JBI100 Visualization

Summary

INTRODUCTION

Recall	Notes
What is the aim of visualization systems? What is used to achieve this?	Computer-based visualization systems provide visual representations of datasets and interactions designed to help people carry out tasks more effectively. For well-defined questions: computations usually enough (ML, statistics), but many analysis problems are ill-defined (What questions to ask? How to answer?). Visualization allows to augment human capabilities.
Are humans good at pattern recognition?	Humans are good at pattern recognition. Data visualization is a cognitive process. Aim: gain understanding - insight. *Example*: Cholera map London - J. Snow (1854) → cholera water borne. Vis as such is rather old. It has an intuitive step - graphical illustration. Large sizes of data → graphical approach necessary. Infographics, business graphics.
Name 5 disciplines involved in data visualization.	Data vis is multidisciplinary: application domain, numerical analysis, perception, computer graphics, optimization, algorithmics, human-computer interaction, ML, infographics, image processing.
Name the differences between InfoVis ans SciVis. Consider the type of visualized data (general type and data types/kinds), number of dimensions.	(see table below)

InfoVis	**SciVis (MedVis, FlowVis)**
Abstract data, no spatial reference.	Spatial data and spatial reference.
N-dimensional, heterogenous.	2-3 dimensional mostly.
Numerical, text, images, multimedia.	Scientific, engineering, biomedical.

Summary: Vis representations of datasets and interactions allow people to do tasks more efficiently - augment human capabilities. Vis is a cognitive process - humans are good at recognizing patterns. Multiple disciplines involved: numerical analysis, perception, computer graphics, optimization, algorithmics, ML, image processing, infographics. InfoVis - visualization of abstract data (numerical, text, multimedia), can be high-dim. SciVis shows spatial data, 2-3D.

VISUALIZATION DESIGN

Recall

What are the 3 high-level goals of visualization? Explain when they are relevant.

How are visualization guidelines set and refined?

Describe the process of Vis Design – 4 levels of selecting a solution.

Describe Munzner's nested model for Vis Design. What are the levels?

What is done at each level?

What dangers are there?

What methods for validation are applied at each level?

Notes

3 types of **goals** of vis: **explore** ("nothing is known"), **analyze** ("there are hypotheses, verify or falsify"), **present** (everything known, communicate results).

Vis design: many options, tradeoffs abound. Many possibilities known to be ineffective (so can be avoided) - extensive experimentation has been done. Guidelines continue to evolve. Refine iteratively.

Vis Design Process: known space → consideration space → proposal space → selected solution. Should be narrowed down proportionally.

Vis Design - nested model

Domain situation - problem characterization

- understand user, data, tasks
  - user's limitations, skills, needs, workflow
  - provide actionable knowledge (get info, decide)
- use domain specific vocabulary
- gather a set of tasks/questions of target users on the data (diff. levels)
- get info: interviews, observations, reading

DANGER: misunderstood user needs

VAL: observe how target users use existing tools

Data/task abstraction

- data described in generic (vis) terms: table, hierarchy, set, …
- tasks described in generic terms: search, compare, …

DANGER: showing the wrong thing

Visual encoding/interaction idiom

- design space, select visual encodings, define interactions

DANGER: the way you show it doesn't work

VAL: justify design wrt. alternatives

Algorithm

- layout of algorithm, ordering, rendering

DANGER: too slow

VAL: measure system time/memory, analyze comp. complexity

VAL: analyze results qualitatively, measure human time (lab study)

VAL: observe target users after deployment (field study)

VAL: measure adoption

Summary: 3 high-level goals: explore (nothing known)/analyze (check hypotheses)/present (communicate results). The process of selecting a solution is about narrowing down from whole known space to consideration space to proposal space to selected solution. Nested model for VisDesign: 1. Domain situation: define problem, understand user, get info - domain specific voc.), 2. Data/task abstraction: describe data and tasks in generic terms, 3. Visual encoding /interaction idiom - define design space, select encodings and interactions. 4. Algorithm.

VISUALIZATION DESIGN - WHAT

Recall

What are the 3 steps of visualization design analysis?

Of what type can data be?

Of what type can datasets be?

*What data types are contained in a **table**?*

*What data types are contained in a **network**?*

*What data types are contained in **fields**?*

*What data types are contained in **geometry datasets**?*

*What data types are contained in **clusters, sets, lists**?*

How can we divide datasets based on their availability?

Notes

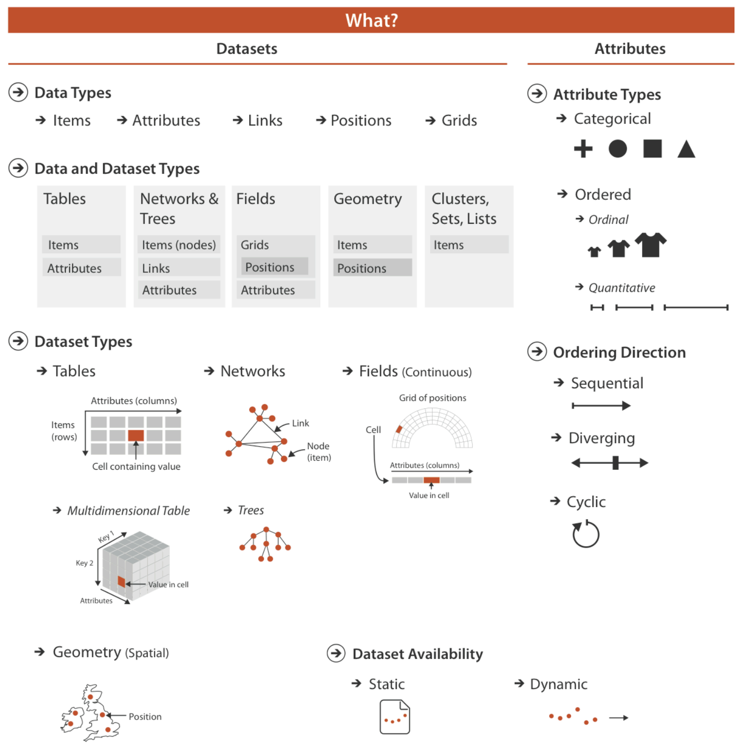
ANALYSIS FRAMEWORK

What *is shown?* - **data abstraction**

Why *is the user looking at it?* - **task abstraction**

How *is it shown?* - **visual encoding and interaction**

Munzner's categorization - WHAT



Datasets

Data types	items, attributes, links, positions, grids
Dataset types	table (can be multidimensional), network/tree, fields (continuous), geometry (spatial), cluster/set/list
Dataset availability	Static (offline - most cases), dynamic (online: streaming - time varying)

Summary: Analysis framework: What (data abstraction), Why (task abstraction), How (visual encoding and interaction). For data abstraction, the proposed framework describes datasets based on data types (item/attribute/link/position/grid), dataset types (table/network/fields/geometry), and dataset availability (static/dynamic).

VISUALIZATION DESIGN - WHAT

Recall	Notes
Of what types can attributes be? *What can be the ordering direction of ordered data?*	**Attributes**

Attribute types	categorical, ordered: ordinal, quantitative
Ordering direction	sequential, diverging, cyclic

This categorization is just a starting point - real data sets might be more complex - sets, groups, combinations of dataset types, new types.

One dataset can be looked at in different ways - the same data can be presented in different views (dataset types).

Give an example of when each dataset type can be used.

Examples - dataset types:

- Table: *products*
    - Multidimensional: *product orders*
- Network: *social network friends*
    - Tree: *Company organogram*
- Fields: *Weather prediction*
- Geometry*: Roads, rivers*

Give an example of each attribute type.

Examples - attribute types:

- Categorical: *fruit names, order ID*
- *Quantitative diverging*: temperature
- *Quantitative sequential:* order date, ship date
- *Ordinal cyclic:* day of the week
- *Ordinal sequential:* order prio, container size

What 2 requirements should domain tasks meet?

Tasks

First, domain tasks are defined. They should be:

- **Relevant** (specify what is important for a goal)
- **Functional** (not solution driven - *so specify what the user should be enabled to do, not what should be shown*)

Summary: Attributes can be categorical or ordered (ordinal or quantitative) – sequential, diverging, or cyclic. For tasks, domain tasks are first defined and they should be relevant (for the goal) and functional (so user-oriented, not vis-oriented).

VISUALIZATION DESIGN - WHY

Recall

What are the 3 main types of actions?

By what actions can data be consumed/produced?

On what do the actions related to search depend on? Name the 4 types.

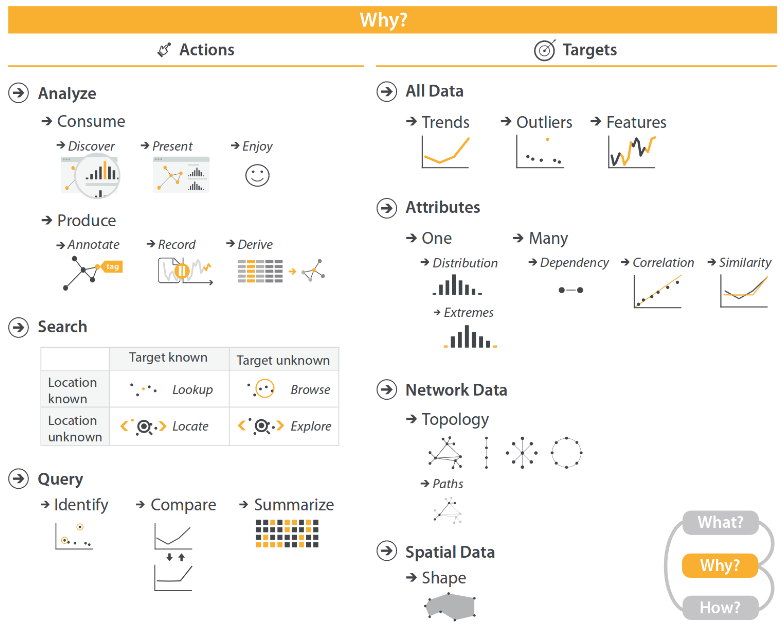
How can we query data?

What targets can relate to all data?

What targets relate to attributes?

Notes

Munzner's categorization - WHY



Actions

Analyze	*Consume*	discover	present	enjoy
	Produce	annotate	record	derive
Search				
Query	identify	compare	summarize	

Search:

Target / Location	Known	Unknown
Known	Lookup	Browse
Unknown	Locate	Explore

Summary : Targets and actions → tuples. General action types are *analyze*, *search* and *query*. Analyze - *consume*: *discover*, *present* or *enjoy*, or *produce*: *annotate*, *record* or *derive*. *Search* depends on whether the target or location are known. If both are known - *lookup*, only location known - *browse*, only target known - *locate*, both unknown - *explore*. Query can mean either *identify*, *compare* or *summarize*. Targets can be for all data (trends, outliers or features) or for attributes (one: distribution or extremes, many: dependency, correlation, similarity).

VISUALIZATION DESIGN - WHY, HOW

Recall	Notes

What are targets?

Targets - aspects of the data interesting to the user

All data	trends	outliers	features
Attributes	*One*	distribution	extremes
	Many	dependency, correlation, similarity	
Network data	*Topology*	*→ paths*	
Spatial data	*Shape*		

Why - {action, target} pairs

Examples: discover distribution, compare trends, locate outliers, browse topology

How

Generating solutions: generate many and fast, look back, sketching (explore design space, find solutions, structure, break constraints, inspire yourself).

Marks and channels

What are marks? Name 5.

Marks - geometric primitives:

- Points
- Lines
- Areas
- Volume
- Links: containment/connection

What are channels? Name 6.

Channels - control and appearance of marks:

- Position (horizontal/vertical/both)
- Color
- Tilt (angle)
- Size
- Shape
- Motion

Summary: Targets are the aspects of the data interesting to the user.

Marks are geometric primitives: lines, points, areas, volume or links (containment/connection).

Channels control the appearance of marks: position, color, angle, size, shape, motion.

VISUALIZATION DESIGN - HOW

Recall

Name and explain 2 principles for selecting visual encodings.

What are rankings of channels based on?

Describe Munzner's visual channel ranking.

Notes

Selection of visual encodings

Expressiveness principle

Show all, but only what is in the data. Match the channel and mark to data characteristics.

Effectiveness principle (salience)

Encode most important attributes with highest ranked channels. Rankings based on: accuracy, discriminability, separability, popout.

Visual channel ranking (Munzner's):

Categorical	*Ordered*	
1. Spatial region	1. Position on common scale	5. Area (2D)
2. Color hue	2. Position on unaligned scale	6. Depth (3D)
3. Motion	3. Length (1D)	7. Color saturation & luminance
4. Shape	4. Tilt/angle	8. Curvature / Volume (3D)

Summary: There are 2 main principles for selecting visual encodings: *effectiveness* principle and *expressiveness* principle. Effectiveness: choose the most effective channels to encode most important attributes. Expressiveness: Show all that is in the data but nothing more, and match the encoding to the data characteristics. There are rankings of channel effectiveness, based on separability, discriminability, accuracy and popout of the channels.

DATA AND TASK ABSTRACTION

<table>
<tr><th>Recall</th><th>Notes</th></tr>
<tr><td></td><td>
Good visualization? Depends on a context:

Who is it for? User (domain expert - speaks domain language)
What data do we have? Data abstraction
To do what? Task abstraction

Data/task abstraction provides a language to describe data in a meaningful and useful way to vis design. They help vis designer to reason about the right encoding.
</td></tr>
<tr><td>
What is the goal of data and task abstraction?

How is data usually described? How does data abstraction change this description?
</td><td>
<h3>Data abstraction</h3>
Data is typically described within domain language. To search for suitable vis, we need to translate them into more abstract structures which we know how to encode. This narrows down the design space.
<h3>Task abstraction</h3>
Tasks provide a constraint on design. No vis representation supports all tasks. Depends on the user and the goal of the vis. Transform domain specific language into abstract form to reason about similarities and differences.
Tasks provide a constraint on design and vice versa.

What info can I extract from this representation?What questions should the user be able to answer with this vis?

</td></tr>
<tr><td>
How do tasks and design affect each other?
</td><td>
<h3>Task - tuple (action, target)</h3>
Examples:

Convince the boss that the models are improving: (Present Trends)
Find out if there are money laundering structures: (Discover Topology)
Understand if all products have the same quality: (Summarize Similarity)

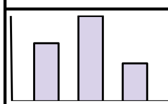
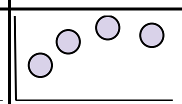
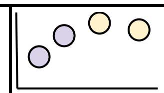
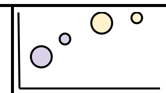
</td></tr>
<tr><td colspan="2">
Summary: Data and task abstraction - transform domain specific language into abstract form for vis design (domain-independent). Data abstraction: translate the description of the data from domain language to more abstract structures which we know how to encode - narrows down design space. Task abstraction: transform user's needs, expressed in domain-specific language, to independent tasks from which we can reason about encodings.
</td></tr>
</table>

ENCODINGS

Recall	Notes

What is a visual encoding?

Visual encoding - combination of marks and channels showing data attributes

	Bar chart	Scatter plot		
				
Channel	Length (vertical position)	Vertical and horizontal position	V and h position, Color	V and h position, Color, Size
Mark	Line	Point	Point	Point

Principles - selecting visual encodings

Expressiveness principle

Show all but only what is in the data.
Match the channel/mark to data characteristics.

Effectiveness principle (salience)

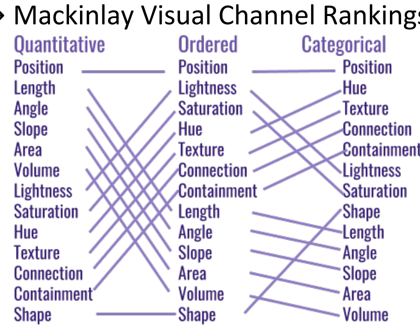
Encode most important attributes with highest ranked channels. Rankings based on:

- Accuracy
- Discriminability
- Separability
- Popout

What is the most effective channel for all types of data, according to Mackinlay?

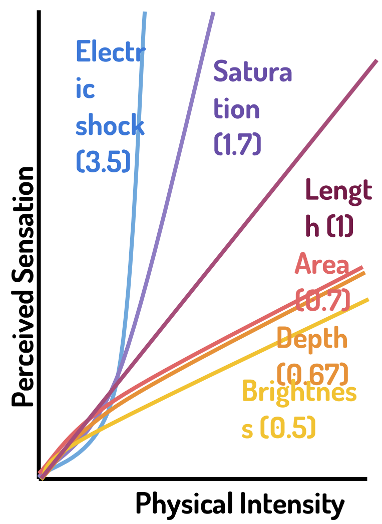
For what kind of data do color-related channels perform the worst?

→ Munzner's Visual Channel Rankings (↑)
→ Mackinlay Visual Channel Rankings



Summary: A visual encoding is a combination of marks and channels showing data attributes. The selection of visual encodings should be based on the principles of expressiveness and effectiveness.

ENCODINGS

Recall	Notes
Based on accuracy (crowdsourced experiments), how do channel types rank?	### Creating rankings *Accuracy* → experiments - crowdsourced results: - Positions > angles > circular areas > rectangular areas
What is Steven's Psychophysical Power Law? State and explain the formula. What channel is "one-to-one"? What channel has the worst noticeable increase?	Steven's Psychophysical Power Law: $S = I^N$ 
What is the discriminability of channels?	*Discriminability - How many usable steps?* Example: line width - only a few separable steps
What is the separability of channels? What is the different ("opposite") name for this feature? Name 2 separable channels and 2 integral channels.	*Separability/Integrality* - Position vs. hue - separable - Size vs. hue - some interference, hard to discriminate small items - Size width vs. height ("elongated circles")- interference, perception of area/size is integral - Red vs. green (scale) - major interference, perception of color and hue is integral

Summary: Based on experiments on accuracy, position is the most effective channel, followed by angles, circular areas and rectangular areas. Steven's Power Law states that the *sensation* can be expressed as *physical intensity* raised to the power of N, where N is a channel-specific coefficient. Based on this formula, the appropriate increase in intensity can be chosen to adjust perceived change to seem linear. Channels have the characteristics of discriminability (usable steps) and separability from another channel.

ENCODINGS

Recall

What is preattentive vision? Give an example of a channel which is perceived preattentively.

Give an example of a channel combination which is not preattentive.

What does the popout characteristic of a channel allow for?

If a channel does not have a popout, what kind of search/processing is performed?

What is the focus of Gestalt principles?
Name 7 Gestalt principles.

Notes

Preattentive vision and popout

Preattentive - attentional system not invoked, search speed independent of "distractor"

- Color hue and shape alone are preattentive

Require attention - search speed linear with "distractor" count

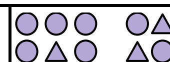
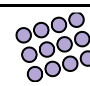
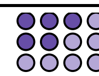
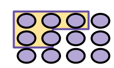
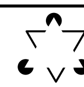
- Combined color hue and shape not preattentive

Most channels have a **popout** - parallel processing possible (sufficiently different item noticed immediately, independent of distractor). For some channels which do not have a popout, serial search is required (e.g. find parallel tilted pairs).

Principles and guidelines

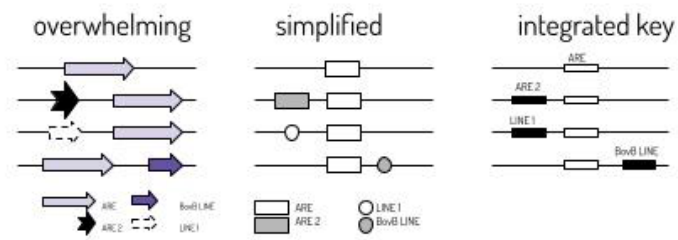
Gestalt principles (1920s)

- Human perception - how do we group elements, see patterns, simplify information

Proximity (Emergence)	Group elements close to each other	
Good figure	Objects grouped together perceived as a single figure	
Similarity	Group elements with similar appearance	
Common region	Group elements in the same enclosed region	
Figure/ground (Multi-stability)	We see depending on the perception of figure/background	
Closure (Reification)	Complete missing parts	
Continuity	Form and group continuous lines	

Summary: Preattentive vision is that not requiring attention - then, search speed is independent from the distractor (e.g. color or shape alone). When processing requires attention, search speed is linear (e.g. color and shape together). Most channels have a popout - preattentive processing possible. For channels without a popout - serial search (linear).
Gestalt principles refer to human perception - grouping, seeing patterns. Principles: proximity, good figure, similarity, common region, figure/ground, closure, continuity.,

ENCODINGS

Recall	Notes
What is Tufte's principle of graphical integrity? What are the 2 design principles of Tufte?	*Tufte's principles* **Graphical integrity:** avoid missing scales, scale distortion **Design principles:** - maximize data-ink ratio: $data\ ink\ ratio\ = \frac{data\ ink}{total\ ink\ used\ in\ the\ graphic}$, - avoid chart junk - remove redundancy 

Name 5 problems resulting from using depth in visualizations.
 	
What alternatives to 3D would be more suitable?
 	
When is it good to use 3D?
	Dangers of Depth (3D)
	We do not really see 3D ("2.05D")
	- 3D shown in planar spatial position:
	  - up/down and sideways - image plane: acquire more info quickly from eye movement
	  - Away - depth into scene: acquire more info only from head/body motion
	Problems:
	- Occlusion
	- Interaction complexity
	- Perspective distortion (interferes with all size channel encodings - power of the plane is lost)
	- Difficult text legibility (labels)
	- Detailed comparison impossible
	Transformation to a suitable abstraction:
	- Clusters (derived data)
	- Multiple views
	Use of 3D must be justified
	- 3D good for true spatial data (to show shape)
	- For non-spatial - a good reason needed

Summary: Tufte proposed the principle of graphical integrity (not to mislead the audience - avoid missing or distorted scales) and design principles: maximize data-ink ratio (data ink/total ink), avoid chart junk.

3D - depth - should be used carefully in visualizations. It can cause occlusion, interaction complexity, perspective distortion (interferes with size-related channels), difficult text legibility and make detailed comparison impossible. Instead, use clusters (derived) or multiple views

ENCODINGS

Recall

Why, in general, is resolution preferred to VR?
What benefits can VR have?

What is the problem with memory and attention for visual search tasks?

How is time shown in animation? And in small multiples?

What is the benefit of side-by-side views?

When is animation a good choice?

Notes

Resolution beats immersion

Resolution is very important (pixels - scarcest resource). Desktop is better for workflow integration.
Alternative: Virtual reality - immersion

- Very difficult to justify for non-spatial data, as sense of presence or 3D not needed there
- Recent attention to advantages of VR:
    - Better perception of dataspace geometry
    - Intuitive data understanding
    - Better retention of relationships in the data
    - Preferred by the users

Eyes beat memory

Memory:

- Long-term: unlimited
- Short-term - working memory: has a limit (reaching the limit - cognitive load)

Attention:

- Very limited for conscious visual search tasks

Animation vs. Side-by-side views

literal	*abstract*
animation	**small multiples**
show time with time	show time with space

Side-by-side views: easy to compare (low cognitive load) by moving eyes
Animation: hard to compare visible items to memory of what was seen

- Good for choreographed storytelling
- Good for transitions between 2 states/datasets
- Give control to the user on the animation (re-play, stop, pause)
- Change blindness!

Alternative to animation: Small multiples (one graph per experimental conditions, same spatial layout)

Summary : Resolution beats immersion (VR) because in general 3D is not needed for spatial data, but VR can allow for better perception of dataspace geometry, more intuitive understanding and better retention of relationships, as well as user engagement. Short-term memory and attention are limited for visual search tasks (cognitive load). To compare, side-by-side views are better than animation as we can compare them by moving eyes. Animation requires comparison with a memory → change blindness (but good for storytelling, transitions).

COLOR

Recall

How do we see? Describe the process of seeing an object.

What is the wavelength range for visible light?

What is the function of cones and rods?

How many colors approximately do humans see?

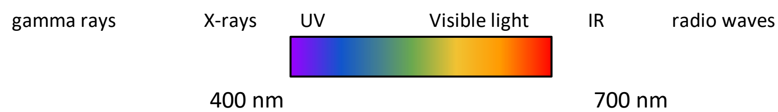
What causes color blindness?

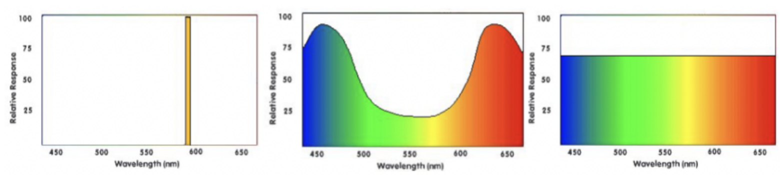
How many men are r-g colorblind? How many women? And b-y colorblind?

Notes

Light → object → stimulus → observer (retina/LGN/visual cortex)

Light spectrum: electromagnetic waves





How do we see?

- **Cones**: **chromatic perception**, resolution: 3x full HD, not used in night vision, **define color**
  - Detect S/M/L wavelength - 3 cone types
- **Rods**: **achromatic perception**, Low-Light vision (night vision), saturated in day vision

→ Visual system combines response

We can see ~10M colors

Color blindness

Different kinds of deficiencies (depends on functioning of cones - only having 2 of 3 types).

Red-green colorblindness:

- 5-8% men, 0.5% women
- Red cone deficiency (protanomaly)
- Green cone deficiency (deuteranomaly)

Blue-yellow colorblindness (tritanopia)

- 1% males-females

Summary: We see the light that is reflected by an object, and then, as a stimulus, reaches our retina and visual cortex. Light spectrum - electromagnetic waves (visible: 400-700 nm). Cones are responsible for seeing colors (detecting wavelength), in high resolution, and rods for achromatic perception. Visual system combines responses from both. Rods are for night vision. We can see around 10 million colors. Color blindness is caused by deficiencies in cone functioning - r-g deficiency (5-8% M, 0.5% F) or b-y deficiency (1% M and F).

COLOR

Recall	Notes
How many dimensions are needed (at least) for representing colors?	**Representing colors** - 3 dimensions (since 3 cones)
Name 2 representations that are device-oriented. Do they represent the full spectrum of human-visible colors?	**Device Oriented**: Physical realization - Screen RGB space (3 lamps) - Printer CMY(K) Simple; each device has its own space. Only a part of human visible colors represented.
What are intuitive representations of color based on?	**Intuitive**: semantic properties of color - Hue (color wheel) - Saturation (how much gray a color has) - Light/Value (how much light a color has)
What is the name of the representation including all human visible colors? How was it built? Is it hard to convert?	All human visible color: - CIE 1931 XYZ - built through experimentation, mathematical formulation, device independent, easy conversion - Other models: RGB, CMY gamut represent only a part of CIE 1931
Name 2 perceptually uniform representations.	Perceptually uniform: - MacAdam's ellipses - indistinguishable color grouped, ellipse indicates noticeable difference - CIELUV, CIELAB, (L*, a*, b*) - deformation of CIE space, euclidean distance corresponds to perceptual difference

What color is better distinguishable? What color is weakest distinguishable?	**Perception**
	Our visual system is biased:
	- Color is not evaluated independently
	  - Green is slightly better distinguishable
	  - Perception of blue is very weak

To what property of color are humans sensitive?	- Humans sensitive to luminance
Is perception context-independent?	- Context is relevant, perception based on brain experience.

Summary: Color representations need at least 3 dimensions (because there are 3 cones). Representations can be device-oriented (RGB - screen, CMY(K) - printer), which represent only a part of human-visible colors, intuitive (based on hue, saturation and light), representing all human-visible color (CIE 1931 XYZ), and perceptually uniform (MacAdam's ellipses, CIELUV, CIELAB). Our perception is biased - we distinguish green better and blue poorly, we are very sensitive to luminance, and perception depends on context and previous experience.

COLOR

Recall	Notes

Recall

What attribute feature should be reflected in colormaps?

What is the discriminability of color – how many colors can be distinguished?

Name 6 color maps. For what data could they be used?

What are the problems with rainbow colormaps?

Name one benefit of a rainbow colormap.

What can be used instead of a rainbow colormap?

Notes

Colormaps

- Categorical - hue
- Ordered: sequential/diverging

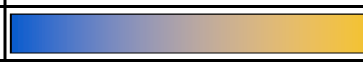
Common color spaces:

- RGB - poor choice for visual encoding, better HSL/Lab

Colormaps for categorical data

Discriminability: Humans can distinguish 12 bins of color (if discontinuous and large area)

Colormaps for quantitative/ordered data

Rainbow		categorical
Luminance - grayscale		sequential
Cool-warm		diverging
Black body		sequential
Isoluminant		
Blue-yellow		

Rainbow can lead to wrong conclusions.

Problems:

- Perceptually unordered, perceptually nonlinear
- Perceptual borders that are not there

Benefits: Nameable regions

Alternatives: few multiple hues with monotonically increasing luminance for fine-grained

Examples: fronts changes (lines) on the map. Proven that use of rainbow map increased diagnostic mistakes.

Summary: Colormap choice should be adjusted to whether an attribute is categorical, sequential or diverging. Humans can distinguish a dozen bins of colors (if discontinuous).

Rainbow colormaps can be wrong: they are perceptually unordered and linear, and there are perceptual borders that are not there. However, it has the benefit of namable regions. Instead of rainbow, few hues with monotonically increasing luminance should be used.

COLOR

Recall

What is luminance?

What is perceived when luminance is higher?

Is the relationship linear?

What is the name of luminance transformation to make it perceived linearly?

How many levels of gray can be discriminated?

When do we perceive changes in brightness better?

How should the luminance be for categorical data?

What is the name of color maps that are perceived linearly?

Notes

Visual perception - luminance

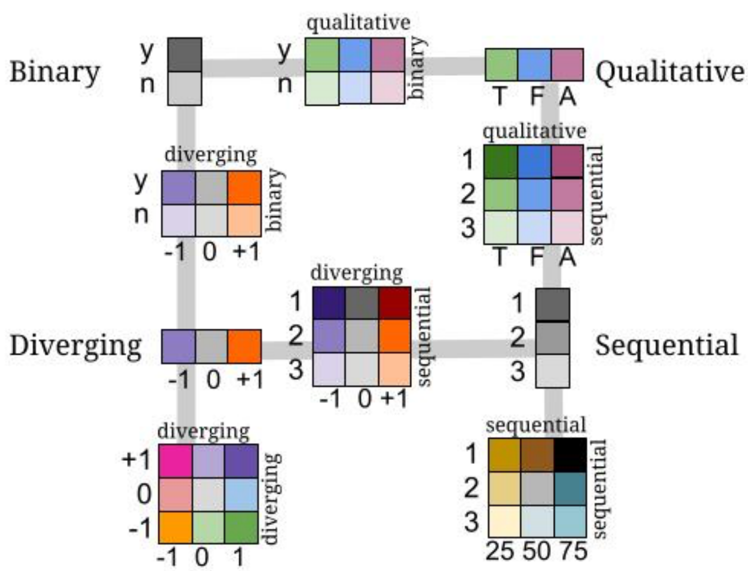
Luminance - physical parameter, light intensity per unit area

Higher luminance → higher brightness perceived, but:

- Non-linear relation between values and perception:
    - linear luminance - perceived as nonlinear lightness
    - Gamma compressed luminance - perceived as linear lightness
- Humans perceive relative brightness changes better in darker areas. We can differentiate between 100 gray levels simultaneously.

Colormaps and luminance

- For categorical data, luminance should be constant.
- Look for **perceptually uniform sequential color maps**



Summary: Luminance is a physical parameter of color - it corresponds to the lightness of a color that we perceive. The relation between luminance and perceived brightness is not linear (linear - Gamma compressed luminance). Changes in luminance are perceived better in dark areas and we can differentiate between 100 levels. For categorical data, luminance should be constant. We should choose perceptually uniform color maps.

PERCEPTION

Recall	Notes
Does the perceptual system base on relative or absolute judgements? *What is the role of the background in stimuli perception?* *Describe the Weber Law.* *What increases accuracy of judgements?*	**Perception - Weber Law** The perceptual system mostly operates with relative (not absolute) judgements. Smallest change in stimuli that is perceived (ΔS) to background stimuli (S) is constant (k): $\frac{\Delta S}{S} = k$. Applicable for example to luminance. ***Relative vs. absolute judgements*** Accuracy increases with common frame/scale and alignment: framed and aligned - easier to compare.

Summary: Perception works based on relative, not absolute judgements. The change is perceived as relative to the background. WeberLaw for perception: change in stimuli perceived to background stimuli is constant.

Accuracy of comparison increases with common scale and alignment.

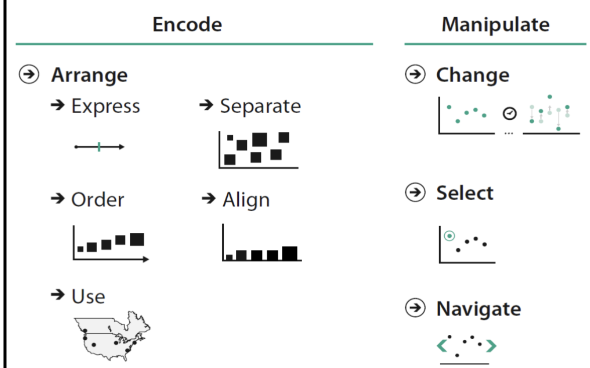
IDIOMS

Recall	Notes

Vis idioms

What design techniques can be used for vis idioms?

Techniques (design choices)



How are idioms described?

Idioms are described with:

- Data (*what is the data behind the chart?)*
    - # categorical attributes
    - # quantitative attributes
    - Semantics of keys and values (key → index, values - in cells of the table)
- Mark (*which visual elements are used?*)
    - Points, lines, glyphs, …
- Channels (*how is the data encoded?*)
    - Arrangement and mapping
- Tasks (*what are the supported tasks?*)
    - Discover trends, outliers, distribution

Describe a bar chart.

Idioms with n keys and 1 value

Bar chart

Data	1 cat. key, 1 quant. value
Mark	Lines
Channels	Length (for quant. val), spatial region
Tasks	Compare, lookup values
Scalability	Hundreds of levels for key

Summary: In vis design, data can be encoded by arranging (express, separate, order, align, use) or manipulating (change, select, navigate).

Idioms are described with data (what's behind? - qual./quant. Attributes, keys/values), marks and channels (visual elements used and how they encode data), and tasks supported by these idioms.

Bar chart - 1 cat and 1 quant, lines - length and spatial region, compare/lookup values. x100 key l.

IDIOMS

Recall	Notes
Describe a stacked bar chart.	**Stacked bar chart**
Describe a line chart.	**Line chart**
For what type of attribute should a bar chart be used? And line chart? *Why cannot a line chart be used for categorical data?*	**Bar vs. line chart** - Categorical → bar, ordered → line (don't use line for categorical attributes! Violates expressiveness principle, implication of trend overrides semantics)

Stacked bar chart

Data	2 cat. keys, 1 quant. value
Mark	Stack of lines
Channels	Length (for quant. val), color region, spatial region (one per mark) - only 1st aligned
Tasks	Compare, lookup values; part-to-whole relationships
Scalability	Hundreds of levels for key, several to dozen for stacked attribute

Line chart

Data	1 quant. key, 1 quant. value
Mark	Points, line connecting marks
Channels	Aligned lengths, separated and ordered into horizontal regions
Tasks	Find trends; connection marks emphasize ordering of key items - show relationships
Scalability	Dozens to hundreds of levels for key

Bar vs. line chart

- Categorical → bar, ordered → line (don't use line for categorical attributes! Violates expressiveness principle, implication of trend overrides semantics)

Summary:

Stacked bar chart - 2 cat and 1 quant, stack of lines - length, spatial region, color, compare/lookup values/part-to-whole. x100 key l, x10 stacked. → for categorical

Line chart - 2 quant, points and line connection - length, spatial region, find trends/show relationships. x10 - x100 key l. → for ordered (only! For categorical - violates expressiveness principle, overrides semantics).

IDIOMS

Recall	Notes

Describe a streamgraph.

Streamgraph

Data	1 cat. (name), 1 ordered (time), 1 quant. value (counts)
Marks and Channels	Derived geometry: layers, height encodes counts
Tasks	Find trends, part-to-whole relationship
Scalability	Hundreds for time, dozens to hundreds for names (more than stacked bars, since most layers don't extend across whole chart)

Describe a heatmap.

Heatmap

Data	1 quant. (value), 2 cat. keys
Marks	Separate and align in 2D matrix, indexed by 2 attributes
Channels	Color by quantitative attribute
Tasks	Find clusters, outliers, patterns
Scalability	1M for items, hundreds of cat. Levels, ~10 quant. attribute levels

Describe a pie chart.

Pie chart/Polar area chart

Data	1 cat., 1 quant.
Marks	Separate colored area
Channels	Color by categorical, Angle for quantitative
Tasks	Part-to-whole judgment
Scalability	1 dozen

Summary:

Streamchart - 1 cat, 1 quant and 1 ordered (time), derived geometry (layers, length), find trends/part-to-whole. x100 for time, x10 - x100 for names (more than stacked bar)

Heatmap - 1 quant and 2 cat, 2D-matrix, - color, find clusters/outliers/patterns. x1M items, x100 categories, x10 quantitative (colors)

Pie chart - 1 cat and 1 quant, area - color and angle, part-to-whole. x10 levels

IDIOMS

Recall	Notes
Describe a histogram.	**Histogram**

Histogram

Data	1 quant. + derive data: keys → bins, values → counts (bin size crucial!)
Marks	Line
Channels	Length encodes frequency
Tasks	Understand distribution

Describe a boxplot.

Boxplot

Data	1 quant. + derive data: 5 quant. attributes (median, min, max, lower and upper quantile) and explicitly show outliers
Marks	Lines + box
Channels	Length encodes derived values
Tasks	Understand distribution
Scalability	Hides details!

Describe a violin plot.

Violin plot

Data	1 quant. + derive data: 5 quant. attributes (median, min, max, lower and upper quantile) and density at each point
Marks	Lines + box
Channels	Length for derived values, width for frequency
Tasks	Understand distribution

Summary:

Histogram - 1 quant and derived data, line - length, understand distribution

Boxplot - 1 quant and derived data (5 + outliers), lines + box - length, understand distribution

Violin plot - 1 quant and derived data (5 + outliers), lines + box - length and width, understand distribution

COMPLEX MULTIVARIATE IDIOMS

Recall	Notes
Describe a scatterplot. How can scatterplot scalability be improved? What is a SPLOM?	**Scatterplot** <table><tr><td>Data</td><td>2 quant., no keys (only attributes)</td></tr><tr><td>Marks</td><td>Points</td></tr><tr><td>Channels</td><td>Horizontal + vertical position</td></tr><tr><td>Tasks</td><td>Find trends, outliers, distribution, correlation, clusters</td></tr><tr><td>Scalability</td><td>Hundreds of items</td></tr></table>Improve scatterplot scalability: - **Change mark**: Reduce point size, Set transparency, Color by category, KDE on diagonal/off-diagonal - Multiscale scatterplot: Smoothing by convolving with Gaussian at different scales (KDE - Kernel Density Estimation) → Find patterns at different scales **SPLOM - Scatterplot matrix** - If there are many attributes - use a matrix showing all relations between variables - Interaction: linked brushing - Symmetric: use diagonal for other idiom

Summary:

Scatterplot - 2 quant, points - position (h + v), find trends, outliers, distribution, correlation, clusters. x100 items (more with: mark changes: smaller size, transparency, color, KDE; smoothing at different scales)

SPLOM - scatterplot matrix: involves more attributes and shows a scatterplot for each pair. Interaction : linked brushing.

COMPLEX MULTIVARIATE IDIOMS

Recall

What is a PCP?

How can positive correlation be detected in PCP and in SPLOM? And negative? How are uncorrelated attributes shown?

How can PCP scalability be improved?

What relationship between axes is shown in PCP and what in SPLOM?

What is the scalability of each?

What is the most important aspect when designing them?

What is the main feature of radar plots?

What is a glyph?

Give an example of iconification in vis.

Notes

PCP - Parallel Coordinate Plot

- All axes parallel, scaled to min/max data range
- Each item/sample is a polyline
- All dimensions shown at the same time

correlation	**positive**	**negative**	**uncorr.**
PCP	Parallel line segments	All segments cross at halfway point	Scattered crossing
SPLOM	Diagonal low-to-high	Diagonal high-to-low	Scattered points

PCP scalability: dozens of attributes, millions of items
Improve by transparency + additive blending.
Reduce clutter by interaction. Order of axes matters.

High dimensional data visualization - PCP vs. SPLOM

PCP	SPLOM
Primarily relationships between adjacent axes	Primarily relationships between pairs of axes
Limited scalability: ~ 50 dimensions, 1-5k samples	Limited scalability: ~ 20 dimensions, 500-1k samples
Important: axis ordering, interaction crucial	Important: brushing crucial

Radar plots or Flexible linked axes

Instead of parallel, arrange axes in star shape - samples become polygons. Shapes can be compared.

→ Iconification
Icon/glyph: map multidimensional data to properties of graphics objects
Example: Chernoff faces (1973) - use human capabilities to recognize faces to convey info.

Summary: PCP – parallel coordinate plot – all axes parallel, each item – polyline. PCP scalability can be improved by adding transparency and additive blending, and clutter can be reduced by interaction. Also, the order of axes is significant. In general, PCP is better for showing relationships between adjacent axes, for <50 dimensions and 1-5k samples, while SPLOM shows relationships between all pairs, for <20 dimensions and <1k samples. Radar plots have star-arranged axes. Iconification – using glyphs and their graphical properties to encode data.

INTERACTION

Recall	Notes

Interaction principles

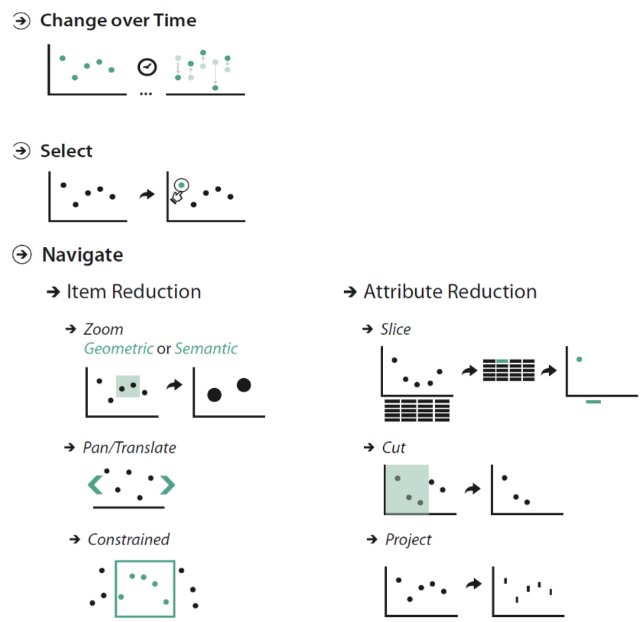
What should interaction be adjusted to?

What interaction techniques can be identified?

- Low-latency visual feedback
- Perceptual processing - 0.1 s
  Immediate response - 1 s
  Brief tasks - 10 s

Interaction should support speed of thinking.
Give overview first → details on demand (zoom/filter)

Manipulate (Taxonomy based on techniques - Munzner)



What can be changed over time in interaction?

What technique supports showing change over time?

Interaction

Change over time

- Advantage of digital over paper
- Change encoding, parameters, viewpoint, aggregation, and within encoding: change sorting order, rearrange layout...
- Consider animation

Summary: Interaction should be adjusted to the speed of thinking. Overview should be given first, and then the details. Among interaction techniques, we can distinguish *change over time*, *select*, or *navigate*. For *navigate*, either item or attribute reduction can be done. For item reduction, we can zoom (geometric or semantic), pan/translate, or constrain. For attribute reduction, we can slice, cut, or project. Change over time can use changing encodings, parameters, viewpoint, and encoding characteristics (order, layout, ...). Animation can be helpful.

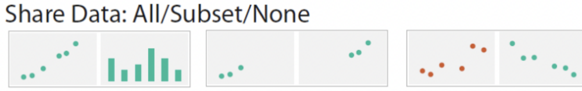
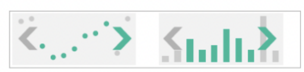
INTERACTION

Recall	Notes
Describe how selection is done and what it "does" to an idiom. *What is the difference between geometric and semantic zoom?* *What is the difference between slice, cut and project?* *What is shown in coordinated multiple views?* *What design choices can be made wrt. coordinated mv?* *How can linking and brushing be implemented in coordinated mv?*	**Select** - Basic operation for all interaction - Click or hover - Highlight: change encoding for item of interest **Navigate** *Item reduction* → camera metaphor **Zoom**: Change viewpoint/visibility of items - Geometric - like camera - Semantic - change visual encoding depending on zoom level (show more detail at zoom in) **Pan/Translate** **Constrained** - constrain to specific area *Attribute reduction* **Slice**: show only items matching specific value for given attr. **Cut**: show only items within some region or attr. range **Project**: dimensionality reduction **Coordinated multiple views** Show relations between items and attributes. Provide different perspective on the same items. Design choices: - View count: few vs. many - View visibility: popout vs. side by side - View arrangement: manual vs. automatic Linking and brushing: - Linked view parameters - Linked highlight/selection - items highlighted or selected in one view are also selected in the other views.

Summary: Selection can be done by clicking or hovering, and the item of interest is highlighted. Navigation can be focused on items (zoom: - semantic: add items when zoom in, geometric: just a closeup on the same view; pan; constrain) or attributes (slice: show only those meeting a criterion, cut: show only those in some range, project: dim. reduction). Coordinate multiple views show relations between items and attributes, from different perspectives - few/many views, visibility (popout/side-by-side), arrangement (manual/automatic). Linking/brushing needed.

INTERACTION

Recall	Notes
What do small multiples show?	Small multiples: same representations of data (change for one attribute)

How can interaction be implemented in coordinated multiple views? What aspects can be shared (or different) between the views?	***Interaction for coordinated multiple views***
	➔ Share Encoding: Same/Different
	➔ *Linked Highlighting*
	
	➔ Share Data: All/Subset/None
	
	➔ Share Navigation
	
	Encoding: same/different
	- Linked highlighting
	Data: all/subset/none
	Navigation: can be shared
	Types of multiple views
	- Multiple views: different visual encodings of same data
	- Small multiples: same representations of data, changing for one attribute
	- Overview and detail:
	    - same visual encoding, same data, different zoom-level
	    - Detail on demand: extra view with more info on selection

What is the focus and context technique? What type of technique is this?	***Focus and context*** Different levels of detail integrated in the same view: - Show area/items of interest (**focus**) in detail and surroundings (**context**) in less detail - Distortion techniques

Summary: Small multiples show the same representations of the data, with only one attribute changed. Interaction for coordinated multiple views can be done by using same/different encoding (and then using highlighting), same data or a subset, and sharing navigation.

Different levels of detail can be integrated in one view, by distortion - showing focus (items of interest) in detail and context (other, surrounding data) in less detail.

INTERACTION

Recall	Notes
What is the taxonomy of interactions proposed by Yi et al. based on? Describe it.	**Taxonomy based on user's intent - Yi et al.** 1. Select - mark something as interesting 2. Explore - show something else 3. Reconfigure - show a different arrangement 4. Encode - show a different representation 5. Abstract/Elaborate - show more/less detail 6. Filter - show something conditionally 7. Connect - show related items

Summary: Taxonomy based on user interest: select (mark as interesting), explore (show something else), reconfigure (change arrangement), encode (change representation), abstract (show less detail) or elaborate (show more detail), filter (show conditionally), connect (related items).

DATA REDUCTION

Recall

What is filtering?

What is aggregation?

What is the goal of item filtering? Which dimension changes and which stays the same?

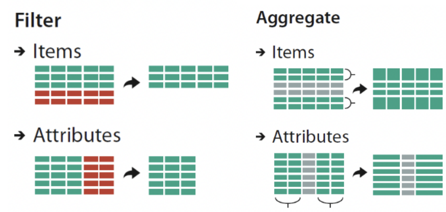
Name and describe 2 methods for item filtering.

What is the goal of attribute filtering? Which dimension changes and which stays the same?

How can attribute filtering be achieved?

What is the Pearson correlation coefficient? How can it be used for attribute reduction?

Notes



Filter: straightforward, intuitive (but easily forgotten)
Aggregate: summarize dataset (but details are lost)

Filter items

Goal: eliminate items based on their values for specific attributes
attributes does not change

Dynamic querying and direct manipulation

- User interacts to process/filter data.
- Immediately show the result.
- Selection/filtering by pointing (visual queries).
- Promotes exploration

Scented widgets

- Show data properties/distribution
- Guide user where to look/filter

Filter attributes

Goal: eliminate attributes
items does not change

Reduce number of attributes

- Create attribute ordering (remove highly correlated attributes - determine similarity measure - e.g. Pearson correlation coefficient:
  $r_{xy} = \frac{cov_{xy}}{sd_x sd_y}$, r=1 - perfect pos. corr., r = -1 - perfect neg. corr.)
- Immediately show the result. Promotes exploration.
- Selection/filtering by pointing (visual queries).

Summary: Filtering - get rid of a part of data, aggregate - summarize the data (details lost). Items are filtered based on their value for a specific attribute. It can be done by dynamic querying and direct manipulation (user interacts and result is shown right away - exploration). Scented widgets can guide the user in filtering, as they show the distribution of features. The number of attributes can be reduced by ordering them and removing the highly correlated ones
(pearson coef. = cov(x, y)/(sd(x)*sd(y))). Attribute filtering can also be done on demand.

DATA REDUCTION

Recall	Notes

What is the goal of item aggregation?
 	
How can the number of items be reduced by aggregation? Name different methods supporting it.
 	
What is the goal of attribute aggregation?
 	
How can attributes be aggregated? Name a simple and a more complicated method.
 	
What is the idea behind dimensionality reduction?
 	
How is dimensionality reduction performed?
What transformations can be done for dimensionality reduction? What characteristics of the data are preserved with this methods?
	Aggregate items
	Goal: reduce items through summarizing/grouping
	# attributes for each item does not change
	Reduce number of items
	Using visual encoding idioms, supported by interactive idioms (dynamically update changes):
	- Clustering: define groups of similar items
	- Navigation between items and clusters
	- Possibilities: k-means, hierarchical clustering
	Represent many data items with a single mark.
	Aggregate attributes
	Goal: summarize attributes
	# items does not change
	Reduce number of attributes
	Strategy: group attributes by similarity measure
	More complex: **dimensionality reduction**
	Preserve meaningful structure of a dataset while using fewer attributes to represent the items. Assume that there is redundancy in the data. Find best (combination of) attributes that represent the n dimensional space. Complexity: $2^n - 1$ → use automatic methods (heuristics).
	
	Method: minimize an objective function that measures the discrepancy between similarities in the data and similarities in the map
	Transformation to a projection/subspace:
	- Linear: resulting attributes are a linear combination of existing ones (→ interpretable):
	  - PCA - preserve variation
	  - LDA (Linear Discriminant analysis) - preserve class separation
	- Non-Linear: resulting attributes don't have a straightforward relation to original attributes: MDS (Multi-Dimensional Scaling) - preserve distances, t-SNE (t-Distributed Stochastic Neighbor Embedding) - preserve neighborhoods, UMAP (Uniform Manifold Approximation and Projection) - Riemannian metric

Summary: Item aggregation allows for their summarizing/grouping. Available methods: clustering (e.g. by k-means, hierarchical clustering), navigation between items and clusters. Attribute aggregation allows to summarize attributes. For this, we can group similar attributes or do dimensionality reduction. With dimensionality reduction, the structure of the dataset is preserved, but items are represented with fewer attributes. Methods: linear (create linear combinations of attributes: PCA, LDA) or nonlinear (MSD, t-SNE, UMAP).

MAPS

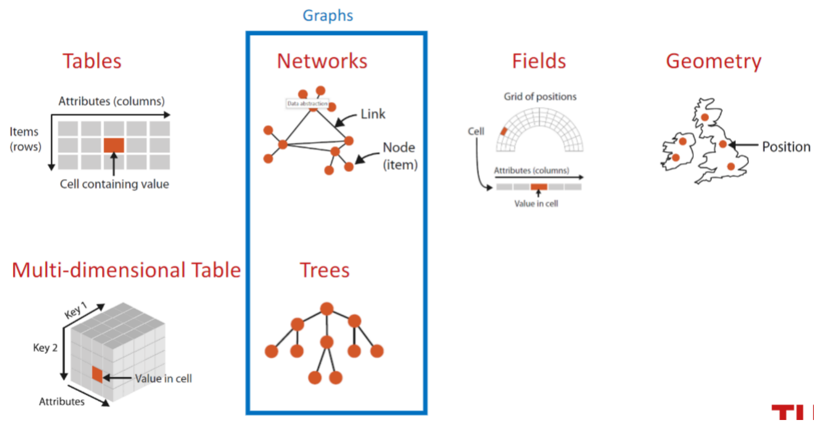
<table>
<tr><th>Recall</th><th>Notes</th></tr>
<tr><td>Describe a choropleth map</td><td>
CHOROPLETH MAP
<table>
<tr><td>Data</td><td>• 1 Quantitative attribute (table with 1 quantitative attribute per region)
• Geographic geometry</td></tr>
<tr><td>Marks</td><td>• Geometric area</td></tr>
<tr><td>Channels</td><td>Color</td></tr>
<tr><td>Tasks</td><td>Understand spatial relationships
(Note that size of objects depends on geography, not on the attributes)</td></tr>
</table>
</td></tr>
<tr><td>Describe a dot map</td><td>
Cartogram - Size represents quantity at cost of familiarity due to distortion.

DOT MAP
<table>
<tr><td>Data</td><td>• 1 Quantitative attribute (table with 1 quantitative attribute per region)
• Geographic geometry</td></tr>
<tr><td>Marks</td><td>Points</td></tr>
<tr><td>Channels</td><td>Color</td></tr>
<tr><td>Tasks</td><td>Understand the presence and distribution density of an attribute
Ex: number of cases of an infection in Africa</td></tr>
</table>
</td></tr>
<tr><td>What are the cautions to be aware of with maps?</td><td>
Caution with maps:
A dataset may contain geographical information
• Yet creating a geographical visualization may not be relevant.
• Ask: “Does spatial arrangement matter for my task?”
• Positioning the most effective visual channel does not waste it, if not relevant.
• A map is not always the best / only solution.
</td></tr>
</table>

MAPS

<table>
<tr><td>Recall:</td><td>Notes</td></tr>
<tr><td>What are the two kinds of maps?</td><td>Absolute VS Relative maps
• Population density vs. per region
• When populations are low, variations tends to be high</td></tr>
<tr><td>Describe a topographic map</td><td>TOPOGRAPHIC MAP
<table>
<tr><td>Data</td><td>• Scalar spatial field (1 quantitative attribute per grid cell)
• Geographic geometry
Derived data
• Isoline geometry (isoline are lines that represent the contours of a particular level of the scalar value, can be derived from a scalar spatial field)</td></tr>
<tr><td>Mark</td><td>lines</td></tr>
<tr><td>Channel</td><td>Shape, position, color</td></tr>
<tr><td>Tasks</td><td>Understand spatial relationships, query shape</td></tr>
</table></td></tr>
<tr><td>Describe a density map</td><td>DENSITY MAPS
Need a data transformation to turn discrete data into continuous data. Typically using density estimation mechanism (e.g., KDE).
can also be used for other type of space (not only geographical), ex: weather forecasts like thunder, heat</td></tr>
<tr><td>What are the advantages of maps?</td><td>ADVANTAGES OF MAPS
Familiarity
• People know where something is on a map (assuming they are familiar with the region)
• Maps act as an index from spatial to semantic information and vice-versa
Spatial -> semantic (How many cases of COVID-19 in Noord-Brabant?)
Semantic -> spatial (where are the lowest # of covid cases?)</td></tr>
</table>

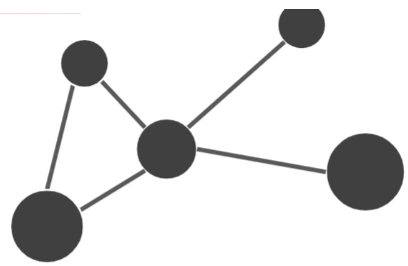
GRAPHS

<table>
<tr><td>Recall:</td><td>Notes:</td></tr>
<tr><td>What are the different data types and which ones are graphs?</td><td>


</td></tr>
<tr><td>Describe a network</td><td>

NETWORKS

Graph – Network
Vertices – Nodes
Edges – Links



Networks describe the relations between objects

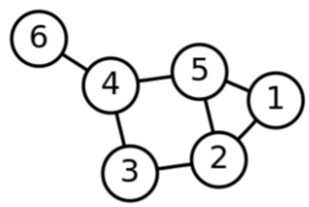
$G = (V, E), E \subseteq V^2$
V set of vertices
E set of edges

A network graph or simply a network is a connection of objects that are linked together through a series of nodes representing those objects. The connection between these nodes is termed as an edge or a link that signifies the connection of two objects. Network graphs can be understood in light of the functioning of the internet.
</td></tr>
<tr><td>What are the 2 types of networks?</td><td>

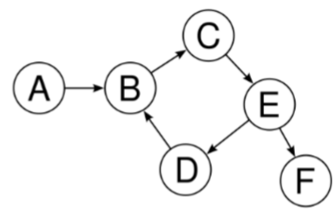
Types of networks:

1. Undirected - relationships are non-directional by their nature. For example, in a Facebook network, if Smith is friends with Jones that means that Jones is friends with Smith.
</td></tr>
</table>

2. **Directed** - connections between nodes are directional. For example, in a Twitter network, Smith might follow Jones but that does not mean that Jones follows Smith

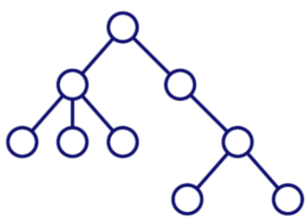


Undirected



Directed

Describe a tree



TREE

- Graph without cycles and one root
    - Every other node exactly one parent acyclic
    - Single root: all other nodes reachable from there
    - E = V – 1
- Hierarchy = “rooted tree”

What are the 3 network types?

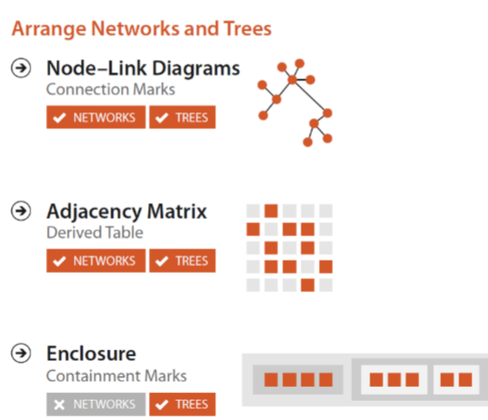
3 types of networks:

1. <u>Static</u> - structure
2. <u>Multivariate</u> data – extra information
3. <u>Dynamic</u> – evolving over time

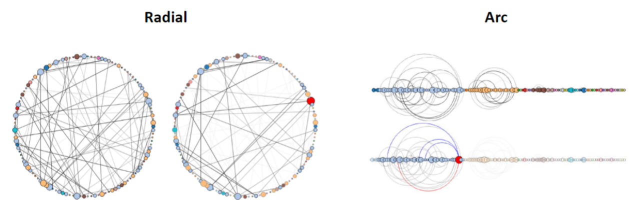
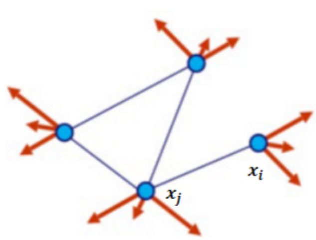
Describe static networks

STATIC NETWORKS

Static networks are networks in which the connections between the nodes are fixed and do not change over time. Ex: transportation networks, such as a city's subway system.



2 kinds of static networks:

What are the 2 kinds of static networks	1. Radial 2. Arc Linearize nodes: ordering plays a very important role 
Elaborate about the visualization techniques for static networks	**Static network visualization** • Node-link diagram easy to understand by non-experts Positioning of nodes is called layout or embedding Compute layouts graph drawing (force directed methods) <u>Readability & aesthetics</u> • Equal edge length • Minimize crossings • Non-overlapping nodes • High-degree nodes should have a central position • Symmetry should be maximized • Communities should be clearly visible Conflicting – NP-hard optimization problems Use heuristics to solve this
What are force-directed algorithms	**Force-directed algorithms** - simulate the behavior of physical systems, such as the repulsion between charged particles, to determine the layout of the graph • Mechanical laws • Model edges as springs, also nodes repel each other • Numerically simulate until stable state is reached
What is vertex force and edge force in a force directed graph? *(vertex force:)* 	**Vertex force** • Repelling force between vertices $ii$and jj • Prevents vertices to come too close to each other Repulsion strength $$g_{ij} = \frac{r_{ij}}{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

<table>
<tr>
<td>

(edge force:)

</td>
<td>

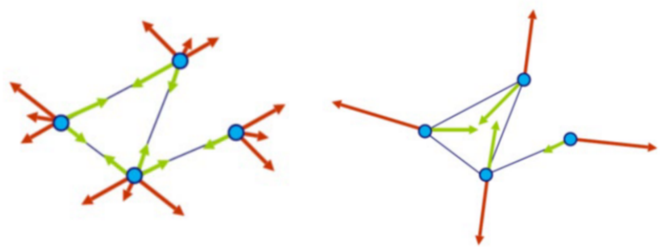
Edge force

- Spring forces on edge
- Attracts vertices connected by edge
- Prevents these vertices from getting too far apart

$$f_{ij} = \overset{\text{Spring tension}}{k_{ij}}\left(d\left(x_i, x_j\right) - \overset{\text{Spring length}}{s_{ij}}\right)\frac{x_i - x_j}{d\left(x_i, x_j\right)}$$

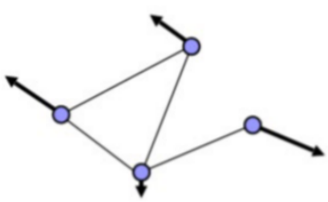
Edge and vertex forces in action

- Compute repulsion and attraction forces for all nodes



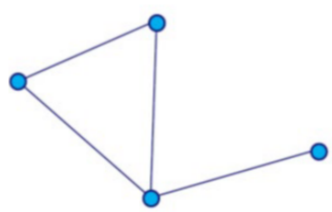
</td>
</tr>
<tr>
<td>

Compute the following forces in action and show the vertices in their new position



</td>
<td>

Vertices in new position:



</td>
</tr>
<tr>
<td>

How does a static network terminate and what are its tasks?

</td>
<td>

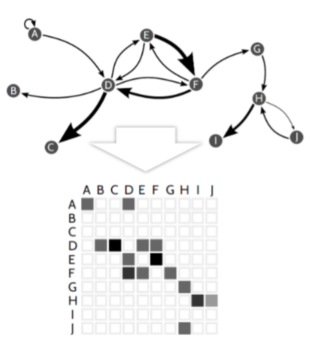
<u>Termination</u>

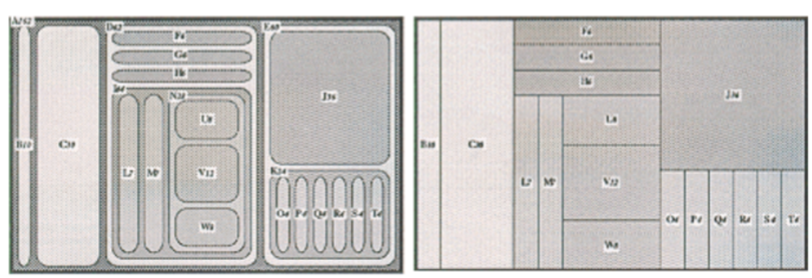
- Fixed number of iteration
- Total energy below some threshold
- Local minimum
- User input

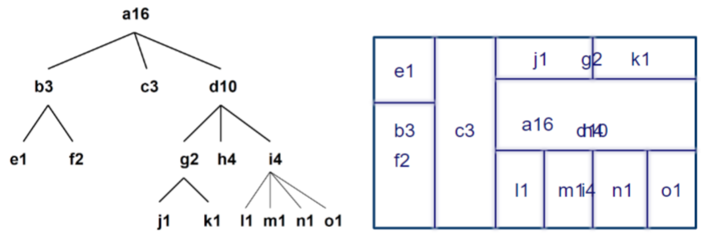
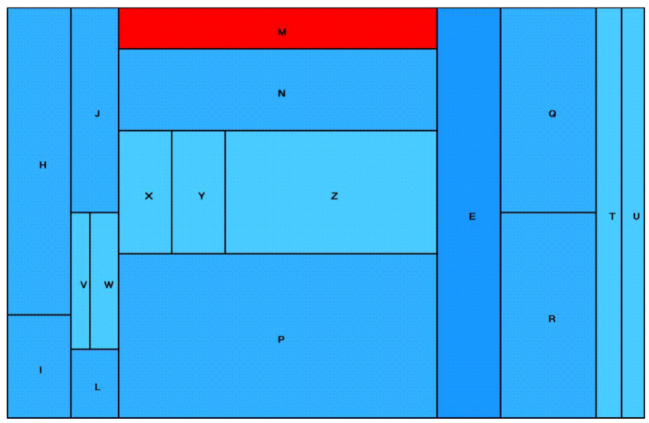
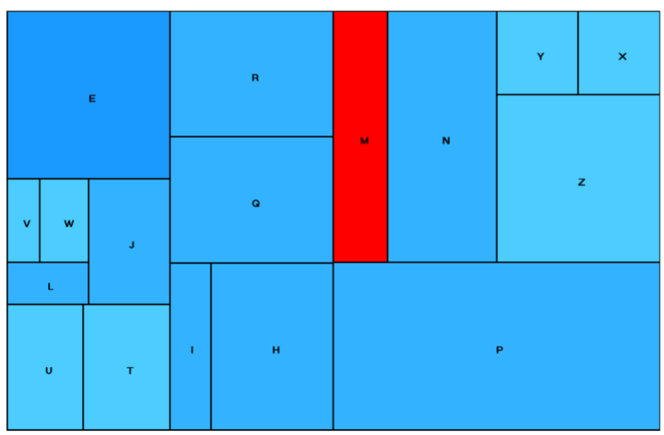
<u>Tasks</u>

- Explore topology, locate paths

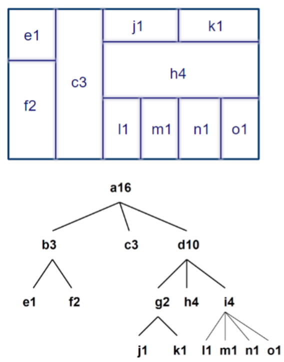
</td>
</tr>
</table>

Describe large network visualizations	**LARGE NETWORK VISUALIZATION** Large network visualization is the process of visualizing complex networks that have a large number of nodes and edges. These networks can be found in many different fields, including social networks, transportation networks, and biological networks. Due to the large number of elements, traditional graph drawing methods, such as force-directed algorithms, become impractical and inefficient.
Techniques to visualize large networks?	Large network visualization typically involves using techniques such as node-link diagrams, matrix-based representations, and hierarchical representations. These techniques are designed to reduce the clutter and complexity of the network, making it easier to understand and analyze.
Describe node-link diagrams and the techniques it uses to declutter	**Node-link diagrams**, for example, represent the network as a set of nodes and edges, where each node is represented by a symbol or icon, and each edge is represented by a line connecting the nodes. However, in large networks, the number of edges can become overwhelming, making the network difficult to interpret. To overcome this problem, techniques such as **edge bundling and edge clustering** are used to group similar edges together, reducing the number of edges displayed in the diagram.
Describe matrix-based representations (like adjacency matrices) 	**Matrix-based representations**, such as **adjacency matrices and heat maps**, represent the network as a matrix of values, where each cell in the matrix represents the relationship between a pair of nodes. These representations are useful for identifying patterns and clusters in the network, but they can be difficult to interpret when the number of nodes is large. **Visual adjacency matrix** • More scalable • $M_{i,j}$: existence/weight of edge i j • Maximize edge visibility, no crossing edges • Cons: difficult to see communities, paths, motifs • Note: sorting extremely important
Describe hierarchical representations (like treemaps)	**Hierarchical representations**, such as **tree maps** and dendrograms, represent the network as a hierarchy of nodes, where each node is grouped with similar nodes at the same level

	in the hierarchy. These representations are useful for exploring the structure of the network and identifying groups of related nodes.
Show the 3 kinds of visualizations for static trees	**TREES** **Visualizations for static trees:**  Node-link diagram Enclosure: Space-filling diagram Combined representations
Describe an enclosure	**ENCLOSURE** - An enclosure, in data visualization, is a technique used to represent hierarchical relationships between data elements by enclosing them within nested containers. A treemap is an example of an enclosure. - In general, enclosure diagrams are useful for showing the hierarchical structure and proportions of a dataset, but they can be difficult to interpret when the hierarchy is deep and/or the number of elements is large.
Describe a treemap	**Treemap - space filling technique** Treemap diagrams use nested rectangles to represent hierarchical data, where each level of the hierarchy is represented by a rectangle. • Long-thin rectangles can occur • Squarified layout generates nicer rectangles
What does a nested vs treemap look like	Nested treemap vs treemap  Nested Treemap Treemap

How does a tree translate to a treemap?	Tree vs treemap 
Show a treemap that also shows a time series	Treemap + time series  
What are some of the applications of treemaps	**Applications of treemaps:** Market share of companies in different industries such as communication services, healthcare, industrials, etc; answering questions like “what did singapore export in 2012?”; geographic data; business intelligence.

Node-link vs treemap



Treemap – Enclosure techniques
- Scalable: very good usage of available space – show attributes
- Difficulty in distinction of the hierarchy (implicit)

Node-link diagram
- Intuitive
- Good at exposing structure of information
- A lot of empty space

Describe dynamic networks and the 3 methods of representing dynamic networks

$$G = (V, E)$$
$$E \subseteq V \times V \times T$$

DYNAMIC NETWORKS - networks that change over time

Structural properties
- Communities, motifs

Temporal properties
- Trends, periodicity
- temporal shifts, anomalies

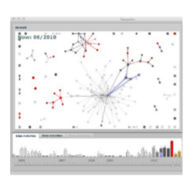
Discovery/exploration of states
- Characterizing the evolution of the network
- Stable states, recurring states, outlier states
- Transitions between them

Dynamic networks can be represented as a sequence of static networks (**snapshots - small multiples**) or as a continuous evolving network.

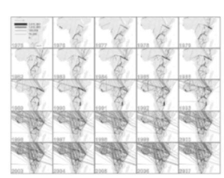
- In the first case, each snapshot represents the network at a specific point in time, and each edge or node has a start and end time.

- In the second case, the network is represented as a continuous process **using an animation**, and the edges and nodes are added or removed over time, and the topology of the network changes over time.

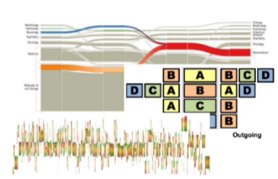
- And the last approach is to have an **integrated approach** such that time is mapped to space.



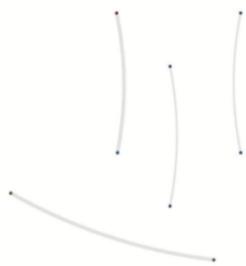
Animation
Map time to time



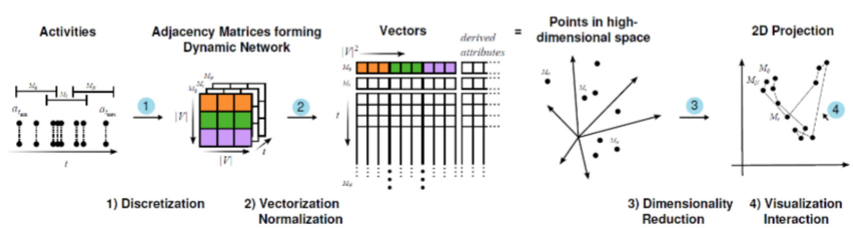
Small Multiples
Split time in intervals



Integrated approaches
Map time to space

Describe Animation as a way of representing dynamic networks 	**1. ANIMATION** Animation can be used to visualize the changes in the network structure, such as the addition or removal of nodes or edges, the alteration of edge weights or attributes, and the change in the topology of the network. For each timestep compute layout Improve: - Keep variation of the layouts as small as possible - Preservation of the mental map - Aggregation (of time) – super-graph. - Add timeline control
Pros and cons of using animation?	**Pros of using animation:** • Simple to implement • Easy to spot big changes • Applicable to all methods for which a single timestep can be visualized. **Cons of using animation:** • Need to focus on many moving or changing items simultaneously • Animation imposes a massive memory load when the amount of change between each frame is complex • Keep track of (multiple) changes over longer time periods. • Change blindness
Describe the phenomenon of change blindness and its implication on vis	**ANIMATION - Change blindness** - The phenomenon of change blindness is that we fail to notice even quite drastic changes if our attention is directed elsewhere. - Our eyes dart around, gathering information just in time for our need to use it, so quickly that we do not typically notice this motion at a conscious level. - The difficulty of tracking complex and widespread changes across multiframe animations is one of the implications of change blindness for vis.
Describe the method using small multiple to show dynamic networks	**1. SMALL MULTIPLES** Multiple views with the same encoding and different partitions of the data between them are often called small multiples. The shared visual encoding means that the views have a common reference frame so that comparison of spatial position between

	them is directly meaningful - Juxtaposed visualizations using a filmstrip or grid layout.
Pros and cons of small multiples	**Pros of using small multiples:** - Independent of the visualization method used - side by side, allowing the user to glance quickly between them with minimal interaction cost and memory load. **Cons of using small multiples:** - Decide on the number of multiples to use. - Operational limit on the number of multiples and display - Multiples might be far apart difficult to spot patterns
What does mapping time to space include and what are its pros & cons?	**Mapping time to space** Provide a static overview of the entire time span of the network in one visualization. Pros: • Complete overview of network is presented. • Global patterns can be directly identified. Cons: • Specialized visual encoding often difficult to interpret to non-experts. • Often restricted to specialized type of network (acyclic, directional, compound etc.)
Describe clusters over time in dynamic networks	**Dynamic networks - clusters over time** <u>Visualization</u>: • Show entire timespan • Scalable in both number of nodes and timesteps • Individual nodes and edges are not visible anymore. <u>Automated method</u>: • Cluster nodes and show change of clusters over time.
Describe massive sequence view in dynamic networks	**Dynamic networks - massive sequence view** <u>Visualization</u>: • Show entire timespan • Scalable in both number of nodes and timesteps • Clutter and overdraw • Difficult to see structure <u>Automated method</u>: • Optimize layout to reduce overdraw and emphasize patterns

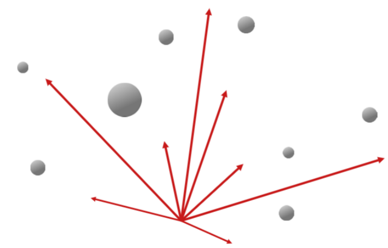
<table>
<tr>
<td>Describe visual adjacency list in dynamic networks</td>
<td>
Dynamic networks - visual adjacency list
<u>Visualization</u>:
• Show entire timespan
• Scalable in both number of nodes and timesteps
• Needs interaction, filtering, zoom-pan.
• Difficult to see structure, paths etc.
<u>Automated method</u>:
• n/a
</td>
</tr>
<tr>
<td>Describe parallel edge splatting in dynamic networks</td>
<td>
Dynamic networks - parallel edge splatting
<u>Visualization</u>:
• Show entire timespan
• Scalable in both number of nodes and timesteps
• Needs interaction, filtering, zoom-pan.
• Difficult to see structure, communities
<u>Automated method</u>:
• Optimize layout to reduce clutter
• Density computation and normalization
</td>
</tr>
<tr>
<td>Describe the method of reducing snapshots to points in dynamic networks and the 3 ways of doing it</td>
<td>
Dynamic networks - reduce snapshots to points
What if a network becomes really large?
<u>Discovery of states</u>
• Stable • Recurring • Outlier
Transitions between them


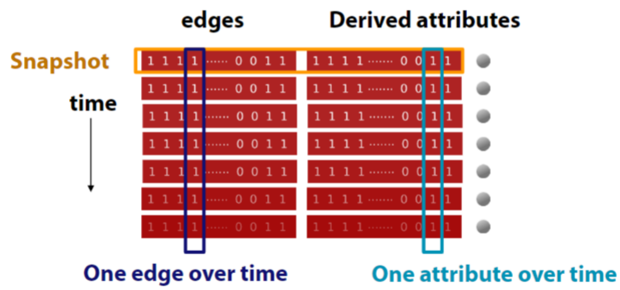
Discretization
<ol type="a">
Discretization is the process of converting a continuous variable into a discrete variable.
Techniques to do it: binning, decision tree discretization, clustering, etc.

Vectorization (Points in high dimensional space)
<ol type="a">
can be used to reduce a sequence of snapshots of a dynamic network to a set of points in a vector space. Vectorization can be used to represent a dynamic network as a sequence of vectors, where each vector represents the network at a specific time step

</td>
</tr>
</table>

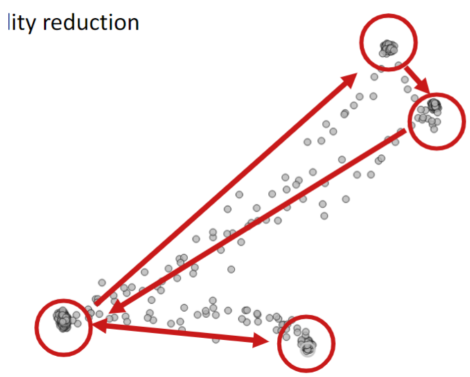
b. node-based/ edge-based/ matrix based (like adjacency matrix)

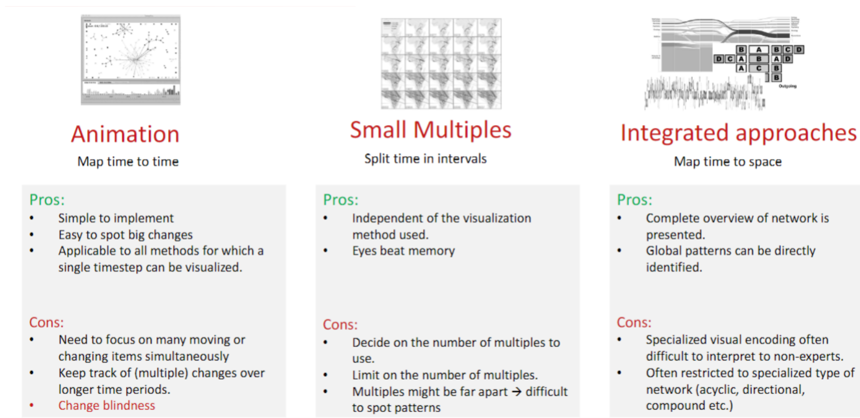




3. Dimensionality reduction

- PCA
- MDS
- t-SNE
- Umap





TIME SERIES

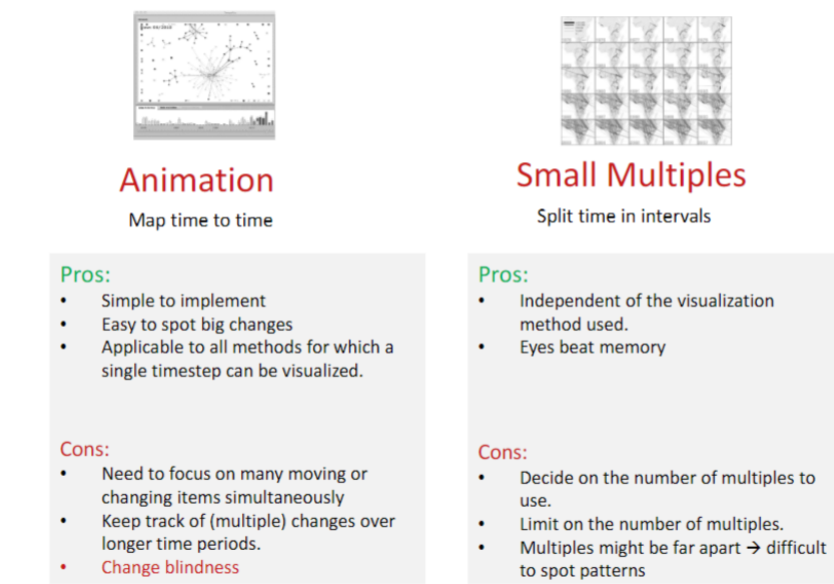
Recall:

2 kinds of time series:

Describe a line chart

Notes:

Animation and Small multiples

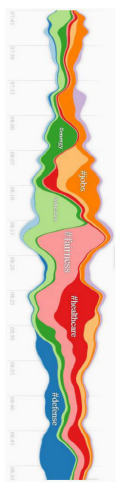


Line chart

Data	• 2 Quantitative attributes • One key, one value
Marks	Points, line connecting marks
Channels	• Aligned lengths to express quantitative value • Separated and ordered by key attribute into horizontal regions

Tasks	• Find trends • Connection marks emphasize ordering of items along key axis show relationship Scalability? Key attribute dozens to hundreds

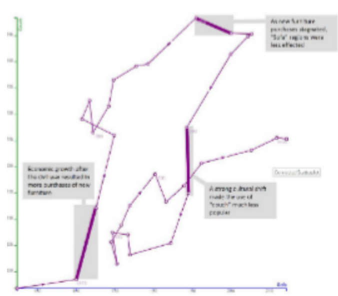
Describe a stream graph



Stream graph

Data	• 1 Categorical attribute (names) • 1 Ordered key attribute (time) • 1 Quantitative value attribute (counts)
Marks & channels	Derived geometry: layers, height encodes counts
Tasks	• Find trends • Part-to-whole relationship Scalability? Hundreds of time keys Dozens to hundreds of (names) keys • More than stacked bars, since most layers don't extend across whole chart

Describe a connected scatterplot



Connected scatterplot

Scatterplot with line connection marks

• Popular in journalism

• Horizontal + vert axes: value attributes

• Line connection marks: temporal order

Empirical study

• Engaging, but correlation unclear

Describe a gantt chart

Gantt chart

Data	• 1 Categorical attribute • 2 Quantitative attributes • One key, two (related) values
Marks	Line, length duration
Channels	• Horiz. Position: start/end times • Horiz. Length: duration
Tasks	• Emphasize temporal overlaps, start/end dependencies between items Scalability? Dozens of key levels, hundreds of value levels