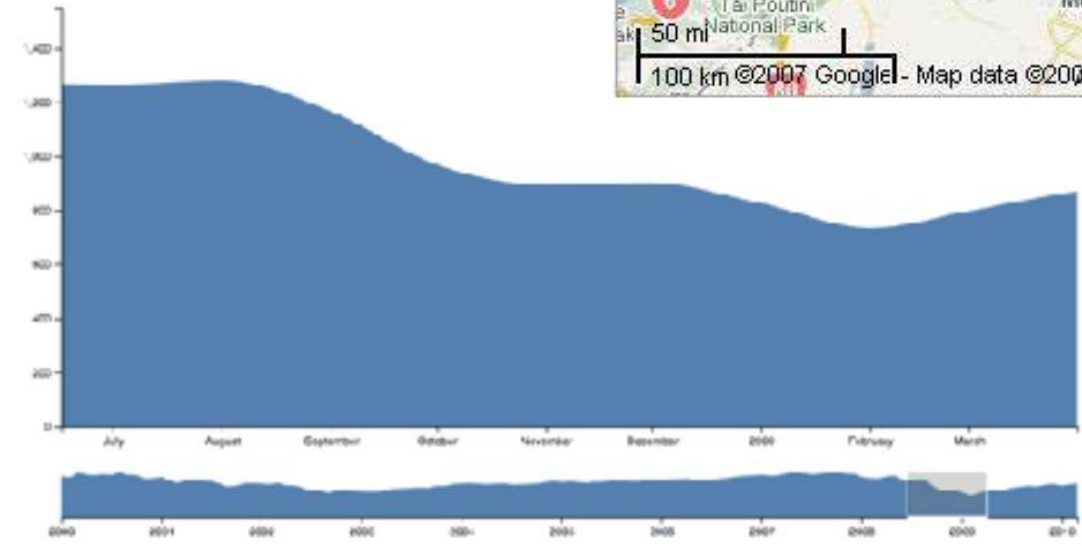
Example: Cross filter

- Very fast javascript library: <http://square.github.io/crossfilter/>



Juxtapose idiom: overview-detail views

- Encoding: same or different
 - e.g. same (birds-eye map)
- Data: shared subset
 - Viewpoint differences
- Navigation: shared
 - Bidirectional linking
- Other differences
 - Window size
 - Level of detail



Juxtapose idiom: tooltips

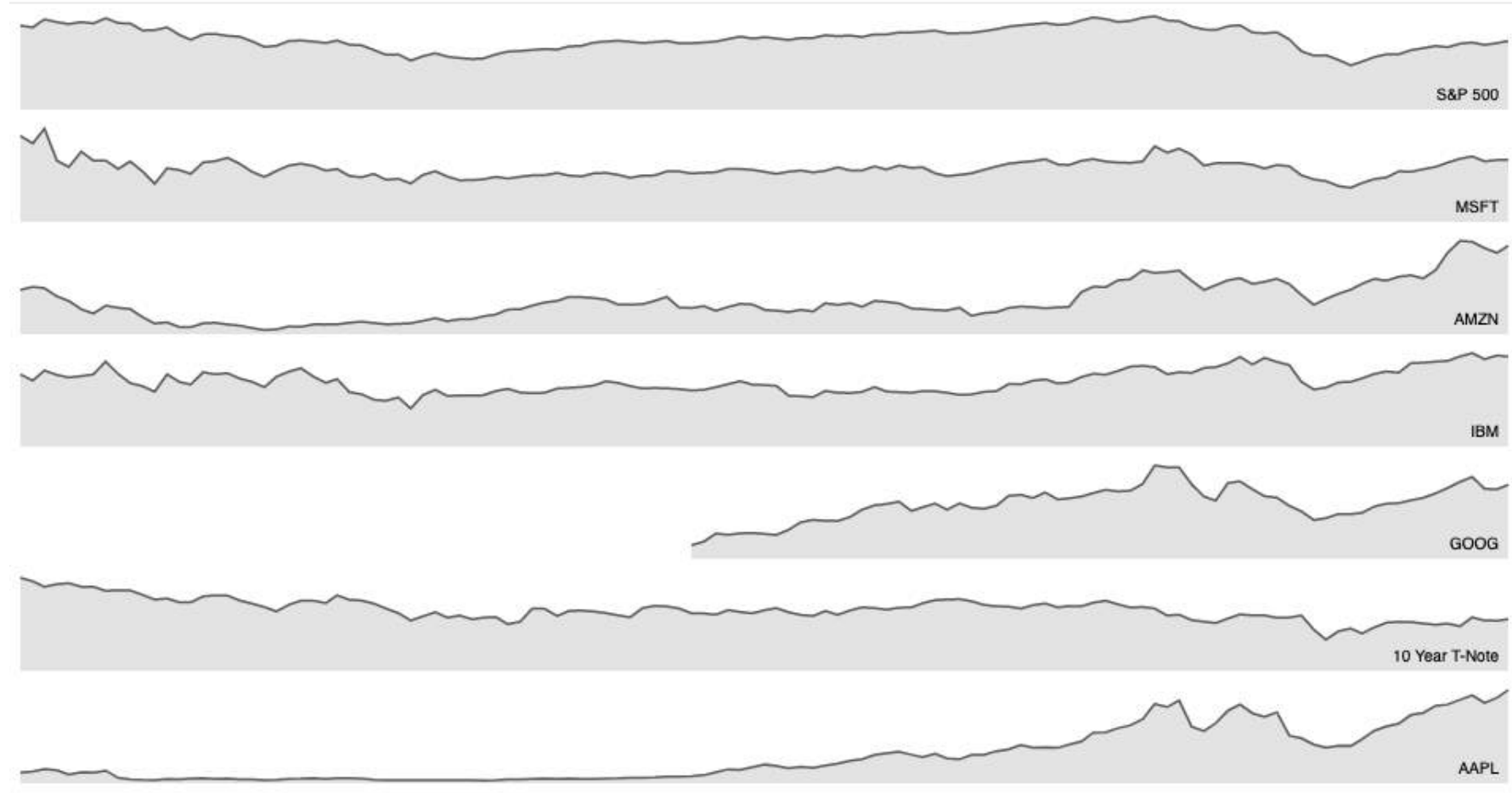
- Pop-up information for selection
 - Hover or click
 - Specific case of “details on demand”
- Beware: tooltips do **not** support overview
 - Possibly add an additional visual encoding to support overview
 - “If you make a rollover or tooltip, assume nobody will see it. If it’s important, make it explicit.”
 - Gregor Aisch, New York Times
- Prefer tooltips that provide more details instead of completely new information



<https://www.highcharts.com/demo/dynamic-master-detail>

Juxtapose idiom: small multiples

- We've seen this before
- Encoding: same
 - E.g. line charts
- Data: none shared
 - Different slices of dataset
 - E.g. stock prices for different companies



Interactive small multiples

- Linked highlighting: analogous item/attribute shown across views
 - E.g. same year highlighted across all charts when hovering in any single chart

The Rise and Decline of Ask MetaFilter

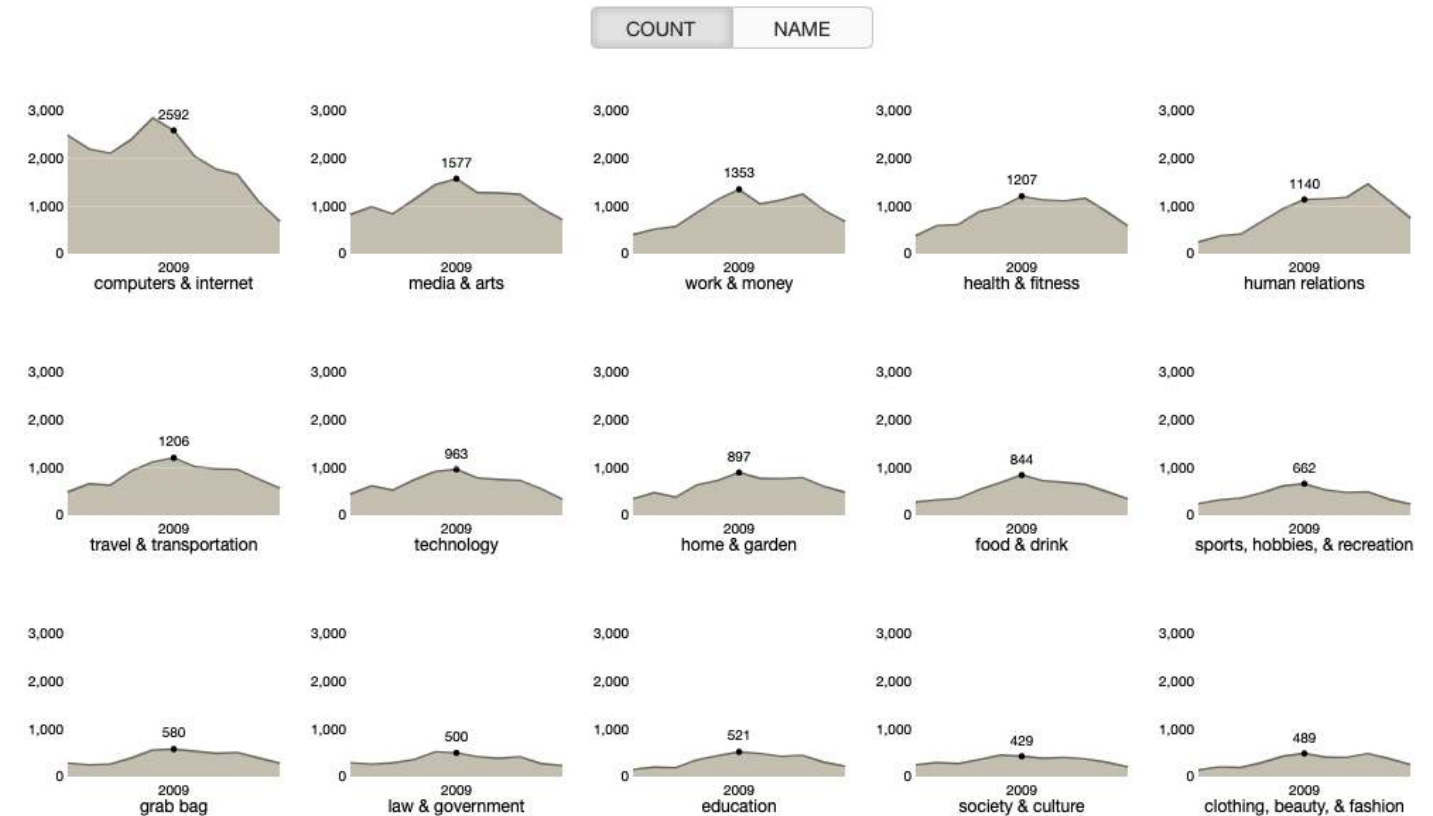
Metafilter's revenue has been on the decline, but has its content dried up as well?

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.

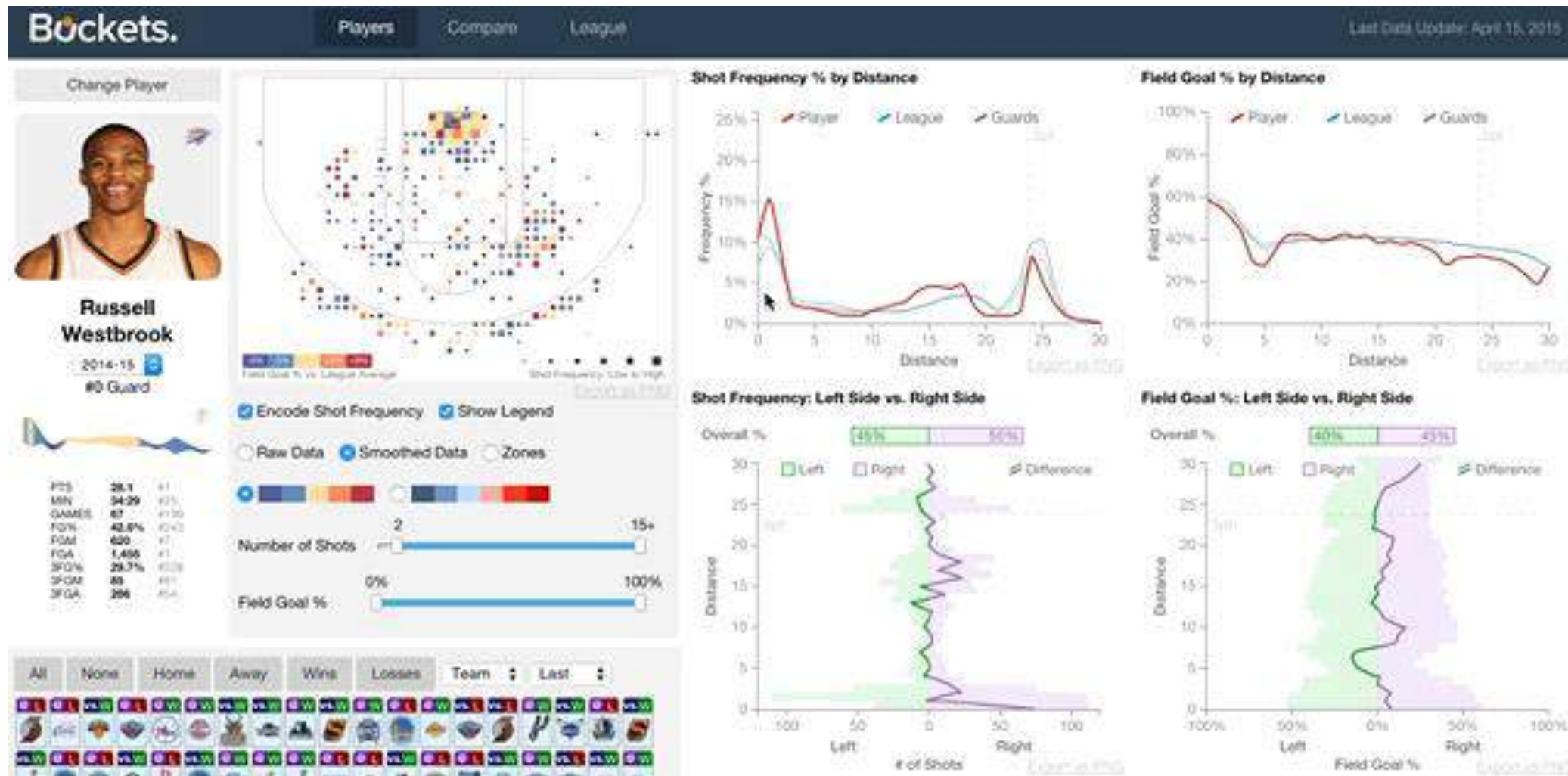
Disclaimer: 2014 is included, even though the year is not over yet.



https://projects.flowingdata.com/tut/linked_small_multiples_demo/

Example: combining many interaction idioms

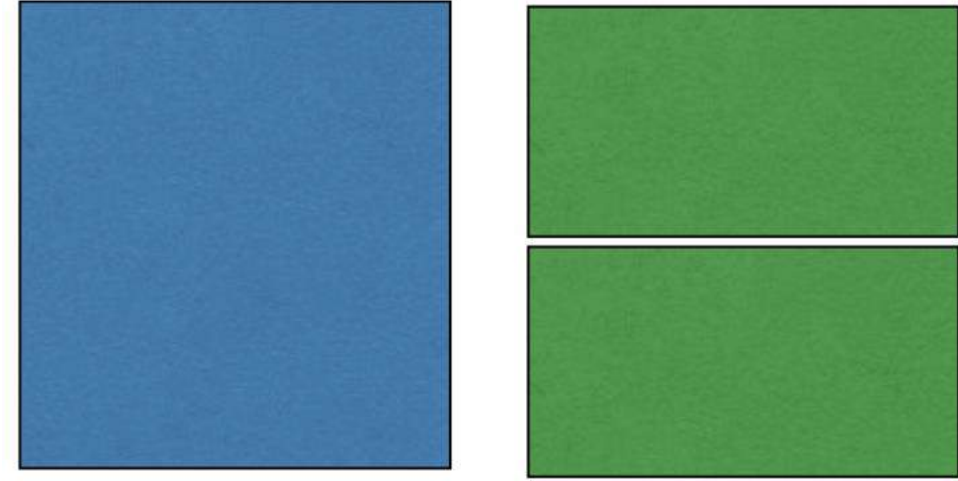
- https://buckets.peterbeshai.com/app/#/playerView/201935_2015



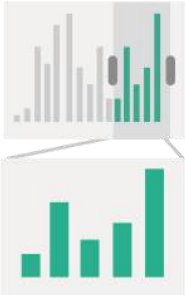

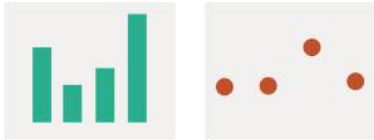
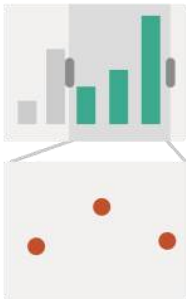
- Multiform
- Multidirectional linked highlighting of small multiples
- Tooltips

Juxtapose views: tradeoffs

- Juxtapose pros:
 - Cognitive load: eyes > memory
 - Lower cognitive load: move eyes between 2 views
 - Higher cognitive load: compare single changing view to memory of previous state
- Juxtapose cons:
 - Display area
 - 2 views side by side → each view has half the area

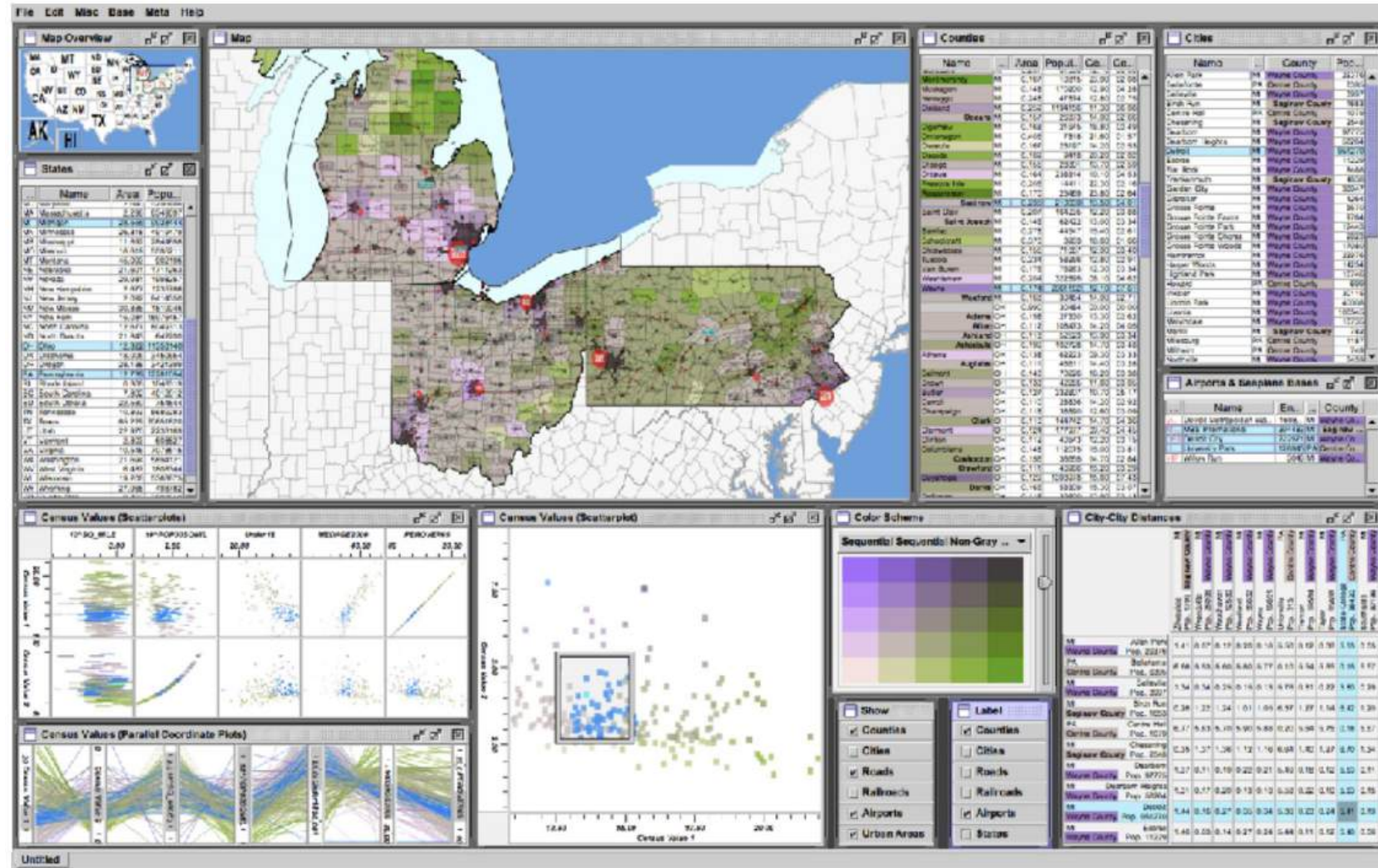


View coordination: design choices

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

Juxtapose idiom: reorderable lists

- List views
 - Easy lookup
 - Useful when linked to other views, not so useful on their own
- How many views is okay vs too complex?
 - Open research question



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

Facet

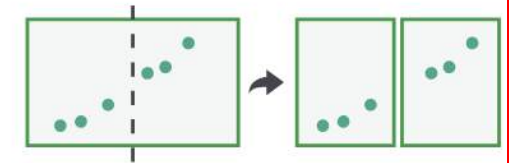
- Another way to handle complexity
- Juxtapose
 - Put multiple views of the same data next to each other
- Partition
 - Put multiple views of the same data next to each other
- Superimpose
 - Store multiple views as different layers that can be displayed on top of one another

Facet

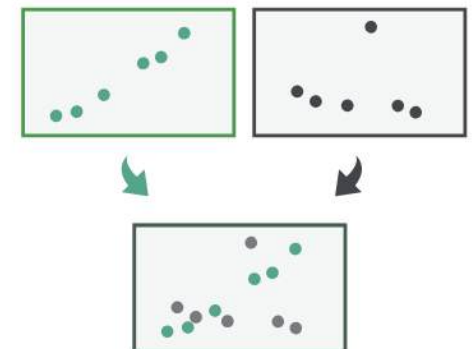
➔ Juxtapose



➔ Partition



➔ Superimpose

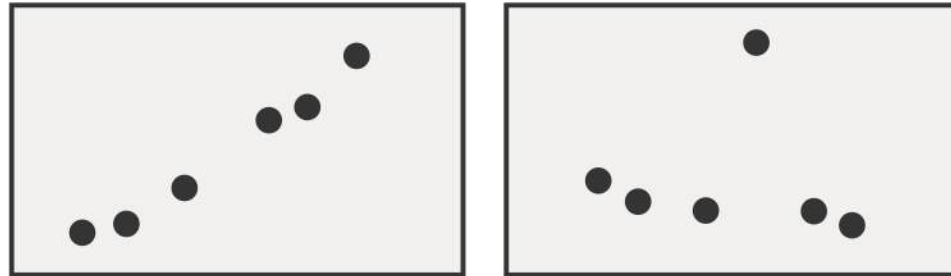


Partition into views

- How to divide data between views
 - Usually split into regions by attributes
 - Encodes association between items using spatial proximity
 - Order of splits has major implications for visible patterns
- Major choices:
 - How many views to split into?
 - How to order the views?

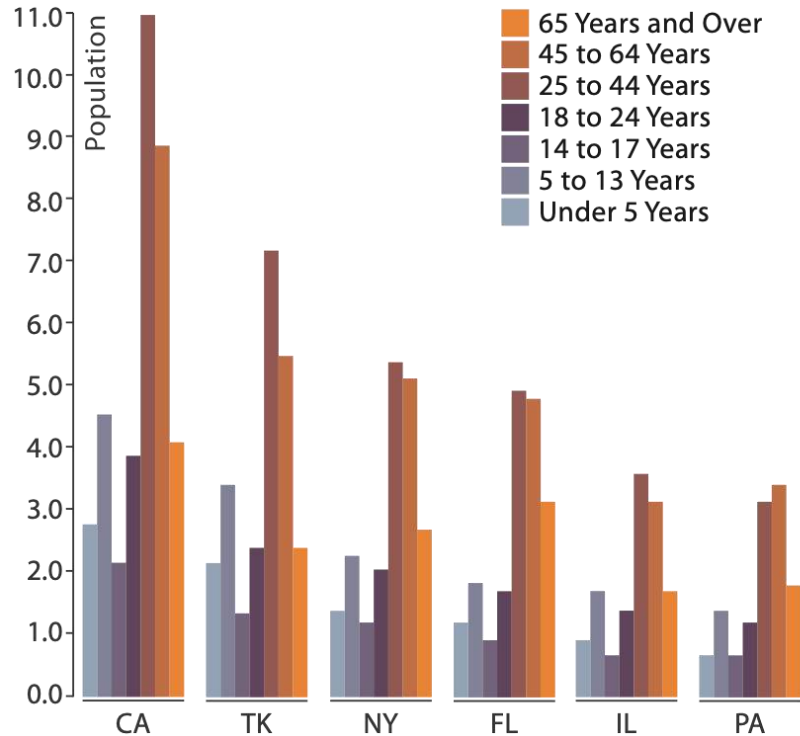


Partition into Side-by-Side Views



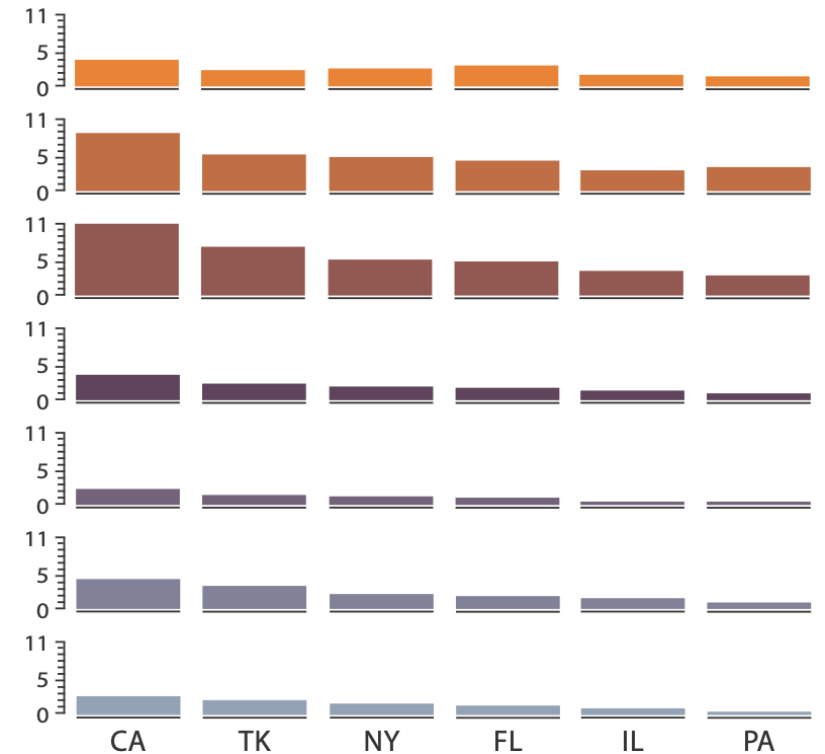
Partitioning: grouped vs small-multiple bars

- Single bar chart with grouped bars
 - Split by state into regions
 - Compare: easy within state, hard across ages



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

- Small-multiple bar charts
 - Split by age into regions
 - Compare: easy within age, hard across states



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

Partitioning: recursive subdivision

- Real estate in London
- Split by neighborhood
- Then by type
 - Flat, terrace, semi-detached, detached
- Then by time
 - Years as rows
 - Months as columns
- Color by price
- What patterns?
 - Where it's expensive
 - Where you pay more for detached type



Configuring Hierarchical Layouts to Address Research Questions.
Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and
Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.

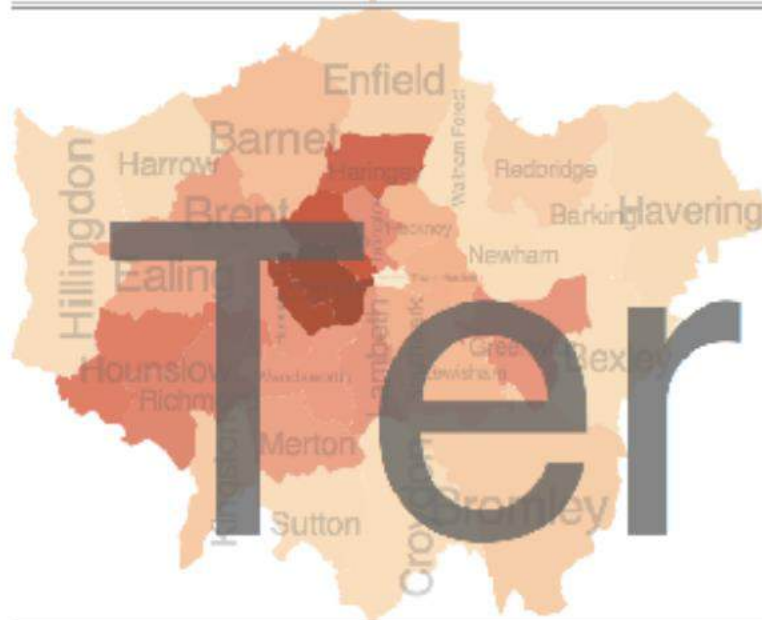
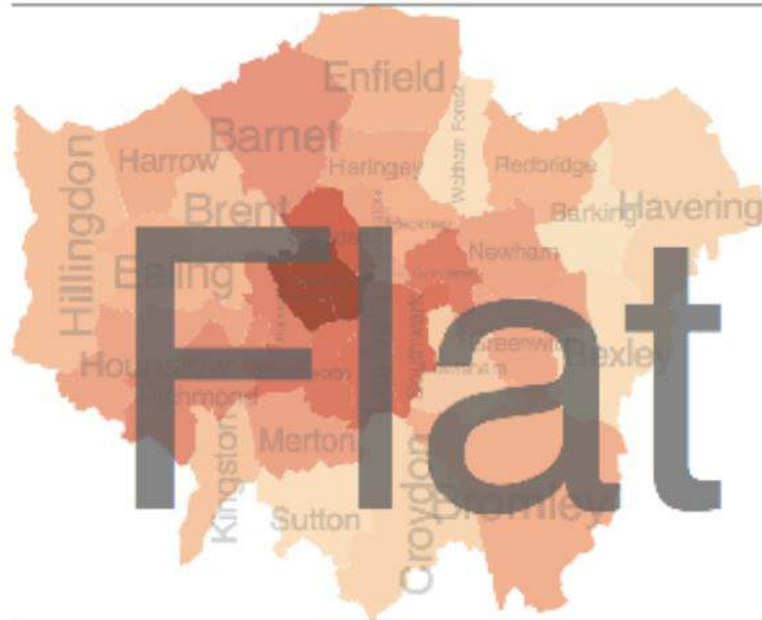
Partitioning: recursive subdivision

- Switch order of splits
 - Type then neighborhood
- Switch color
 - By price variation
- What pattern?
 - Within specific type, which neighborhood is inconsistent
- Order of division matters!



Partitioning: recursive subdivision

- Different encoding: choropleth map
 - Highlights geographic patterns



Facet

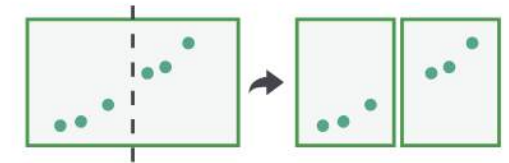
- Another way to handle complexity
- Juxtapose
 - Put multiple views of the same data next to each other
- Partition
 - Put multiple views of the same data next to each other
- Superimpose
 - Store multiple views as different layers that can be displayed on top of one another

Facet

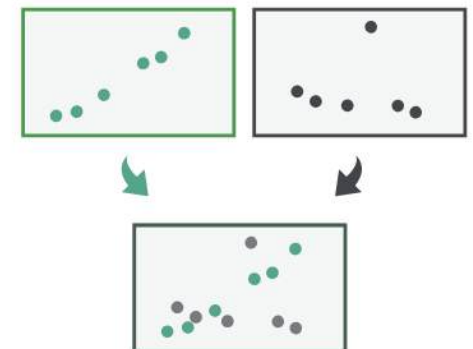
➔ Juxtapose



➔ Partition



➔ Superimpose

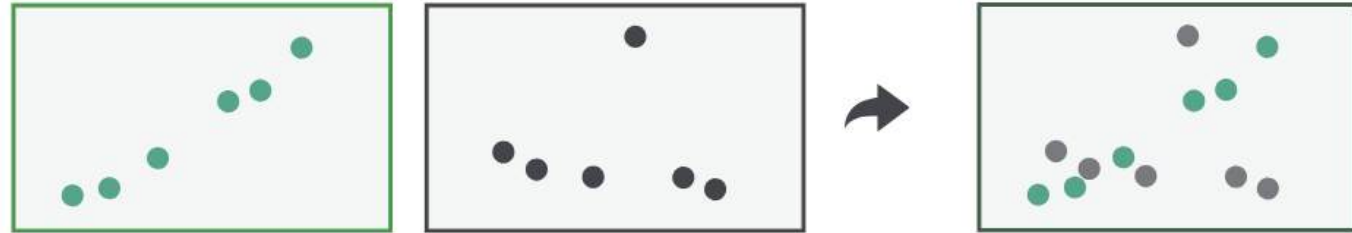


Superimpose layers

- Layer: set of objects spread out over a region
 - Each set is a visually distinguishable group
 - Covers the whole view
- Design choices:
 - How to distinguish layers?
 - Encode layers with different, nonoverlapping channels
 - How many layers?
 - 2 is achievable, 3 with careful design
 - Small static set, or dynamic with interaction?



Superimpose Layers



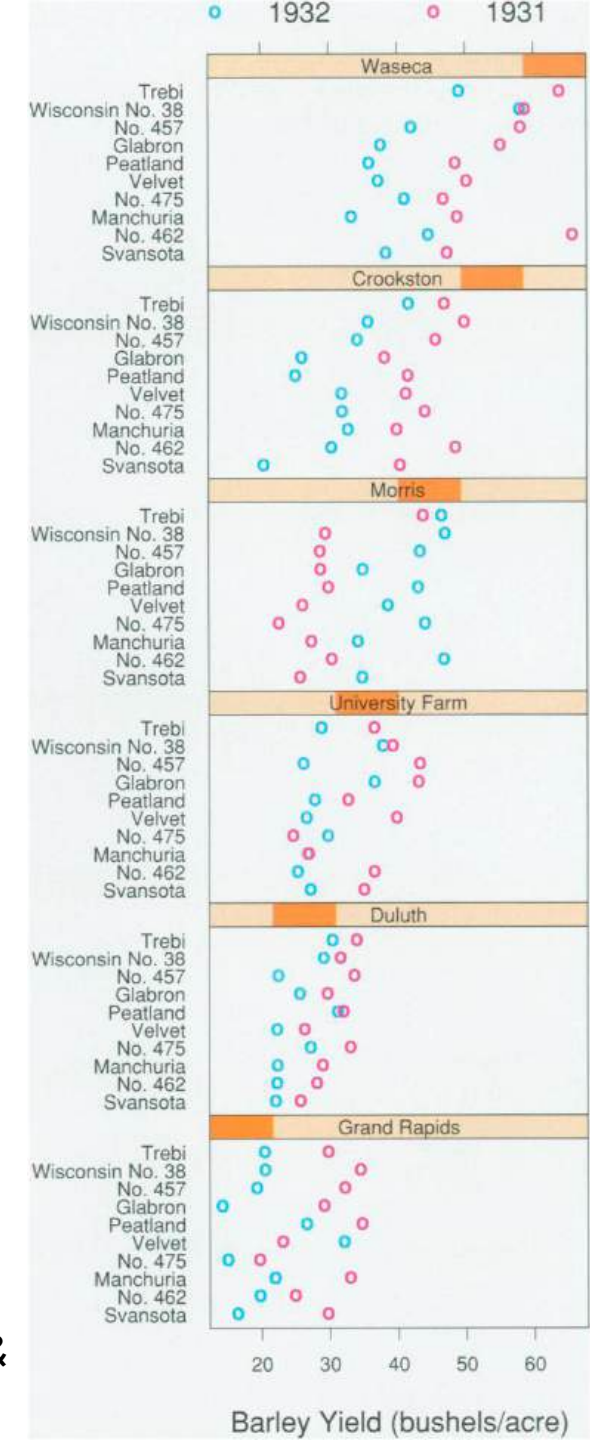
Static visual layering

- Foreground layer: roads
 - Hue, size to distinguish major from minor
 - High luminance contrast from background
- Background layer: regions
 - Desaturated colors for regions
- User can selectively focus attention



Static layering 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
- What patterns emerge?
 - Blue and pink swapped for Morris!

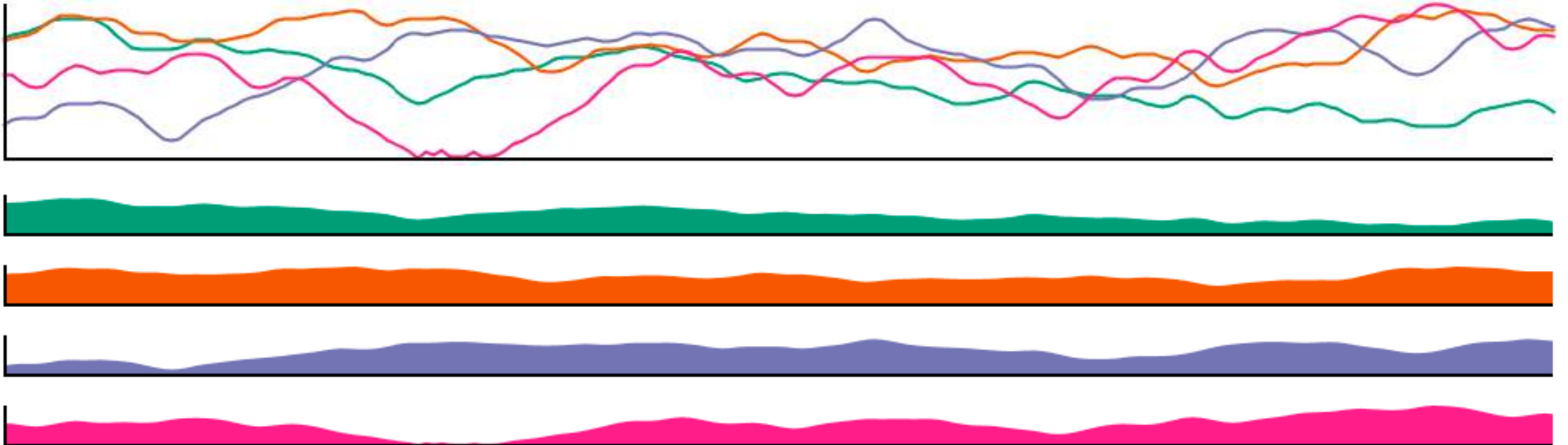


The Visual Design and Control of Trellis Display. Becker, Cleveland, & Shyu.

Limits of superimposing (static)

- Limit on number of layers
 - A few dozen for lines, but not hundreds
- Superimpose vs juxtapose: empirical study
 - Same size: small multiples vs single superimposed
 - Need to have the same number of pixels!
 - Superimposed: good for local tasks
 - Juxtaposed: good for global tasks, esp. for many charts

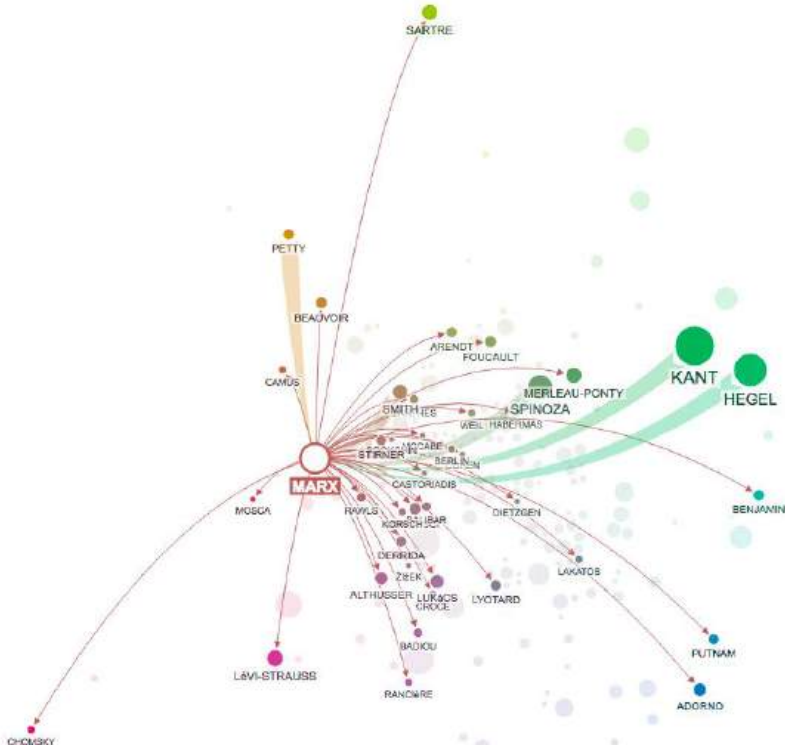
Graphical Perception of Multiple Time Series. Javed, McDonnell, and Elmqvist. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE InfoVis 2010) 16:6 (2010), 927–934



Dynamic visual layering

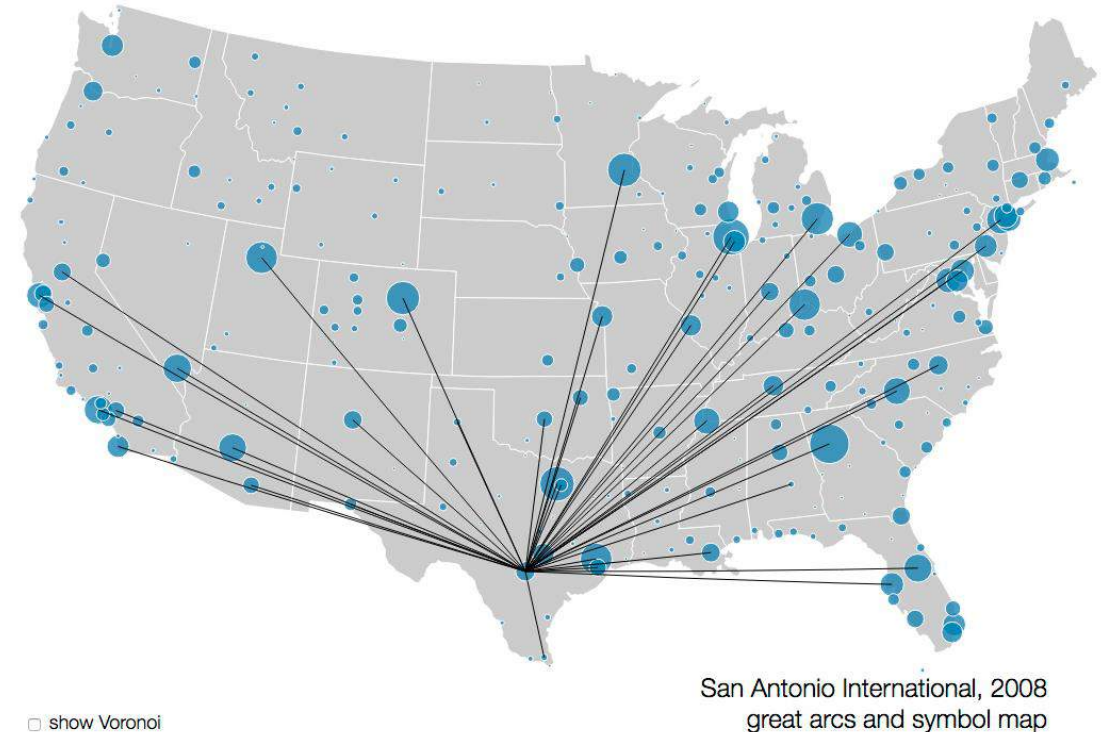
- Interactive, based on selection!
- One-hop neighbor highlighting

Clicking (heavyweight)



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

Hovering (lightweight)



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



Other Options

Reduce items or attributes

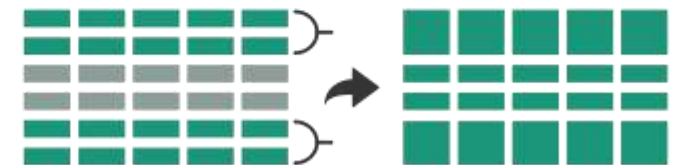
- Within a single view, *show less stuff*
 - With *filtering*, you simply show fewer items and/or attributes
 - Maybe they aren't needed for the particular viz question
 - With *aggregating*, many items and/or attributes are combined

Reduce

➔ Filter



➔ Aggregate

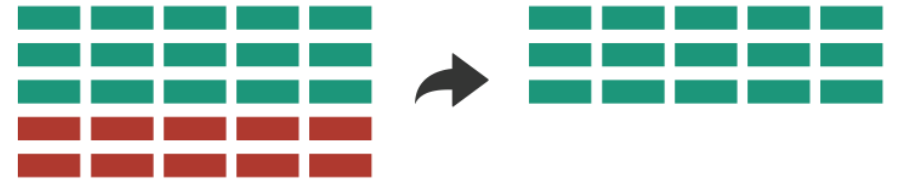


Filter items or attributes

- Pros:
 - Straightforward and intuitive
 - Easy to understand
 - Easy to implement
- Cons:
 - Once data is out of sight, it is often out of mind
 - Data may be filtered unintentionally/erroneously and never looked at again

➔ Filter

➔ Items



➔ Attributes

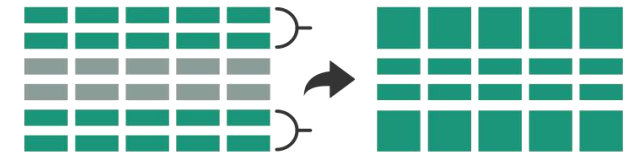


Aggregate items or attributes

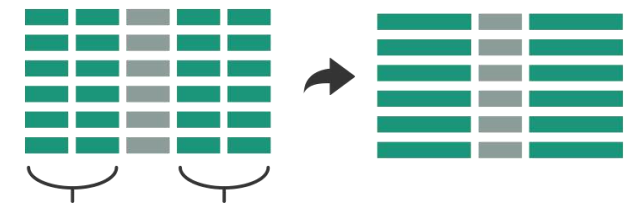
- Pros:
 - Preserves (and even enhances) data about the whole set
- Cons:
 - Difficult to avoid losing "signal" (detail)

➔ Aggregate

➔ Items



➔ Attributes

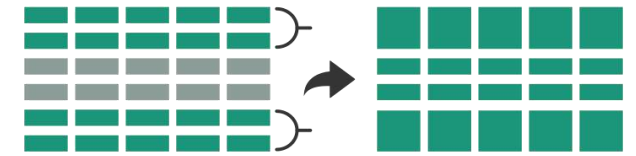


Aggregate items or attributes

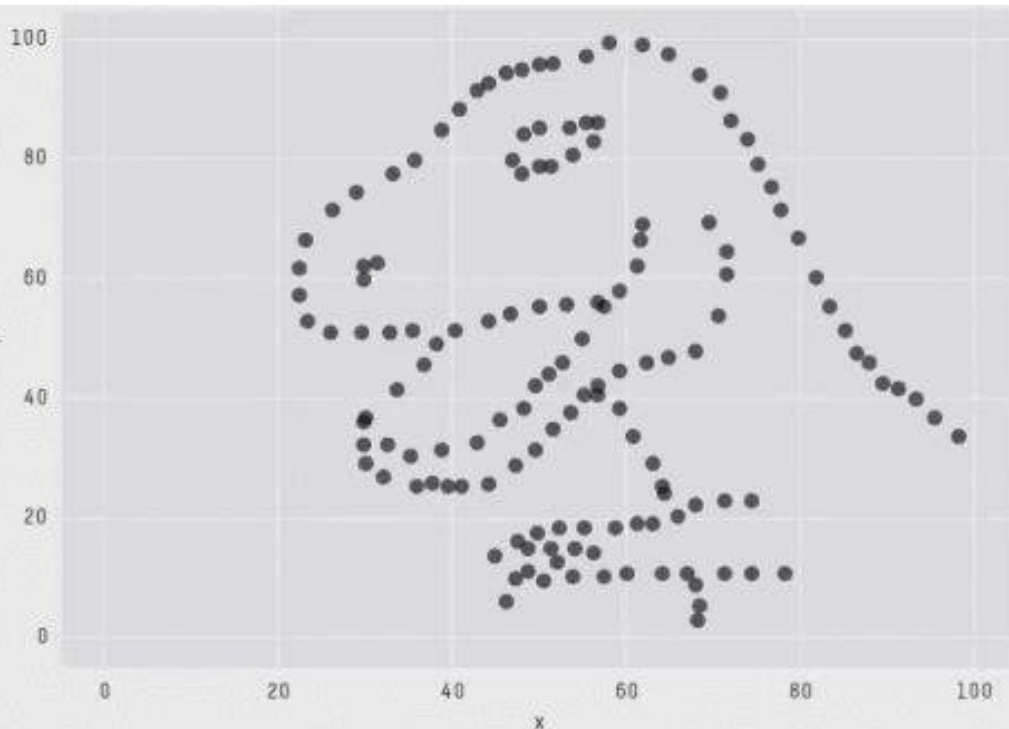
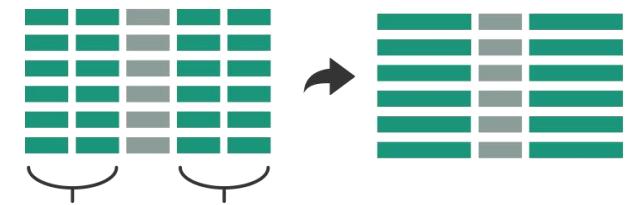
- Pros:
 - Preserves (and even enhances) data about the whole set
- Cons:
 - Difficult to avoid losing "signal" (detail)

➔ Aggregate

➔ Items



➔ Attributes



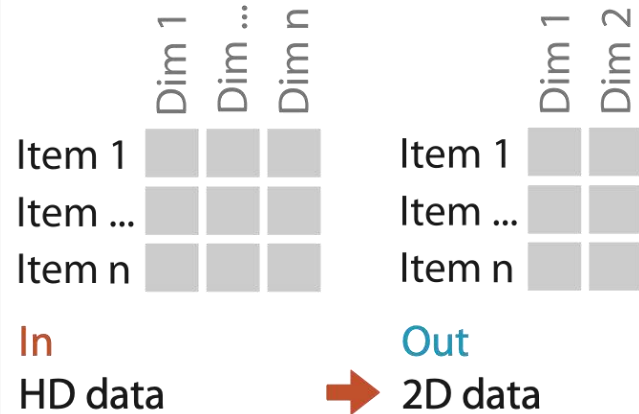
X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

Matejka, Justin, and George Fitzmaurice. "Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing." *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2017.

Aggregation = dimensionality reduction

- Derive a low-dimensional target space from a high-dimensional measurement space
- Goal: Capture maximum variance with minimum error

Task 1



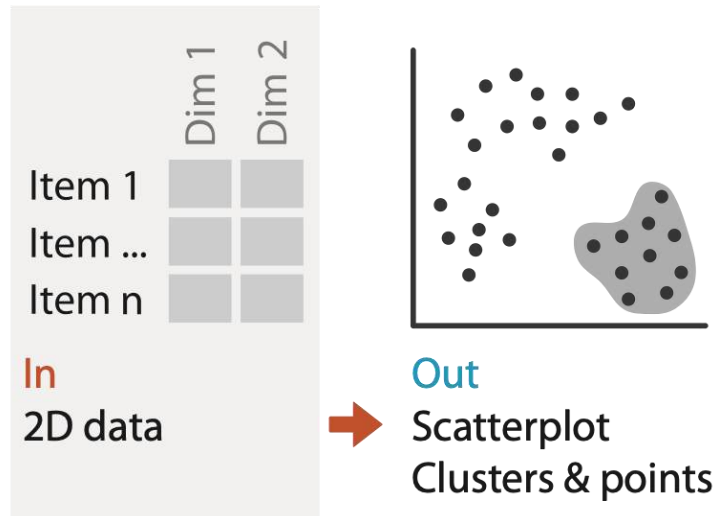
What?

- ➔ In High-dimensional data
- ➔ Out 2D data

Why?

- ➔ Produce
- ➔ Derive

Task 2



What?

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

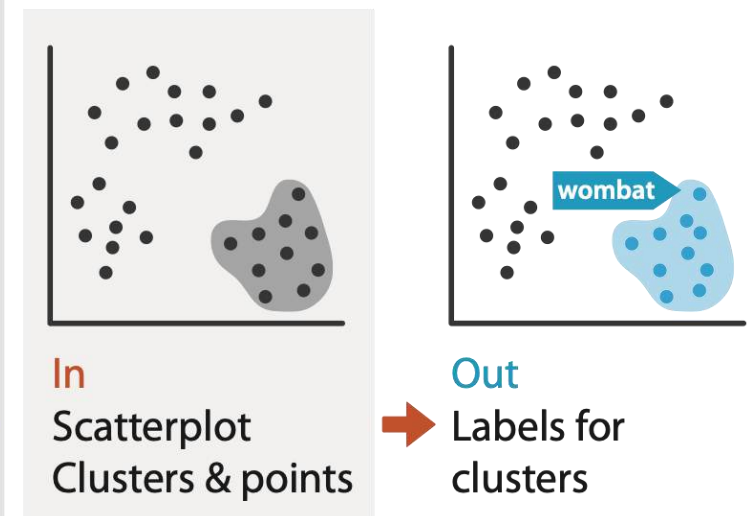
Why?

- ➔ Discover
- ➔ Explore
- ➔ Identify

How?

- ➔ Encode
- ➔ Navigate
- ➔ Select

Task 3



What?

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

Why?

- ➔ Produce
- ➔ Annotate

Not mutually exclusive

- A visualization can combine filtering and aggregation
- It can also combine reduction, manipulation, and faceting
- Your responsibility to use them as you see fit



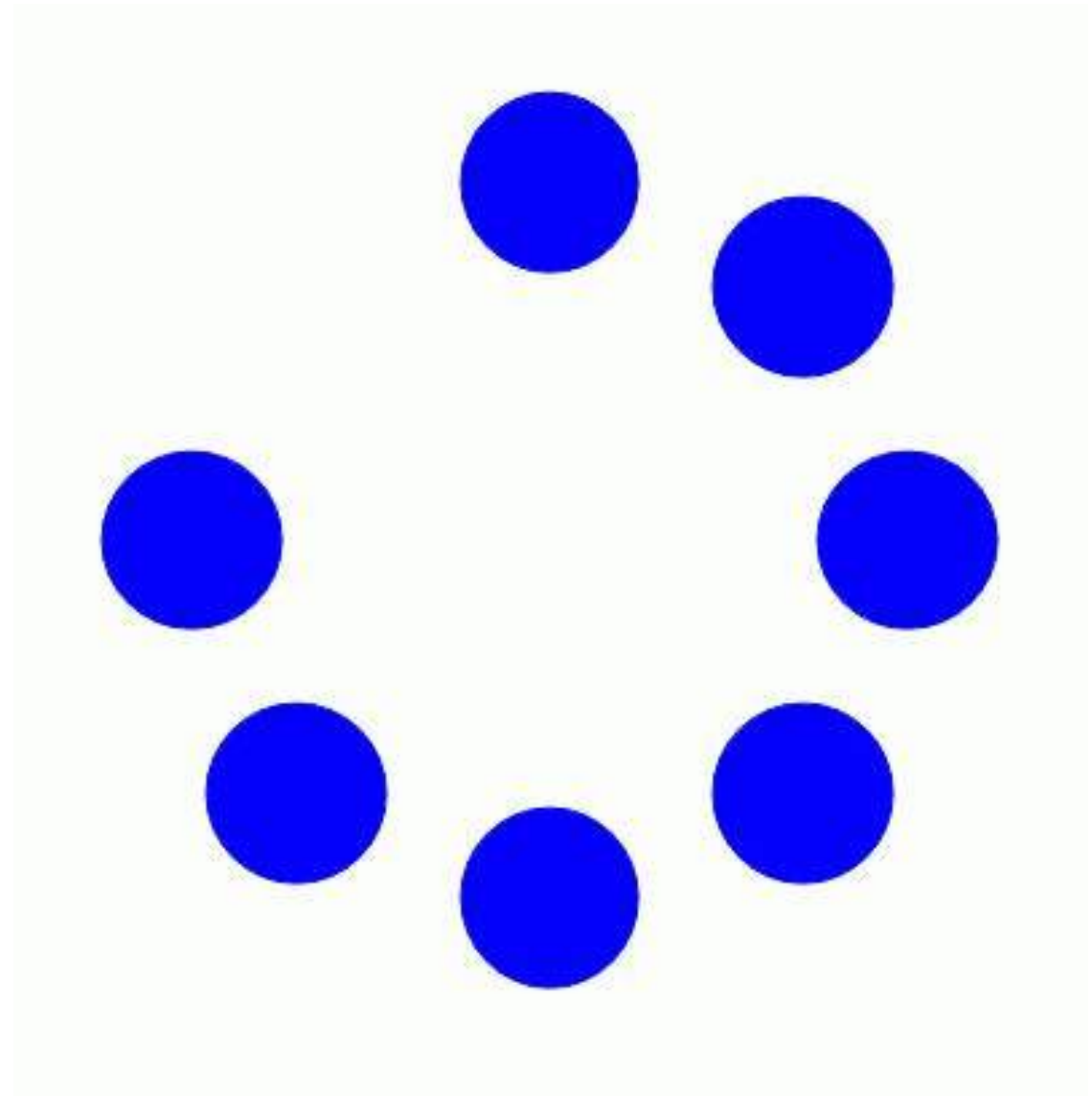
Animation

Why use motion?

- Visual variable to encode data
- Direct attention
- Understand system dynamics
- Understand state transition
- Increase engagement

Perceiving animation

- Motion is perceived at about ~ 10 frames/sec (100 ms).
 - But this does not have to be smooth motion! We can tell that frames are discrete yet still perceive movement.
- More generally, phi phenomenon

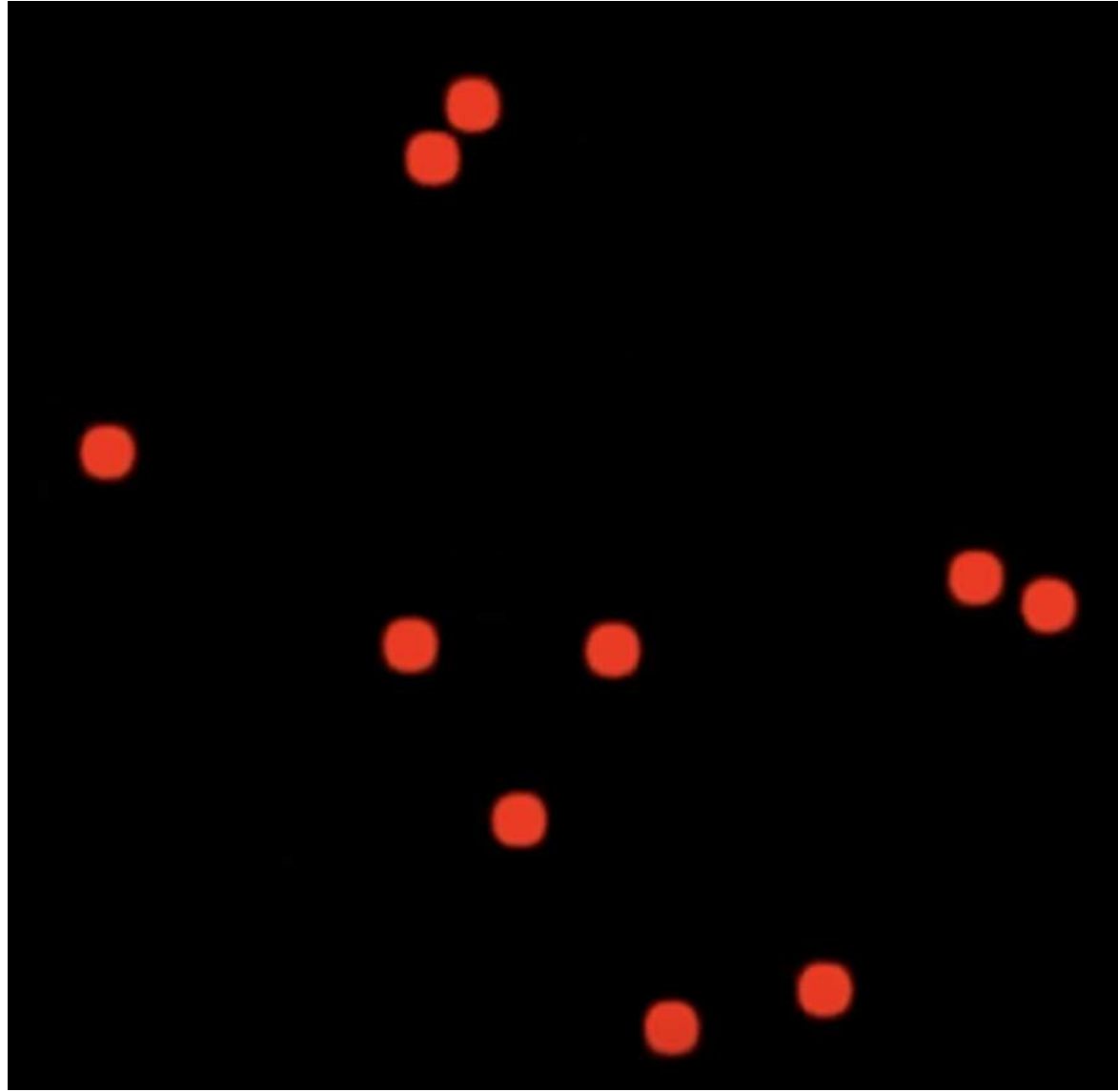


Motion as Visual Cue

- Pre-attentive, stronger than color, shape, ...
- More sensitive to motion at periphery. Why?
- Similar motions perceived as a group

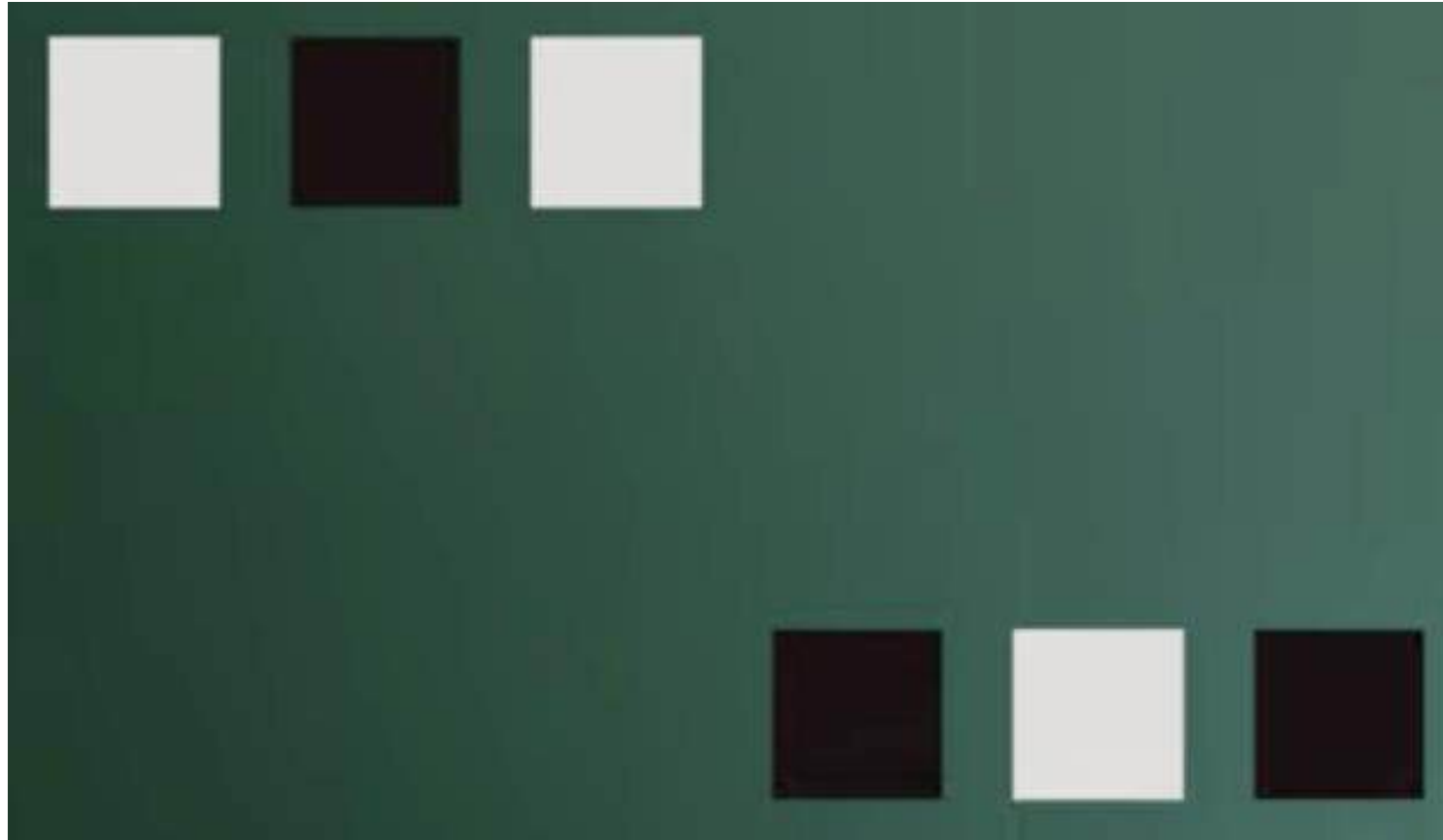
Multiple object tracking

- <https://www.youtube.com/watch?v=V3-HL7MgzzA>



Similar motion gets grouped together

- <https://youtu.be/WUfpM8pvhrk?t=66>



Animation pros and cons

Animation...	Helps...	Hurts...
Attention	Direct attention	Distraction
Constancy	Track changes	False relations
Causality	Cause and effect	False agency
Engagement	Increase interest	Too slow: boring Too fast: errors



Break