Data Exploration & Visualization

Module 8

Coordinated Multiple Views

Dr. ZENG Wei

DSAA 5024

The Hong Kong University of Science and Technology (Guangzhou)

Foreword

- If you learn nothing from this class, just learn to build Coordinated Multi-Views (CMVs)
- If you are good at building these CMV systems, you will have a job
  - at least for your group project

Interaction taxonomy

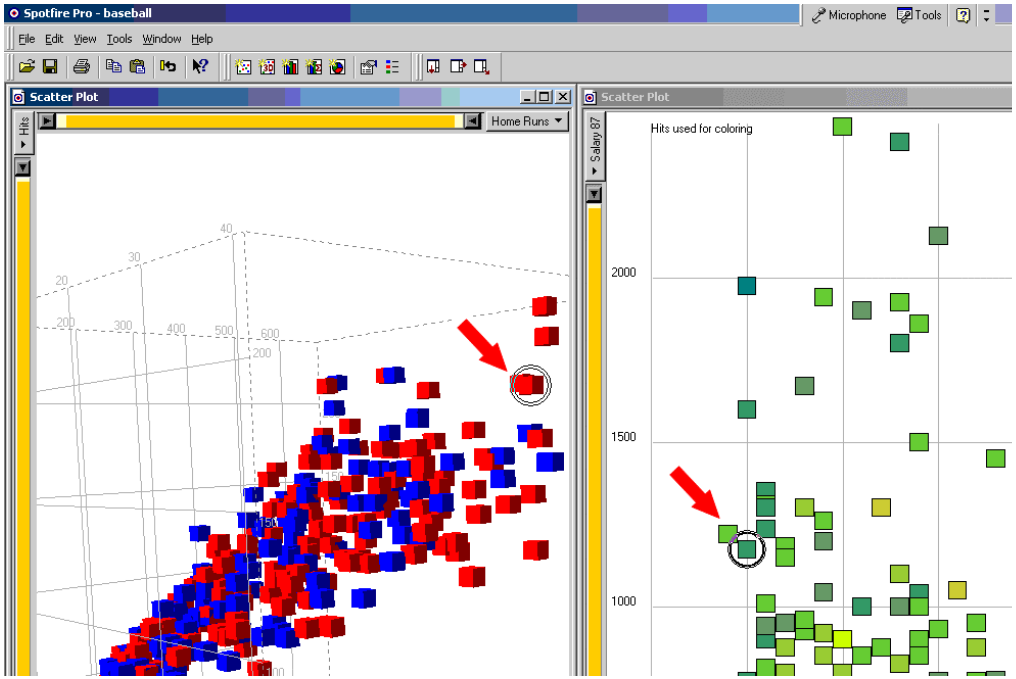
- Last week we discussed interactions in visualization and visual analytics
  1. Select:
     - Mark something as interesting
  2. Explore:
     - Show me something different
  3. Reconfigure:
     - Show me a different arrangement
  4. Encode:
     - Show me a different representation
  5. Abstract/Elaborate:
     - Show me more or less detail
  6. Filter:
     - Show me something conditionally
  7. Connect:
     - Show me related items

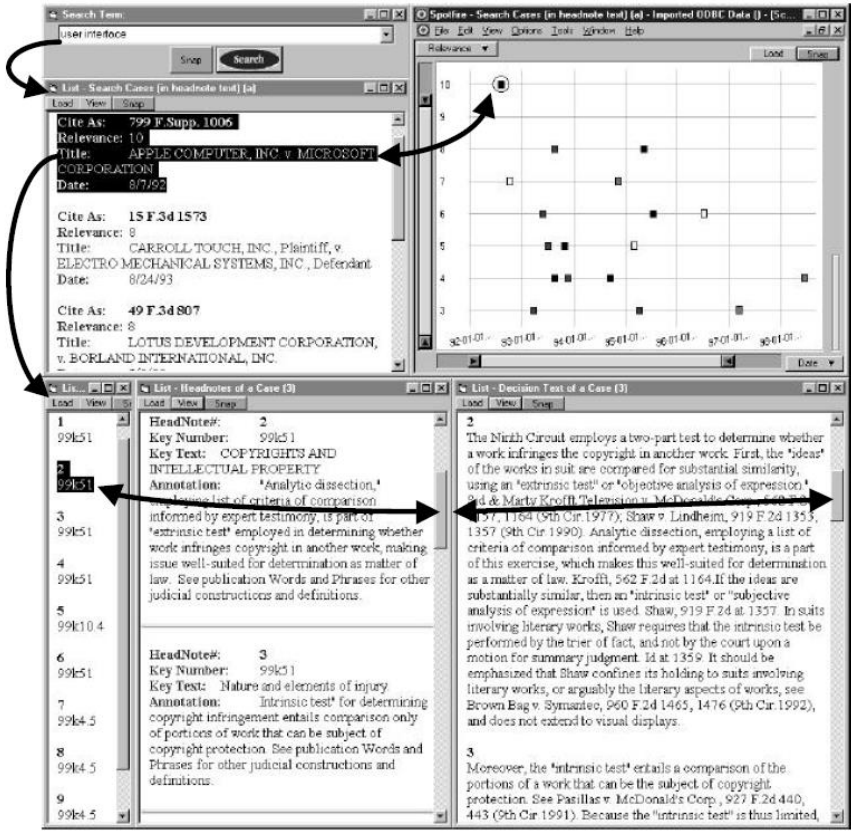
Connect

- Show me related items
  - Brushing and Linking
  - Coordinated Multiple Views (CMV)

Connect

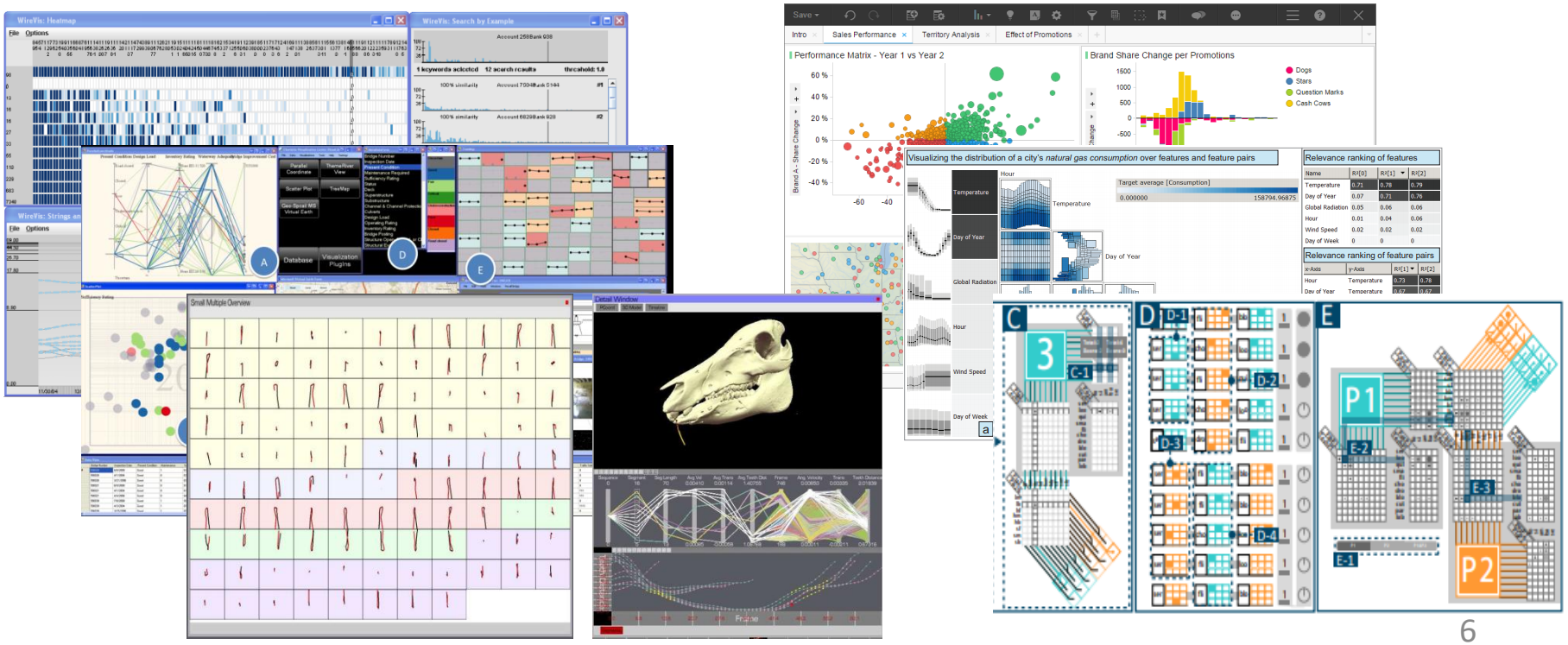
- Show me related items
  - Spotfire
  - Snap-Together Visualization





Coordinated Multi-Views (CMVs)

- Coordinated Multi-Views (CMV) is the “bread and butter” of visual analytics.
  - In fact, most visual analytics systems are CMVs…



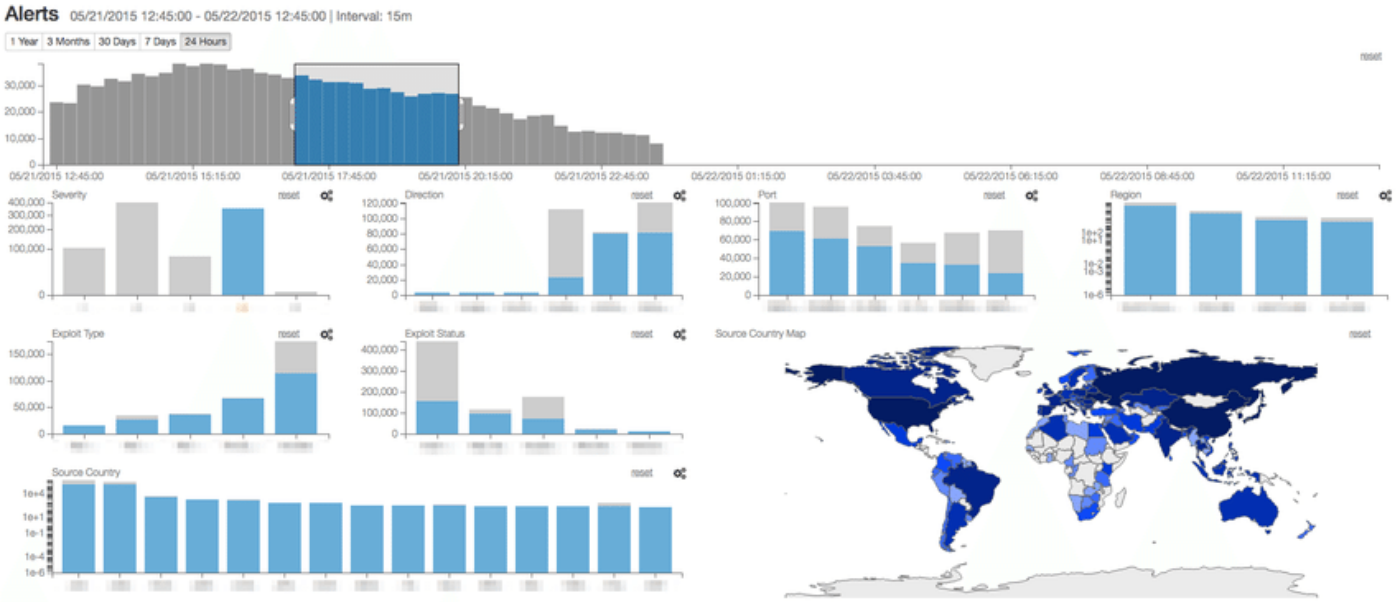
Coordinated Multi-Views (CMVs)

- As the name suggests, CMV is a visual analytics system that ***juxtapose*** multiple views that interoperates through a user's interactions
  - Sometimes referred to as "Multiple Coordinated Views", or…
  - Multiple "linked views"

- Regardless of the naming, in a CMV, a user's interactions with one of the views will trigger other views to respond

Coordinated Multi-Views (CMVs)

- Consider the following simple example:
  - A user highlights time range of events
  - The other views update to only show information about those selected events



Data Exploration & Visualization

Module 8: Coordinated Multiple Views

- Type of coordination
  - Multiple perspective, Overview + detail, comparison, controller + dependent, small multiples
- Interaction techniques
  - Consistent objects, ranges, related objects
- Models of communication between views
  - Binary marking, push-based notification, bounded variables
- Design consideration
  - Consistency
  - Composition and Configuration

Type of Coordination

- How are the views connected to each other?

- There are many ways in which visualizations can be coordinated:
  1. Multiple perspective
  2. Overview + detail
  3. Comparison
  4. Controller + Dependent
  5. Small multiples

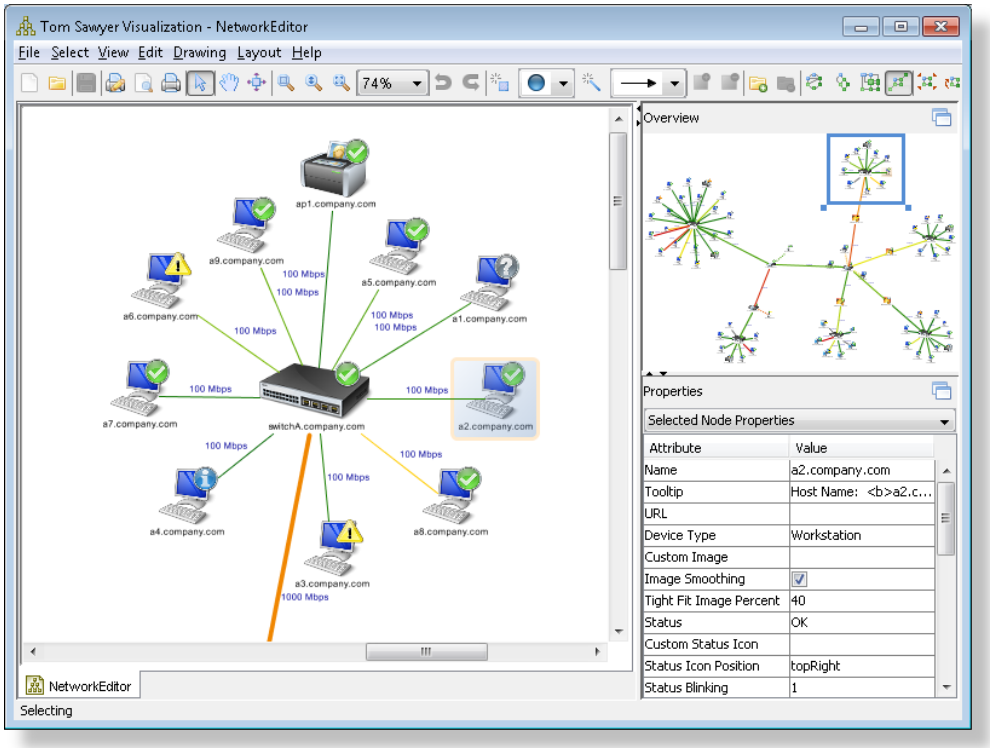
Adopted from Roberts, Jonathan C. (2007) State of the Art: Coordinated & Multiple Views in Exploratory Visualization.

Multiple perspective

- The most common CMV coordination technique
  - Each view encodes a different aspects (or dimensions) of data
  - Selection of an object in one view results in other views highlighting the same object

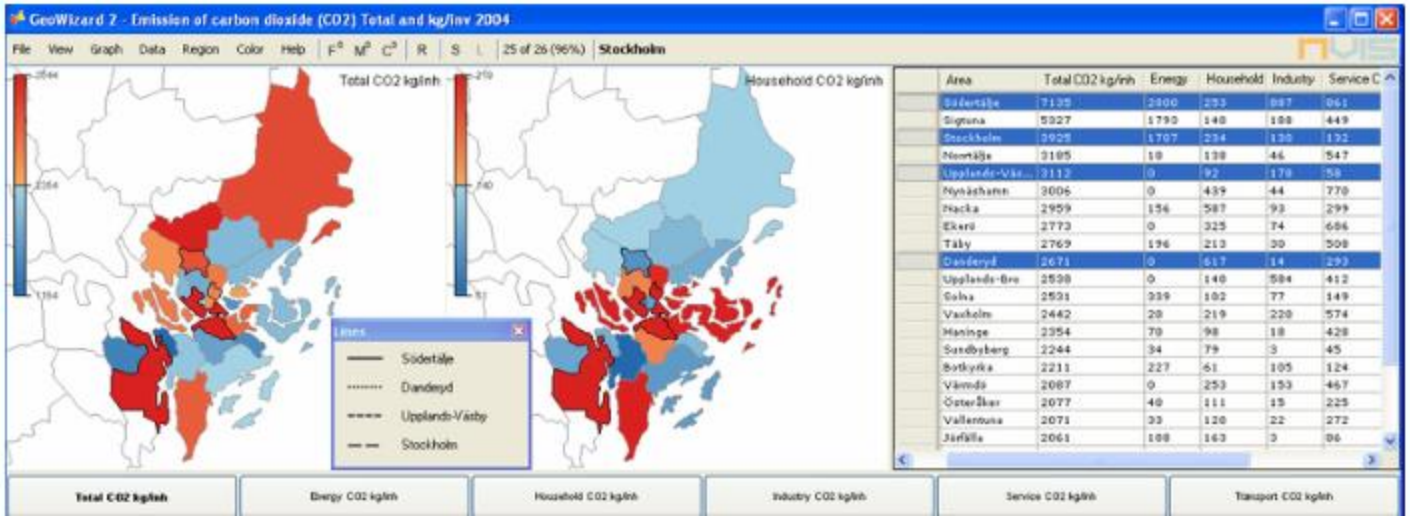
Overview + detail

- The upper right hand view is the “overview” of the entire dataset
- The main view is a detail view that allows for closer inspection
  - Note that unlike “multiple perspective”, the two views show the same visualization (just at different zoom levels)



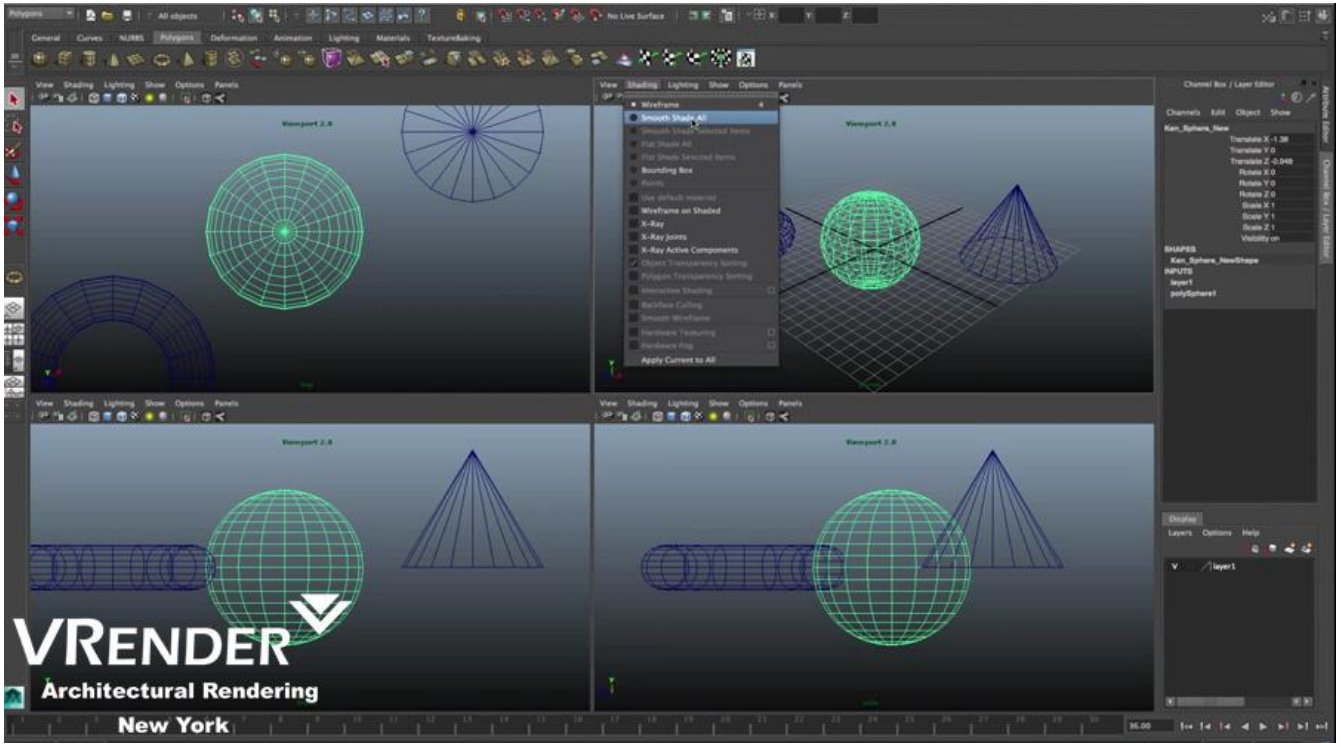
Comparison

- The views show different datasets to allow for comparison
  - … Or the same dataset in a different context (e.g. different year)
  - Unlike the previous cases, how to respond to the selection in one view can be ambiguous



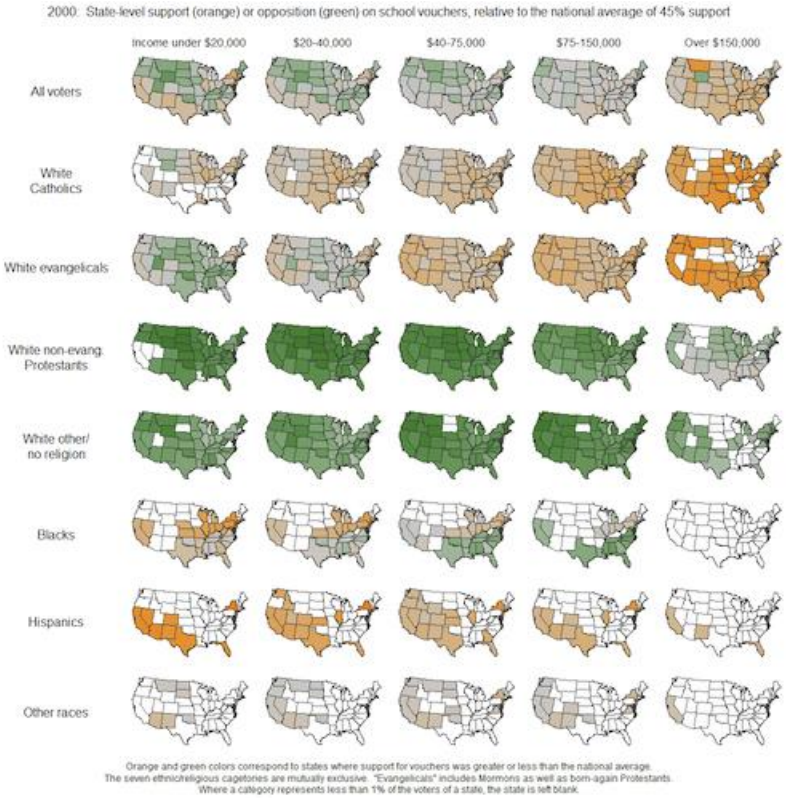
Controller + Dependent

- The multiple views show the exact same scene (from different camera angles)
  - When a user zooms into one view, all views zoom in



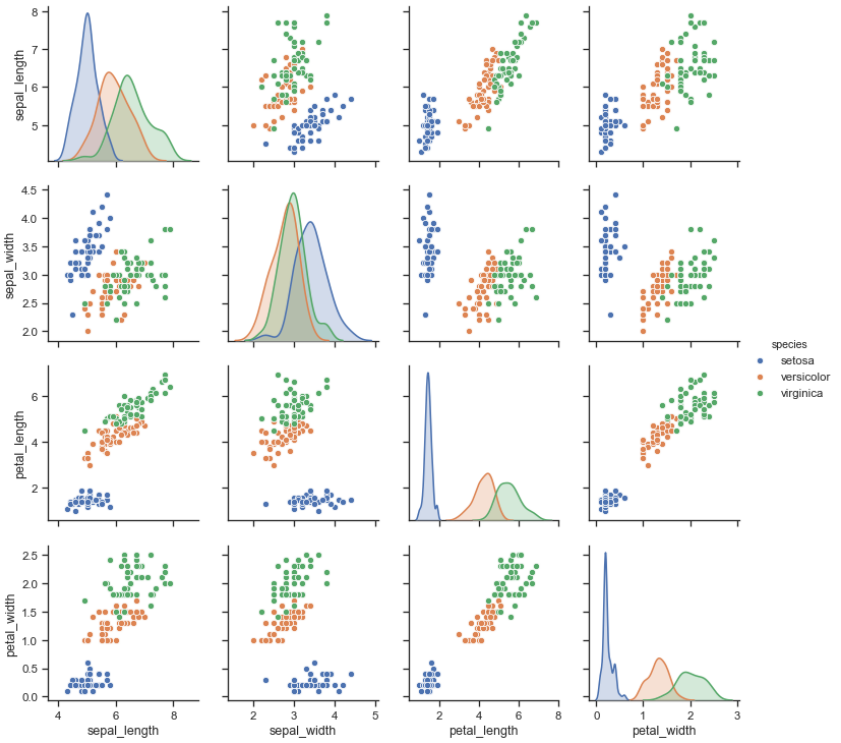
Small Multiples

- Small multiples can be seen as a special case of some of the previous types of coordination
  - The challenge is in the large number of views
- In the simple case, we have a special case of "comparison"



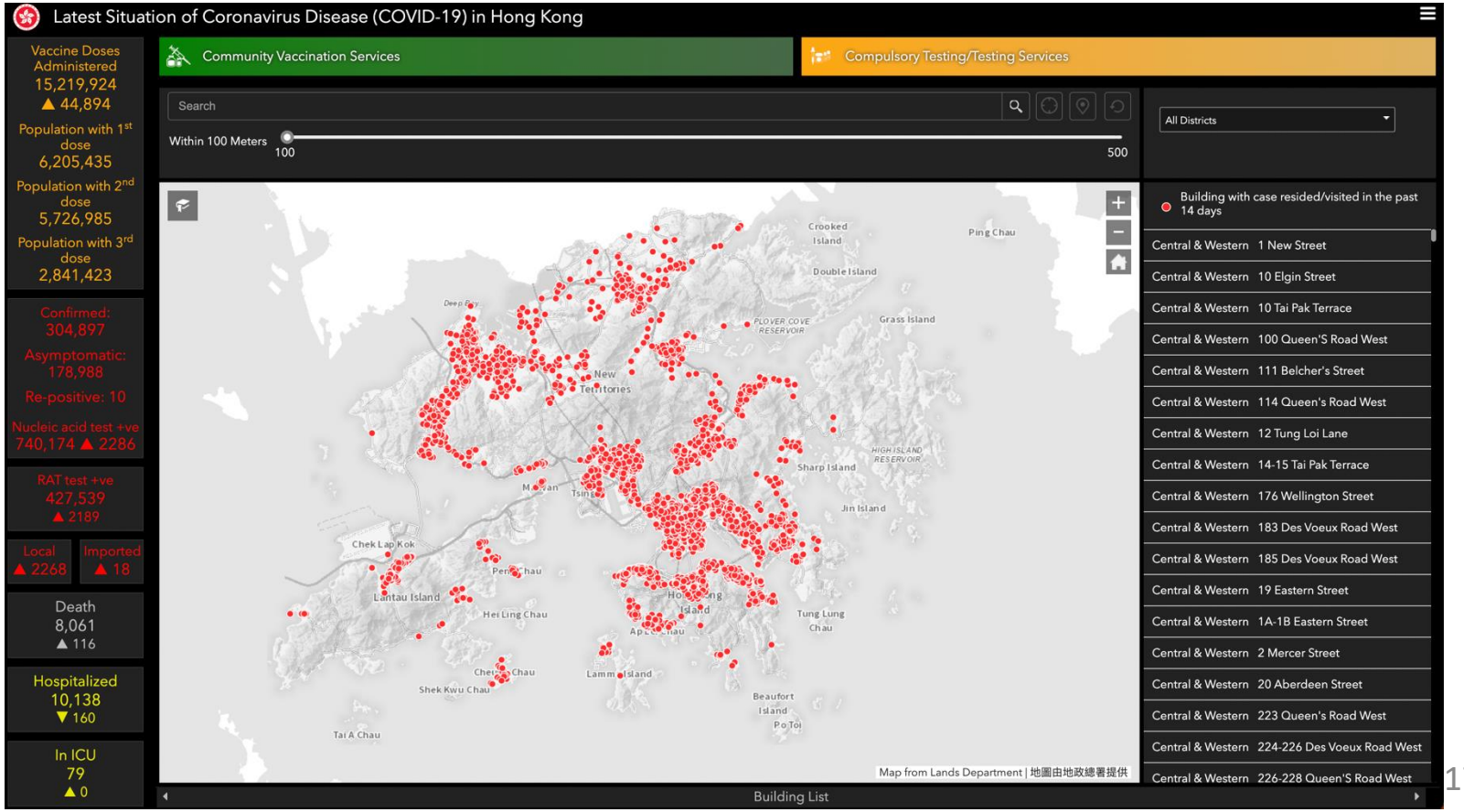
Small Multiples

- In a more complicated case (e.g. a scatterplot matrix), each view represents the same data, same visualization technique, but different x-y axes
  - Diagonals can be complicated…
- When interacting with small multiples, the coordination can be like the “Controller + Dependent” paradigm
  - Panning in one of the scatterplots also pans the other
  - https://altair-viz.github.io/gallery/scatter_matrix.html



In-class exercise

- What is the type of coordination?
  - Covid-19 dashboard HK



Data Exploration & Visualization

Module 8: Coordinated Multiple Views

- Type of coordination
  - Multiple perspective, Overview + detail, comparison, controller + dependent, small multiples
- Interaction techniques
  - Consistent objects, ranges, related objects
- Models of communication between views
  - Binary marking, push-based notification, bounded variables
- Design consideration
  - Consistency
  - Composition and Configuration

Type of coordination: Summary (part I)

- These five examples of coordination techniques are useful, but they also highlight some challenges

1. The CMV might be showing:
   - The same dataset
   - Different parts of the same dataset
   - Different datasets
     - e.g. in comparison
   - Different parts of different datasets
     - e.g. comparison + small multiples

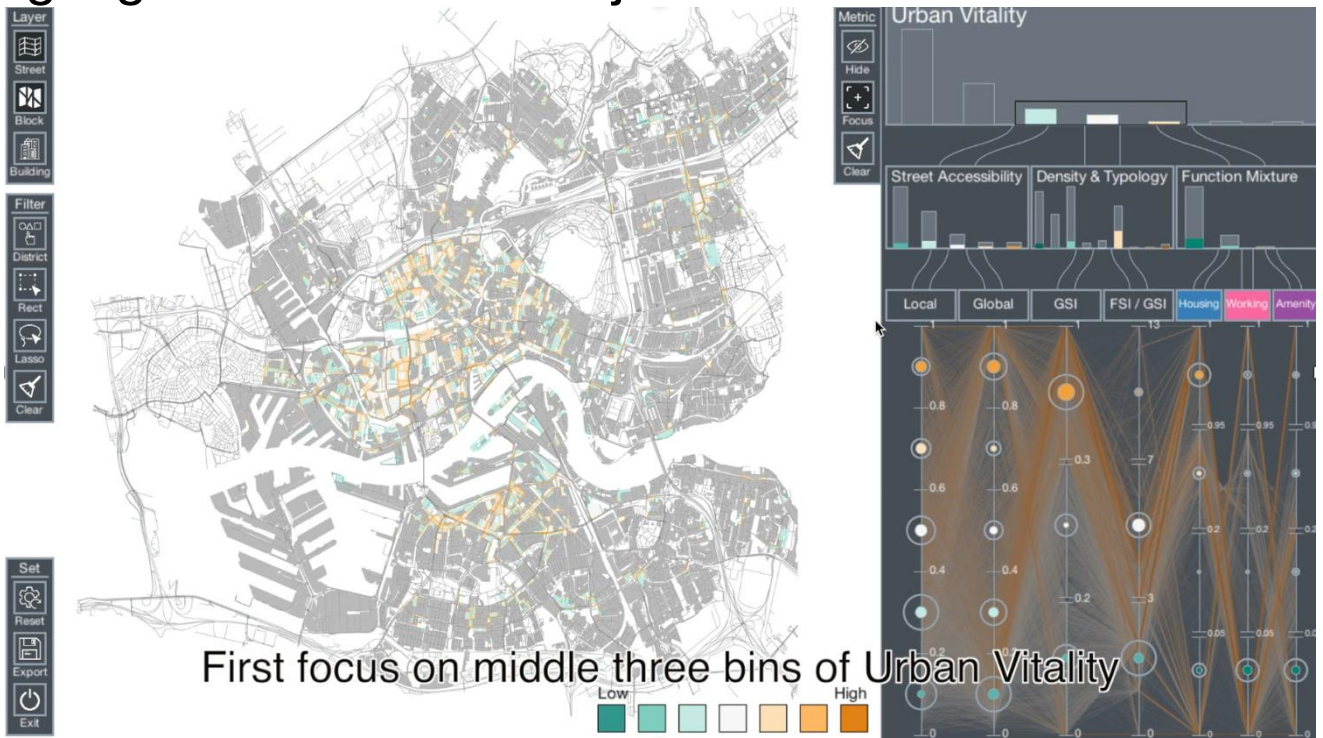
Type of coordination: Summary (part II)

- These five examples of coordination techniques are useful, but they also highlight some challenges

2. The user's selection can manifest in different ways. A user can select:
   - Consistent Objects
     - select the same object that appears in different views
   - Ranges
     - select a range (e.g. a range in time)
   - Related Objects
     - Different views show different objects, but the objects are related in some ways

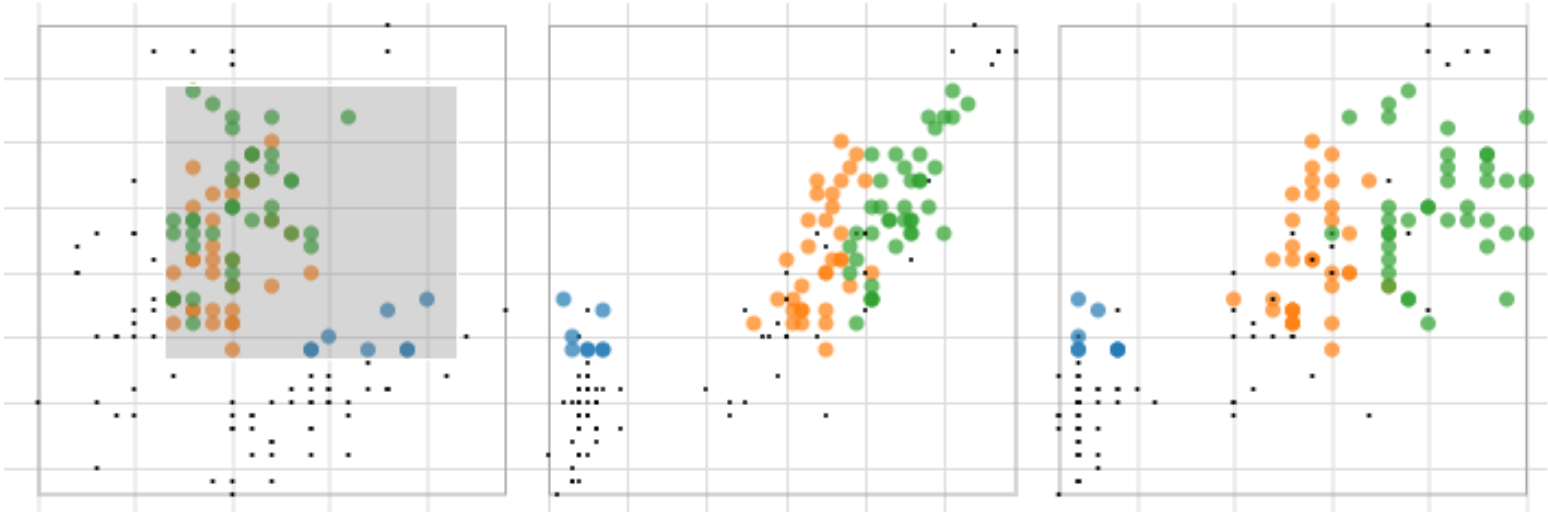
User selection: Consistent Objects

- In many cases, the same data element appears in all the views
- Response to user interaction is simple:
  - Highlight the selected object in all views



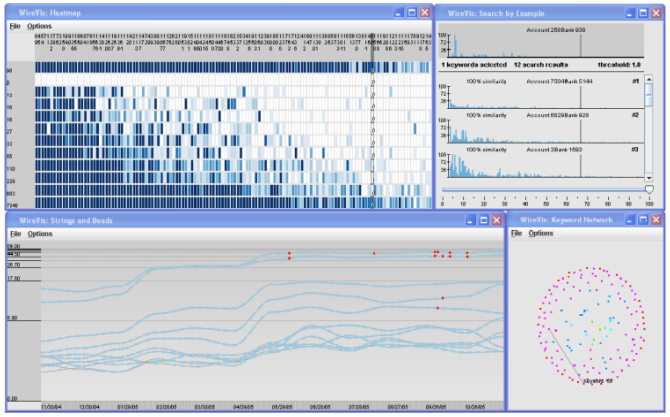
User selection: Ranges

- In the Scatterplot Matrix example, the user can select ranges in the scatterplot
  - Instead of selecting specific data items, the user can drag this range-box around
    - https://observablehq.com/@d3/brushable-scatterplot-matrix
  - Objects are only highlighted if they fit within the bounds of this range-box



User selection: Related Objects

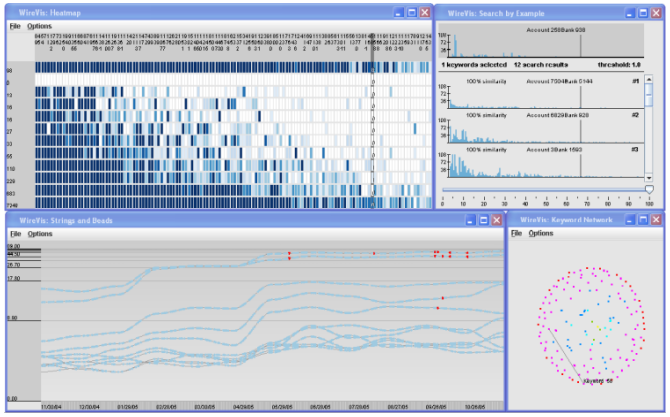
- In complex CMV systems, an object selected in one view does not immediately correspond to another in a different view



- The design of the WireVis system uses 3 types of data
  - Accounts
    - An account contains one or more transactions
  - Transactions
    - Transactions contains date, amount, and “keyword”
    - A transaction is between two accounts
  - Keywords
    - Each transaction is coded with the keywords that describe them (e.g. country name, types of business, commodity, etc.)

User selection: Related Objects

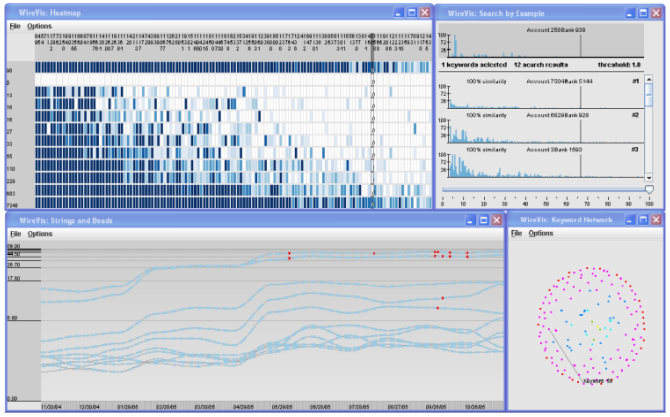
- In complex CMV systems, an object selected in one view does not immediately correspond to another in a different view



- The WireVis system has 4 views
  - Account-Keyword view (upper left)
    - As a heatmap, shows how clusters of accounts (rows) based on their frequency of the appearance of keywords (columns)
  - Transaction-Time view (lower left)
    - For each cluster of account (each row in previous view is shown as a curvy-line), this view shows the total amount over time
  - Search by Example view (upper right)
    - Find similar accounts based on their use of keywords
  - Keyword relation view (lower right)
    - A force-directed layout to show the co-occurrences of keywords. The closer the keywords appear to each other, the more common they appear together in transactions

User selection: Related Objects

- In complex CMV systems, an object selected in one view does not immediately correspond to another in a different view



- Take the heatmap view in WireVis for example
  - A cell in the heatmap corresponds to the transactions relating to a particular keyword
  - No view in WireVis shows a transaction
- In this sense, to update the other views, a system needs to identify the relations between the objects in the views.
  - For example, each dot in the temporal view will highlight itself if it contains one of the transactions highlighted in the heatmap

Data Exploration & Visualization

Module 8: Coordinated Multiple Views

- Type of coordination
  - Multiple perspective, Overview + detail, comparison, controller + dependent, small multiples
- Interaction techniques
  - Consistent objects, ranges, related objects
- Models of interaction
  - Binary marking, push-based notification, bounded variables
- Design consideration
  - Consistency
  - Composition and Configuration

Model of interactions

- Models of communication between views
  - When a view is updated, how does the update propagate to the others?

- There are many ways to do this, from simple to complex in implementation
  1. Binary marking of selected objects
  2. Push-based Notifications
  3. Bounded variables and functions

Model of interactions: Binary marking

- If you have a simple CMV where each view shows the same data items across views (*Consistent Objects*)
  - The easiest coordination mechanism is to “mark” each data element
    - This can be done by adding a new column (of type binary) in your dataset (called “selected”)
  - A user’s interaction marks the data elements as selected (or not).
    - Each view renders the dataset by first seeing which data items are “marked”
    - If it’s marked, draw it red; otherwise draw it with the normal color

Model of interactions: Binary marking

- PROS:
  - Easy to implement

- CONS:
  - This works in the basic cases, but fails when:
    - The relations between the views are complex (e.g. in the case of WireVis)

Model of interactions: Push-based notification

- A more robust approach is for each a to broadcast (i.e. push notifications) to all other views when a user interaction occurs
- In this paradigm, each view is responsible for how it wants to handle these notifications
  - For example, the timeline view in WireVis sends a notification of (time {11/12/2019, 12/31/2019}) to indicate that a user has selected all dates from now till end of the year
  - In the case of the “keyword relation view”, it doesn’t care about time, so it just ignores the notification

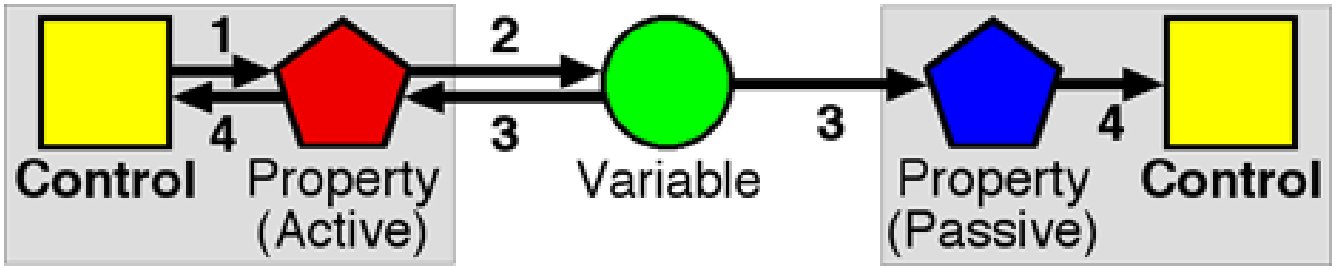
Model of interactions: Push-based notification

- PROS:
  - Robust to handling different types of messages
  - Design of each view is modular from the other

- CONS:
  - Adding a new view means implementing handlers for all types of notifications
    - If there's a new type of notification, all views need to be modified
  - There are potentially n^2 relations (n is the number of views) if each view has its own unique type of notification
  - Each view might need to repeat some expensive operation (e.g. linear scan of the entire dataset)

Model of interactions: Bounded variables

- To alleviate the problems of Push-Based Notification, one can abstract the types of notifications to bounded variables or functions



- The idea is that instead of pushing notifications, each view is observing a bounded variable (the green circle).
  - This bounded variable can be an attribute in the data,
  - or, a derived variable,
  - or, a function over a range of variables

Model of interactions: Bounded variables

- PROS:
  - The use of observables makes this approach more efficient
  - It can support a large number of coordination mechanisms without explicit programming for each

- CONS:
  - Complicated and difficult to implement
  - Depending on implementation, it might break the “model-view-controller” paradigm (in that the view cannot be separated from the model)

Data Exploration & Visualization

Module 8: Coordinated Multiple Views

- Type of coordination
  - Multiple perspective, Overview + detail, comparison, controller + dependent, small multiples
- Interaction techniques
  - Consistent objects, ranges, related objects
- Models of communication between views
  - Binary marking, push-based notification, bounded variables
- Design consideration
  - Consistency
  - Composition and Configuration

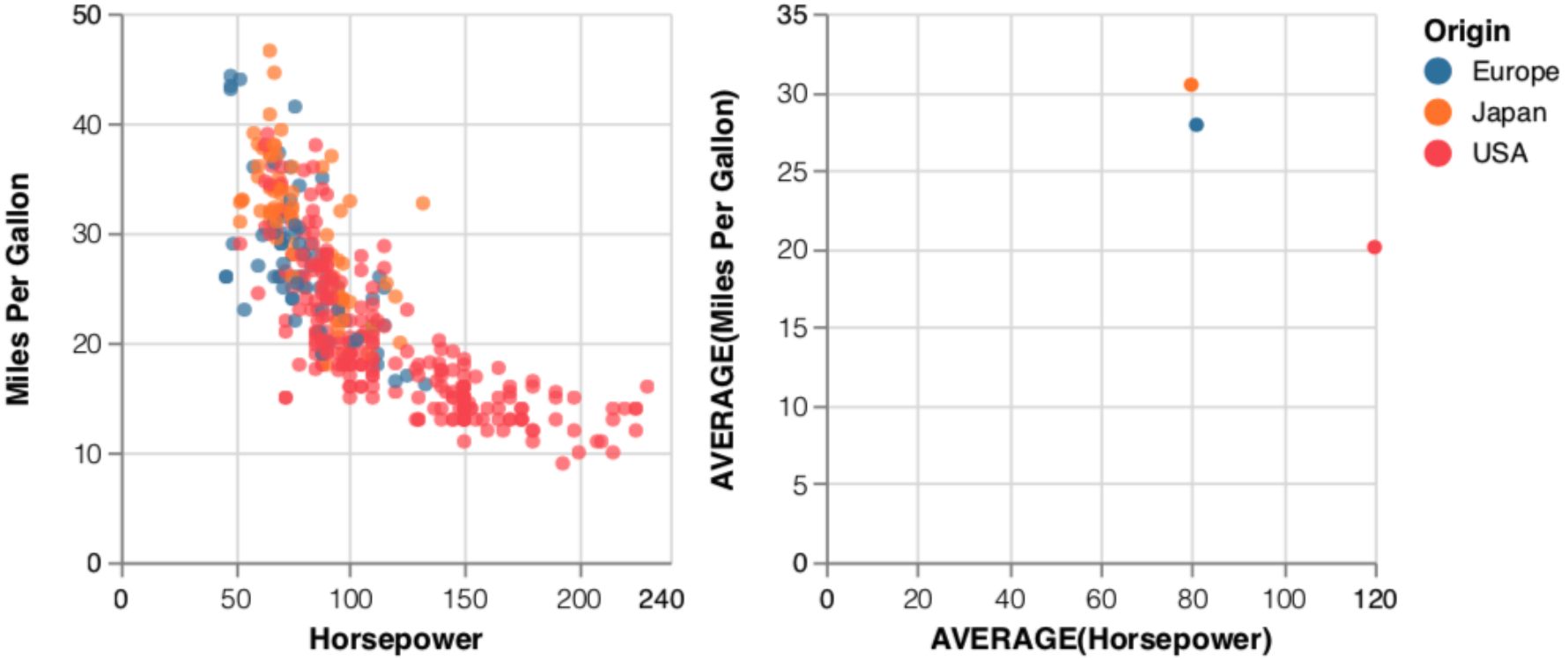
Design consideration: Consistency

- A specific visualization technique that composites multiple views of different *view types* in the *display space*.
  - **View types**: bar chart, line chart, pie chart, scatterplot, etc.
  - **Display space:** mobile phones, desktop displays, video walls

- Maintain a consistent design
  - Consistent appearance puts emphasis on data, not the visual design
  - Changes in design can distract from irregularities in the data

Design consideration: Consistency

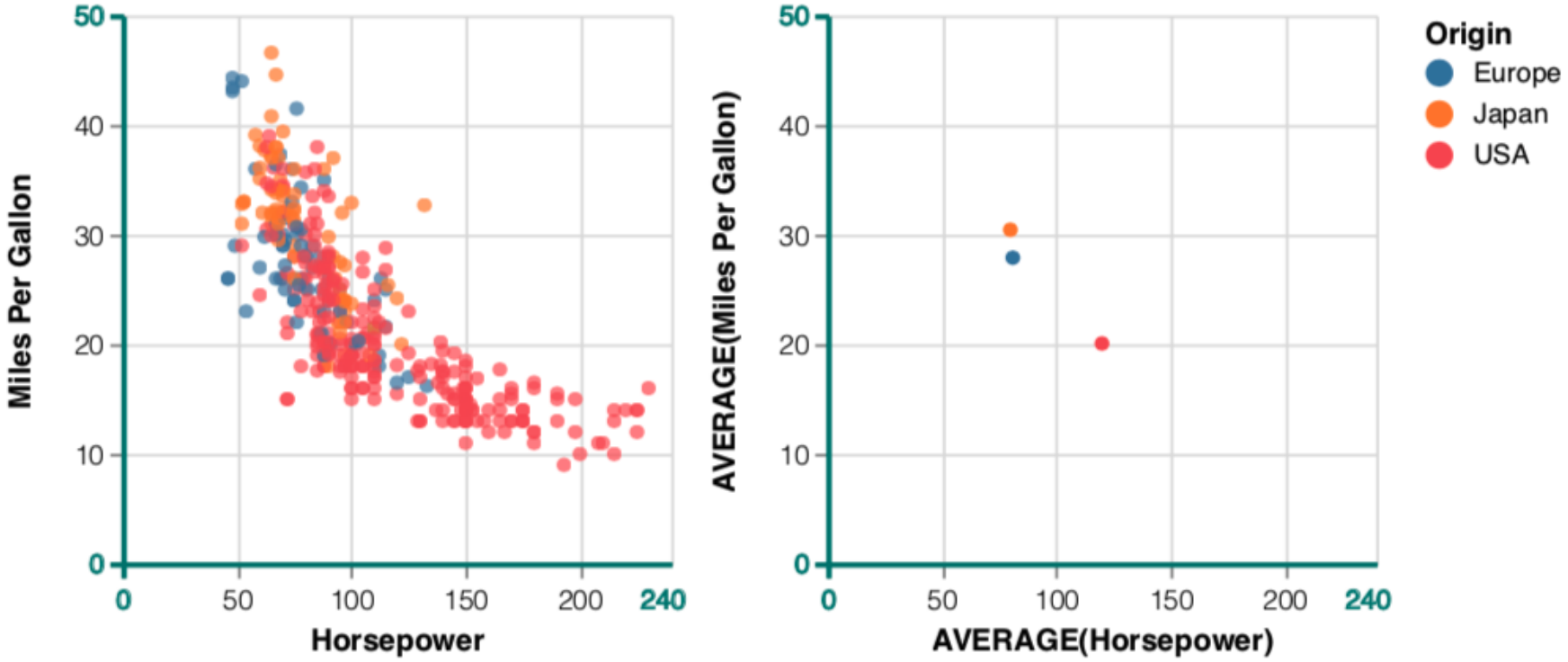
- Multiple inconsistent views



Credit: Qu and Hullman

Design consideration : Consistency

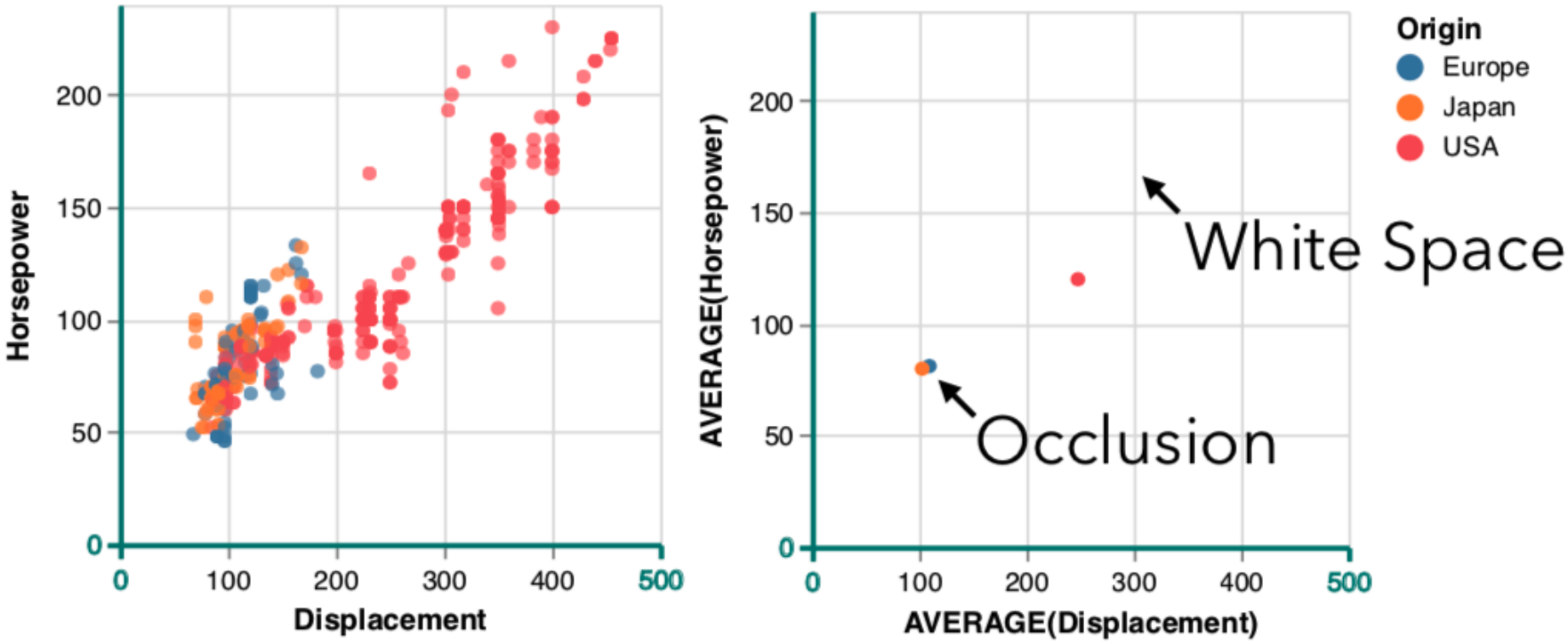
- Having consistency can help audience see view relations, avoid misinterpretations...



Credit: Qu and Hullman

Design consideration: Consistency

- Multiple Views Consistency vs. Single View Effectiveness



Credit: Qu and Hullman

Design consideration: Consistency

- High level constraints

Same Rule

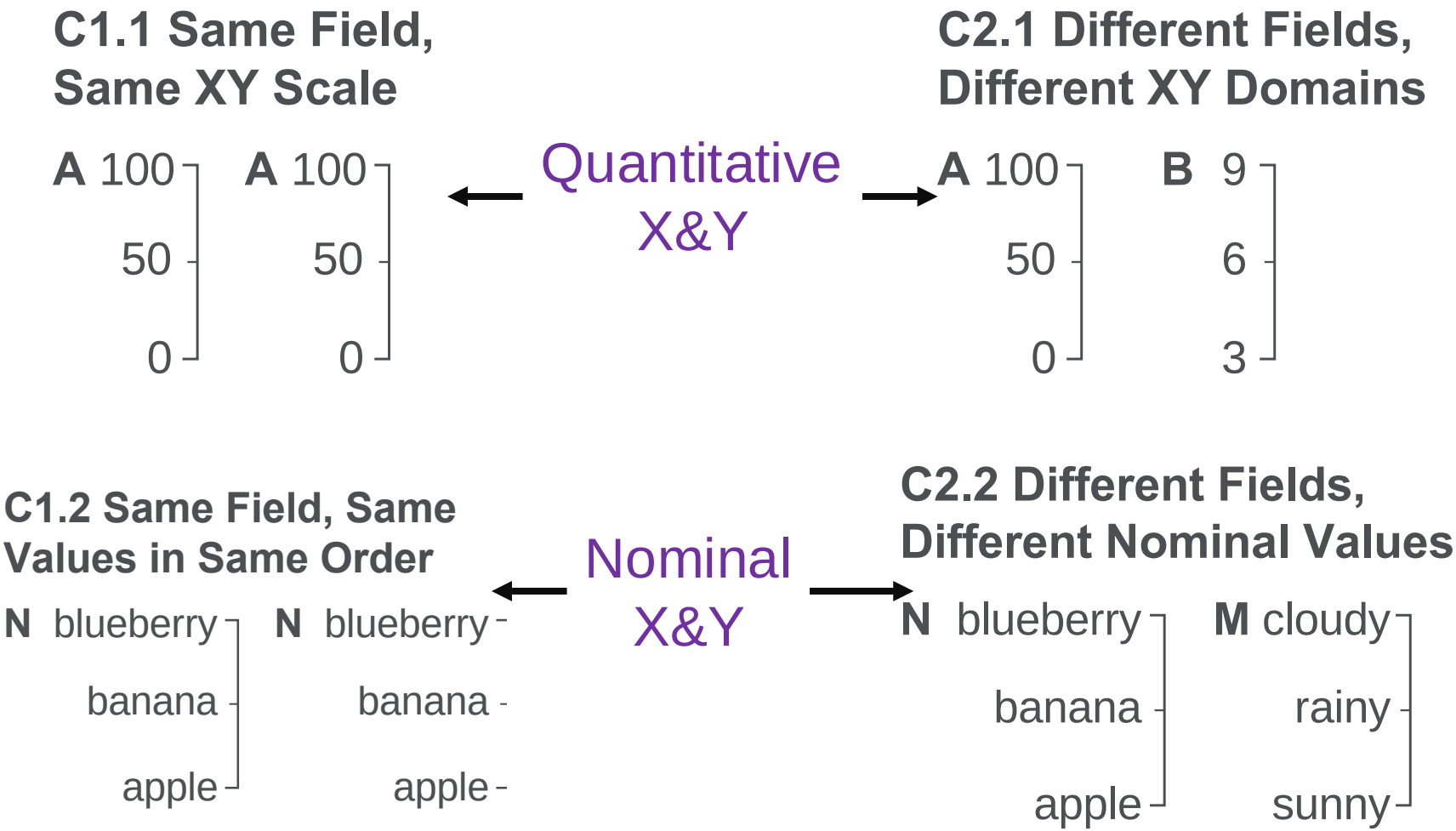
If two views contain the **same** data field, that field should be encoded in the **same** way

Different Rule

If two views contain **different** data fields, the two fields should be encoded **differently**

Credit: Qu and Hullman

Design consideration: Consistency

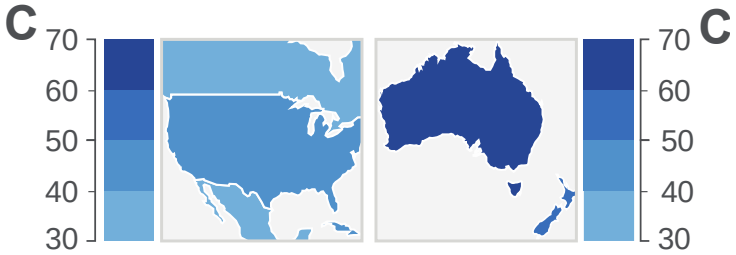


Credit: Qu and Hullman

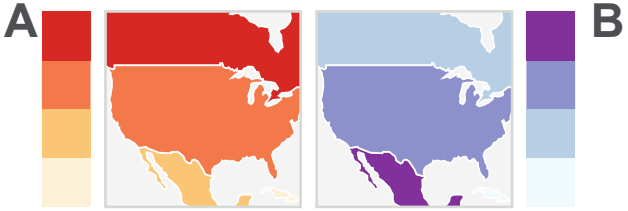
Design consideration: Consistency

Quantitative Colors

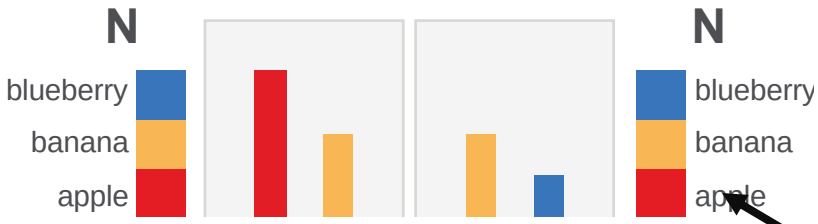
C1.3 Same Field, Same Quantitative Color Scale



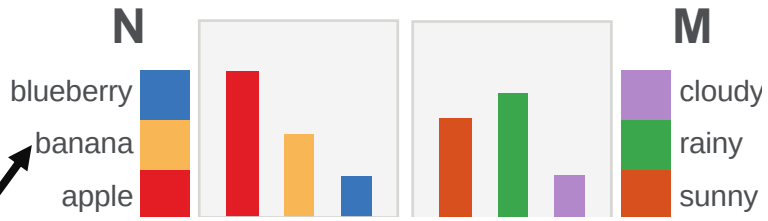
C2.3 Different Fields, Non-Overlapping Hues



C1.4 Same Field, Same Value-Color Mapping



C2.4 Different Fields, Non-overlapping Palettes



Nominal Colors

Credit: Qu and Hullman

Design consideration: Composition and configuration

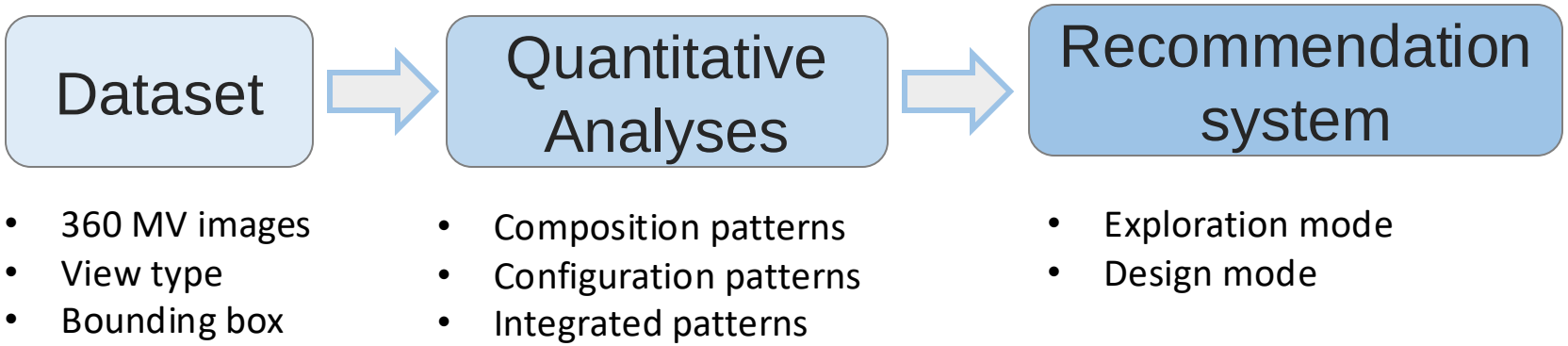
- The design of a CMV first depends on answering the first question of
  - ***What aspects of the data does the user need to see?***

- The following question is harder to think through, but can be just as relevant
  - What views, when combined, will give the user the biggest bang for the buck?
    - For example, a map view + a timeline view is classic because the combined view provides an integrated view of temporal geospatial information

Design consideration: Composition and configuration

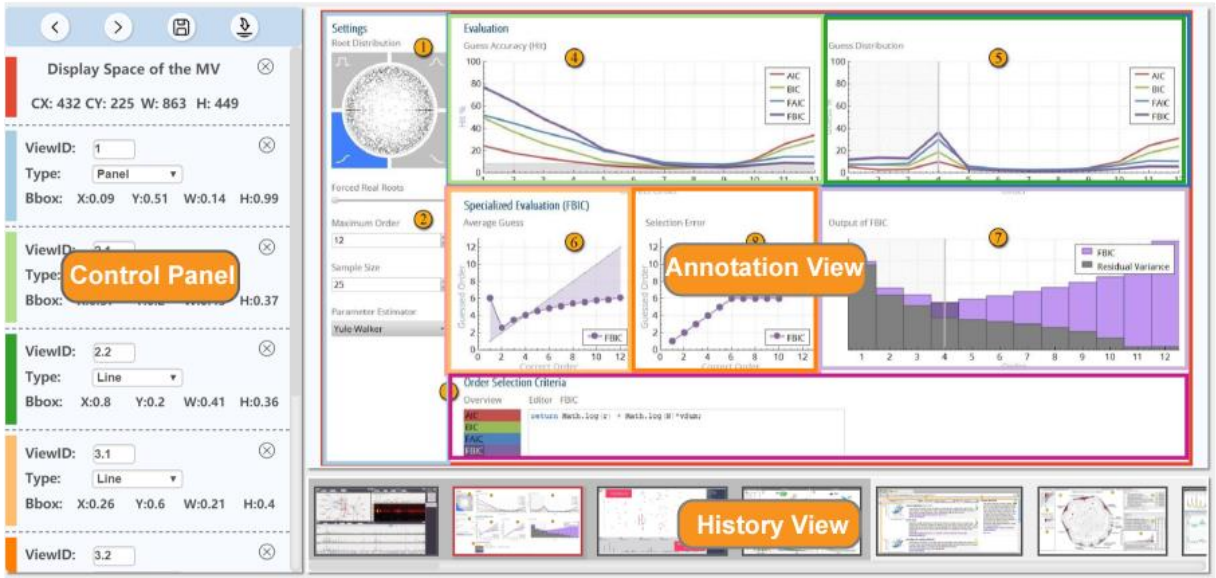
- What's the common practice in MV visualizations?
  - Which view types are frequently used together?
  - Where to position each view?
  - What is the size of each view?

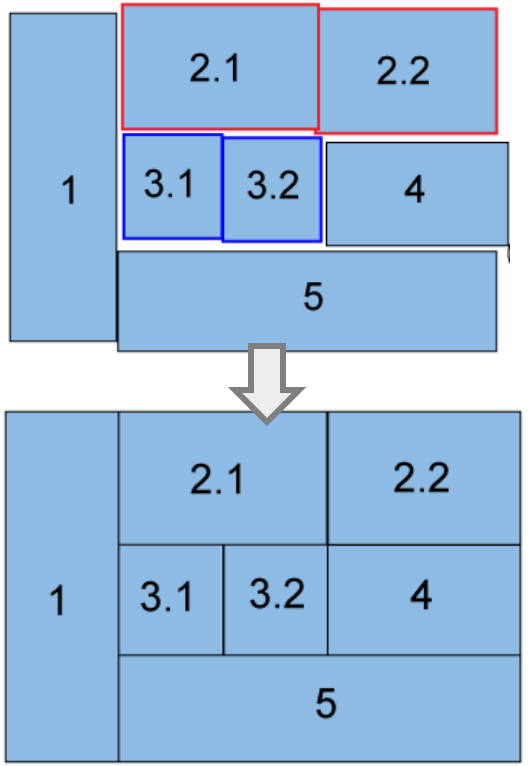
- To answer these questions, we conducted quantitative analyses on existing MV visualizations.



Design consideration: Composition and configuration

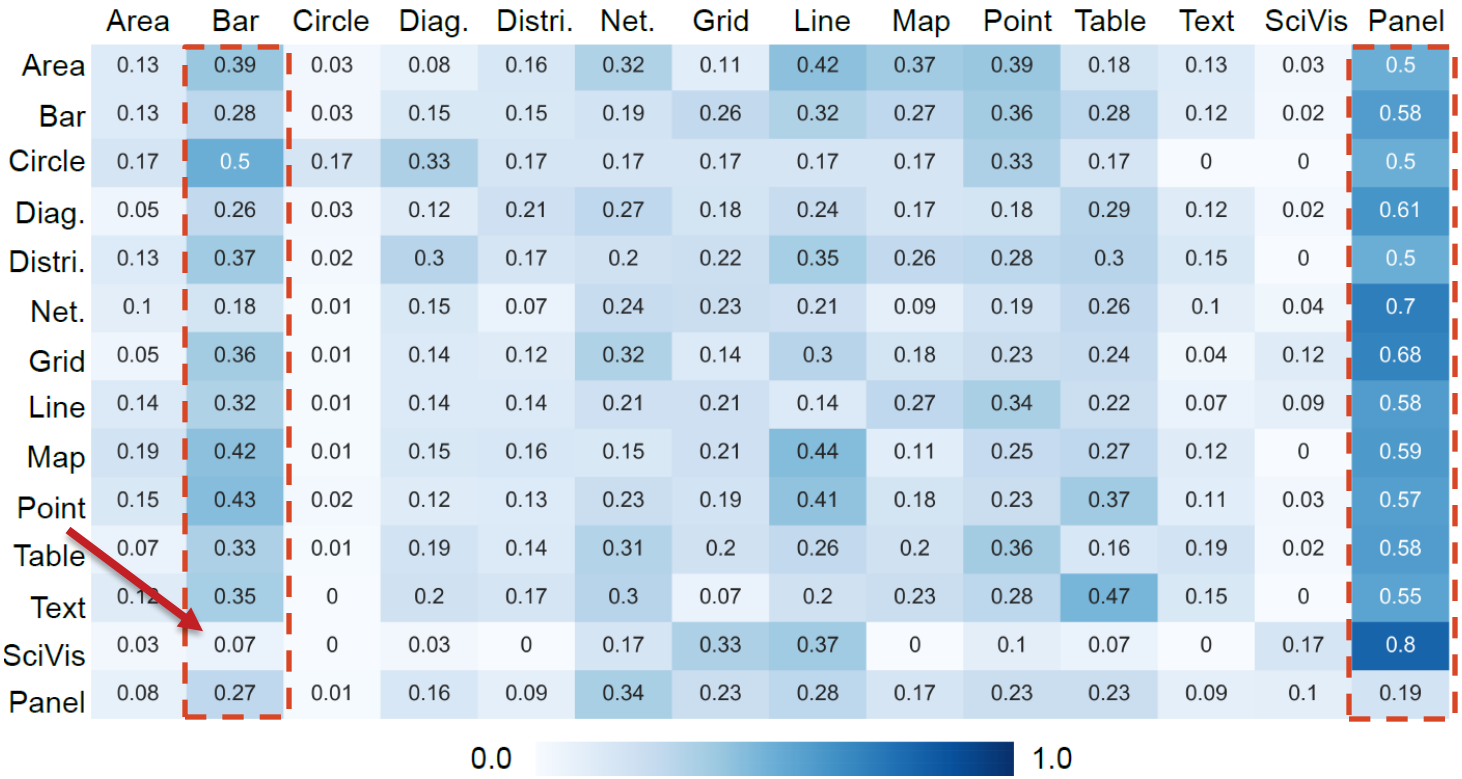
- Dataset Construction
  - Collect 360 images of multiple-view visualizations from IEEE VIS, EuroVis, and PacificVis publications 2011 to 2019.
  - Annotation tool: label attributes of types and bounding box of each view.
  - Layout Refinement





Design consideration: Composition and configuration

- Composition Pattern: Conditional probabilities of view types (columns) given other view types (rows) are employed.
  - Panel and bar are frequently used together with other view types.
  - Few SciVis visualizations incorporate Bar Charts

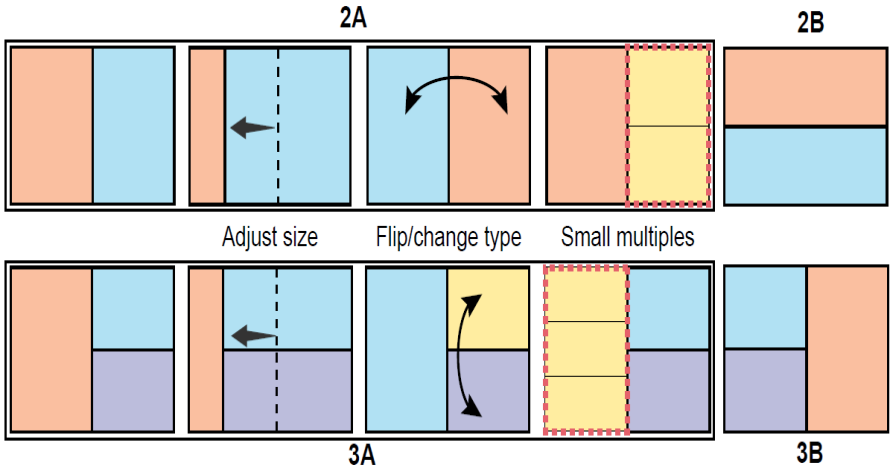


	Area	Bar	Circle	Diag.	Distri.	Net.	Grid	Line	Map	Point	Table	Text	SciVis	Panel
Area	0.13	0.39	0.03	0.08	0.16	0.32	0.11	0.42	0.37	0.39	0.18	0.13	0.03	0.5
Bar	0.13	0.28	0.03	0.15	0.15	0.19	0.26	0.32	0.27	0.36	0.28	0.12	0.02	0.58
Circle	0.17	0.5	0.17	0.33	0.17	0.17	0.17	0.17	0.17	0.33	0.17	0	0	0.5
Diag.	0.05	0.26	0.03	0.12	0.21	0.27	0.18	0.24	0.17	0.18	0.29	0.12	0.02	0.61
Distri.	0.13	0.37	0.02	0.3	0.17	0.2	0.22	0.35	0.26	0.28	0.3	0.15	0	0.5
Net.	0.1	0.18	0.01	0.15	0.07	0.24	0.23	0.21	0.09	0.19	0.26	0.1	0.04	0.7
Grid	0.05	0.36	0.01	0.14	0.12	0.32	0.14	0.3	0.18	0.23	0.24	0.04	0.12	0.68
Line	0.14	0.32	0.01	0.14	0.14	0.21	0.21	0.14	0.27	0.34	0.22	0.07	0.09	0.58
Map	0.19	0.42	0.01	0.15	0.16	0.15	0.21	0.44	0.11	0.25	0.27	0.12	0	0.59
Point	0.15	0.43	0.02	0.12	0.13	0.23	0.19	0.41	0.18	0.23	0.37	0.11	0.03	0.57
Table	0.07	0.33	0.01	0.19	0.14	0.31	0.2	0.26	0.2	0.36	0.16	0.19	0.02	0.58
Text	0.12	0.35	0	0.2	0.17	0.3	0.07	0.2	0.23	0.28	0.47	0.15	0	0.55
SciVis	0.03	0.07	0	0.03	0	0.17	0.33	0.37	0	0.1	0.07	0	0.17	0.8
Panel	0.08	0.27	0.01	0.16	0.09	0.34	0.23	0.28	0.17	0.23	0.23	0.09	0.1	0.19

0.0 1.0

Design consideration: Composition and configuration

- Configuration Pattern
  - Most multiple-view visualization designs adopt simple layouts.
  - 2A occupies higher percentages in InfoVis and SciVis than the other three conferences.



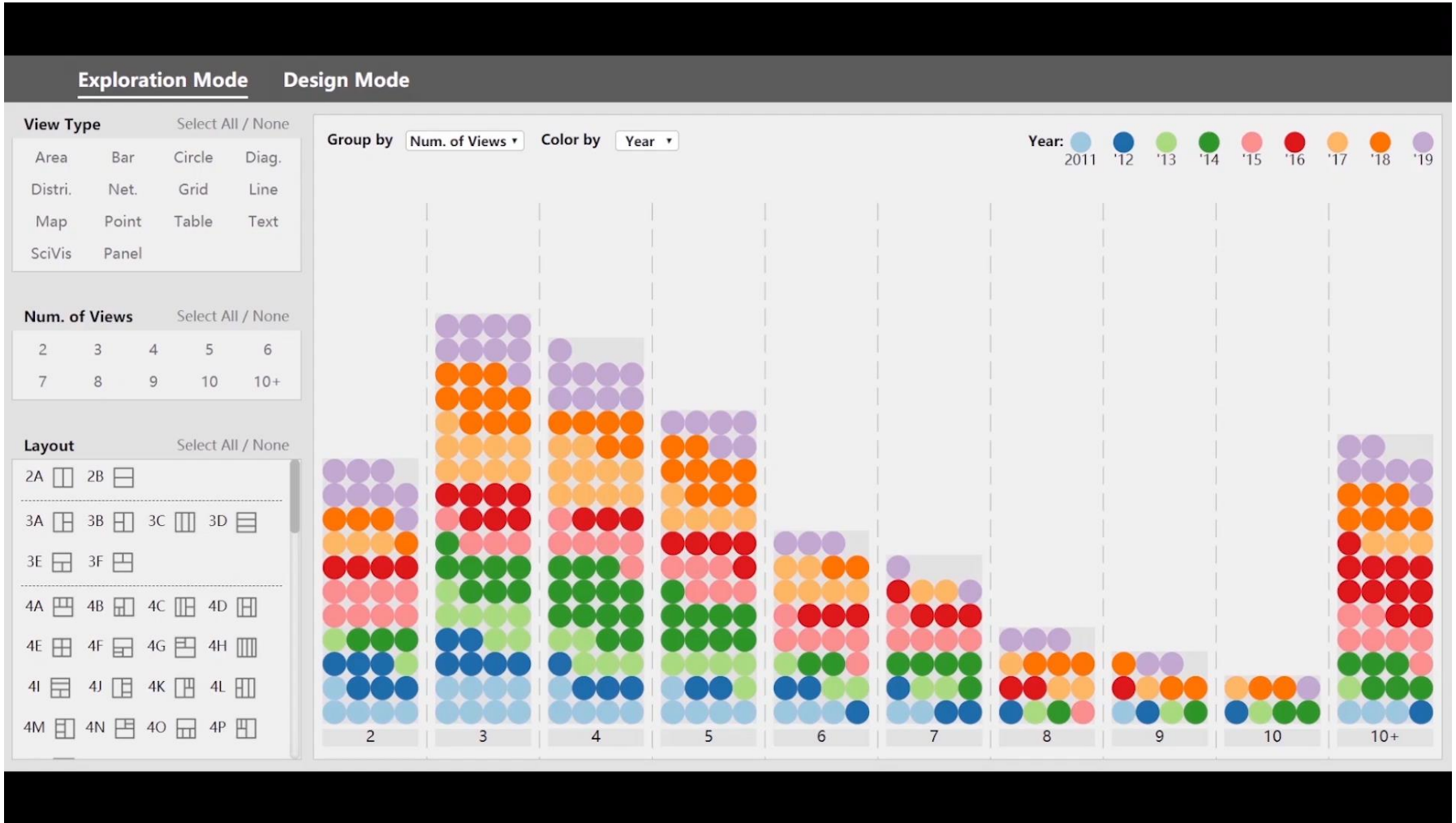
Coding rule:

- Numeric value: the number of "top-level" views
- Letter: a specific type of layout

Layout	VAST	InfoVis	SciVis	EuroVis	PacificVis	Total
2A	17 8.6%	18 36.7%	6 22.2%	7 14.9%	2 5.0%	50 13.9%
3C	17 8.6%	8 16.3%	1 3.7%	8 17.0%	2 5.0%	36 10.0%
3A	16 8.1%	4 8.2%	0 0.0%	1 2.1%	2 5.0%	23 6.4%
3B	6 3.0%	5 10.2%	4 14.8%	2 4.3%	2 5.0%	19 5.3%
4E	10 5.1%	0 0.0%	1 3.7%	2 4.3%	4 10.0%	17 4.7%
3F	6 3.0%	1 2.0%	0 0.0%	4 8.5%	0 0.0%	11 3.1%
3E	6 3.0%	2 4.1%	0 0.0%	1 2.1%	1 2.5%	10 2.8%
4H	4 2.0%	1 2.0%	1 3.7%	2 4.3%	1 2.5%	9 2.5%
4C	4 2.0%	1 2.0%	1 3.7%	1 2.1%	1 2.5%	8 2.2%
2B	1 0.5%	2 4.1%	1 3.7%	2 4.3%	1 2.5%	7 1.9%
Total	87 44.2%	42 85.7%	15 55.6%	30 63.8%	16 40.0%	

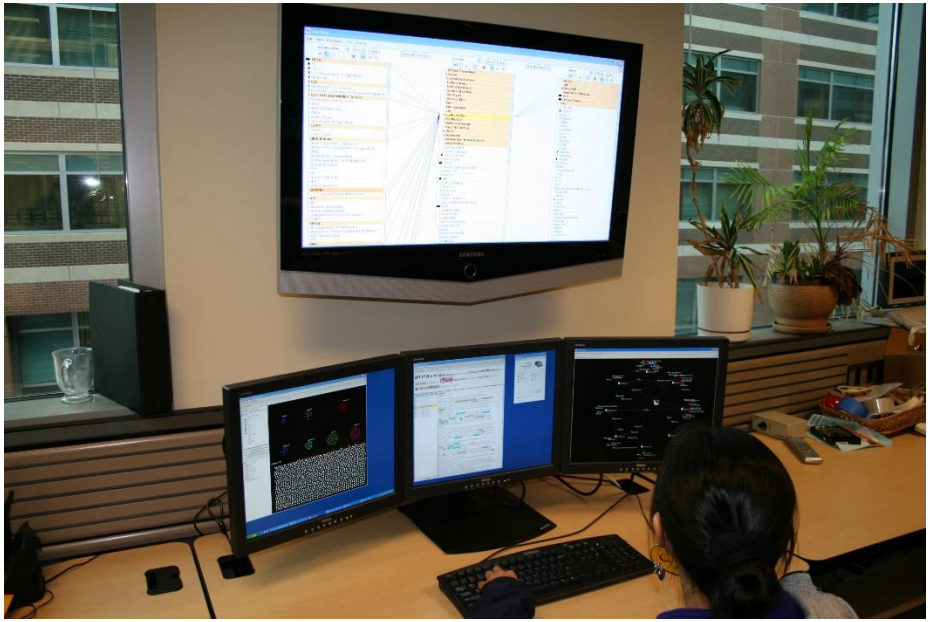
Design consideration: Composition and configuration

- Recommendation system (https://mvlandscape.bitbucket.io/)



Design consideration: Display viewport

- The question of how to design CMV is still debated
  - Many influencing factors including views, coordination, display viewport, and designers.
  - Responsive design is needed.



Design consideration: CMV for collaboration

- Lastly, one note about CMVs:
  - In principle, it is about “showing data from different perspectives”

- Turns out that the same idea can be used to foster collaboration between users (“different perspectives”)

- Collaborative Brushing and Linking
  - http://www.youtube.com/watch?v=E9izFMJ5yms

Summary

- Personally, I think CMV as a “poor man’s approach” to visualization design
  - Really good visualization designers can integrate all aspects of data exploration and analysis into a single, coherent visualization
  - For people who aren’t as good (like me), we rely on CMVs for its modular design while achieving a view of complex and high-dimensional data
    - E.g. think VitalVizor – there probably exists a better-designed visualization that can show all relations in one view
    - Using CMV, we can achieve a similar goal (with the sacrifice of needing the use of interaction to coordinate the views)