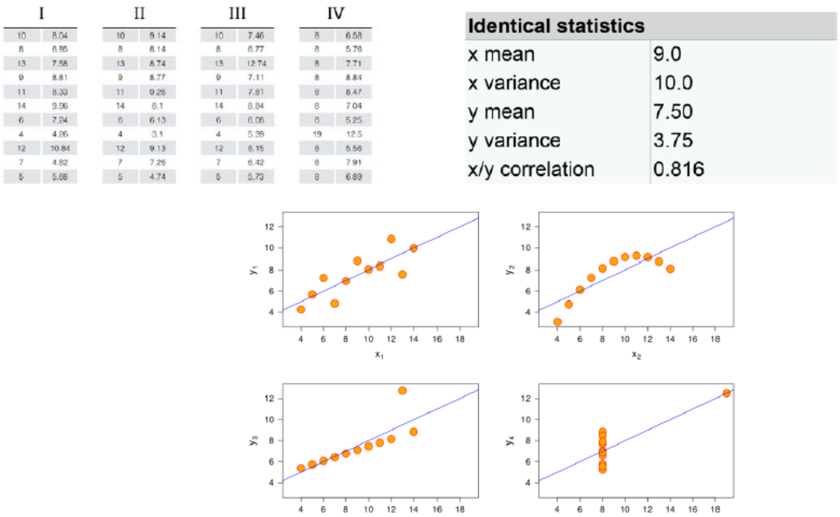
Lecture 1 - Introduction

Book	☐
Completed	☑
Date	@November 17, 2021
Notes	Chapter 1
Type	Lecture
Week	1

- Numbers do not tell the whole story



- **Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively**
- For well-defined questions, computational techniques from statistics and machine learning often suffice
- Many analysis problems are ill-specified; it is unclear what questions to ask upfront or how to answer them
- **We use visualization to augment human capabilities**

Data Visualization

Goal - a cognitive process used to gain understanding: insight

- Provides useful images (graphical illustrations)
- Graphical approach is necessary as data continuously increases in size
- A lot of knowledge used comes from infographics and business graphics (e.g. Excel)

3 Types of Goals in Visualization

1. to explore

- nothing is known
- visualization used for data exploration

2. to analyze

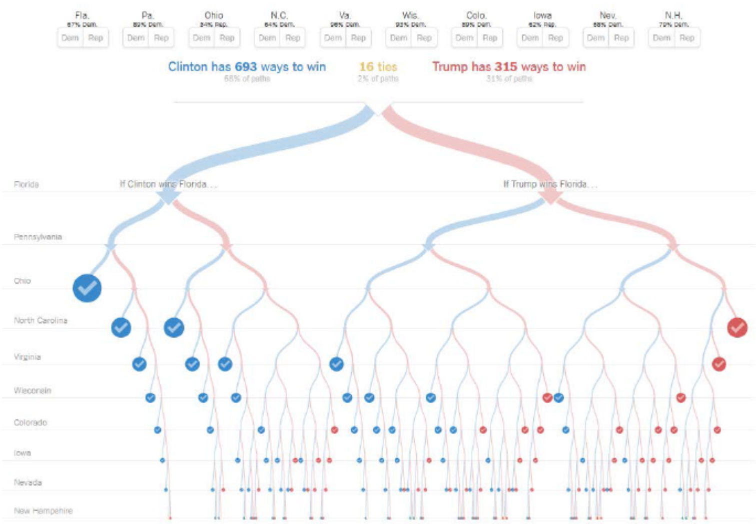
- there are hypotheses
- visualization used for verification or falsification
- e.g. doctors taking scans to be sure if there is lets say a tumor

3. to present

- showing "everything" that is known about the data
- visualization is used for communication of results

- We find non-spatial data everywhere (infographics)
    - for example, coordinates

- Interactions play an important role in understanding the data.



- Data Visualization is a multidisciplinary field of study, entailing of other domains like:
  - optimization
  - application domain
  - numerical analysis
  - perception
  - computer graphics
  - image processing
  - infographics
  - machine learning
  - algorithmics
  - HCI

Visualization field: InfoVis vs. SciVis

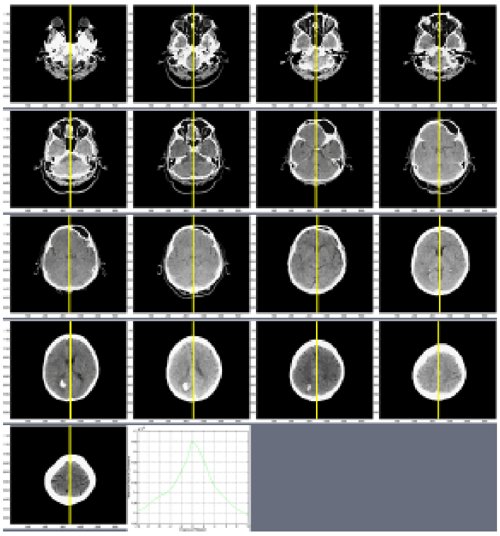
- So information visualization vs. scientific visualization (this course only focuses on Information Visualization)

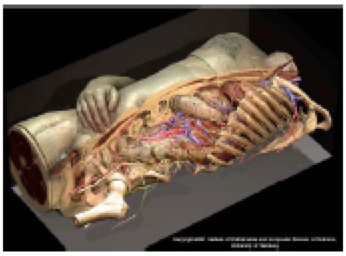
InfoVis	SciVis (VolVis/MedVis/FlowVis...)
Abstract Data No spatial reference.	Spatial Data Spatial reference.
N-dimensional. Heterogeneous.	Mostly 2 or 3-dimensional
Numerical, text, images, multimedia.	Scientific, engineering, biomedical.

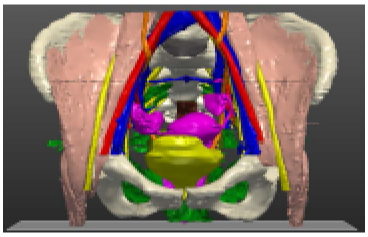
Scientific Visualization Spatial Data (representing physical meaning on space) -

EXAMPLES:

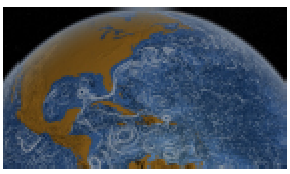
- Medical data: MedVis-VolVis!,

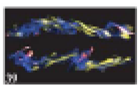


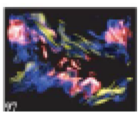


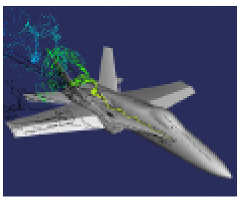


- Flow data: FlowVis!,









- GIS data,
- Microscopic data (molecular physics),

- Macroscopic data (astronomy),

Visualization Design

- There is a huge space of design alternatives (there are a lot of options) and obviously, interactions.
  - Tradeoffs abound
- Many possibilities are known to be ineffective
  - avoid random walk through possibilities
  - avoid some of known mistakes
  - extensive experimentation has already been done
- Guidelines continue to evolve
  - we reflect on lessons learnt in design studies
  - iterative refinement usually wise

The design process

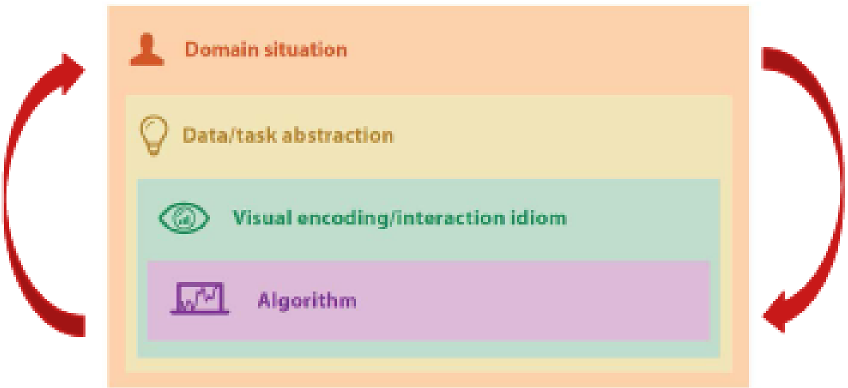
- By doing an analysis of the problem, we can efficiently exclude a lot of clearly ineffective methods.
- Similarly, out of the selected ones, experimentation can help finetune to one solution.

- **Validation is difficult** since its hard to evaluate if the design works (better? faster? more effectively? get more insight?) or what sort of tasks are required to evaluate a system (intended users? benchmark datasets?)

Lecture 2 - Design Space (What/Why) | Visual Encodings | Perception

Book	☐
Completed	☑
Date	@November 19, 2021
Notes	Chapter 2,3,4
Type	Lecture
Week	1

Visualization Design (Nested Model)



- an iterative/refinement process (not linear!)
    - each part/stage depends on the previous level e.g. what happens in data abstraction is largely impacted by what happens with the domain situation

Domain situation

— problem categorization

- understand the user, data and the tasks (what users want to do with the visualization tool)

Knowing your users:

- what are their needs/wants/limitations/skills?
- what is their workflow? and how your tools helps it or integrates within?
- how to provide **actionable knowledge?**
  - what decisions
  - which information is relevant
- How to satisfy the users?

Knowing the data

- exploration of data
  - looking for patterns, missing values, size of data, relevant parts, and other useful aspects

Use domain specific vocabulary

- For example, for us, it would be cars/accidents related vocabulary

Create set of tasks/questions of target users on the data, on different levels

- For example, a user would want to know what the geographical distribution of accidents looks like. So similar questions on different aspects of the data

Information is usually obtained through interviews, observation, reading or general interactions with users.

Data/Task Abstraction

- We take all the tasks and data from our domain and translate it into a more generic, universal language
- Using generic terms for the data, like:
  - table, hierarchy, sets, numerical data, etc. instead of specific attributes
- Using generic terms for tasks, such as:
  - search, compare, observe trends, etc.

Visual Encoding

(usually the most creative part)

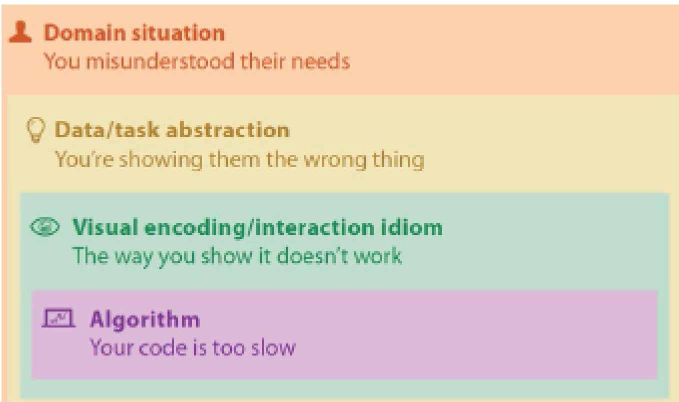
Designing the visual space and selecting visual encodings

- How to display the data to the user
- What techniques to use to make the representation more informative and effective with respect to the task
- How would the visualizations look like, e.g. what kind of plots

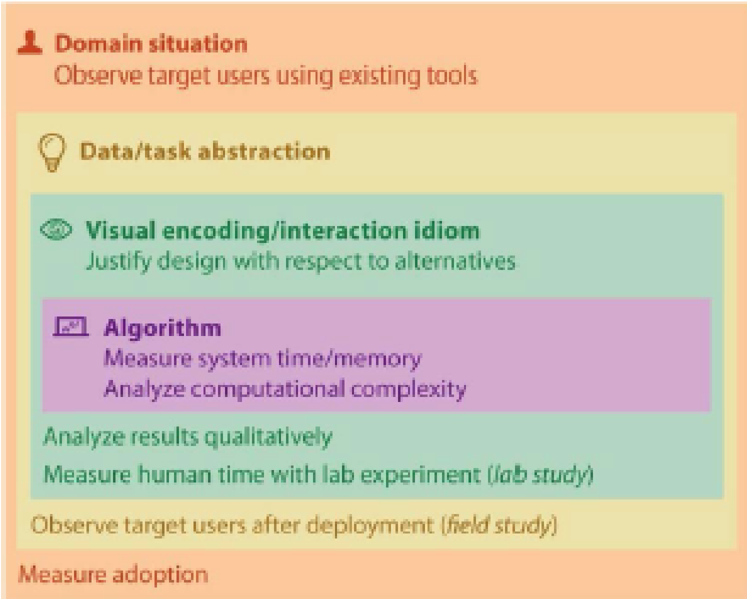
Algorithm

- this level is about implementing the design from the visual encoding stage (using frameworks, algorithms, etc.)
    - the coding part, basically

- Since this is a nested model, one major concern should be **constantly correcting for mistakes**
    - Because a mistake at higher level can not be corrected at a lower level



Validation/Evaluation in the nested model



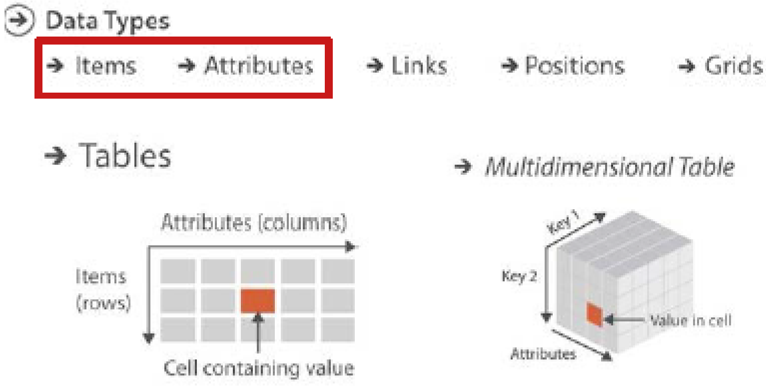
- We can check at level 3, if our visual encoding match our data tasks
- At level 2, we can check if the tools is working or not or how well it is working; is it functioning as planned?
- At level 1, we can check if the tool is suited/adapted to the domain; is it being used in daily life?

Analysis framework

- **What** is shown? (data abstraction)
- **Why** is the user looking at it? (task abstraction)
- **How** is it shown? (visual encoding and interaction)

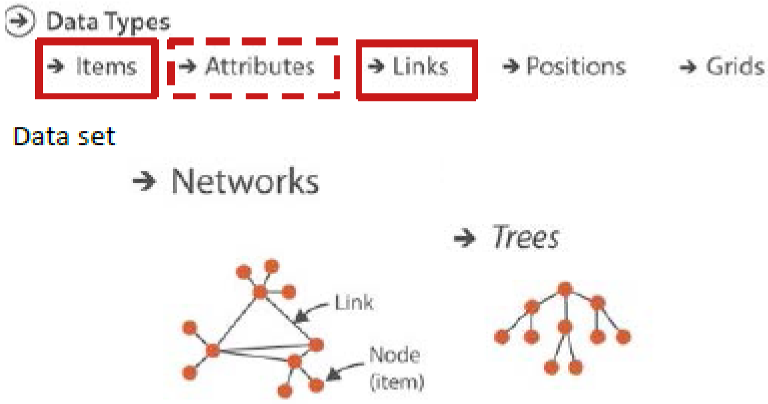
WHAT

Munzner's categorization — DATA TYPES

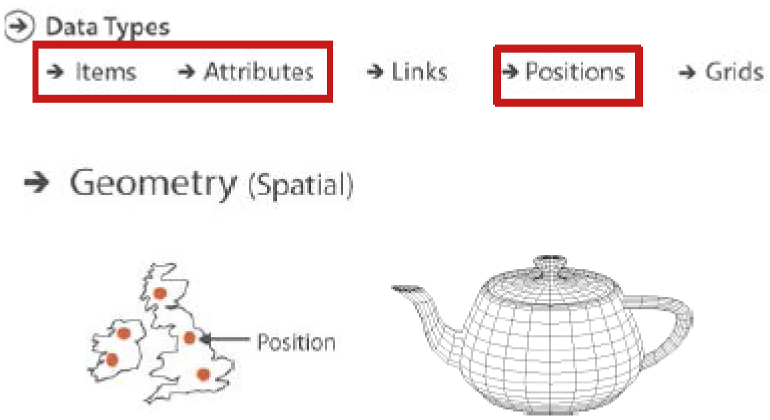


Keys help identify each item uniquely

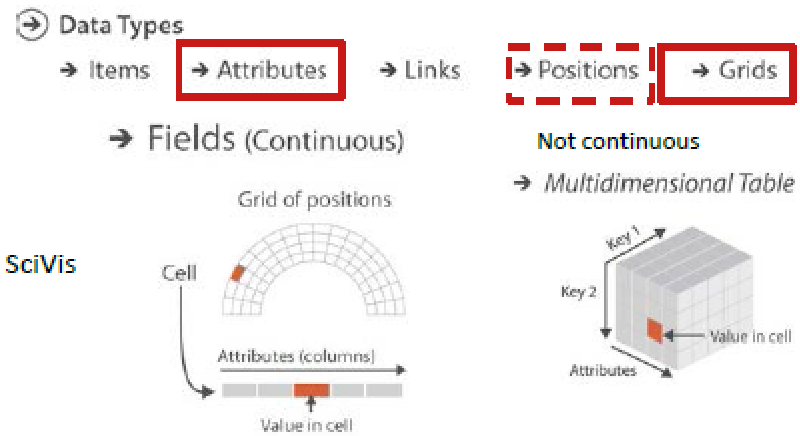
- In networks, there are items and links between them
    - each item may also have an attribute
- Trees represent hierarchical data usually



- With geometric objects (spatial data), items and attributes are also associated with positional data

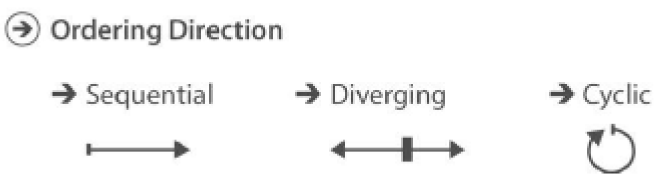


- Data type ‘grid’ ensures continuity in data and therefore, represents a continuous domain
  - the aspect of continuity differentiates between field data and multidimensional tables



> We mainly focus on **tabular data**.

__Munzner’s categorization — ATTRIBUTES__



- Within ordered data, the 2 types are:
  - **Ordinal:** for example, small, medium, large
  - **Quantitative:** numerical values (only if they are meaningful) that can be ordered

- **Sequential ordering** refers to unidirectional ordering where you can only go in one direction (e.g. ascending or descending)
- However, in **diverging ordering,** you can go in both directions. e.g. temperature at 0 degree or taking population average for a particular attribute so that it can be used as reference for upper or lower values
  - so, the mid point should be meaningful (like in the average example)
- **Cyclic ordering** could refer to something like time in hours, days of the week, months, etc.

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

ID is categorical here since it does not really mean anything.
Name - categorical, age - quantitative, size - ordinal, and fruit - categorical

- **Ordering direction could also depend on the viewpoint**, for example date can be cyclic if you are looking at weekly patterns and it can be sequential if you are interested in the future or just the past

Dataset availability —

Offline (most cases)

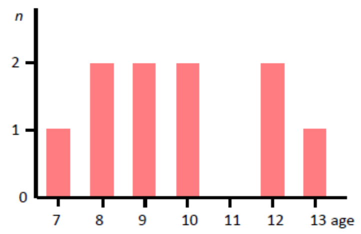
vs.

Online (streaming — time varying)

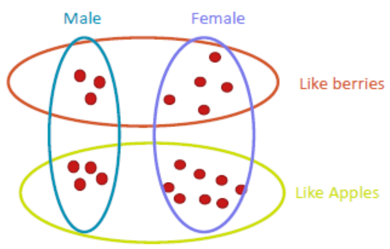
- new add/remove/change attribute and items
- for example, financial data, sensor data, etc.

Categorizations — starting point

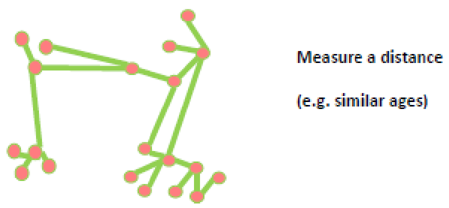
- Real data can be more complex than the presented basic types
  - could be sets/groups, combinations of previously defined datasets, or even new types
- A dataset should be observed in a different way as well, for example **tables for multivariate data**
  - One dataset, multiple views
- The following methods can be considered for a start:
  - **distribution per attribute**



  - **set**



  - **network**



WHY

Domain tasks

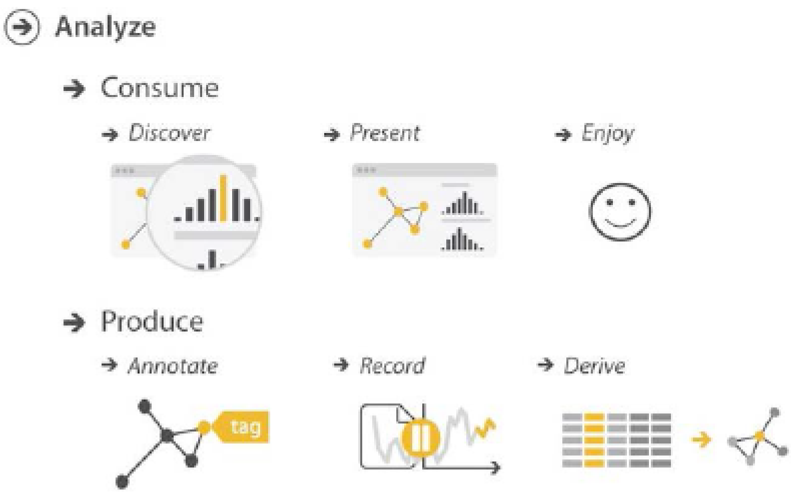
- domain tasks are defined at first

1. Relevant — do they really specify what is important for the goal?
2. Functional, not solution driven
    - OK — user must be enabled to understand the variation in attribute distribution, for example.
    - Not OK — Just showing a histogram
3. Is it complete? (kind of a hard to know so just try to be as complete as possible)

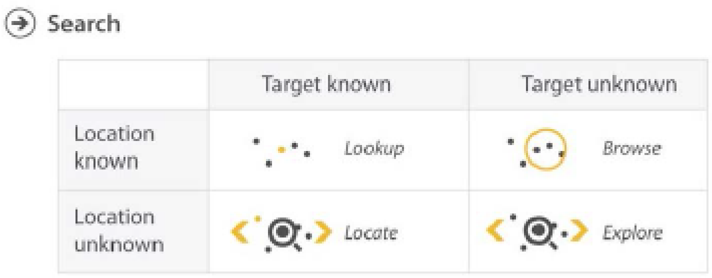
Actions

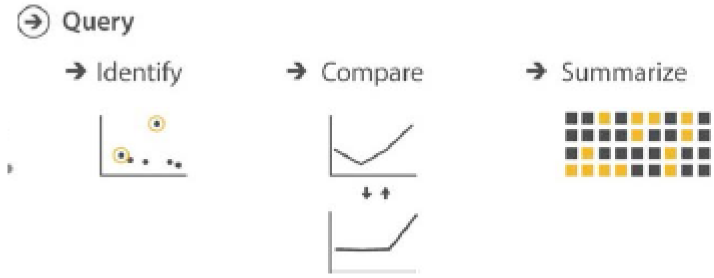
<u>Target —</u> aspect of the data that is interesting to the user

- <u>Types of tasks/actions :</u>



- Discover covers exploration (formulation of hypothesis maybe or even validation of one)
- In 'Produce new data', annotate covers using visualization to label data. Derive refers to using data to produce new data attributes, through visualization.
- ^These are **higher level goals**

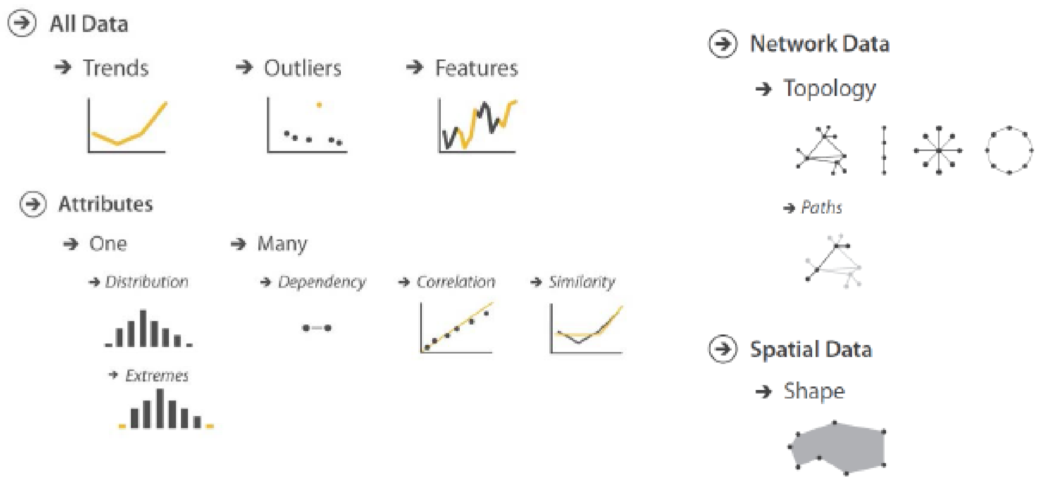




- Search is organized depending how much knowledge the user already has.
- Query is a low-level task, and therefore carries out more basic tasks.

Targets

- <u>Examples of targets:</u>



- Targets that exist for network data are less relevant since we are focusing on tabular data

{action, target} pairs — used in abstraction

- discover distribution

- compare trends
- locate outliers
- browse topology

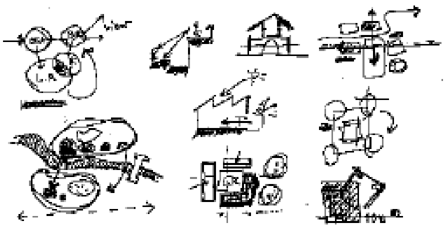
Generating solutions

- Explore design space

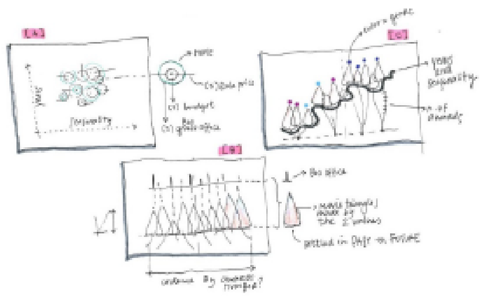
Approach

- Generate as many visualization as you can, as fast as you can
- do not be critical at first
- use sketching
- since it is a nested model, be careful for errors

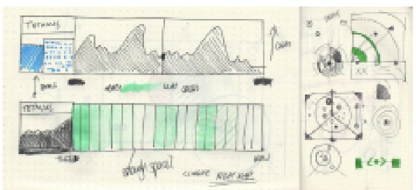
Sketching



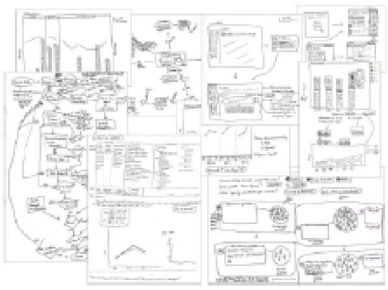
http://depts.washington.edu/napkin/drawing.html



https://medium.com/accurat-studio/sketching-with-data-opens-the-mind-s-eye-92d78554565



http://beyondwordsstudio.com/our-work/global-health-check/



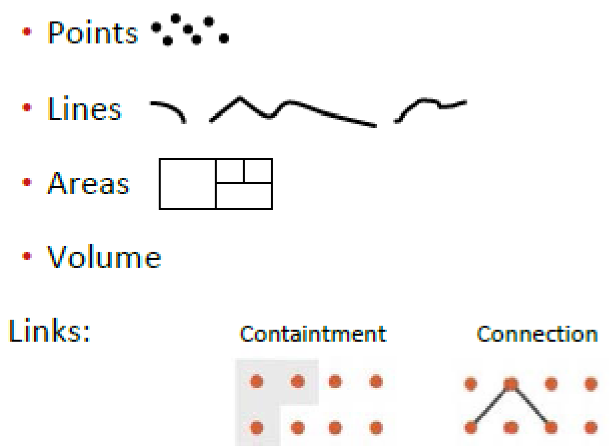
http://karminmauritz.com/data-viz-for-group-health-research-institute/

- find solutions
- find structure
- find and break constraints

HOW

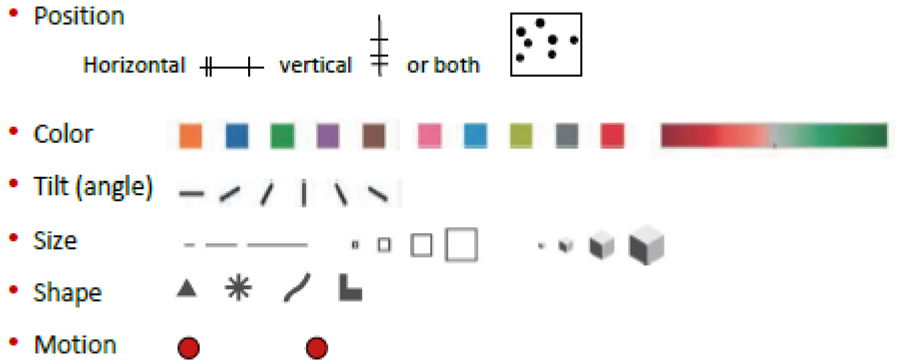
Marks and channels

Marks — geometric primitives



Links are types of marks

Visual channels — control appearance of marks



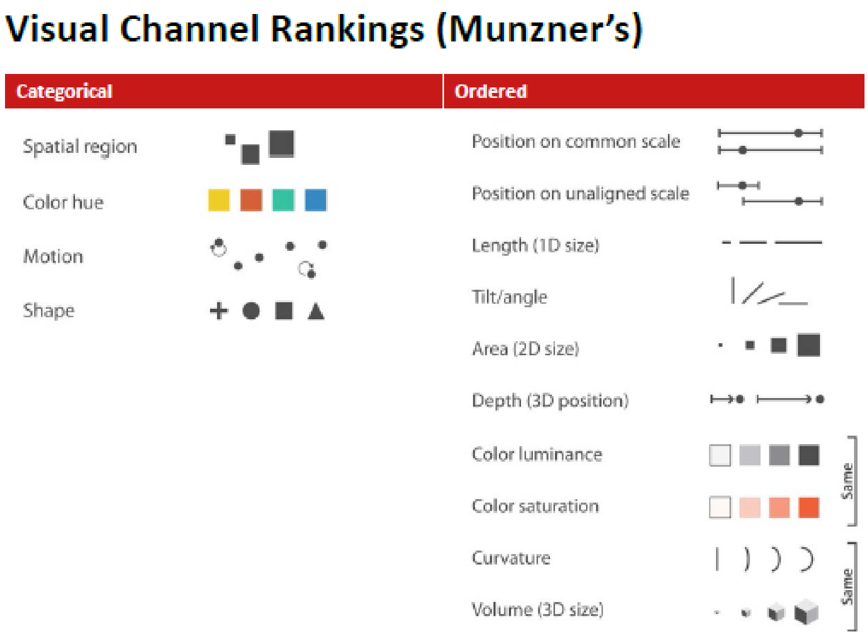
How can we select visual encodings?

Expressiveness principle

- show all but only what is in the data
- match the channel/mark to data characteristics
    - visual of a fruit for example; shape, motions, color, etc.

Effectiveness principle (SALIENCE)

- encode most important attributes with the highest ranked channels



Rankings for the 2 types of data

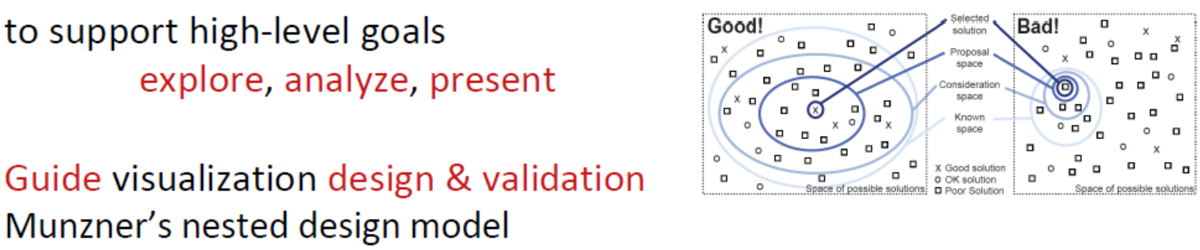
Rankings based on:

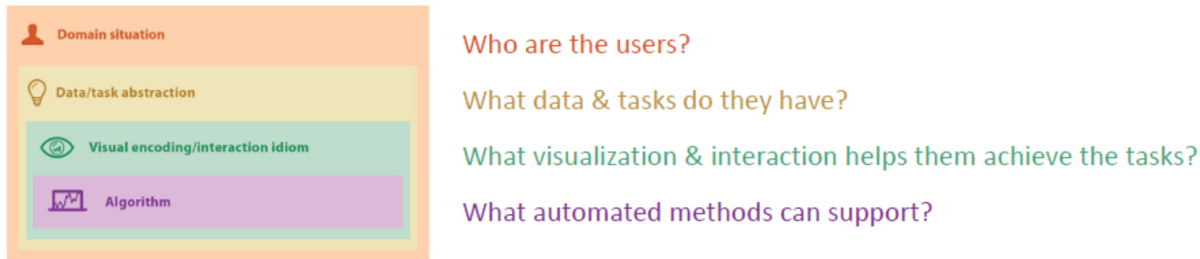
- accuracy, discriminability, separability, pop-out (refer to the human ability to process in parallel and not sequential)

Lecture 3 - Discussion on tasks and data analysis

Book	☑
Completed	☑
Date	@November 24, 2021
Notes	-
Type	Discussion
Week	2

Summary

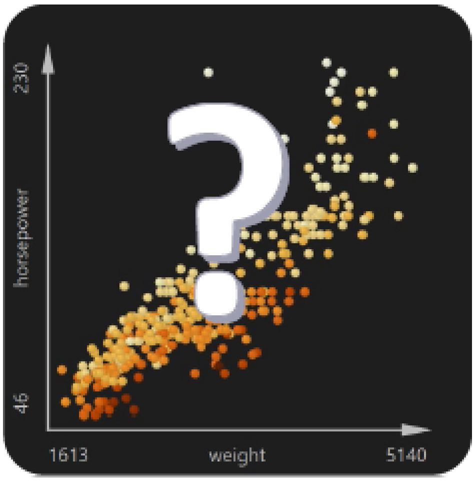




Data and Task Abstraction

Visualization

Is this a good visualization?



- Depends on the context!

Who is it for? → user (domain expert)
What data do we have? → Data abstraction
To do what? → Task abstraction

Provides a language to describe data in a way that is meaningful and useful to visualization design.

Helps visualization designer to reason about a right visualization (encoding).

- The domain expert speaks a domain language

Data Abstraction

- Data is typically described with domain language, for example:

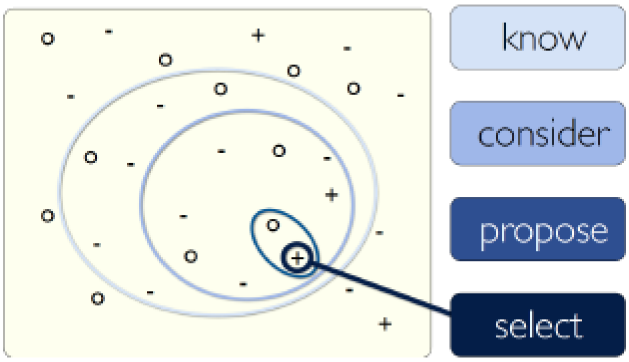
Interactions between proteins
Connection between terrorists
Transaction between bank accounts

Network data

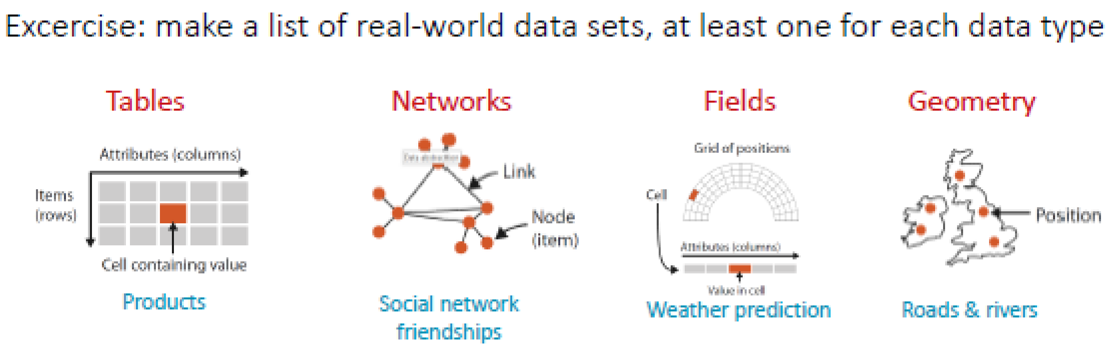
Tracking of animals
Election results in municipalities
Simulation of air turbulence

Spatial data

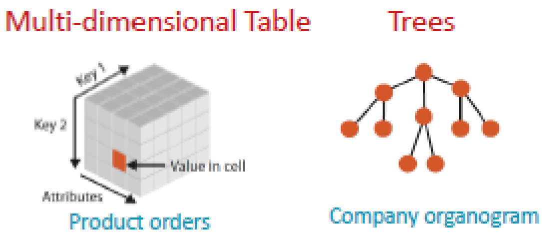
- But in order to search for suitable visual representations, we need to translate them into more abstract structures that we know how to encode
    - this narrows down the design space



Data types — exercise

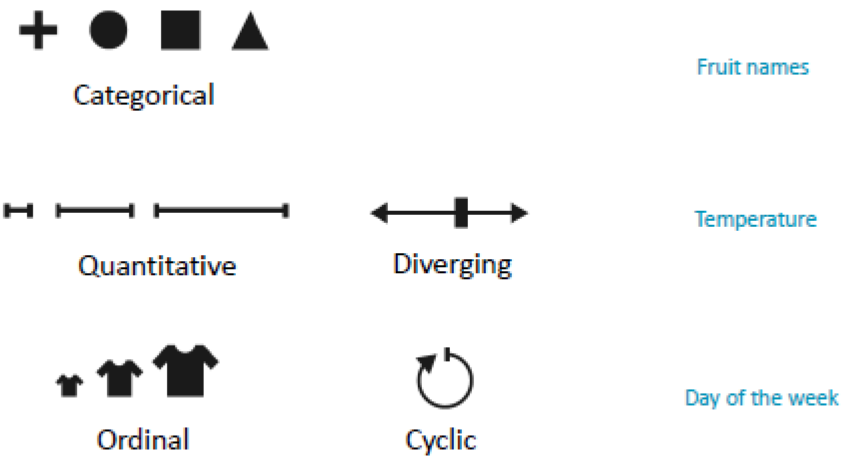


- Difference between field data and geometrical data is that in field data (which is spatial), you should be able to extrapolate but in case of geometrical data, that is not possible.
- Geometry can be represented by a network but its not always possible to map geometric data using networks. This because networks are abstract so can't really have a location.



Data type attributes

Examples - (quantitative isn't always diverging)



SUMMARY ON DATA ABSTRACTION

- provides a **language** to describe data in a way that is meaningful and useful to visualization design
- In order to search for suitable visual representations, we need to **translate domain language into more abstract structures** we know how to encode.

Task Abstraction

Why do we need it?

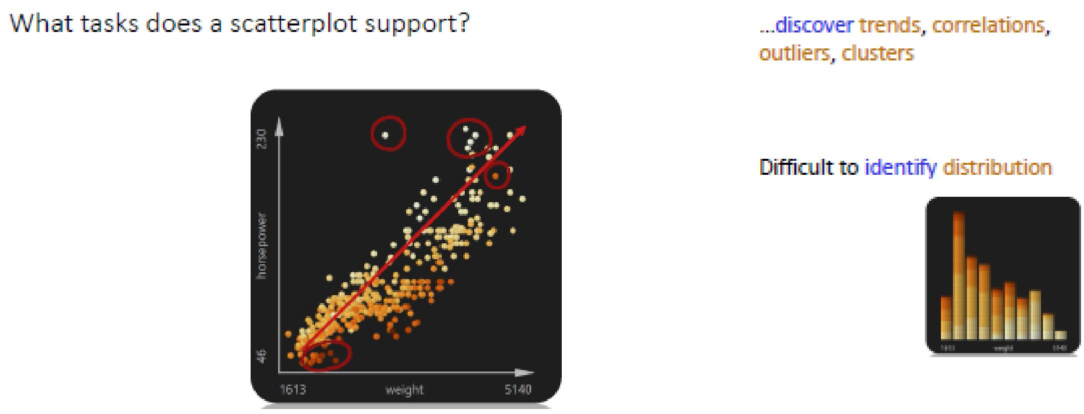
- Helps visualization designer to reason about a right visualization encoding
- Transforms domain specific language into abstract form to reason about similarities & differences
- Example tasks —

> "contrast the prognosis of patients who were intubated in the ICU more than one month after exposure to patients hospitalized within the first week"
>
> "see if the results for the tissue samples treated wih LL-37 match up with the ones without the peptide"

  - Now these tasks compare values from two groups.

- However, that and how to visualize it is not clear from reading it like this. **(Need to turn this into abstract form)**

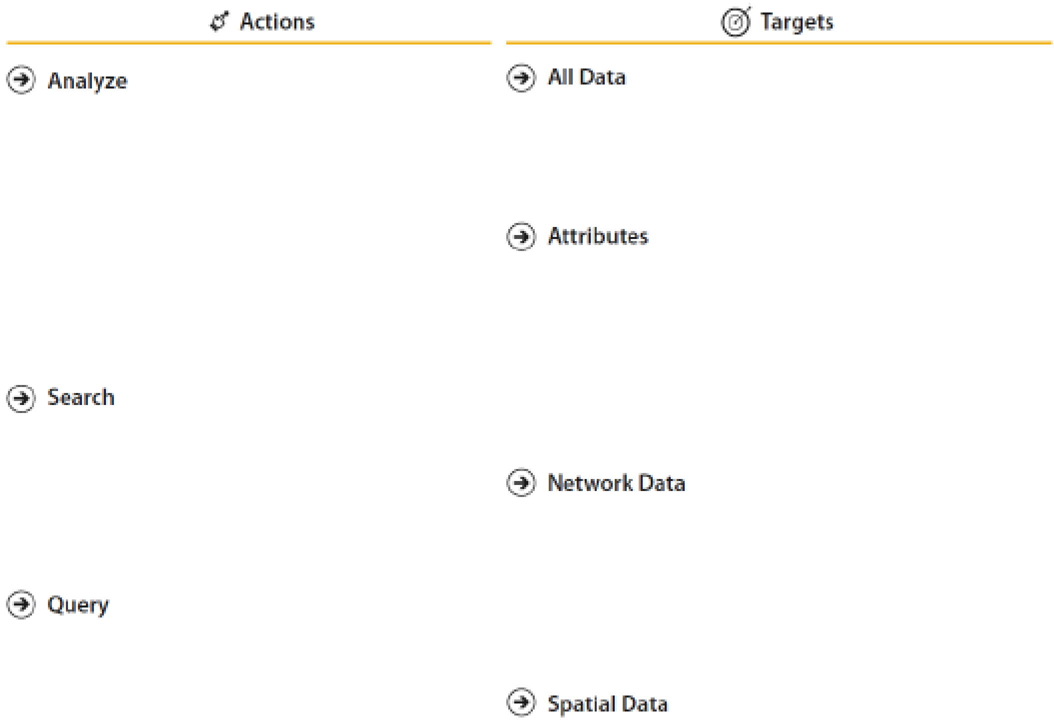
- **Tasks provide a constraint on design (& vice versa)**
  - No visual representation supports all tasks
  - Different for users, one might work in one situation and not the other
  - Depends on the <u>goal</u> of the visualization
  - Example —

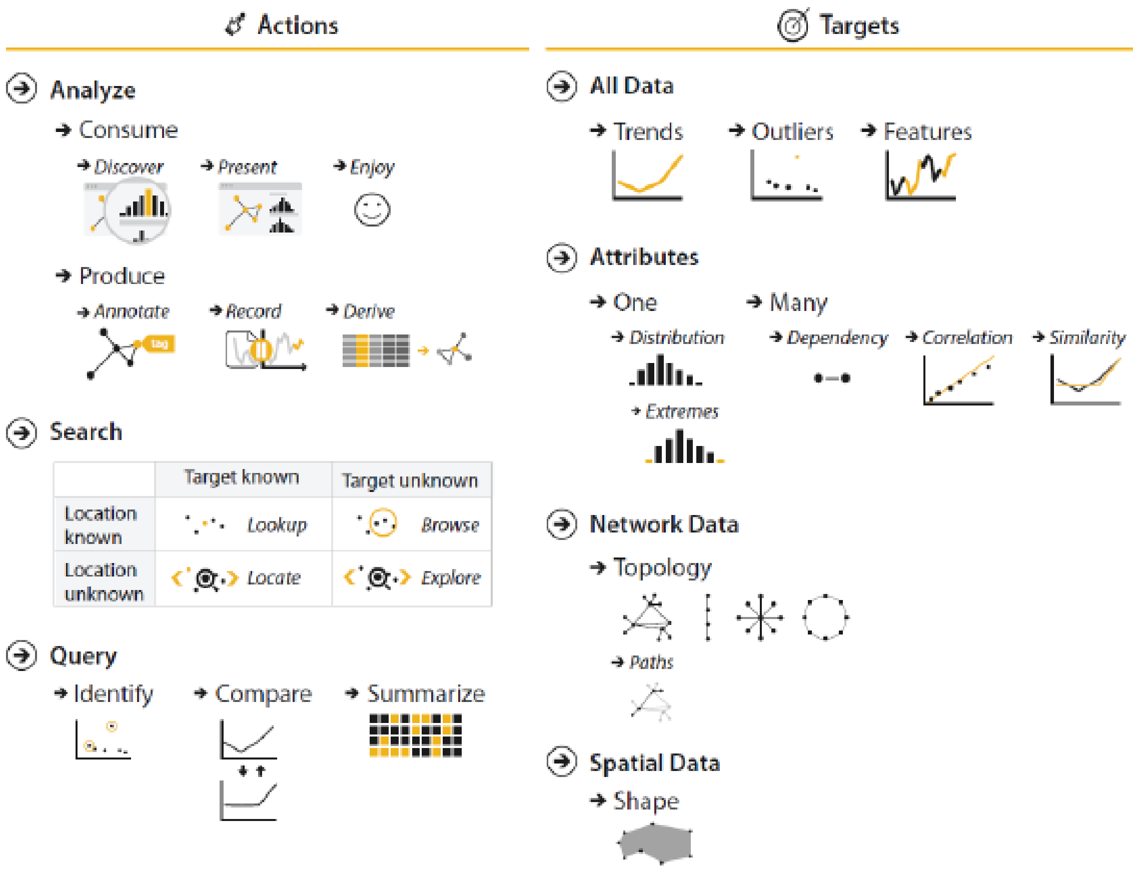


- A histogram can be used to identify distribution

- The design you choose influences the tasks a user can do
- For every visualization you design, ask yourself:
  - what information can I extract out of this representation.
  - what is the problem that I am trying to solve?
  - what questions do I want my user to be able to answer looking at this?
- Task abstraction helps achieve this in a more structural way —

- **Task is a tuple** (Action & Target)

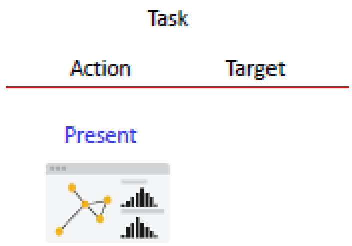




Task abstraction examples

<u>EXAMPLE 1 — </u>

User: data scientist
Task: *"I am developing machine learning classification models and I need to convince the executive board that my time is well spent because the performance of my model keeps getting better."*



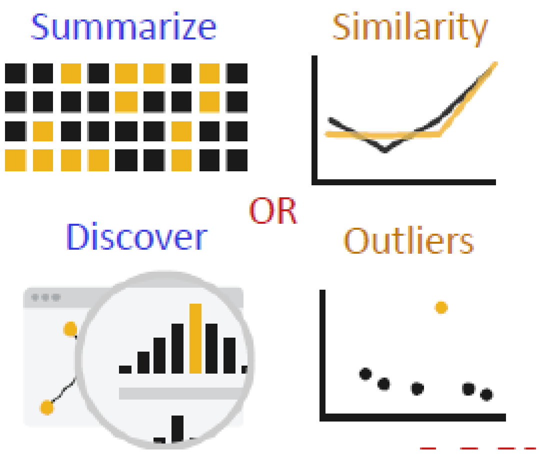
User: data scientist
Task: *"I am developing machine learning classification models and I need to convince the executive board that my time is well spent because the performance of my model keeps getting better."*

EXAMPLE 2 —

User: bank employee
Task: *"I want to find out if money laundering structures are happening in our financial transaction data."*

EXAMPLE 3 —

User: quality assurance manager
Task: "I need to understand if all our products are of the same quality."



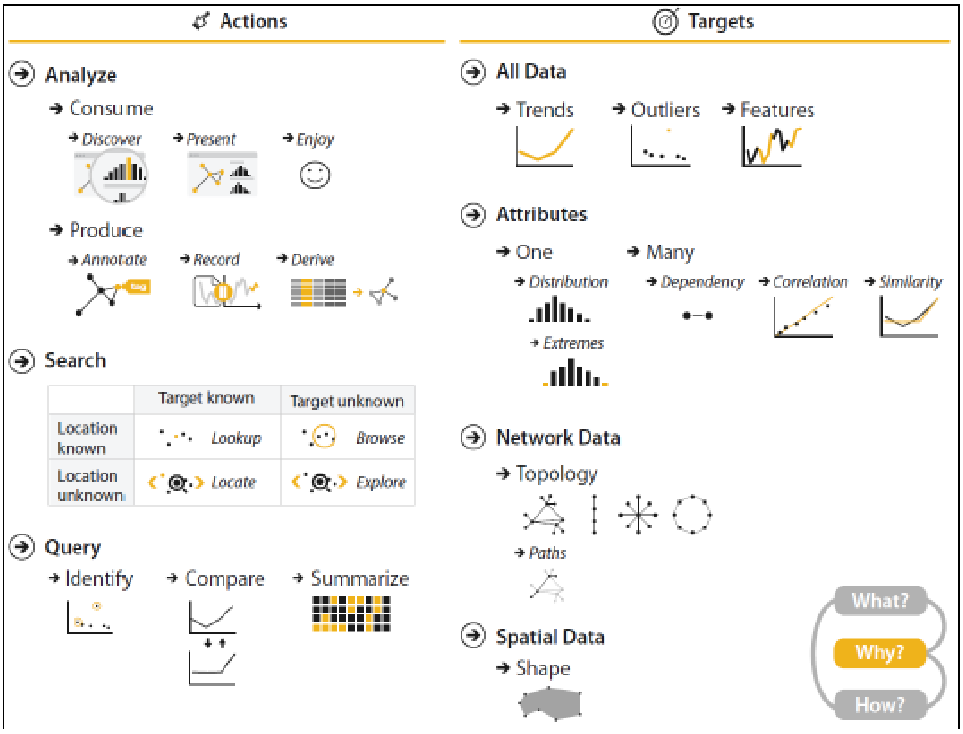
Task Abstraction Exercise

User: Movie enthusiast
Task: *"I am interested in making a short list of horror movies that I want to view that are highest rated by users."*

User: Investor
Task: *"I want to contrast the prognosis of stocks of the past year in the technical domain to the entertainment industry to determine where I can best invest in."*

User: Biology scientist
Task: *"We measured for three different groups of penguins on Antarctica the culmen length, culmen depth, flipper length and body mass, we are wondering if there is a difference between the species, and on what level."*

User: Virologist
Task: *"I need to find the municipalities in Noord-Brabant with the highest number of Covid infections to prevent further spread."*



- <u>Additional:</u> with Identify, you already know what you are looking for. So it would be kind of the opposite of exploration.

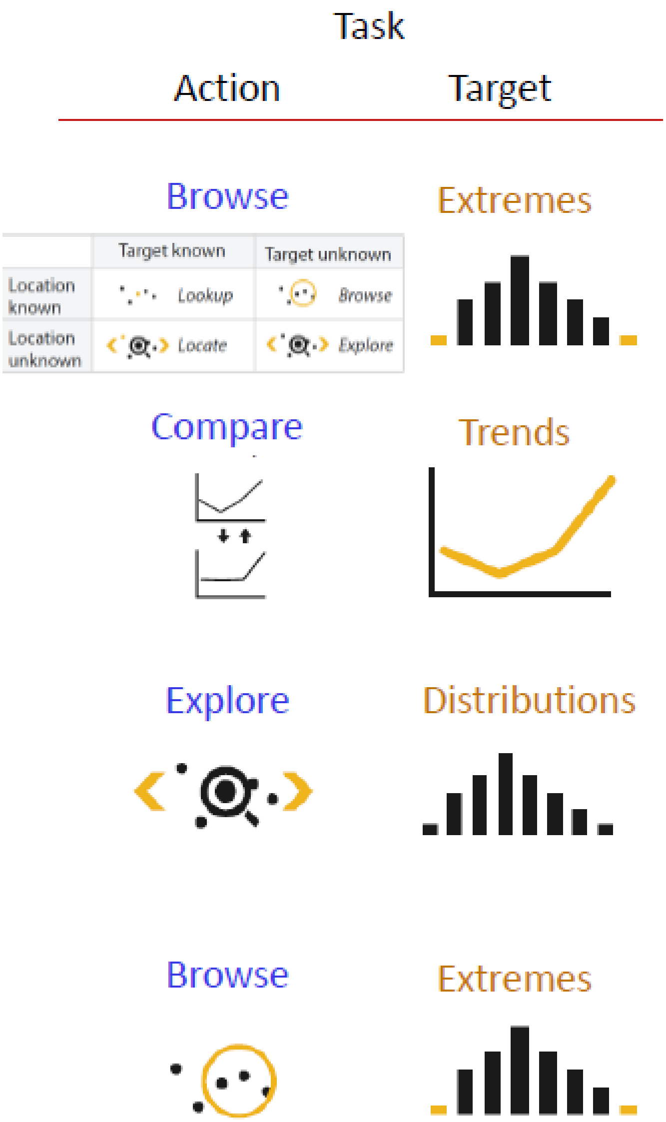
<u>Tip</u> — Look for keywords

User: Movie enthusiast
Task: *"I am interested in making a short list of horror movies that I want to view that are highest rated by users."*

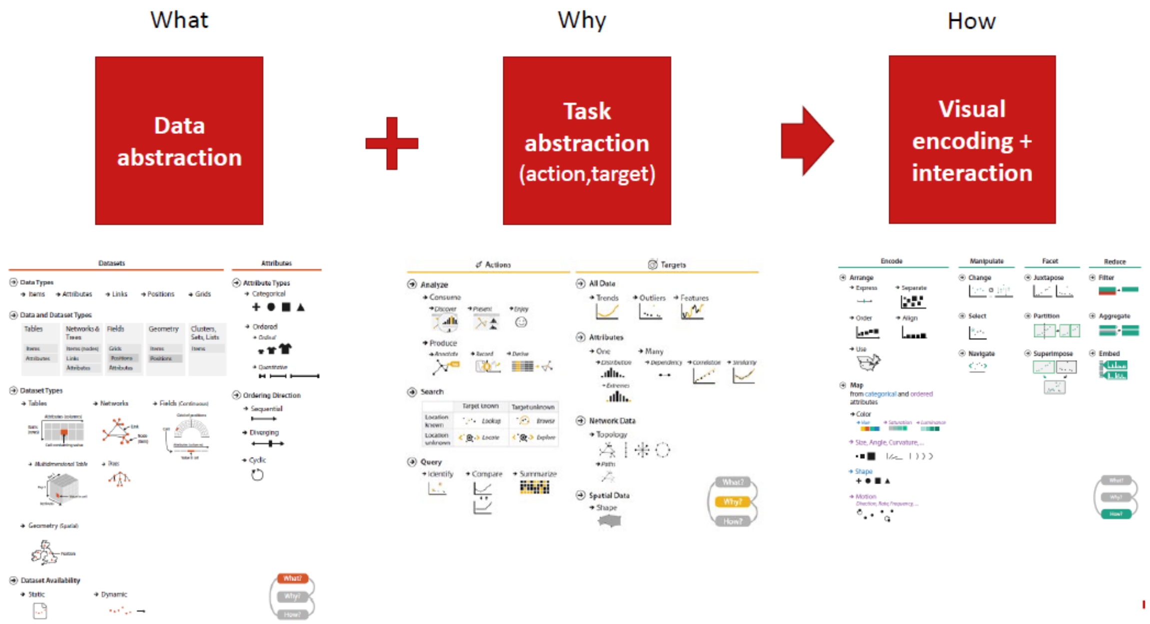
User: Investor
Task: *"I want to contrast the prognosis of stocks of the past year in the technical domain to the entertainment industry to determine where I can best invest in."*

User: Biology scientist
Task: *"We measured for three different groups of penguins on Antarctica the culmen length, culmen depth, flipper length and body mass, we are wondering if there is a difference between the species, and on what level."*

User: Virologist
Task: *"I need to find the municipalities in Noord-Brabant with the highest number of Covid infections to prevent further spread."*



- For the first one and the fourth one, target could also be 'Outliers' since we have one variable
- For the third one, {action, target} could be Compare Similarities and also Compare Correlation (when each variable is checked differently for correlation).



SUMMARY OF TASK ABSTRACTION

- Transform domain specific language into abstract form to reason about similarities & differences
- Helps visualization designer to reason about a right visualization(encoding).
- Not complete, complex operations, combinations, sequences…
  - Since there are multiple correct answers for a problem

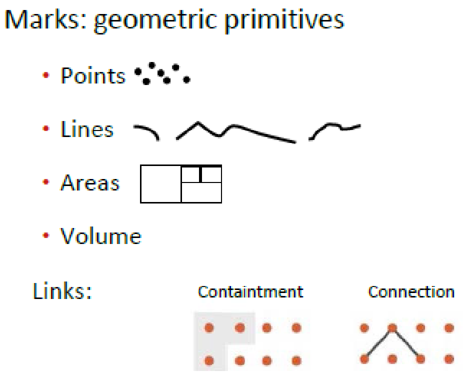
Lecture 4 - Visual encodings | Perception | Rules and principles

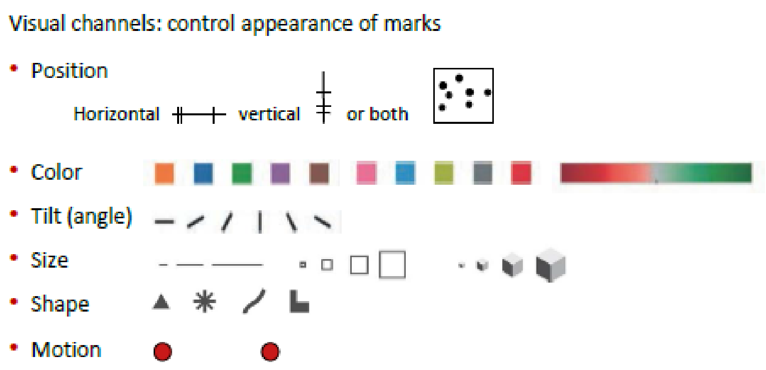
Book	☐
Completed	☑
Date	@December 1, 2021
Notes	Chapter 5,6
Type	Lecture
Week	3

Project: User / Goal Description

- Define user for the whole visualization tool (could be let's say law-makers). (No workflow analysis needed)
- Goal should be high-level for the whole tool. After that, you divide the goal in specific tasks. — Munzner's high-level tasks.

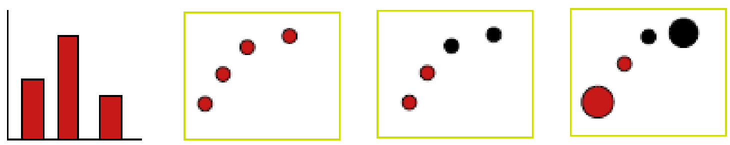
Marks & Channels





Visual Encoding

- Analyze as combination of marks and channels showing data attributes

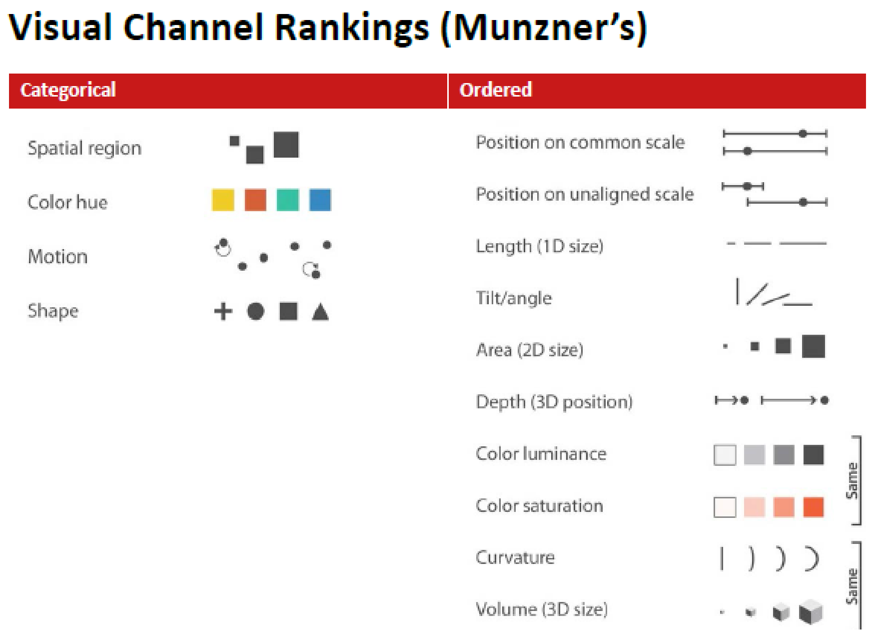


Channel	Length (Vertical pos.)	Vertical pos. Horizontal pos.	Vertical pos. Horizontal pos. Color	Vertical pos. Horizontal pos. Color Size
Mark	Line (Bar charts)	Point (Scatter plots)	Point	Point

How do we select Visual Encodings?

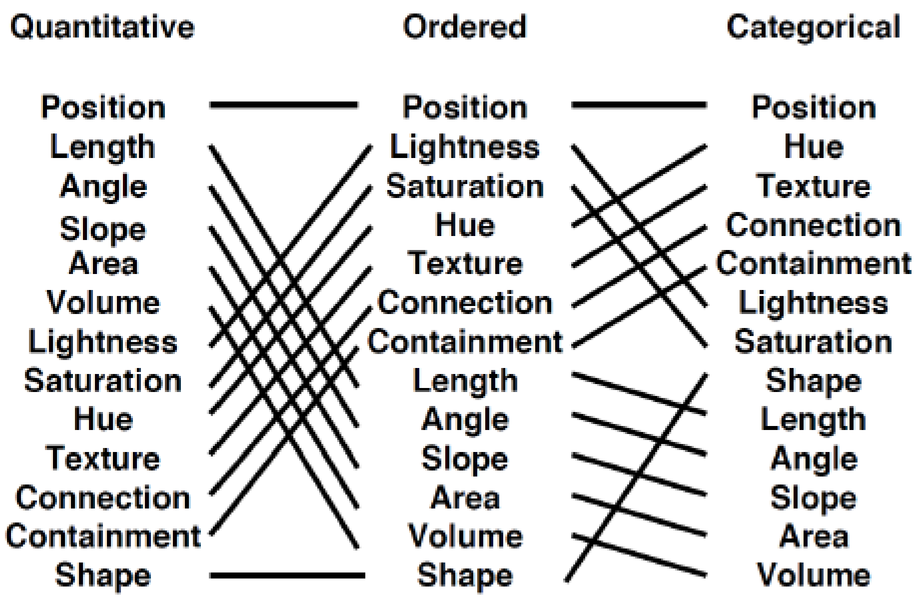
- **Expressiveness principle**
  - show all but only what is in the data
  - match the channel/mark to data characteristics.
- **Effectiveness principle (salience):**
  - effectiveness is the user getting the data as fast as possible
  - encode the most relevant attributes
  - encode most important attributes with highest ranked channels

- **Rankings based on:**
    - **accuracy**, **discriminability**, **separability**, **pop-out**



There are many other rankings as well. (based on perception — derived from perception studies)

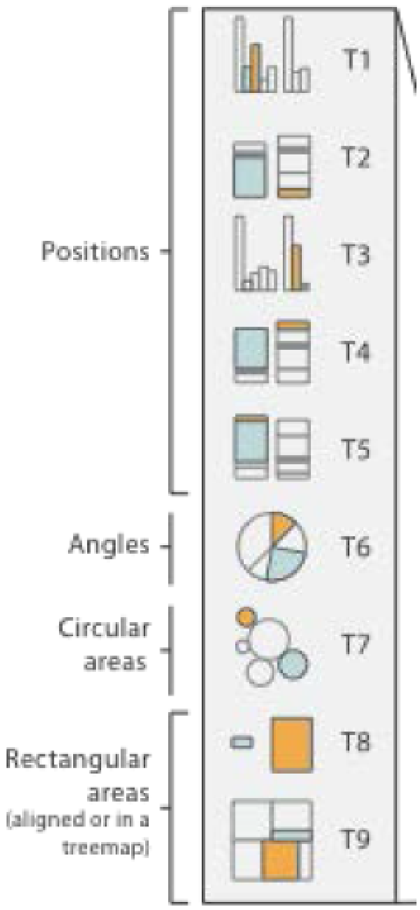
Visual Channel Rankings (Mackinlay)



Position is on top (most important) for each

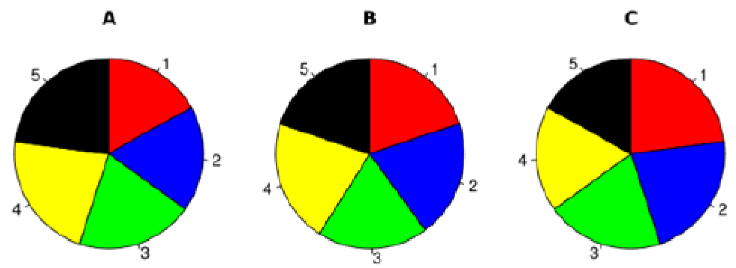
Accuracy: experiments

- Position was found to be most efficient in



evaluation, when compared to angles (pie charts), circular or rectangular areas.

- **Angles** are quite less accurate so rarely used. For example, it is hard to differentiate between the following pie charts:

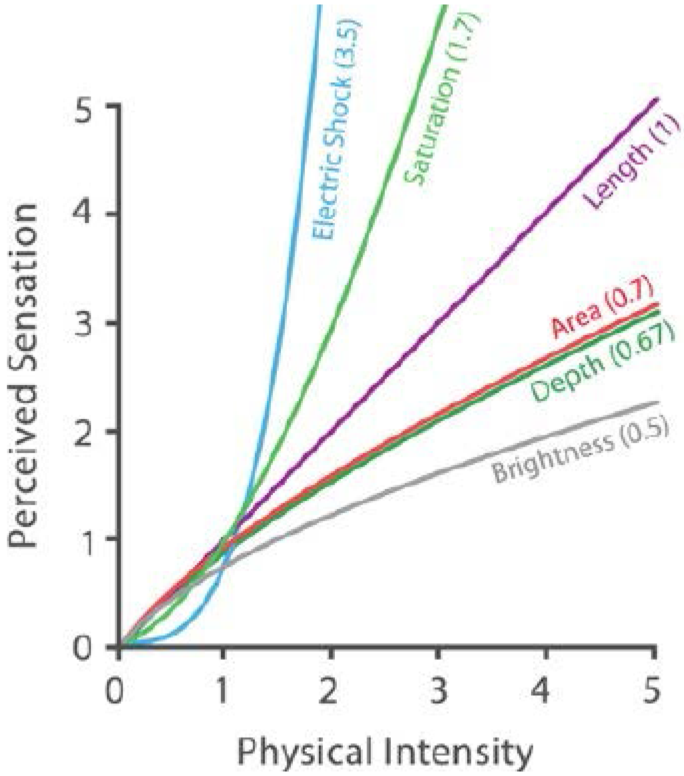


Accuracy

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

S – *perceived sensation*

I – *intensity stimuli*

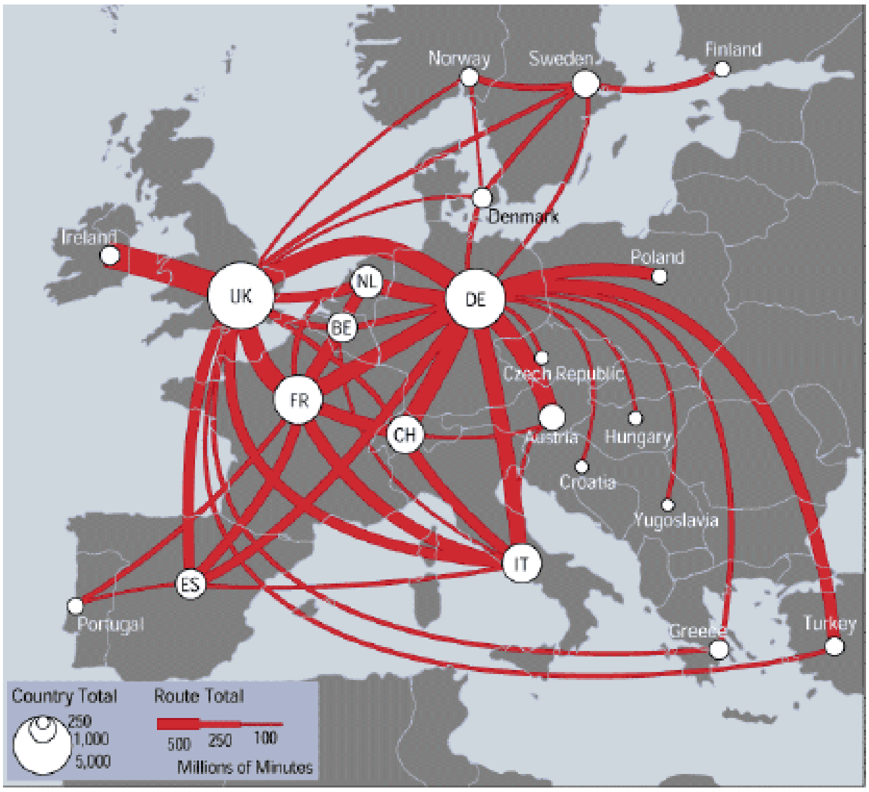


Discriminability

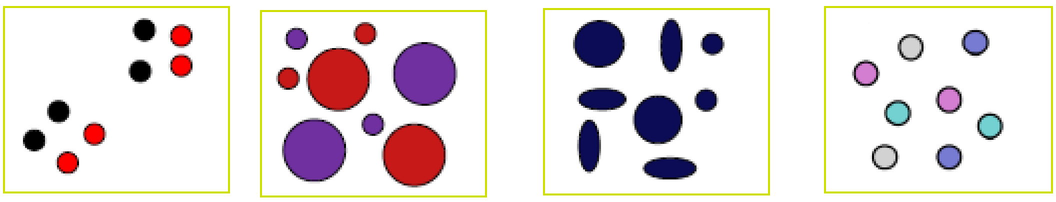
- Quite an important property of channels and also a determining factor in perception

Example — Line Width

- can only discriminate a few line widths
- In this example, there are 3. It does not allow for more.



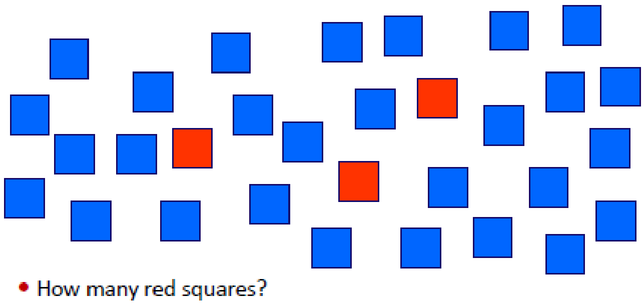
Separability vs. Integrality

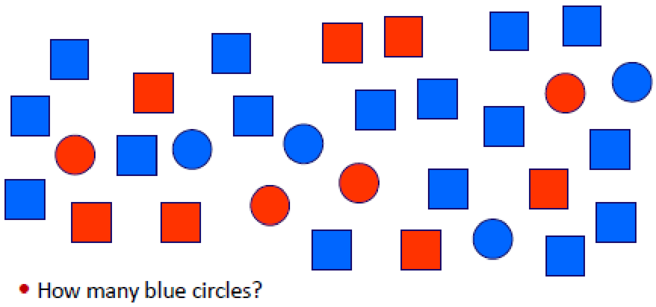


Channel	**Position** **Hue color**	**Size** **Hue color**	**Size width** **Size height**	**Red** **Green**
	seperable	some interference difficult to discriminate Small items	some/significant interference Integral perception area (planar size)	Major interference Integral perception: color/hue
	2 groups each	(2 groups each)	3 groups	4 groups

Pre-attentive vision / Pop out

- We process components of images / visualizations in parallel





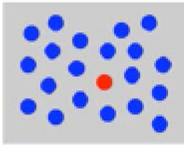
- These examples show the impact of color and shape

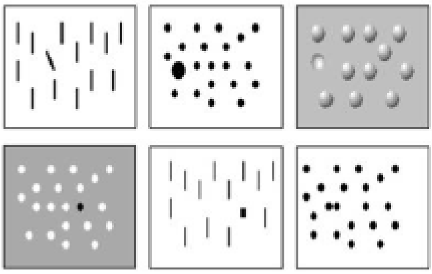
<u>Pre-attentive Vision:</u>

- Color hue or shape alone: **pre-attentive**
  - attentional system not invoked
  - search speed independent of distractor count
- Combined hue and shape are not
  - requires attention
  - search speed linear with distractor count

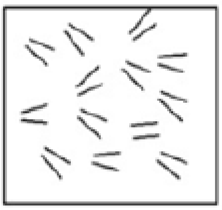
<u>Popout: Most Channels</u>

- Parallel processing on most channels
  - sufficiently different item noticed immediately, independent of distractor





- Some channels have no popout
  - Serial search is required, for example in parallel titled pairs from parallel:



Principles, Advice, and Guidelines

1. **Gestalt Principles**
2. **Tufte's Principles**

Dangers of Depth (3D)

Resolution beats Immersion

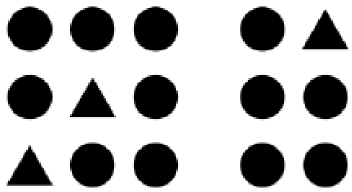
Eyes beat Memory

Gestalt's Principles (1920s)

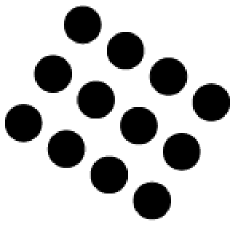
- How human perception group elements, see patterns and simplify information
- How humans **create a whole from parts**

- Commonly used in design (e.g., web design)
- Multiple principles —

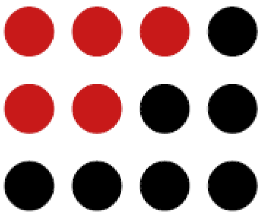
a. Proximity (Emergence) — we group elements that are close to each other



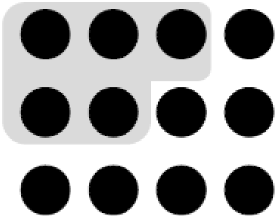
b. Good Figure — objects grouped together tend to be perceived as a single figure



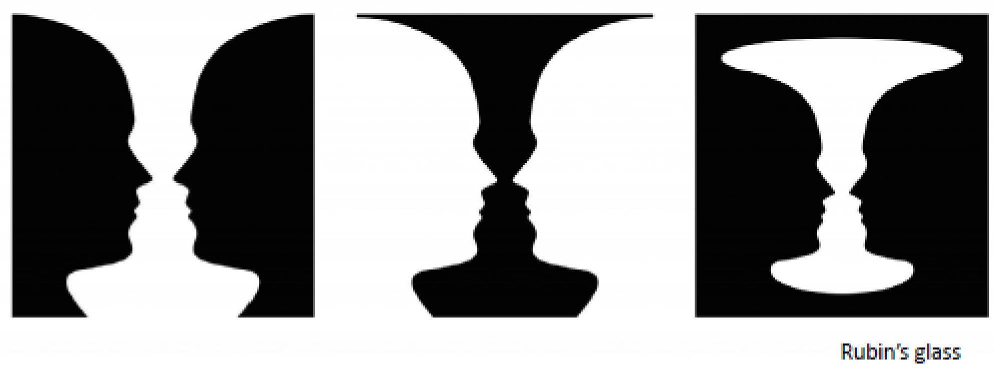
c. Similarity — we tend to group elements with similar appearance



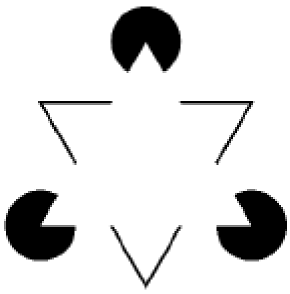
d. Common Region — we group elements that are in the same enclosed regions



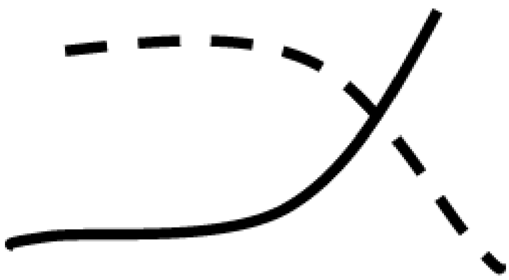
e. Figure/Ground (Multi-Stability) — we see depending on our perception of figure and background. We cannot see both at the same time



f. Closure (Reification) — we complete missing parts



g. Continuity — we tend to form and group continuous lines from pieces



Tufte's Principles

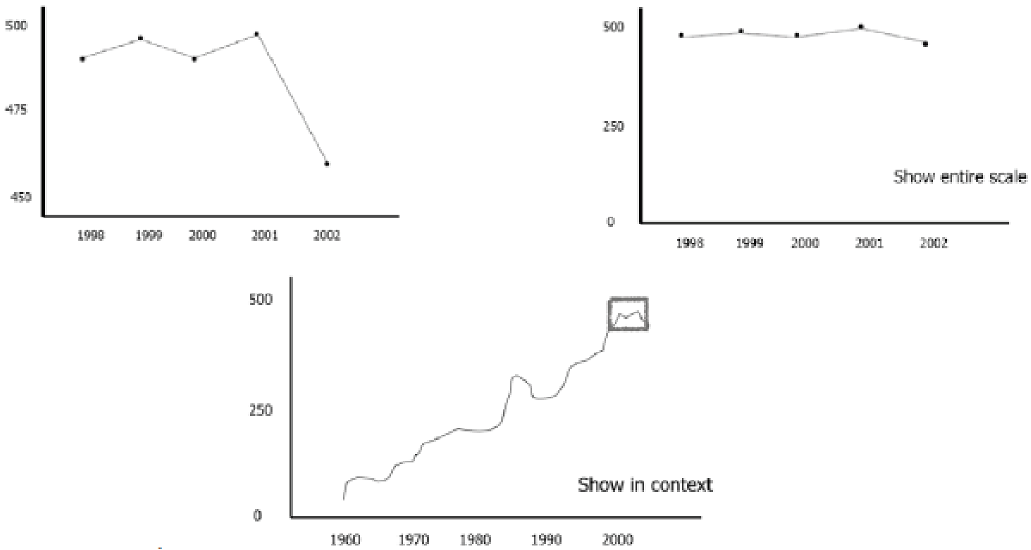
- **Graphical Integrity**
  - Missing scales
  - Scale distortion
- **Design Principles**
  - Maximize data-ink ratio

- Avoid chart junk

Missing scales — Scale distortion

One on the right is better (more honest)

Scale distortion



The context should be well-understood, otherwise it could be misleading

Maximize data-ink ratio

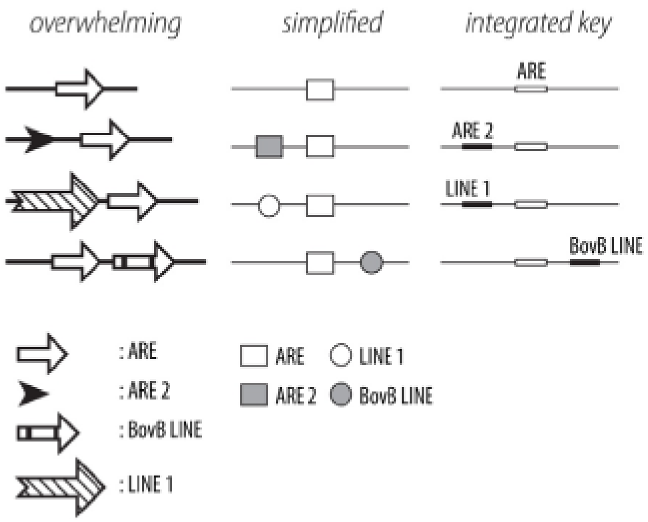
$$data\ ink\ ratio = \frac{data\ ink}{total\ ink\ used\ in\ the\ graphic}$$

- More color can be distracting

- 'Remove to improve' — remove unnecessary stuff like bolding, shades, outer boxes, etc.
- Less is more attractive, effective, and impactful

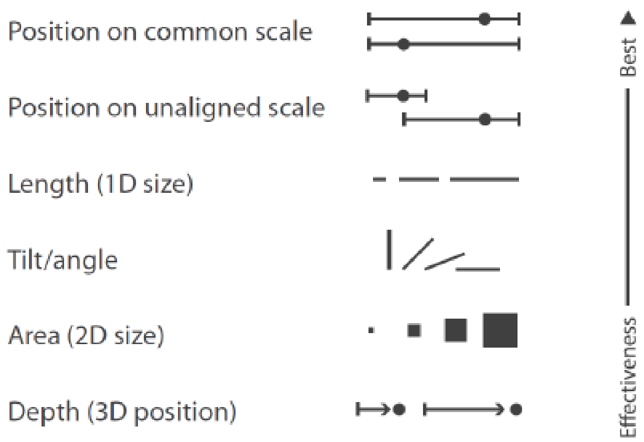
<u>Avoid chart junk: remove redundancy</u>

- Remove information that is not needed



Dangers of Depth

- Ranking for **planar** spatial position, not depth

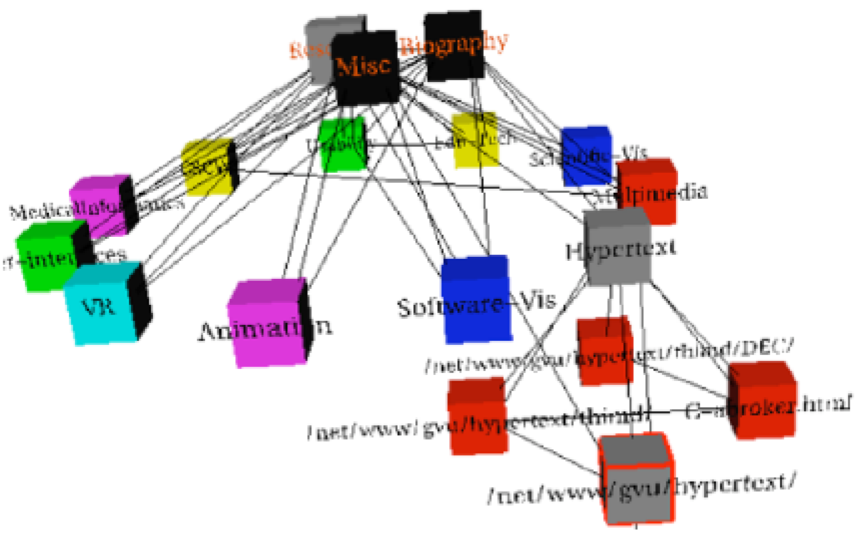


- We do not really see 3D, we see 2.05D
  - Up/down and sideways: image plane

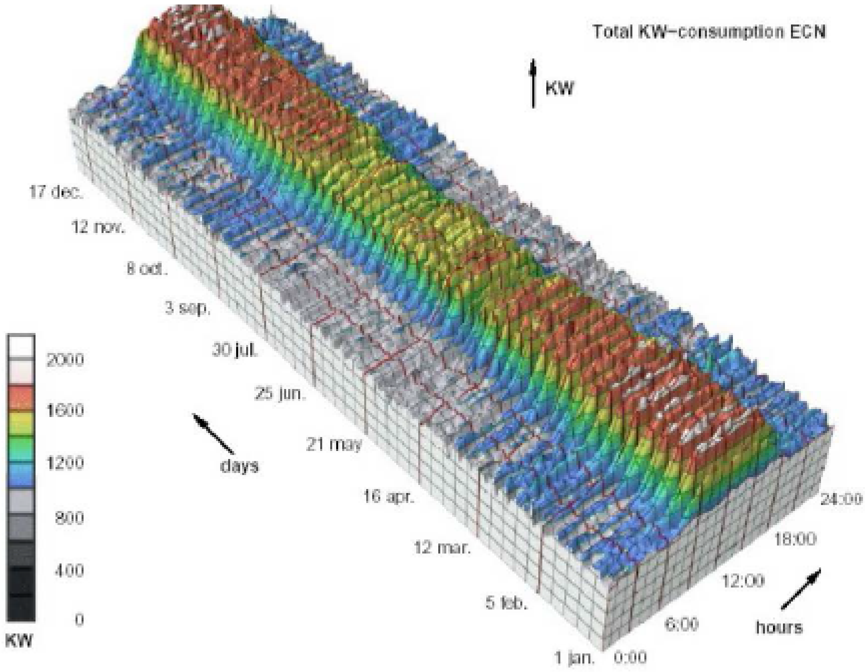
        - acquire more information quickly from eye movement
    - Away: depth into scene
        - only acquire more information from head/body motion

Difficulties of 3D

- **Occlusion** — one object being in front of the other (blockage)
- **Interaction complexity**
- **Perspective distortion**
    - interferes with all <u>size</u> channel encodings
    - power of plane is lost
        - we lost position and length when we acquire depth
- Difficult text legibility
    - worse when titled from image plane



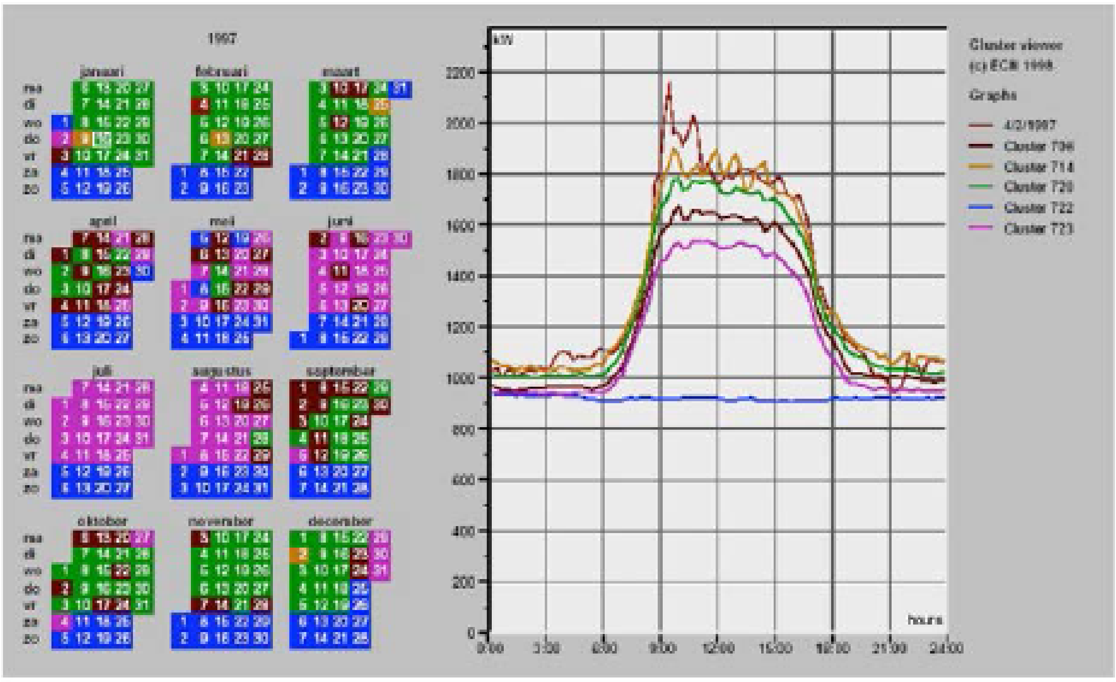
Dangers of depth example

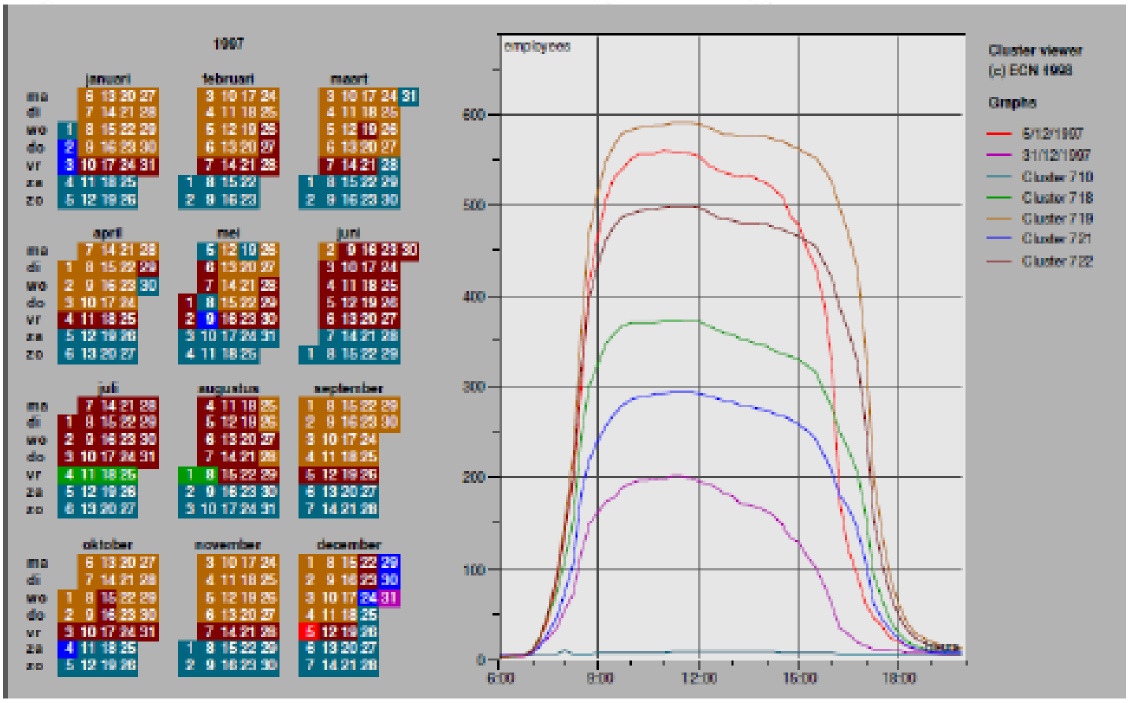


- This is a good figure but it is impossible to make detailed comparisons as you can't really see individual points. There is also quite a lot of occlusion

Transformation to a suitable abstraction

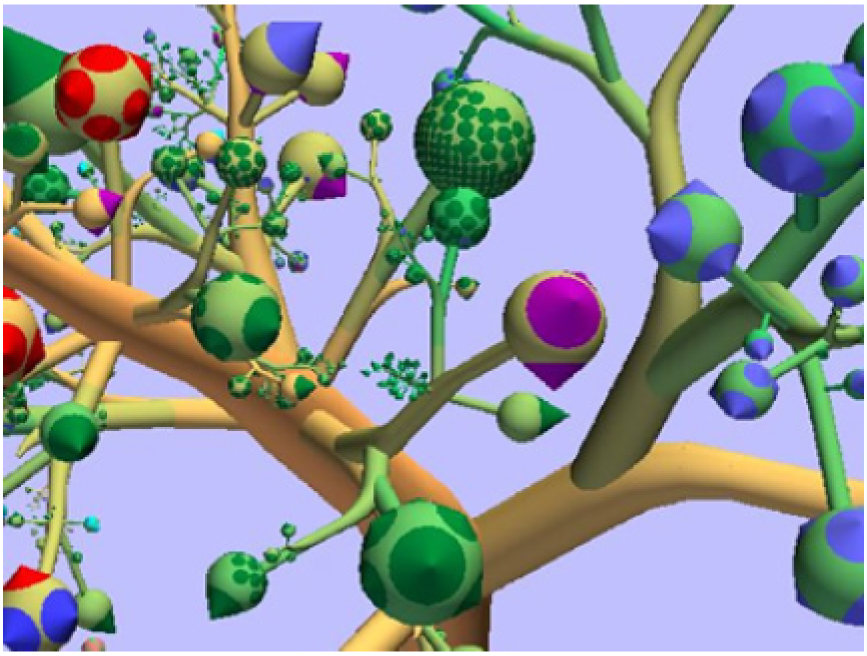
- Derived data (you can group clusters, as is done for graphs in the transformation example below)
- Multiple views: calendar, superimposed 2D curves





Must Justify

- Some cases really require 3D, otherwise they cannot be properly visualized (shape understanding). Like:



- However, 3D needs very careful justification for **non-spatial data**

Resolution beats Immersion

- Resolution much more important (pixels scarcest resource)

Screen resolution	1024x768 = 786.432
Number of cellular phones in Austria (2005)	8.160.000
Transmitted Emails Every Hour (World-wide)	35.388.000

- Desktop over VR for workflow integration as VR setting can be complex to have.
- Virtual reality - Immersion
  - very difficult to justify for non-spatial data as there is not really much need for sense of presence or stereoscopic 3D

Recently attention on HCI advantages of VR vs Desktop

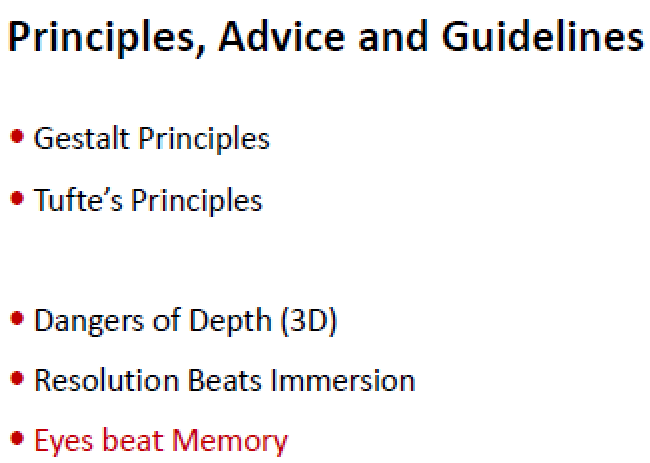
- Better perception of a dataspace geometry
- Intuitive data understanding
- Better retention of the perceived relationships in data

- Preferred by the users
- ...

Lecture 5 - Color and Perception

Book	☐
Completed	☑
Date	@December 3, 2021
Notes	Chapter 10
Type	Lecture
Week	3

… **CONTINUED**



Eyes beat Memory

Memory

- Long-term memory — unlimited
- Short-term memory — working memory
  - has a limit
  - reaching the limit implies cognitive workload
  - we want to **avoid this cognitive load** with visualization

Attention

- Very limited attention for conscious visual search tasks
    - there is a limitation to how long we can remain focused

Animation vs Side-By-Side Views

- Easy to compare (low cognitive load) by moving **eyes (**side-by-side views)
- Animation — hard to compare visible items to **memory** of what you saw
    - because with animation, you are using working memory

Animations

- great for choreographed storytelling (paying attention to localized action — at a specific part)
- Great for transition between two states / datasets
- give control to user on the animation (re-play, pause, stop)
- poor for many states with changes everywhere — **change blindness**

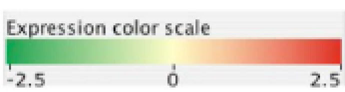
Change blindness

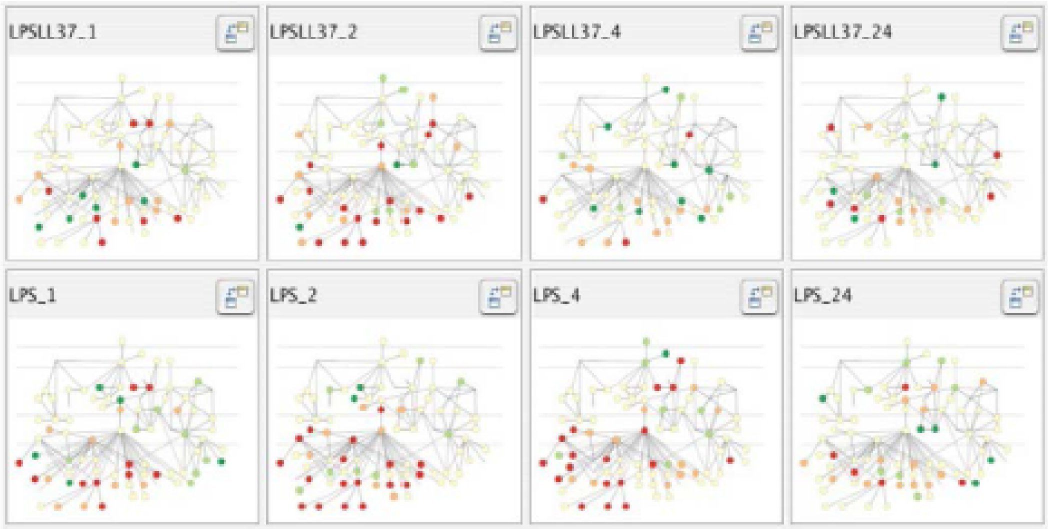
- With a larger numbers of states, and you focusing on specific aspects of the animation, you can be blind to other aspects of the animation and possible changes within.

Small Multiples alternative

EXAMPLE — Cerebral

- One graph per experimental condition
    - Same spatial layout
    - Color differently, by condition

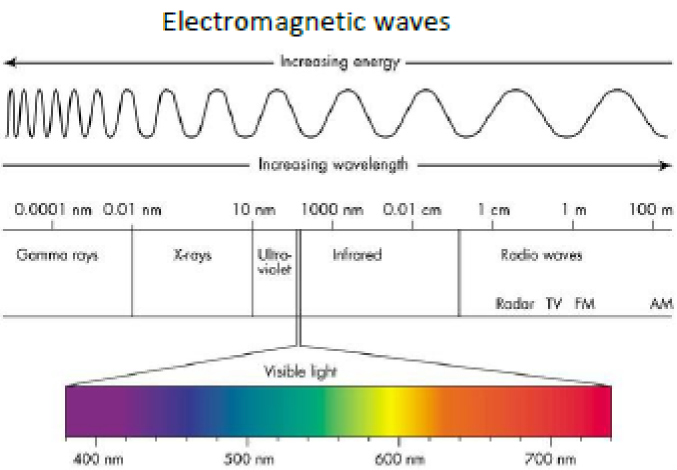




- 8 different characteristics of the graph
- The spatial layout is the same but the properties of the nodes are changing
- You could use animations but since a lot of things are changing at the same time here, that would be ineffective.

Color

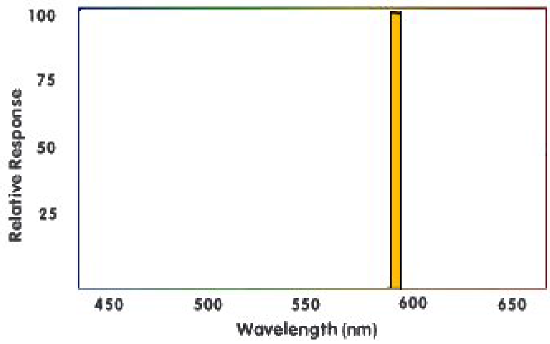
Light spectrum



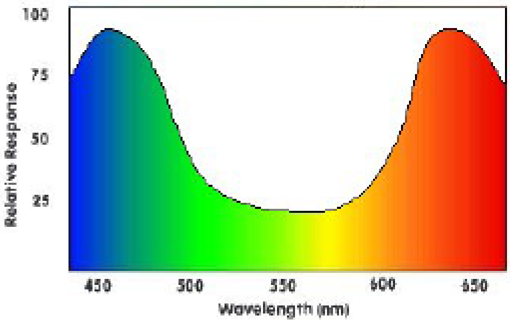
Color physical definition

— Distribution of power over the spectrum

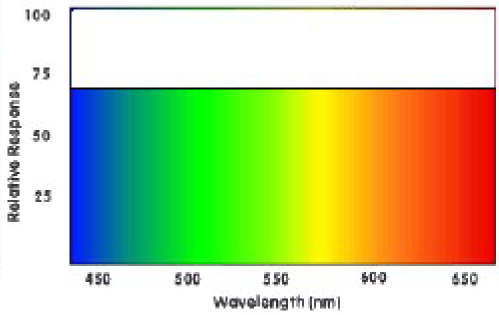
- One of the properties is **Hue**



Hue Orange



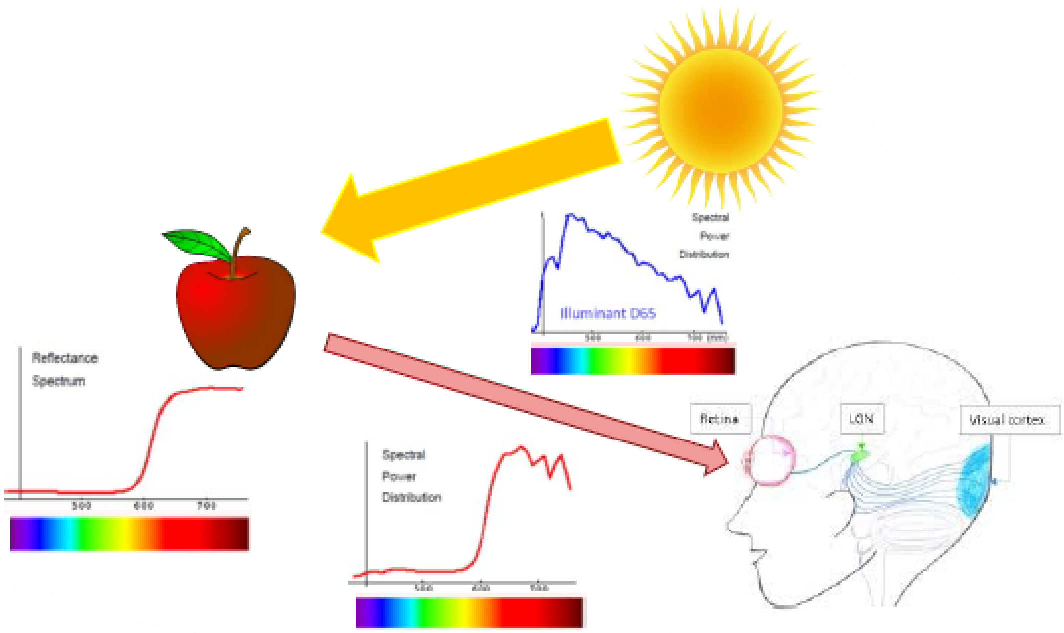
Hue Magenta



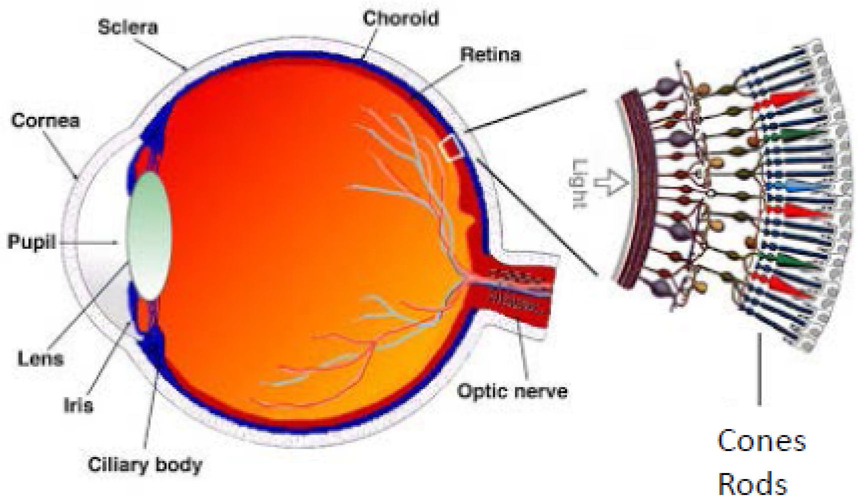
White light

Combination of blue and red in the first one (magenta). Combination of all in second (white)

What do we see?

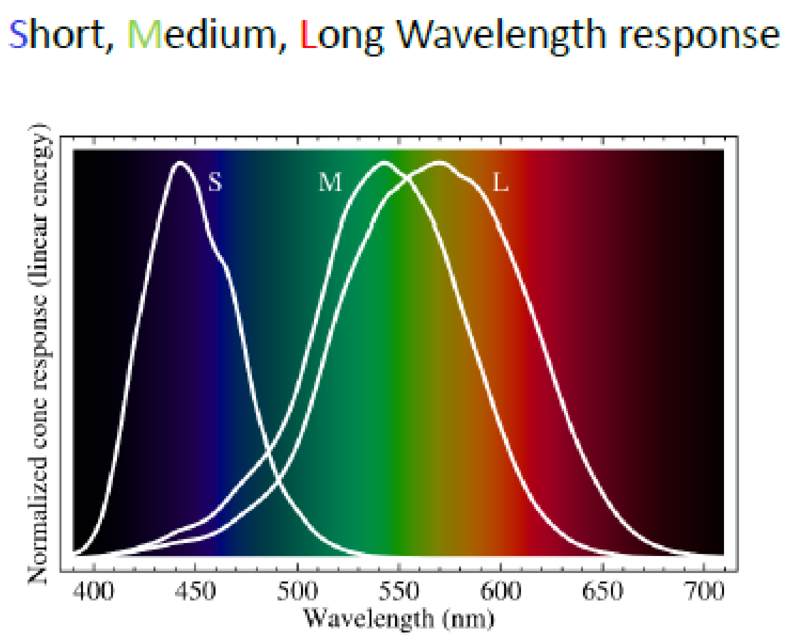


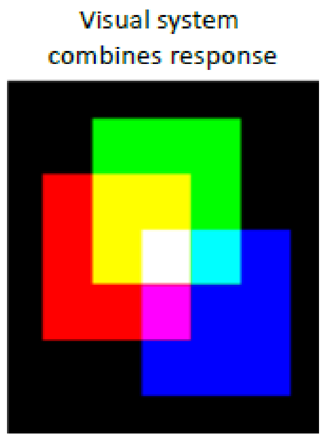
- ^Spectrum of the light coming towards the object is multiplied by the properties of the object (what part of the spectrum is it absorbing and what part is it reflecting), to obtain the spectrum that goes out of the object and is received by our retina as stimulus.



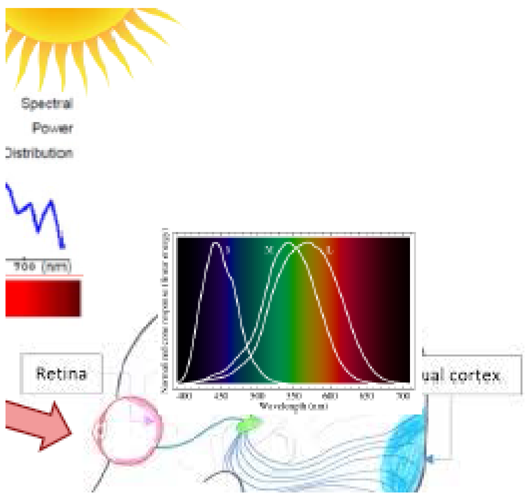
- **CONES — (focused on, in this lecture)**
  - chromatic perception (types: L,M,S)
  - 6 to 7 million present in the retina
  - the limitation to our resolution is 3 times full HD
  - Cones not used in night vision so we don't see color
    - a threshold amount of light is required for activation
  - Color defined by cones
- **RODS —**
  - achromatic perception
  - low-light vision (night vision)
  - saturated in day vision (not used at that time)

Cones spectral sensitivity





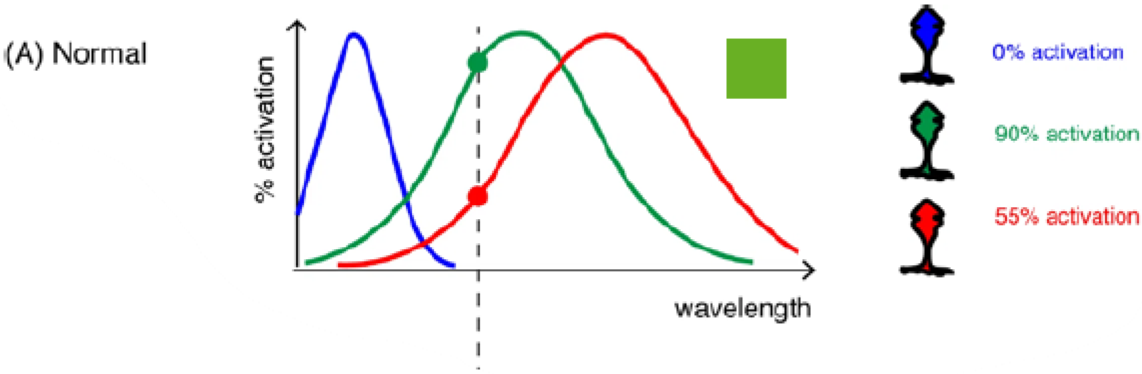
- So, added to the big picture:



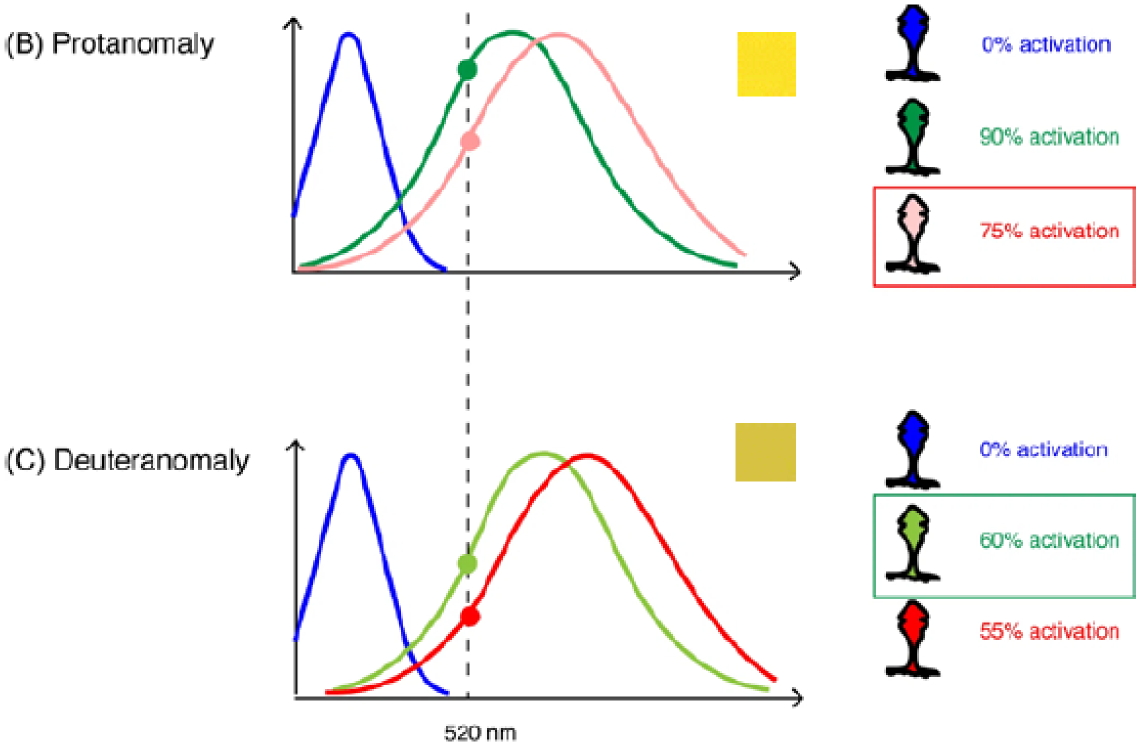
- **We can see approximately 10M colors.**

Color blindness

- Different kinds of deficiencies on the functioning of the cones (e.g., having two cones instead of three)

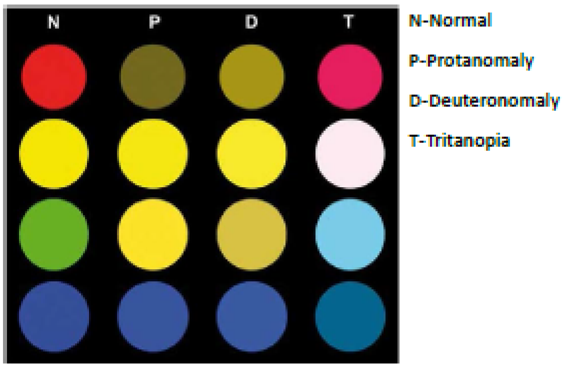


- Red-green colorblindness 5-8% of men and 0.5% of women
  - Red cone deficiency (**protanomaly**)
  - Green cone deficiency (**deuteranomaly**)



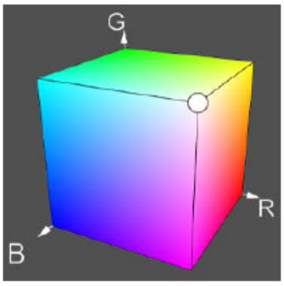
- Blue-yellow colorblindness (**Tritanopia**)
  - 1% males-females

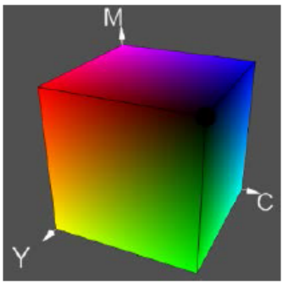
Color blind people see different colors



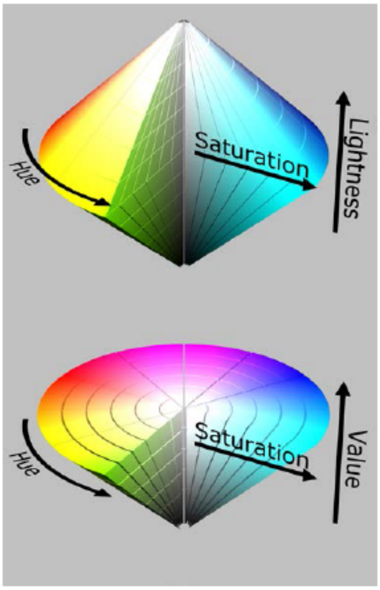
How do we represent color?

- 3 dimensions, since there are 3 cones
- Device Oriented: physical realization
  - screen RGB space (3 lamps)
  - printer CMY(K)
  - Simple; each device has its own space; represents a part of the human visible colors





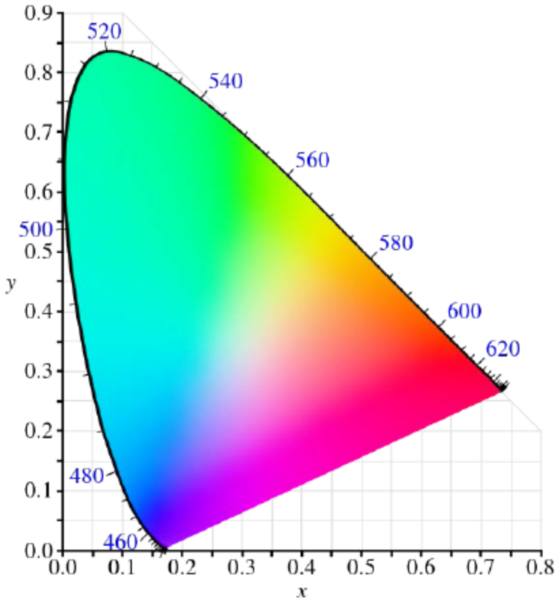
- **More intuitive way —** using semantic properties of color
  - Hue: the color wheel
  - Saturation: how much grey a color has
  - Light/value: how much light/brightness a color has



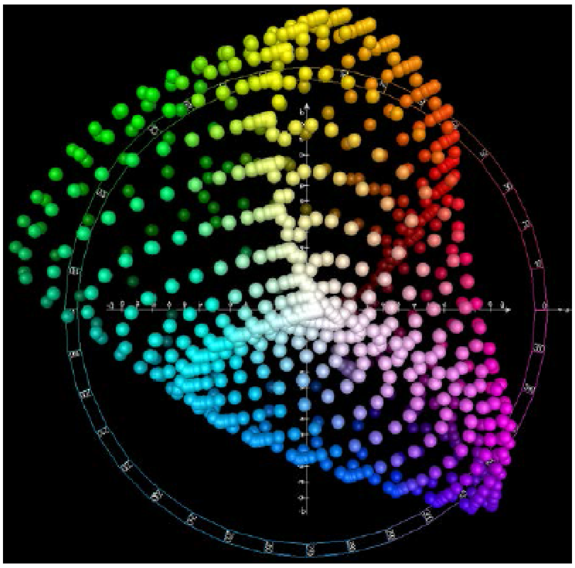
How do we represent all human visible color?

CIE 1931 XYZ Color

- built through experimentation
- device independent
- mathematical formulation of the color space
- easy conversion from these colors from what we want to show



- **CIELUV, CIELAB, (L*, a*, b*)**
  - deformation of CIE space
  - Euclidean distance corresponds to perceptual difference



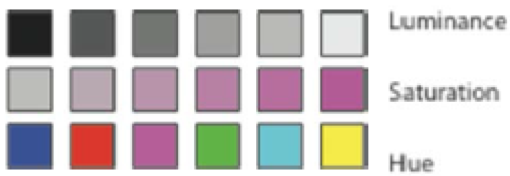
Perception

- Our visual system also includes bias
  - green is slightly better distinguishable
  - blue perception is very weak
  - we are very sensitive to luminance

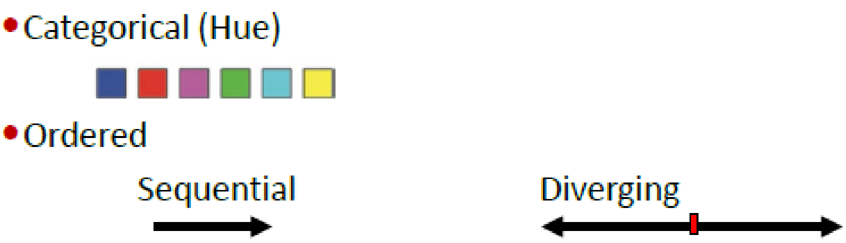
EXAMPLE — blurring one channel

Color: Luminance, Saturation, Hue

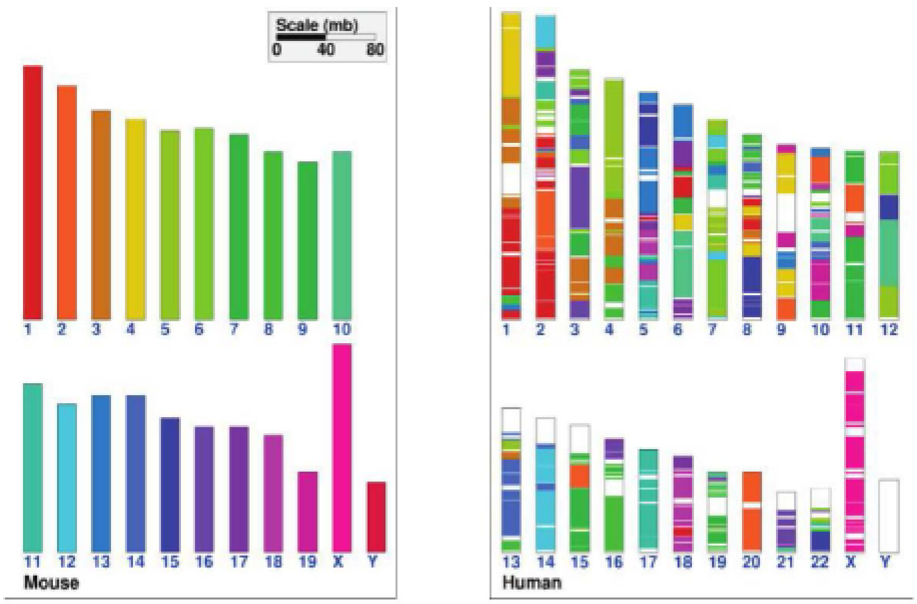
- Common color spaces
  - **RGB** — poor choice for visual encoding
  - **HSL** (Hue-Saturation-Luminance), **Lab** (the one where Euclidean was used) , … are better choices!



Colormap



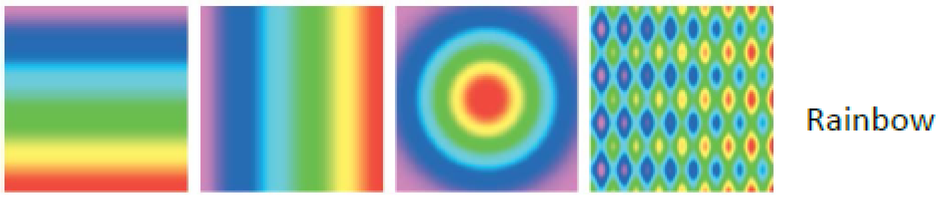
Discriminability: Ineffective categorical color maps



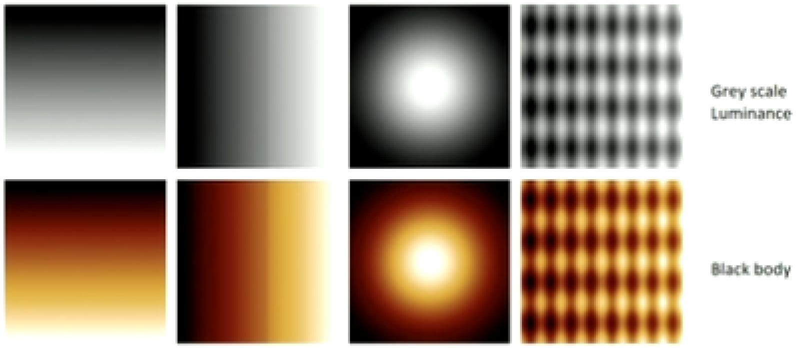
- When looked at separately, the human plots are difficult to differentiate within.

- So when there is a large number of variables, hue is not a good way of mapping

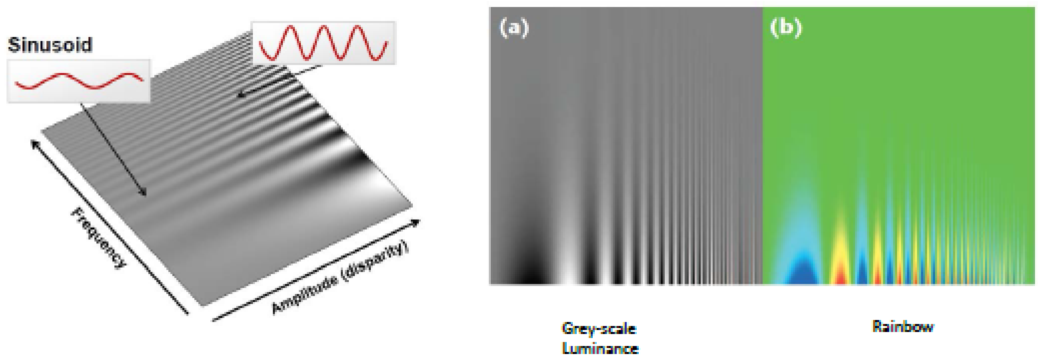
Colormaps for quantitative/ordered



- The colors are not intrinsically ordered so order would vary based on perception. Hence, this too is not the best mapping.
- Could also be a problem for color blind people.

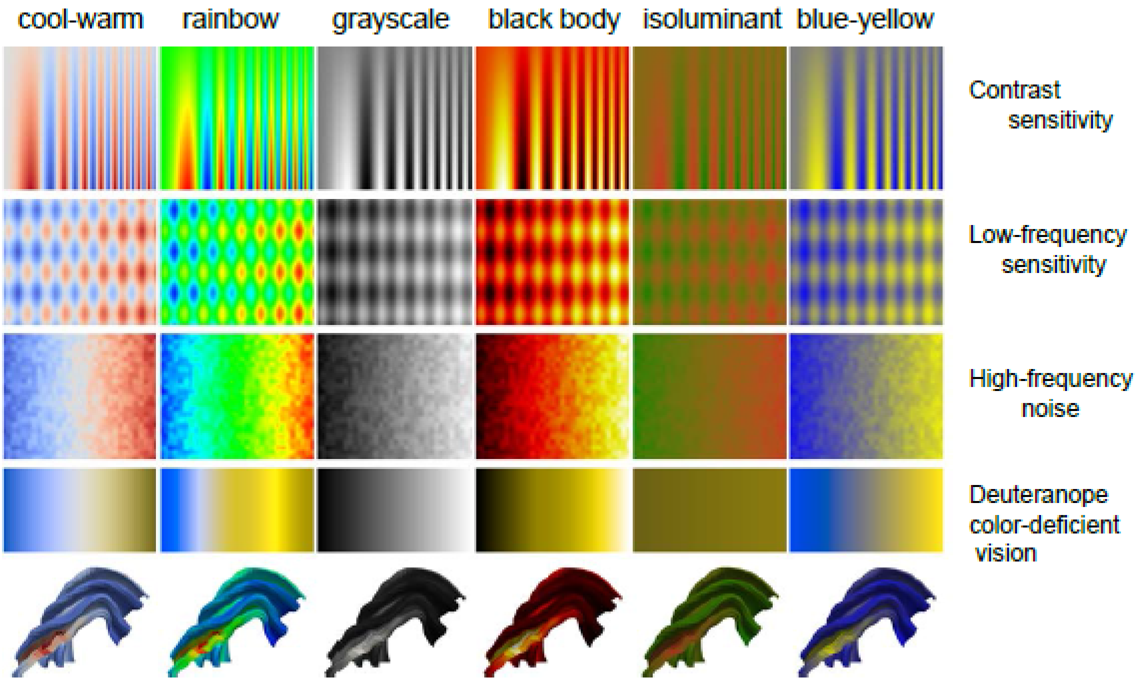


- Using different color hues get us a larger range of perceptual difference



- We are receiving more detail with grey-scale luminance: amplitude is lower with rainbow

<u>Different properties to consider:</u>

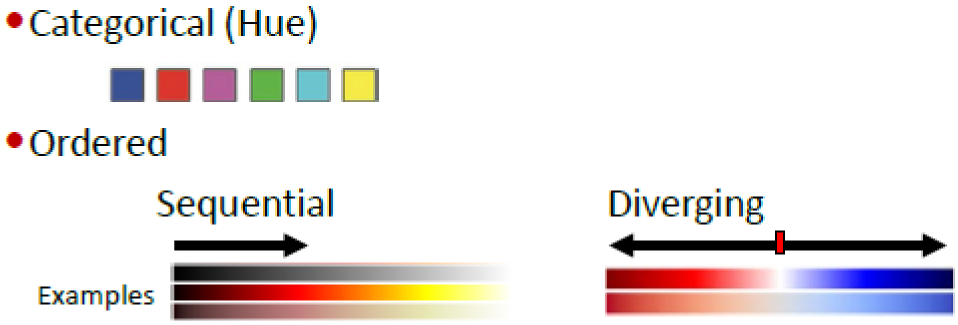


- **Rainbow can lead to wrong conclusions**
  - usually happens because of the number of colors involved
  - an artifact of the colormap may appear in the visualization while not being present in the data
  - for example, a color change could happen even when there is no change in gradient. Or there could be borders identified in areas where there are none.

Rainbow is a poor default

- PROBLEMS —
  - perceptually unordered
  - perceptually non-linear
  - perceptual borders are not there
- BENEFITS —
  - nameable regions
- ALTERNATIVES —
  - few multiple hues with monotonically increasing luminance for fine-grained

Colormaps

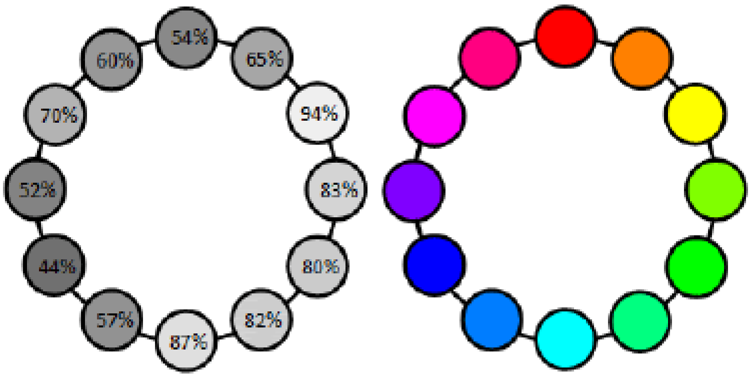


- We cannot use diverging color maps with sequential data because we would have to choose a random midpoint which would have no spatial meaning
- And also vice versa: a point of importance would become more invisible if it was mapped sequentially
- To add rainbow to sequential, make sure:
    - luminance is preserved (remains constant), and
    - there is an intrinsic order to the colors

Visual Perception — Luminance

- **Physical parameter luminance: light per unit area**
- Higher luminance → higher brightness level perceived, but:
    - it does not have a linear relation between values and perception
    - Humans are better at perceiving relative brightness changes in darker areas
- A transformation like gamma makes the map more homogeneous

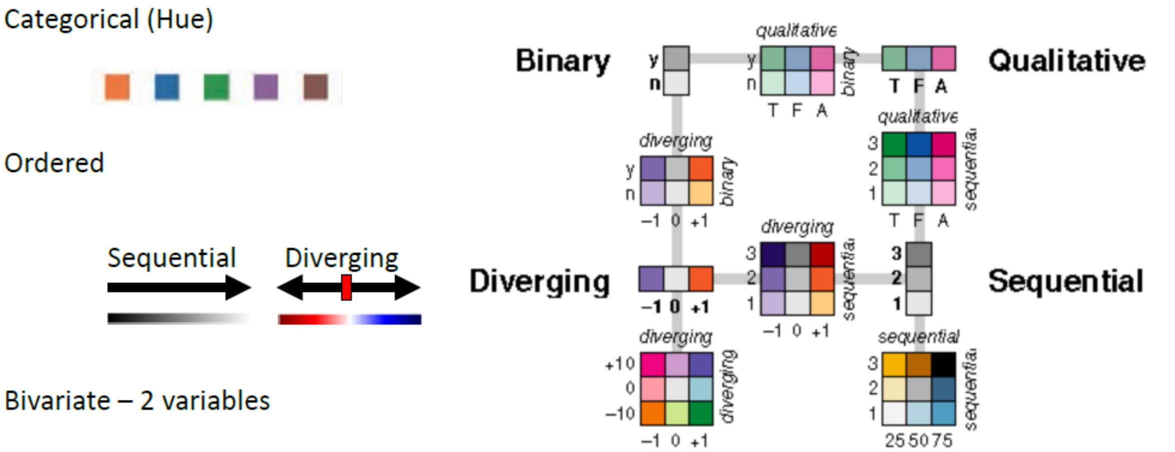
Different Hue → Different Luminance



Colormap and luminance

- Humans are sensitive to luminance
- Constant luminance for categorical data
    - we don't want luminance attracting more attention to one particular category

Colormaps



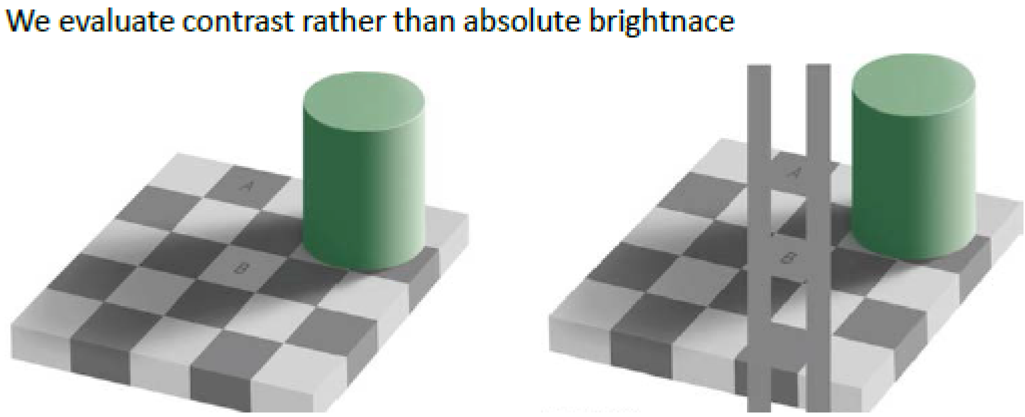
Choose your colormap wisely

- Some resources for help:
    - ColorBrewer & SciVisColor

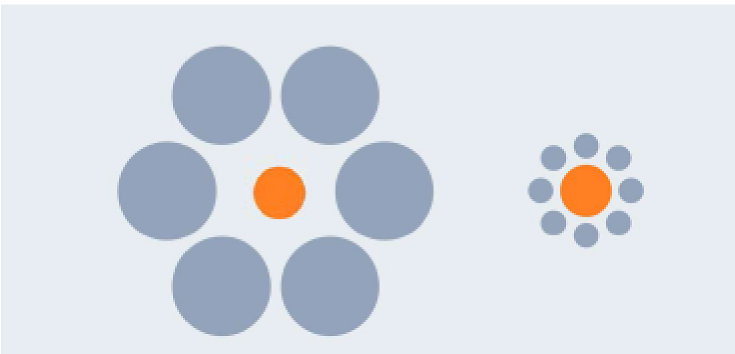
Perception

- Colors are not evaluated independently
- Perception is complex
- Our brain uses information according to its experience.
- Context is relevant

Luminance example



- We also perceive channels other than color, relatively: (both orange circles are of the same size)



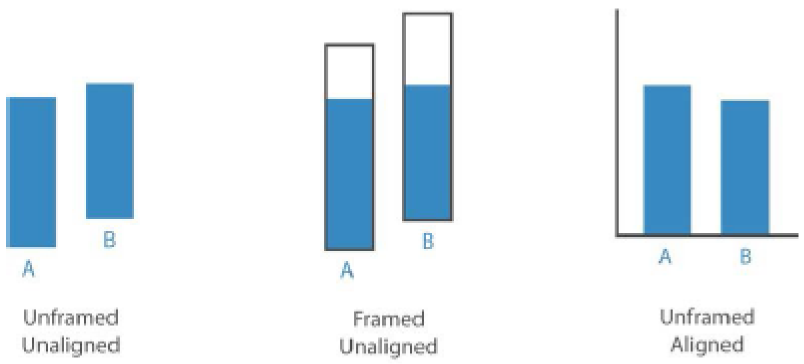
Weber-Law

- Perceptual system mostly operates with relative judgments, not absolute
- **Smallest change in stimuli that is perceived (ΔS) to background stimuli (S) is constant (k)**

$$\frac{\Delta S}{S} = k$$

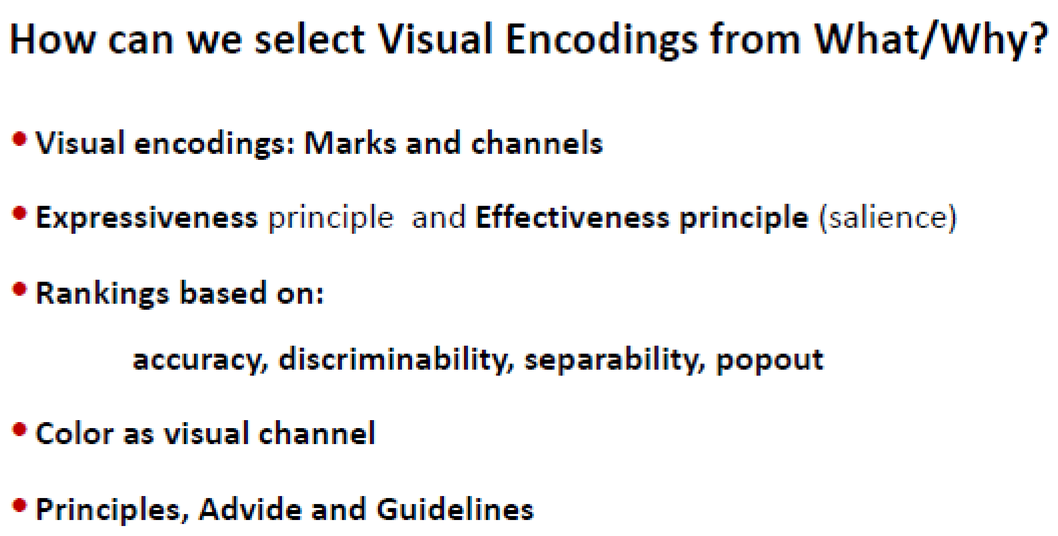
Relative vs. Absolute Judgements

- Accuracy increases with common frame/scale and alignment
  - filled rectangles differ in length by 1:9, **difficult judgement**
  - white rectangles differ in length by 1:4, **easy judgement**



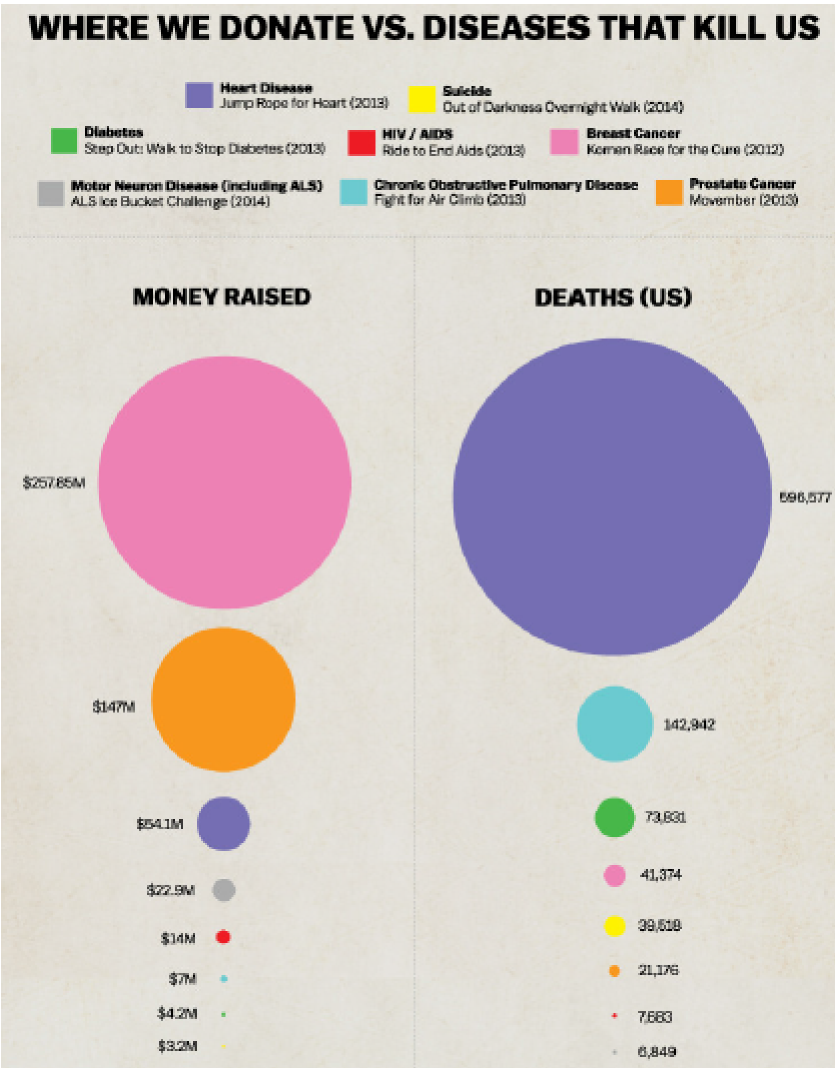
  - Easier to judge the difference relatively, when boxes are aligned

---



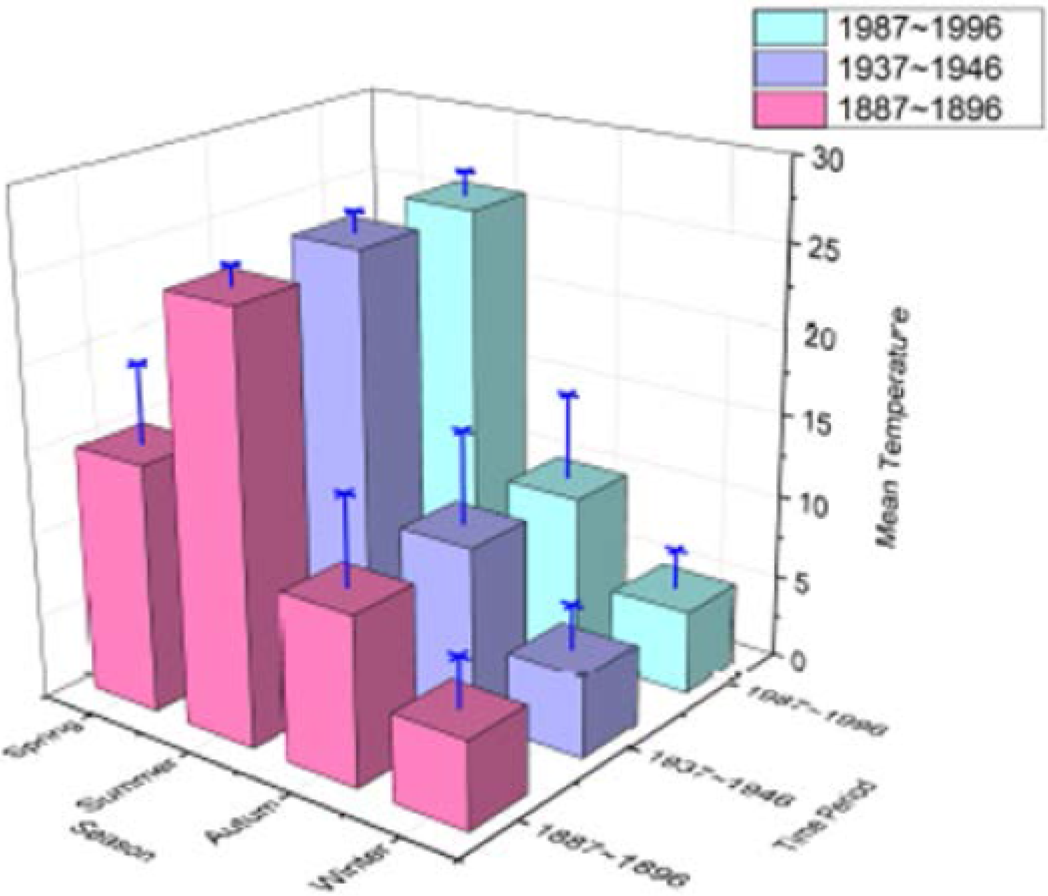
Lecture 6 - Discussion on good/bad visualizations

Book	☑
Completed	☑
Date	@December 10, 2021
Notes	-
Type	Lecture
Week	4



Question — Name one of the elements in the visualization shown to the right that should be considered wrong or inadequate by visualization standards? What would you suggest as an improved visualization?

- Area is being used for encoding, which is not being scaled adequately. We are not perceiving it linearly at the moment. (separability & integrability)
- It should use the most effective encoding for the most important attributes (based on rankings). Area is not on top of the rankings.
- Could use position (bar charts, for example) since it is a much better encoding
- Or to observe the relation better, a deaths-money scatterplot could be made (also uses position for for attributes)
    - better to use scatter plot because it shows correlation much better than lets say, bar charts.



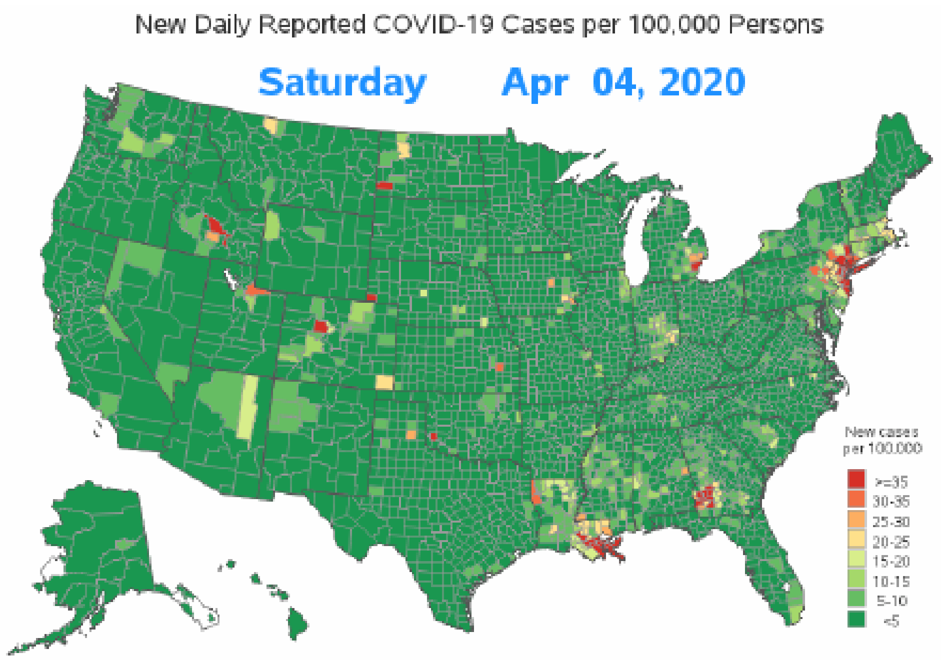
- Assuming this is tabular data, how many attributes are encoded in this visualization? Which data types are they (categorical, ordinal, or quantitative)?

- **Temperature** — quantitative (sequential)
- **Time Period** — ordinal / quantitative (sequential)
- **Season** — ordinal (cyclic) (cannot be categorical because order exists)

- Name two elements on the visual encoding choices (marks/visual channels) that can be considered wrong or inadequate ? Motivate each choice in one sentence.
  - It is a **3D** representation and **volume** itself is not the best visual encoding
  - **Occlusion** — data for the spring season is not visible; it is being blocked
  - **Interaction** is more complicated — we have a perspective issue; it is quite difficult to evaluate as it also depends on projection
  - **Redundancy** — time period uses 2 encodings: color and position, so we are misusing one channel and making the graph unnecessarily complex.
  - **Color encoding** does not demonstrate the order in Season.
- Which visual encoding would you use? Motivate each choice in one sentence.
  - There many good alternatives but for each, there is a need for good justification. For example, for a bar chart, you could argue how it makes reading numerical values and comparing them an easier task

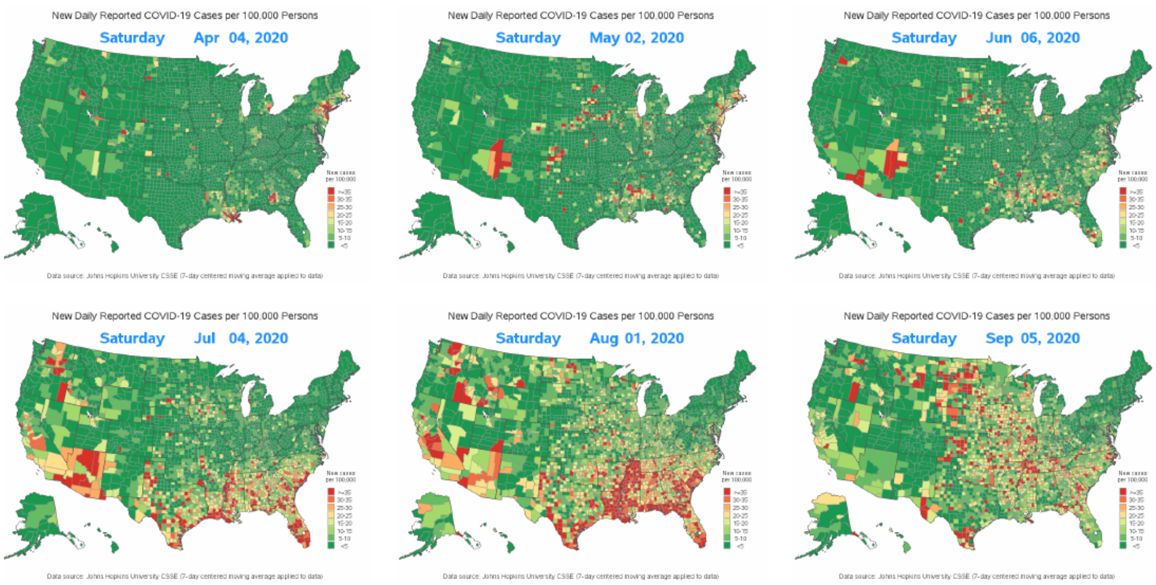
---

A Covid-19 Visualization

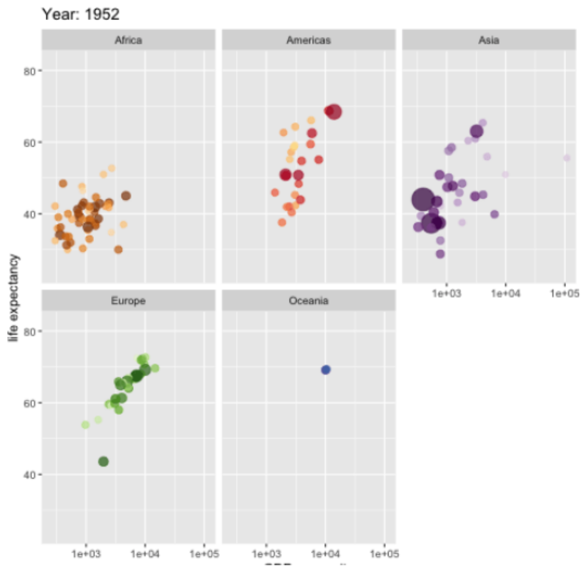


- The good thing here is that we are using a map to display geographical data (expressiveness)
- Color coding is also quite expressive (green associates with good and red with bad)
    - However it is also not a good color coding since we are sensitive to luminance. When using sequential color coding, we expect the luminance to be progressive. But here, it kind of bumps
    - Also, it is not color-blind compatible
- Interference between color and area is also a downside
- Considering this is an animation (which it doesn't play here), we have to use short term memory. Alternative —

Small Multiples (Eyes beat Memory)



- In some cases, animations are much better. For example where trends are shown:

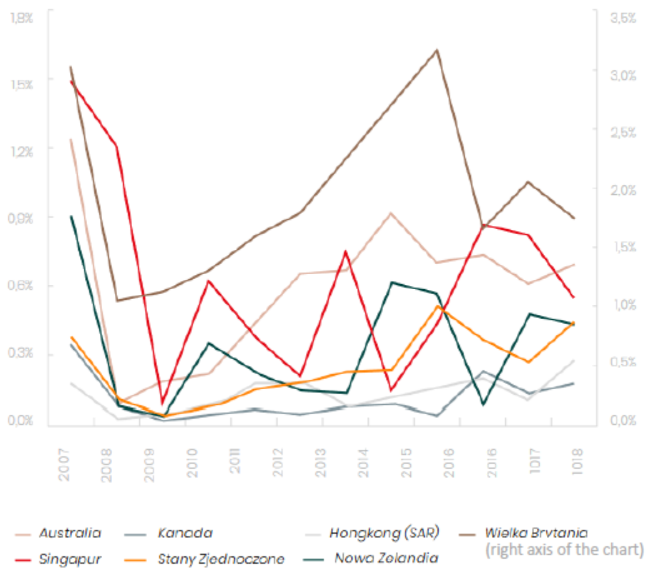


Check the animation in slides for better understanding

- A user slider could be a good addition to animations so we can play specific time frames

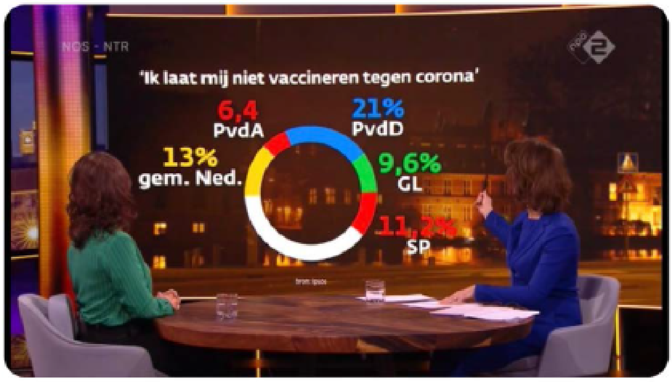
---

Good or Bad Visualization?



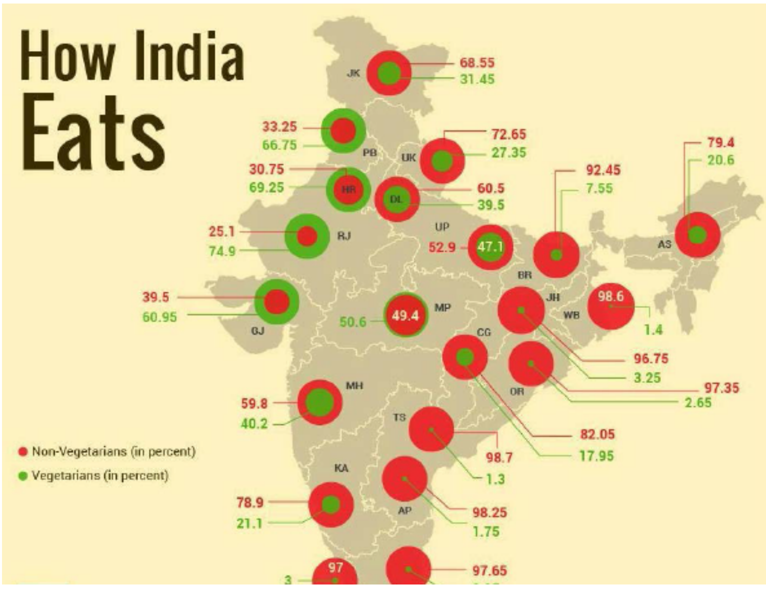
What's wrong?

- double axes (could only use these when trends are being compared)
- the color scheme is not the best
- no title

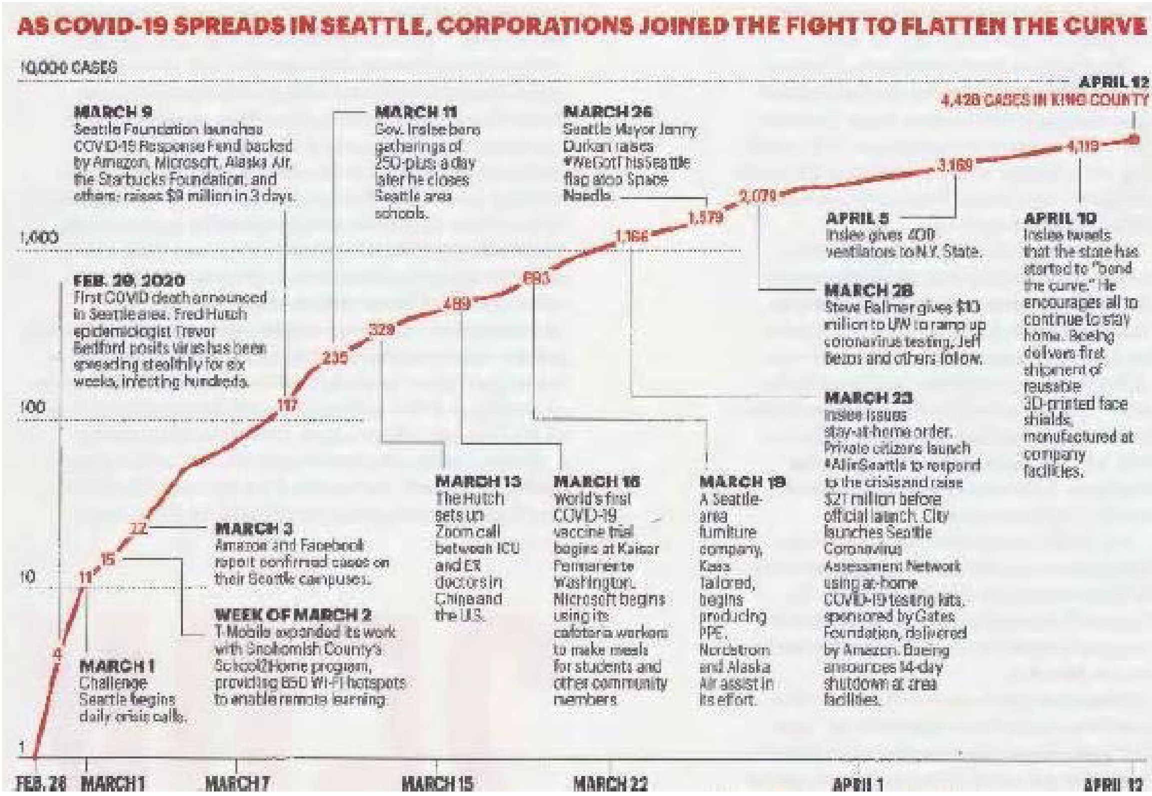


Question: good use of pie/donut chart?

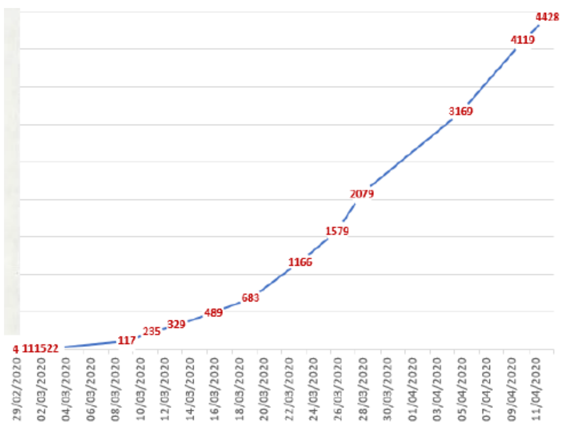
- Percentages do not add to a 100
- There is a white area that is not labelled
- Some values are missing the %
- A correct pie chart could work better

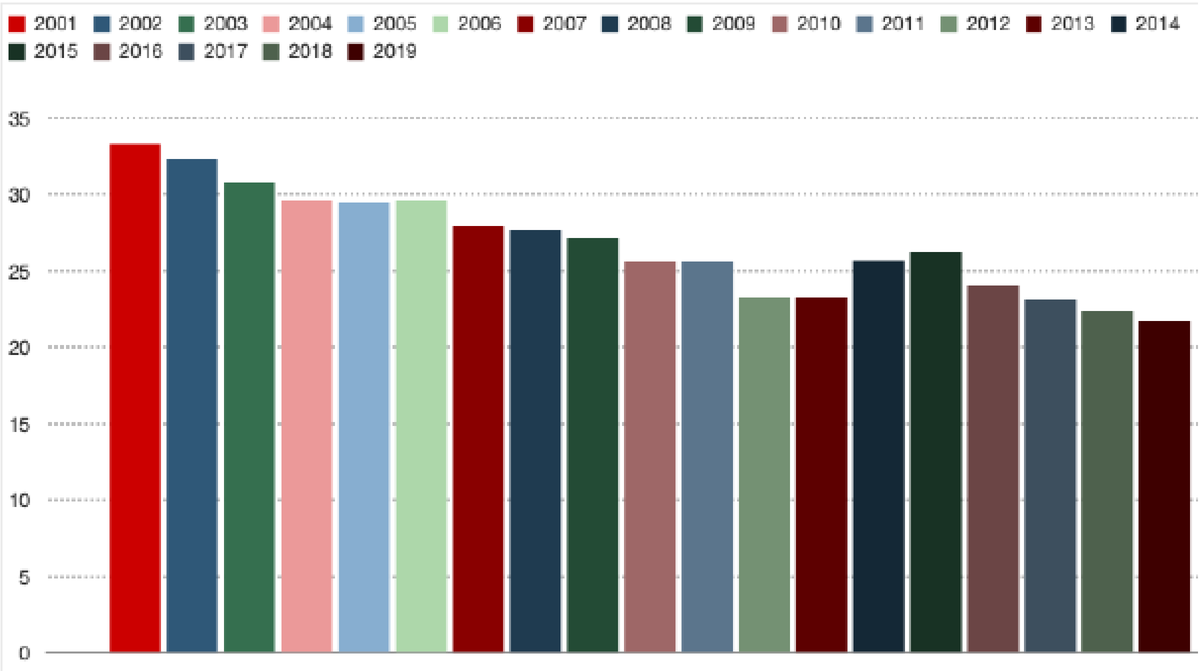


- Not color-blind compatible since green and red are the most common deficiencies
  - Percentage labels are a good addition
- It is difficult to contrast populations (discriminability)



- y-axis is distorted
    - the change in values is not even linear or exponential
- The message conveyed by this visualization would change if the axis was scaled properly:

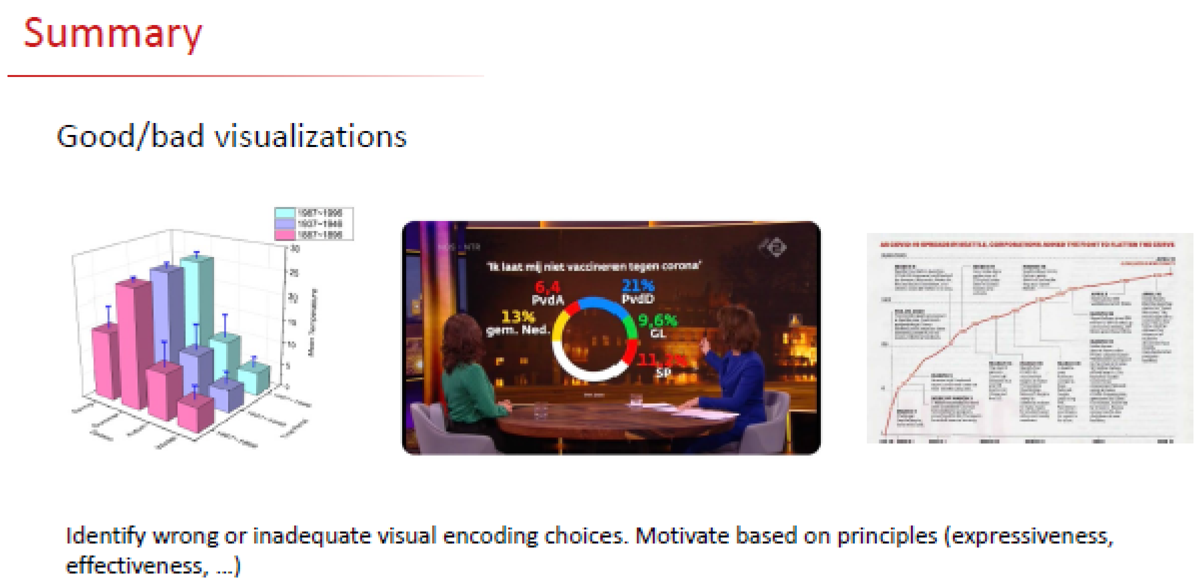




- The use of color is unnecessary here
- Y-axis could be labelled for encoding

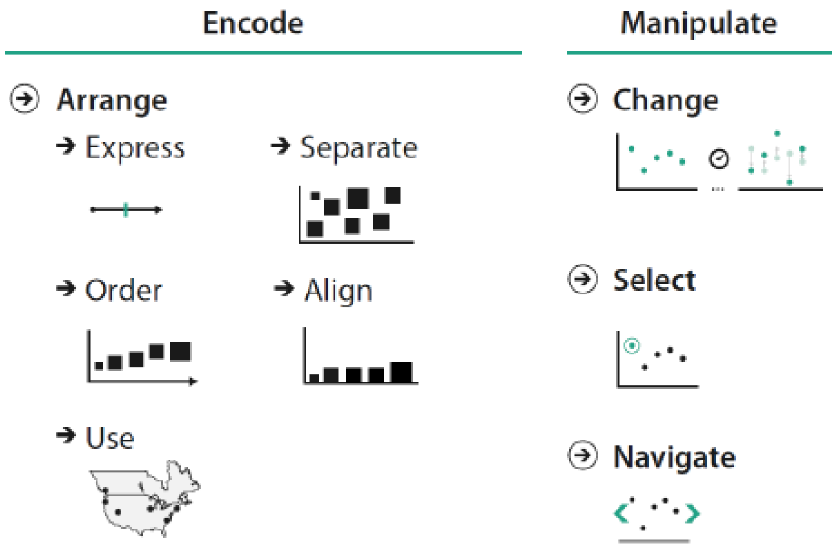
Lecture 7 - Idioms and Interaction

Date	@December 15, 2021
Type	Lecture
Completed	☑
Book	☐
Notes	Chapter 7
Week	5



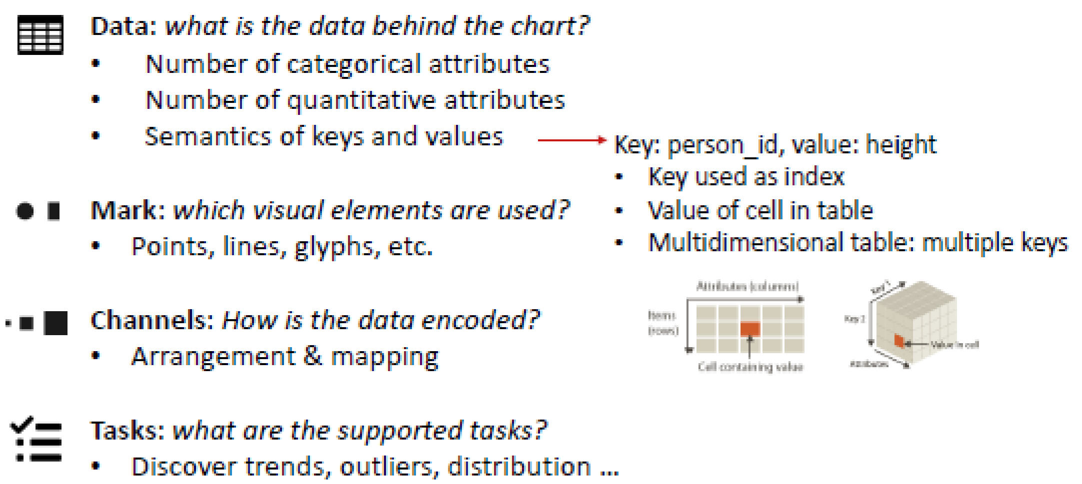
Visualization Idioms (Analysis framework < HOW < Visual Encoding)

- Idioms are different chart types
- Information visualization techniques (design choices)



The different idioms

- Idioms put restrictions on tasks. so **choose carefully**
- Idioms are **described** with **4 properties —**



  - <u>Channels</u> tell us things like: How do we plot our marks? Do we plot them over each other? (Arrangement) How do map the shape to the quantitative value? What do we use (size, position, etc.)? (Mapping)
  - <u>Tasks</u> tell us what idioms are supported by what tasks (for example, we can use a line chart or scatterplot to discover trends)

Idioms with n Keys and 1 Value

Bar Chart

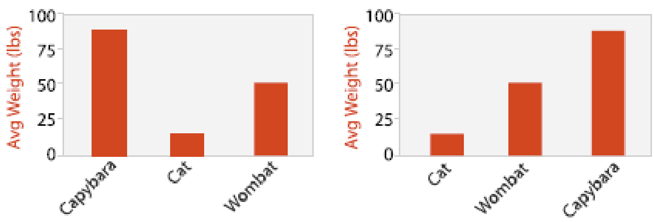
Data: *what is the data behind the chart?*
- 1 Categorical attribute (key)
- 1 Quantitative attribute (value)

Mark: *which visual elements are used?*
- Lines

Channels: *How is the data encoded?*
- Length to convey quantitative value
- Spatial regions: one per mark

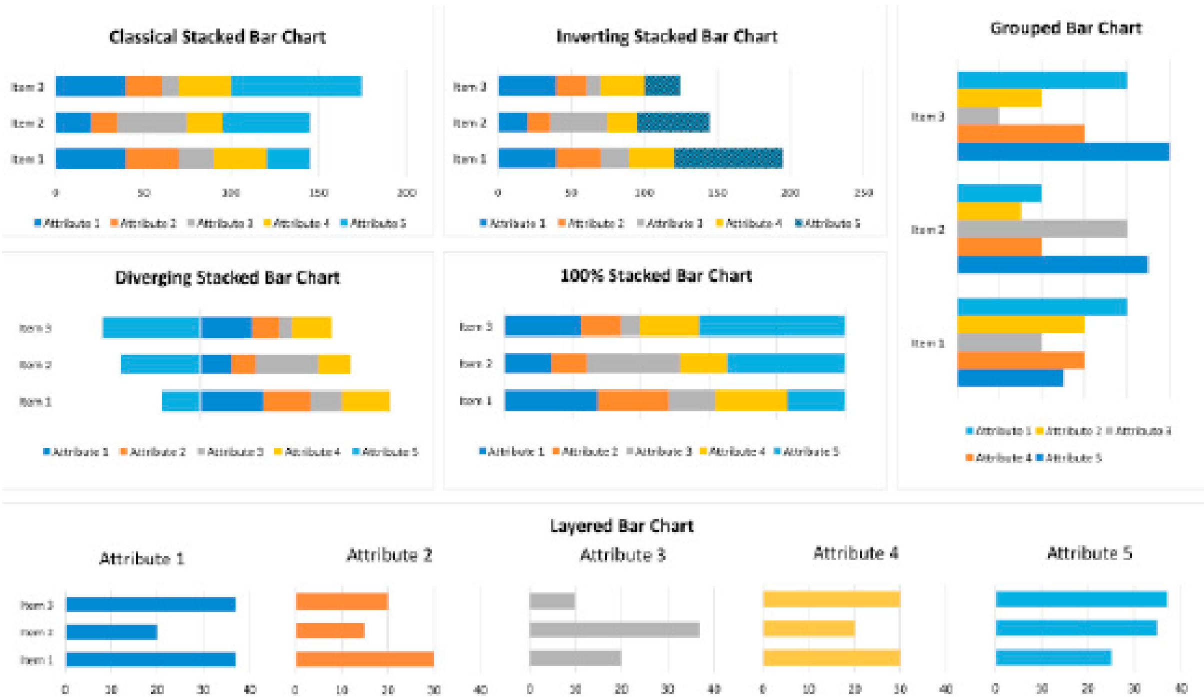
Tasks: *what are the supported tasks?*
- Compare, lookup values

Scalability? → Hundreds of levels for key attribute



Ordered by key vs. ordered by value (data-driven)

Stacked Bar Chart



- Scalability?

Data:
- 2 Categorical attribute (keys)
- 1 Quantitative attribute (value)

Mark:
- Stack of line marks

Channels:
- Length and color hue
- Spatial regions: one per mark
    - Aligned: first bar
    - Unaligned: other bars

Tasks:
- Compare, lookup values
- Part-to-whole relationship

- Hundreds of levels for key attribute (same as bar chart obviously)
- <u>Stacked attribute:</u> several to one dozen (since colors are the main factor)

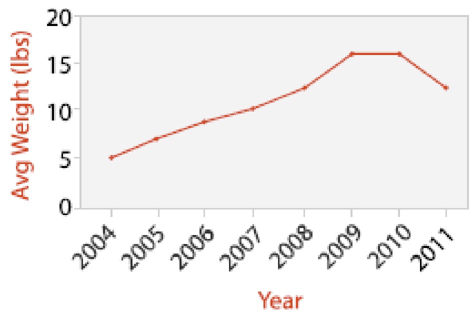
Line Chart

Data:
- 2 Quantitative attributes
- One key, one value

Mark:
- 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

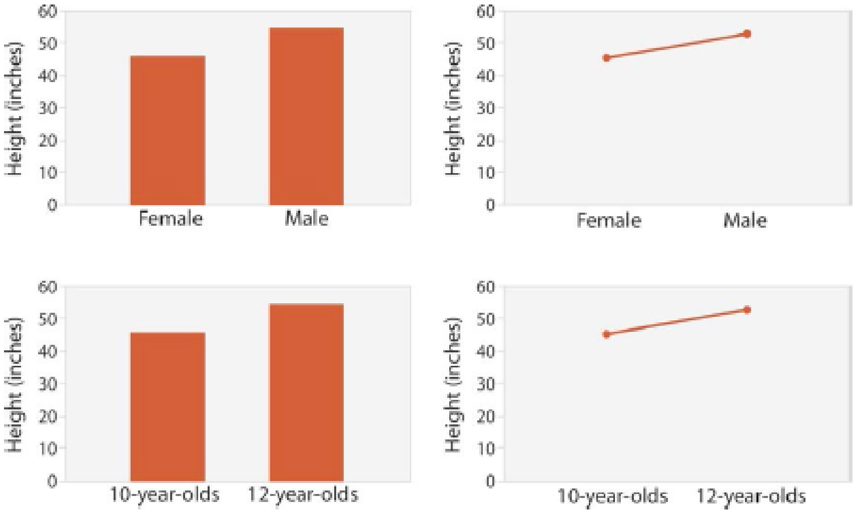


- Scalability?
    - Key attributes dozens to hundreds

Choosing Bar vs. Line Chart

- Depends on the type of key attributes
    - bar charts if categorical
    - line charts if ordered
- Do not use line charts for categorical key attributes
    - violates expressiveness principle

- implication of trend so strong that it overrides semantics. “The more male a person is, the taller he/she is”



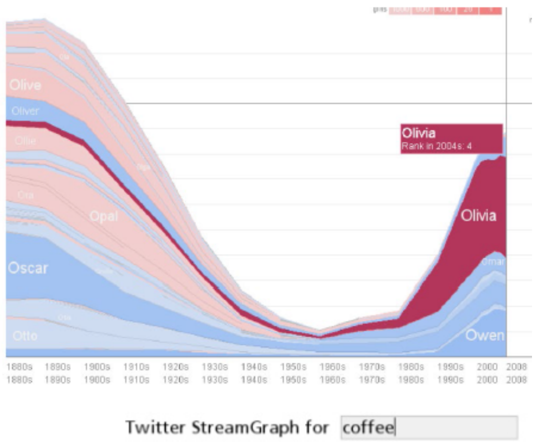
Streamgraph

Data:
- 1 Categorical attribute (names)
- 1 Ordered key attribute (time)
- 1 Quantitative value attribute (counts)

Marks & channels:
- Derived geometry: layers, height encodes counts.

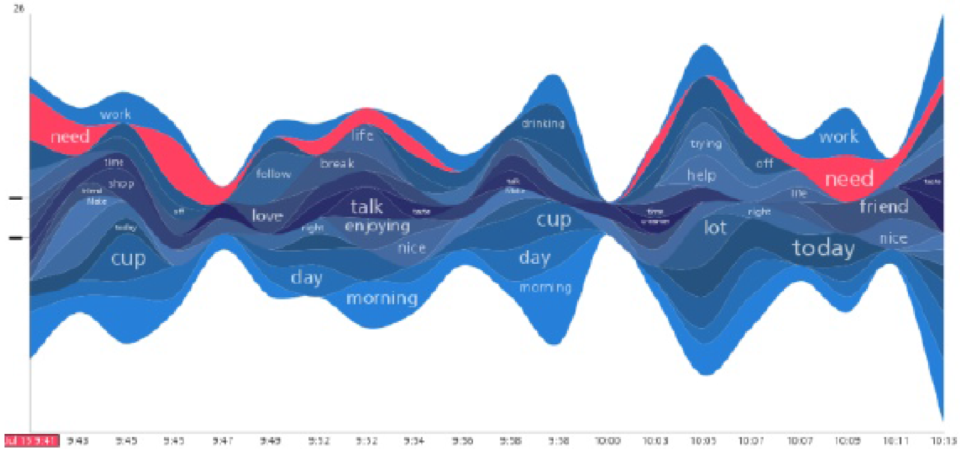
Tasks:
- Find trends
- Part-to-whole relationship

Generalized stacked graph
Emphasizes horizontal continuity



- Scalability? →
  - Hundreds of time keys
  - Dozens to hundreds of (names) keys

- more than stacked bars, since most layers don't extend across whole chart



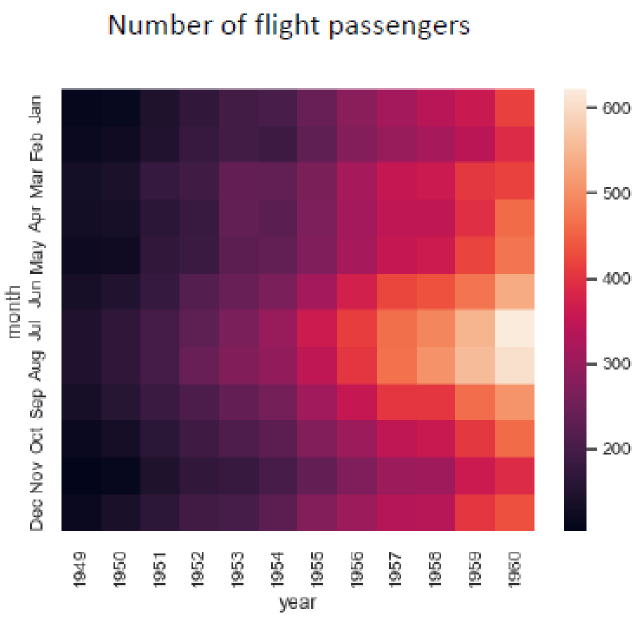
Heatmap

Data:
- 1 Quantitative attribute
- Two keys, one value

Mark:
- Separate and align in 2D matrix
- Indexed by 2 attributes

Channels:
- Color by quantitative attribute

Tasks:
- Find clusters, outliers, patterns



Scalability? → 1M items, 100s of categorical levels, ~10 quantitative attribute levels

Pie Chart | Polar Area Chart

- Scalability → One dozen

Data:
- 1 Categorical attribute
- 1 Quantitative attribute
- One key, one value

Mark:
- Separate colored area

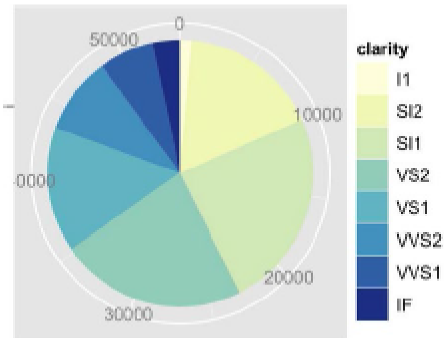
Channels:
- Color by categorical attribute
- Angle for quantitative attribute

Tasks:
- Part-to-whole judgement

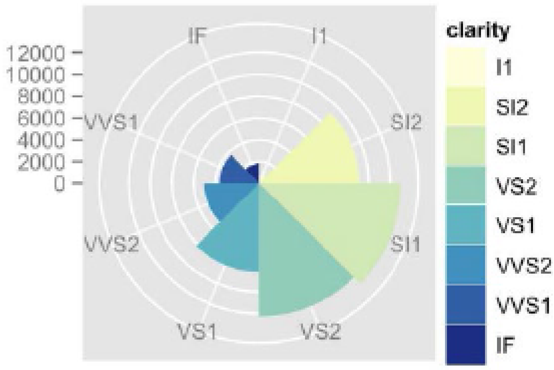
(since we are talking colors)

- Key → categorical
- Value → Quantitative

Pie chart



Polar area chart
- Area marks with length channel
- “Radial bar chart”



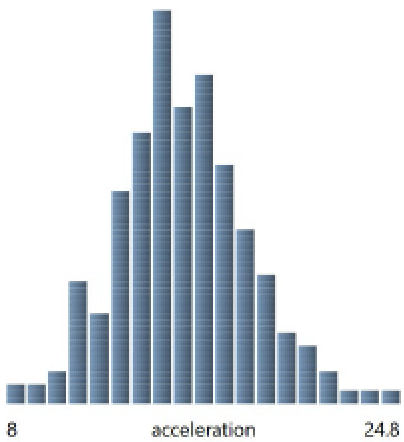
Histogram

Data:
- 1 Quantitative attribute
- **Derive data**:
  - Keys are bins, values are counts
  - Bin size is crucial

Mark:
- Line

Channels:
- Length encodes frequency

Tasks:
- Understand distribution



Boxplot

Data:

- 1 Quantitative attribute
- Derive data:
  - 5 quantitative attributes
  - Median, min, max, lower + upper quartile
  - Explicitly show outliers

Mark:

- Lines (+ box)

Channels:

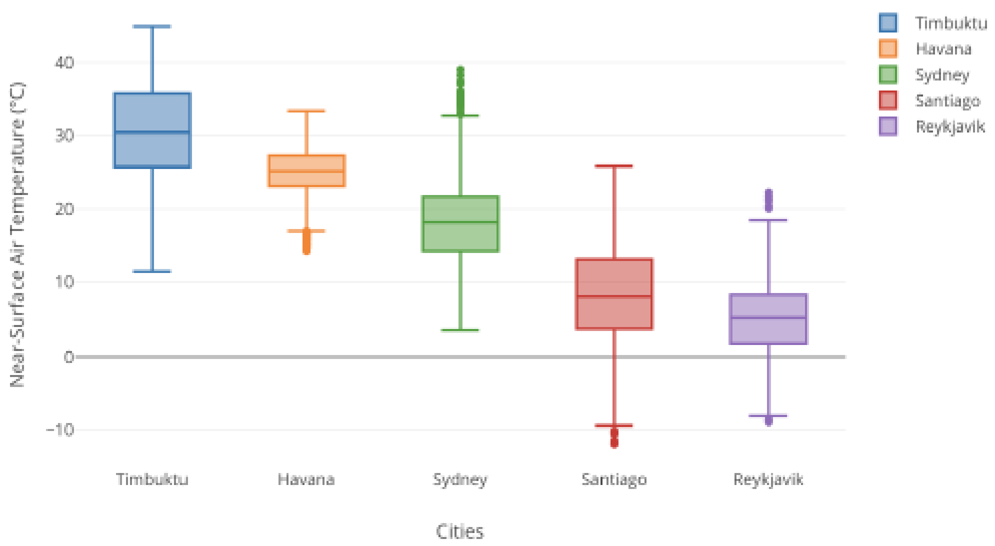
- Length encodes derived values

Tasks:

- Understand distribution

There is 1 categorical key as well

Points/Dots are also included in Marks



https://cds.climate.copernicus.eu/toolbox/doc/gallery/54_box_plots.html

Be aware boxplot hides details

This is because we are using derived data/values. For example, we cannot see what happens below the 25th or above the 75th percentile. We don't know how the distribution really looks like

Violin Plot

Data:

- 1 Quantitative attribute
- Derive data:
  - 5 quantitative attributes (boxplot)
  - Density at each point

Mark:

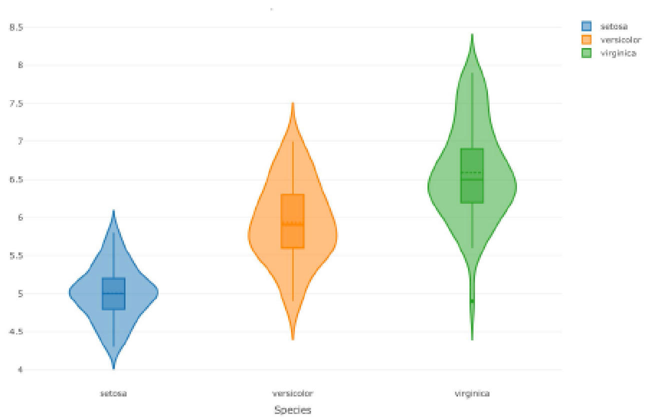
- Lines (+ box)

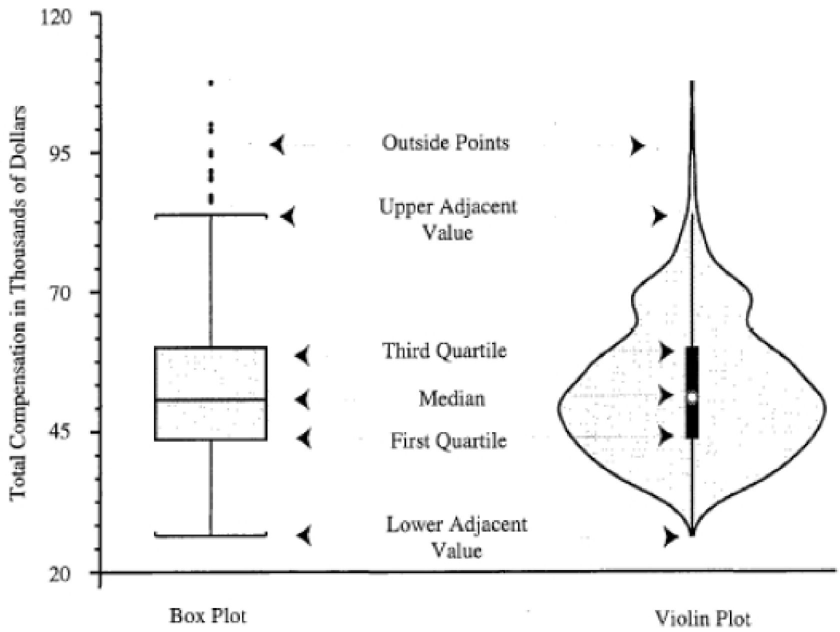
Channels:

- Length encodes derived values
- Width encodes frequency

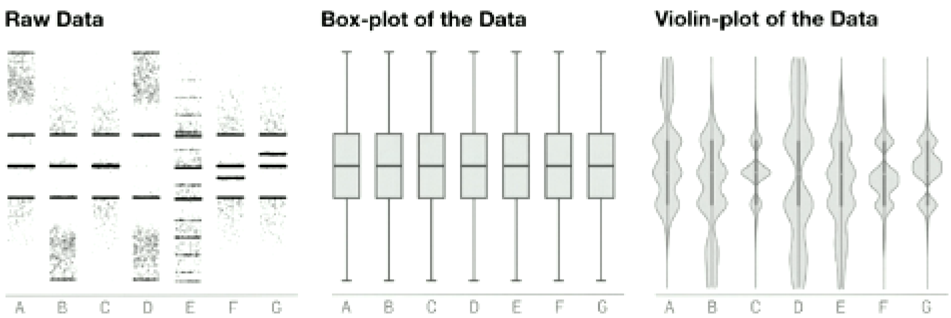
Tasks:

- Understand distribution





Comparison: Boxplot vs. Violin Plot



- Violin plots provide more detail of the data: especially which part of the data has a higher or lower concentration of values.

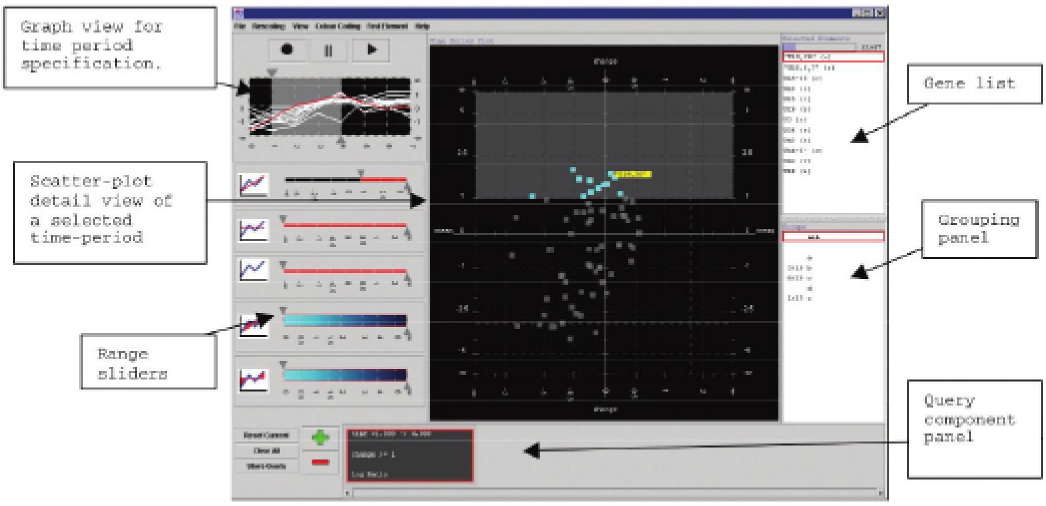
Interaction Principles (Analysis framework < HOW < Interactions)

- For interaction, low-latency visual feedback is needed (should be more real-time)

Time Constant	Value (in seconds)
perceptual processing	0.1
immediate response	1
brief tasks	10

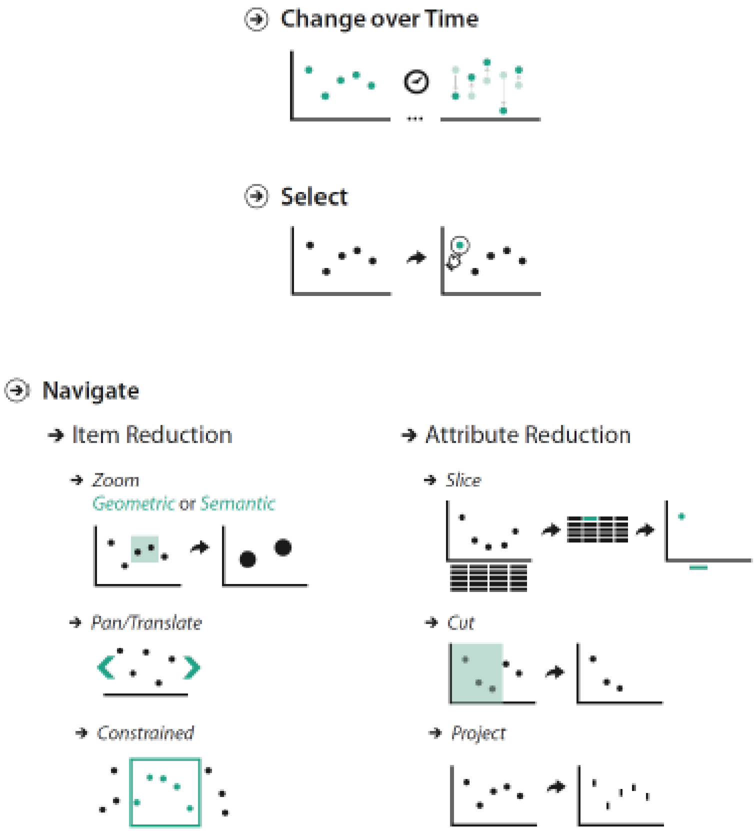
Time taken more than this is not considered real-time anymore. People will think something is wrong (it hangs or something)

    - Manipulate, e.g., selection and highlighting



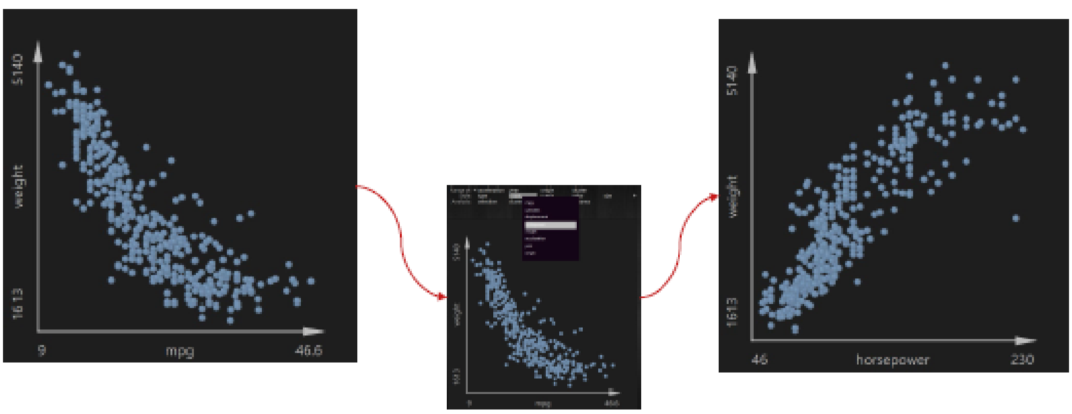
Manipulate

- How interaction/manipulation techniques are categorized? (according to Munzner) —



Interaction: Change over Time

- **Advantage of digital over paper**
    - you can now change encoding, parameters, viewpoint, aggregation, ...



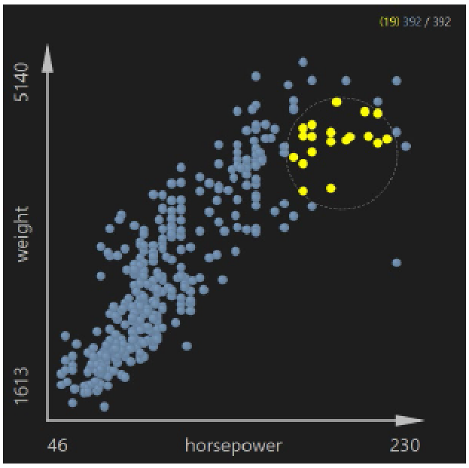
Interaction: change x-axis attribute

- **Within encoding**
    - change sorting order, rearrange layout, ...
- **Consider animations**

- to show different attributes or parameters or how a dataset changes over time (as seen in previous lectures)

Interaction: Select

- Basic operation for all interaction techniques
- **Design choice:** click vs. hover

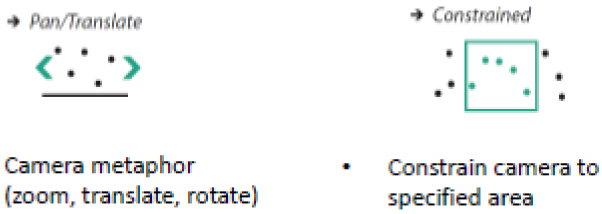


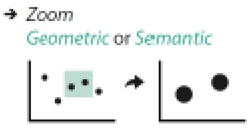
Selection tool; can move around the selection circle

- **Highlight:** change visual encoding for items of interest (ones you have selected or are hovering over)
  - Color
  - Border
  - Explicitly link items
  - ...

Interaction: Navigate

Item reduction — decreasing the number of items

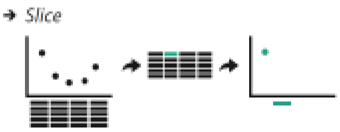




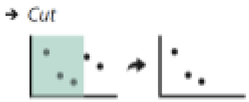
Change viewpoint / visibility of items
Camera metaphor
(zoom, translate, rotate)
Geometric – camera
Semantic – change visual encoding
depending on zoom-level

- **Semantic zooming** means removing detail based on level of zoom. This is not the case for **geometric zooming**

Attribute reduction



Show only items matching specific value for given attribute



- Show only items within some region or attribute range

- Dimensionality reduction (e.g., 3D → 2D)

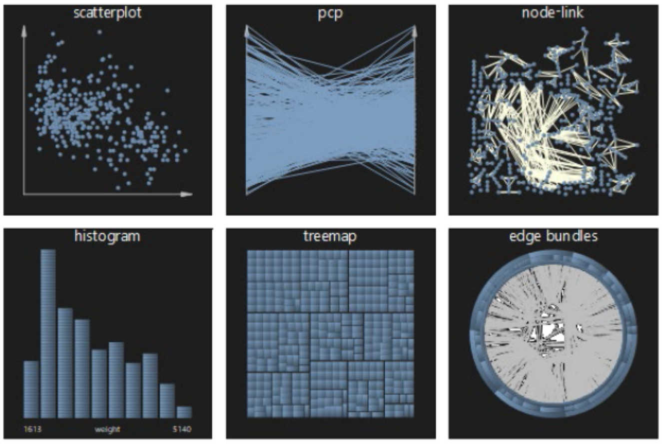
- Slicing example — show only cars with 3 cylinders

Coordinated multiple views

Why do we have multiple views?

- Show relations between items and attributes (multiple visualizations might offer new knowledge)
- Provide different perspective on same items
- **Design choices**
  - View count: few vs. many
  - View visibility: popup vs. side-by-side
  - View arrangement: manual vs. automatic
- **Linking & brushing**
  - Linked view parameters

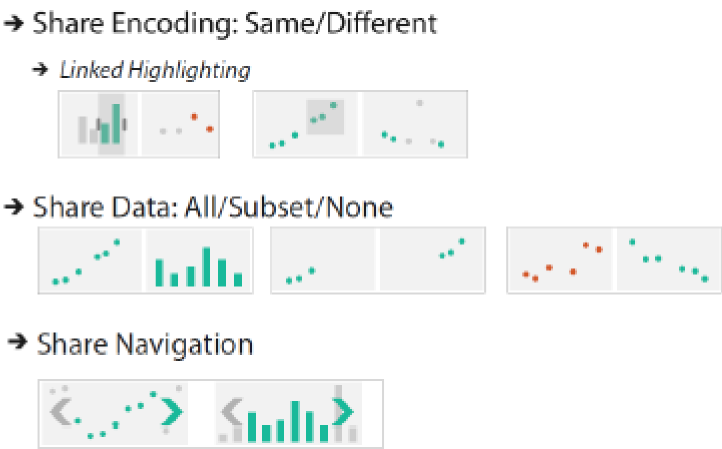
- **Linked highlight/selection** — items highlight/selected in one view are also selected in the other views.



Different visual encodings of same data

- You can also see the impact of changing one attribute in a visualization to the other visualizations

Interaction coordinated multiple views



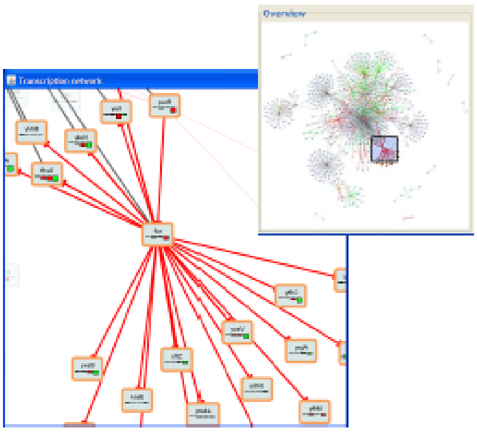
- Shared navigation — If you pan or zoom in one visualization, you can zoom or pan in the other visualization at the same time.

Coordinated multiple views

Types of multiple views

- **Multiple views:** different visual encodings of same data.

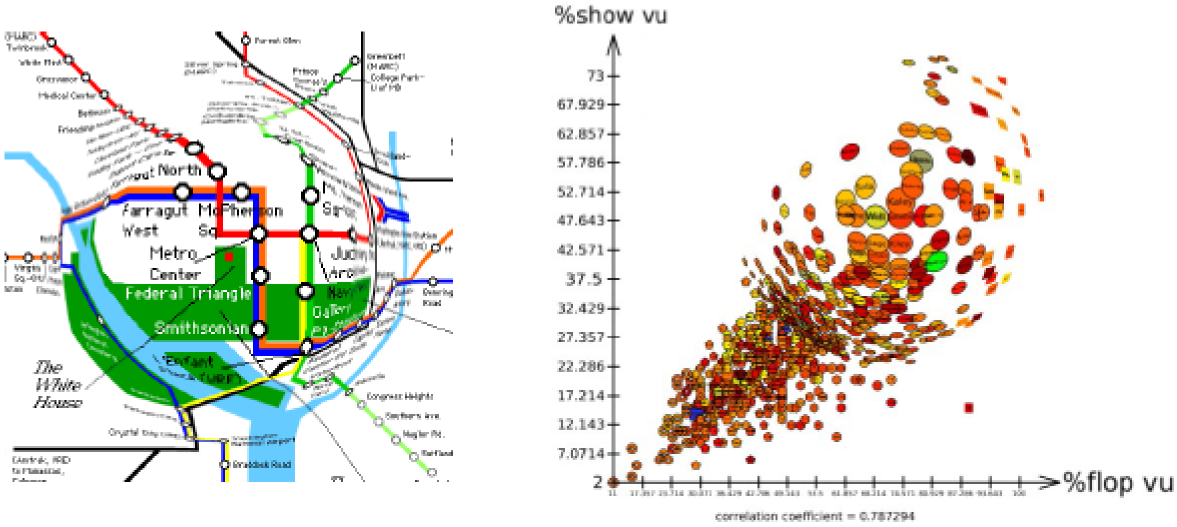
- **Small multiples:** same representations of data, changing for one attribute.
- **Overview & detail:**
    - Same visual encoding, same data, different zoom-level
    - **Detail on demand:** extra view with more info on selection (this is for context; to know where exactly you are looking at):



Focus + Context Visualization

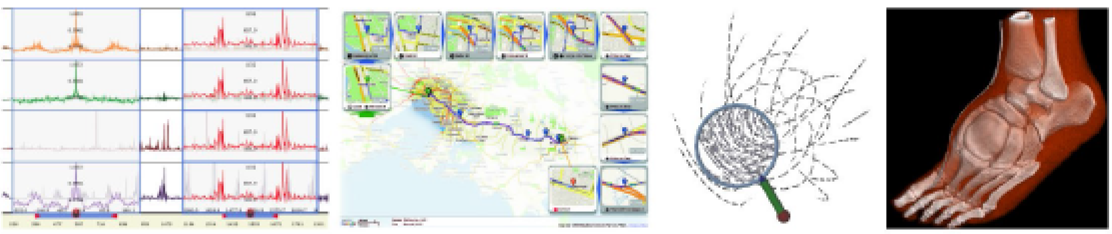
Different levels of detail integrated in the same view

- Show area/items of interest (focus) in detail and surroundings (context) in less detail
- Distortion techniques are used quite often

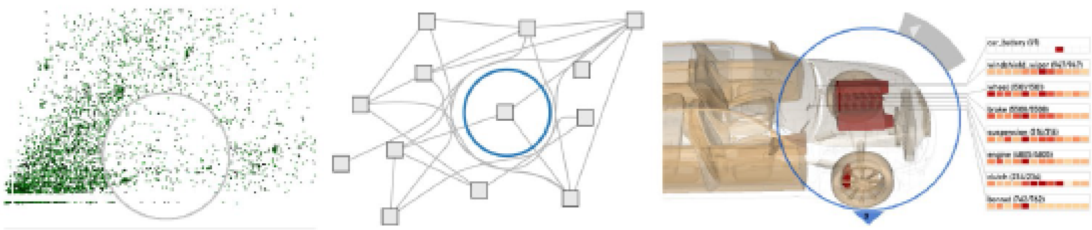


Detailed + fisheye view

- These techniques usually concern **lens**
  - you can hover over the area of interest to view in more detail; can also be used to filter out certain data



(a) Temporal data (b) Geospatial data (c) Flow data (d) Volume data



(e) Multivariate data (f) Graph data (g) Text and document data

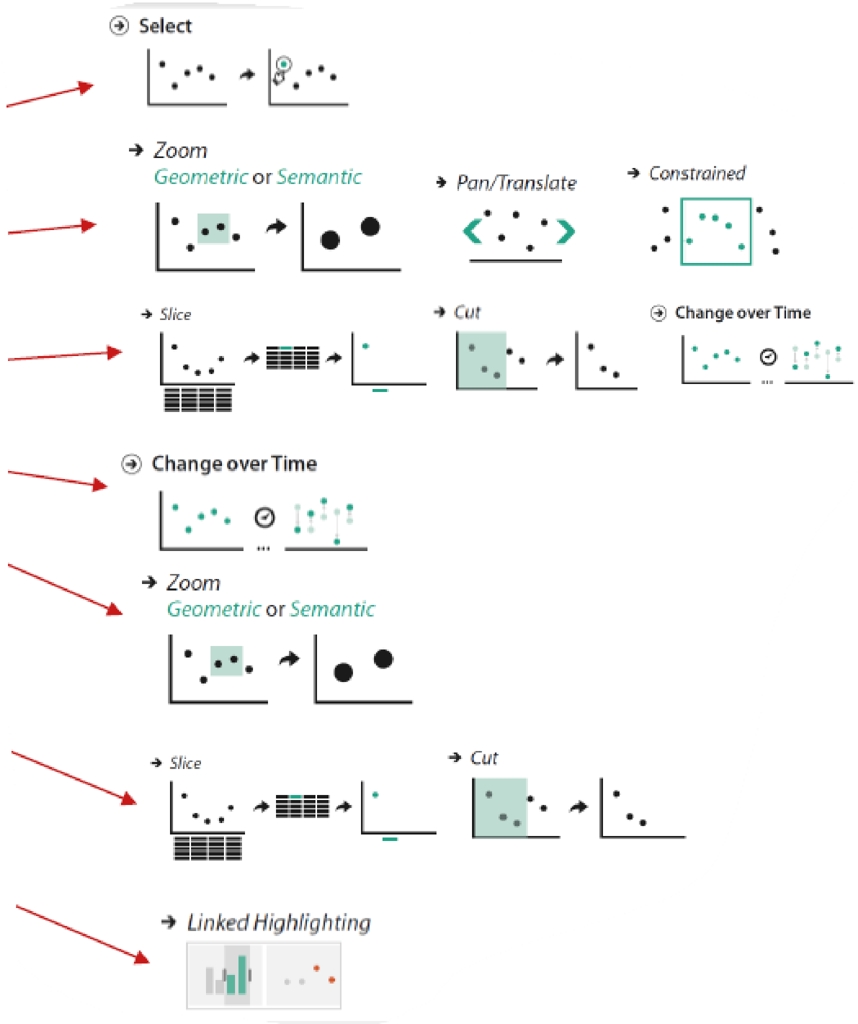
---

Taxonomy based on user's intent

Yi et. al – seven categories of interaction techniques based on user's intent.

1. **Select** — mark something as interesting
2. **Explore** — show me something else
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



IDIOMS & INTERACTION summary

- Idioms put restrictions on tasks (and vice versa), so **choose carefully**
- Idioms are described with data, marks, channels, tasks

- Interaction should support speed of thinking
- Overview first, zoom & filter, details on demand

- Munzner's taxonomy based on techniques
- Yi et al. taxonomy based on user intent

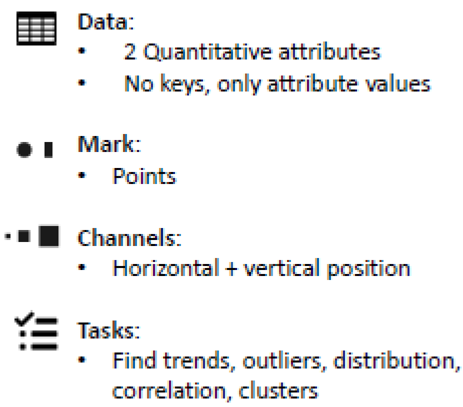
- Justify choices and be creative!

Lecture 8 - Complex multivariate idioms & Data reduction

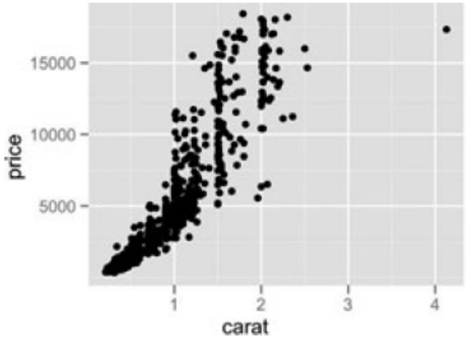
Date	@December 17, 2021
Type	Lecture
Completed	☑
Book	☑
Notes	-
Week	5

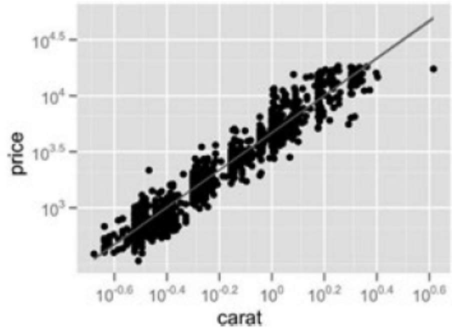
Visual Encoding < HOW < Analysis Framework

Scatterplot — another idiom



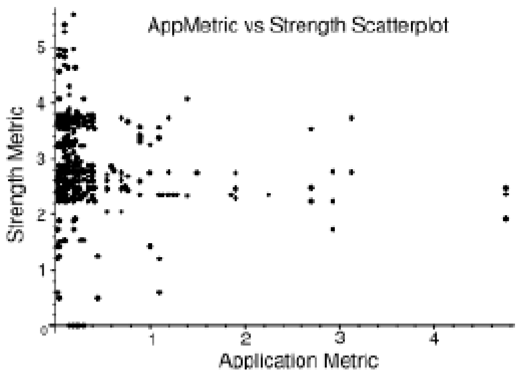
- **Scalability** → Hundreds of items

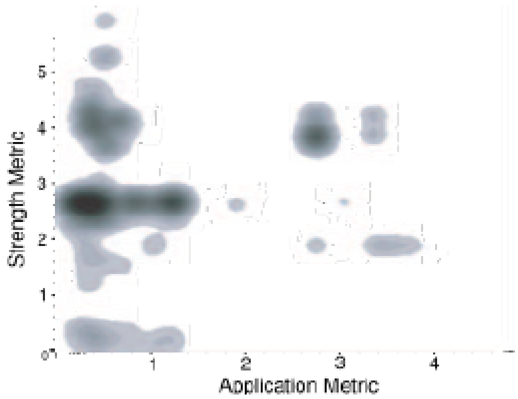


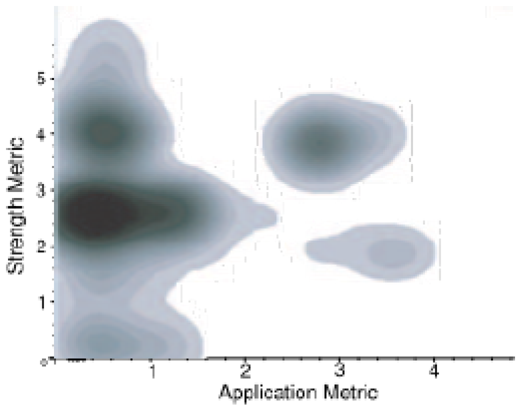


Scatterplot scalability

- change (size of) mark or add transparency so view can be improved
- **Multiscale scatterplot —**
    - smoothing by convolving with Gaussian kernel at different scales (KDE - Kernel Density Estimation)
        - using larger kernel gives less detail but you get a better view of what's happening.
    - Facilitates finding patterns at different scales

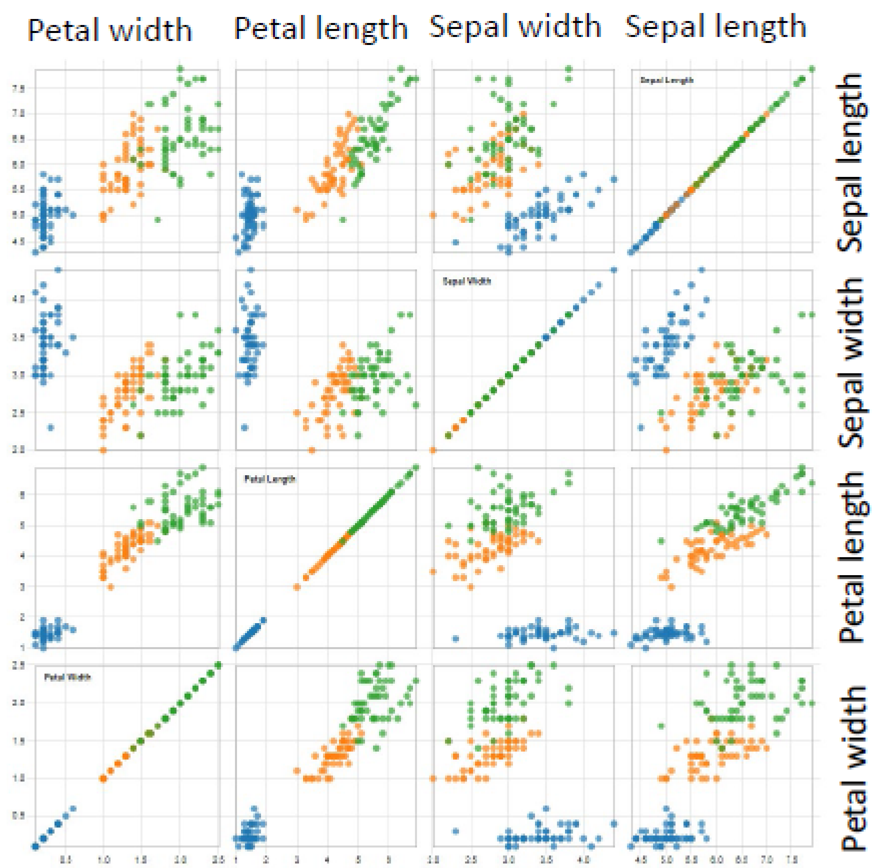






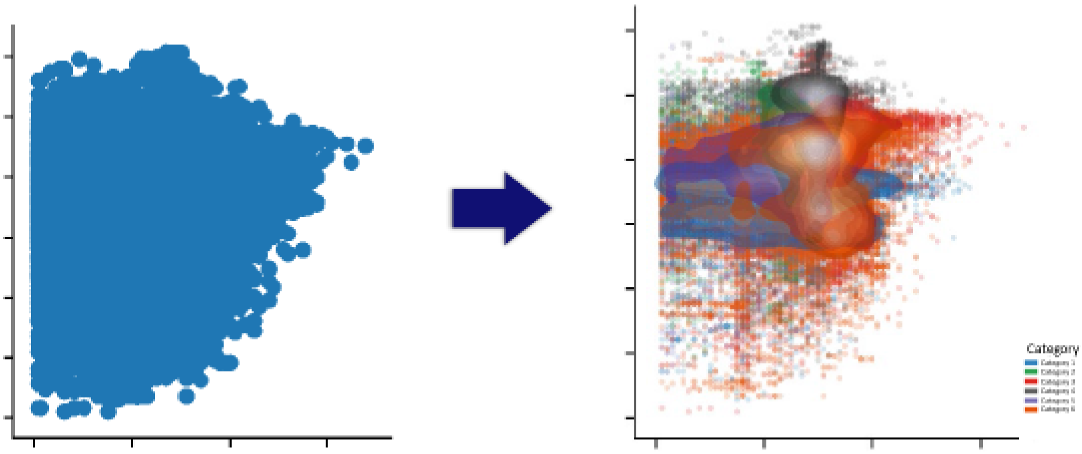
<u>WHAT if we have many attributes?</u>

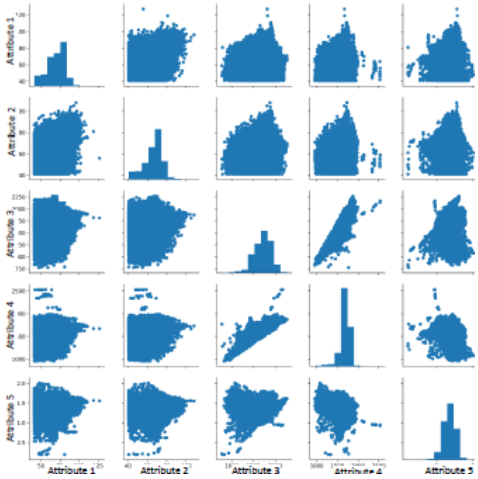
- We use **Scatterplot Matrix (SPLOM)**



- Matrix **showing all relations** between variables
- **Interaction**: linked brushing
- **Symmetric**
    - we can use a diagonal for other idiom

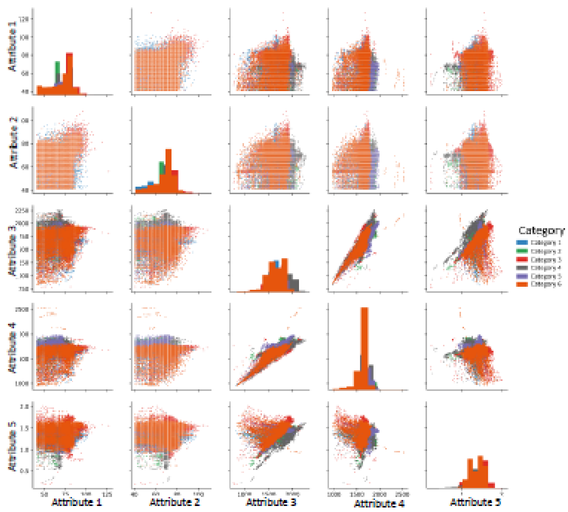
Improving plots



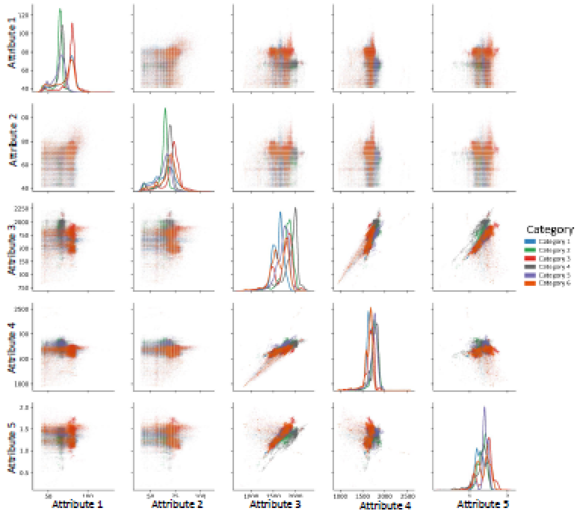


Heavy overdraw here

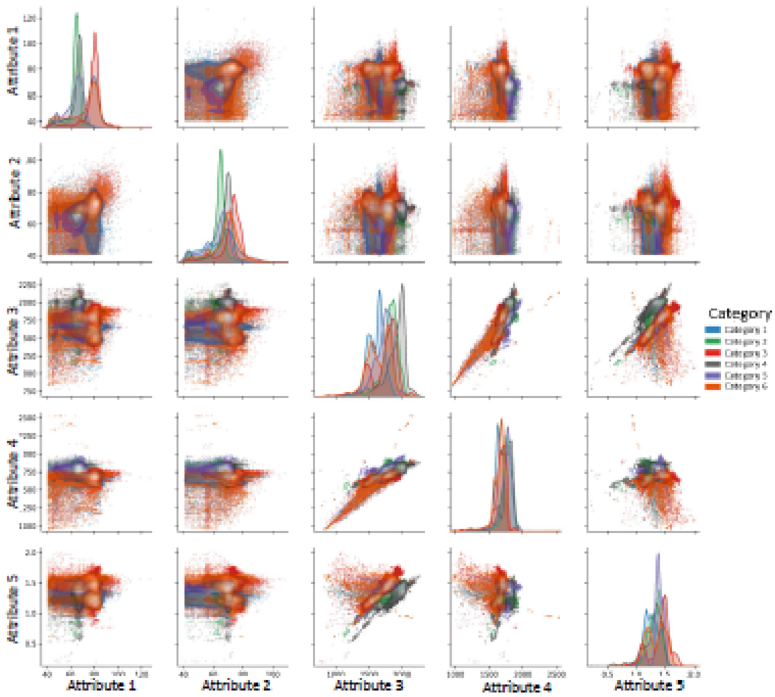
- So we color by category and set transparency:



- Then, we reduce point size, add some more transparency, and add KDE on diagonal:



- Then we add KDE on off-diagonal and further reduce with interaction (zoom, filter):



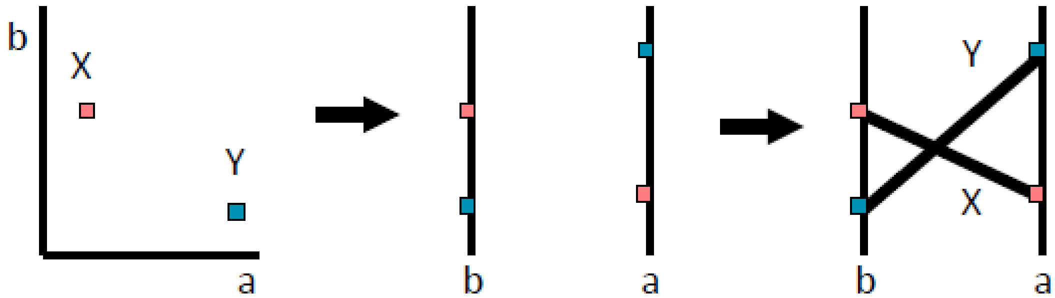
Parallel Coordinate Plot (PCP)

- All axes plotted in parallel

- Each item/sample is a **polyline**

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

- Axes are scaled to min/max range of data
- All dimensions shown at the same time

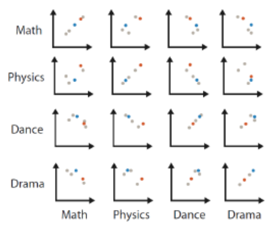


- **EXAMPLE —**

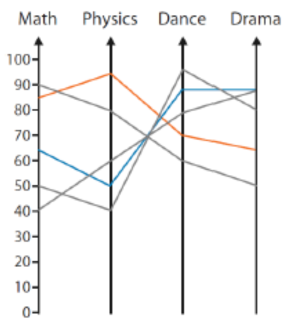
Table

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

Scatterplot Matrix

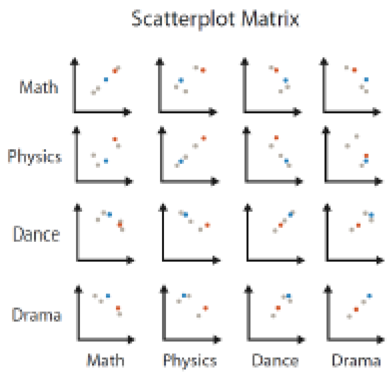


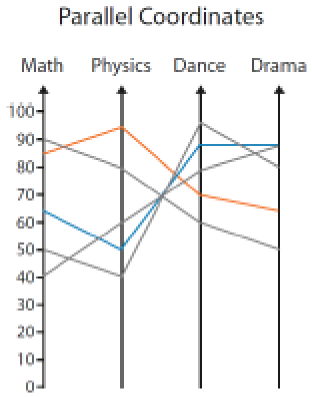
Parallel Coordinates



- Helps observe pairwise correlation

Scatterplot — PCP comparison





Positive correlation
- Diagonal low-to-high

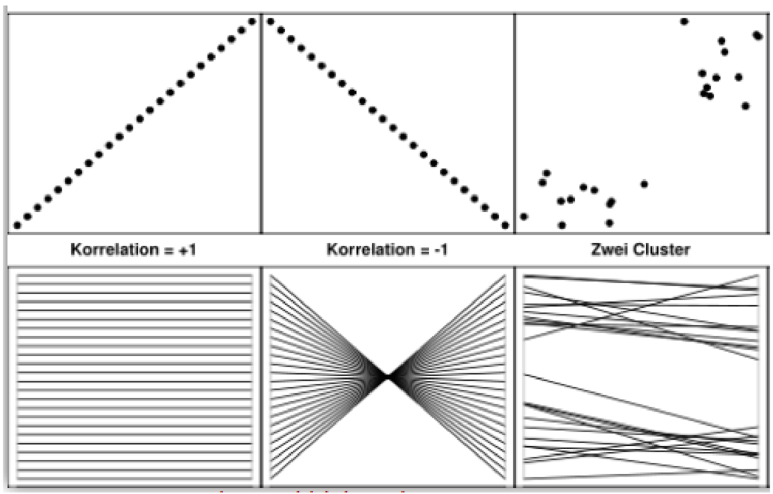
Negative correlation
- Diagonal high-to-low

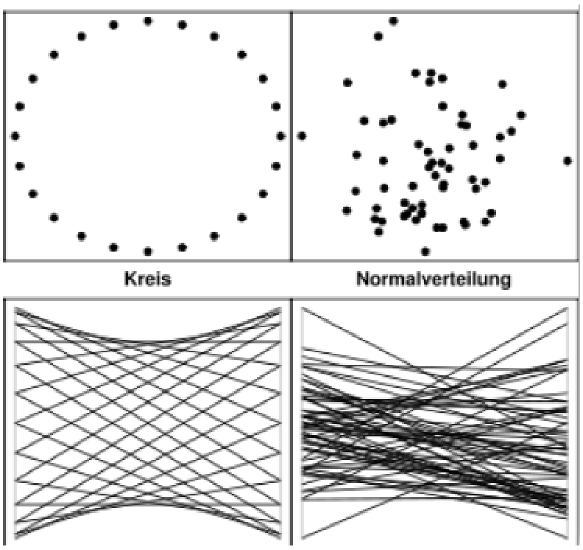
Uncorrelated
- Scattered points

Positive correlation
- Parallel line segments

Negative correlation
- All segments cross at halfway point

Uncorrelated
- Scattered crossings



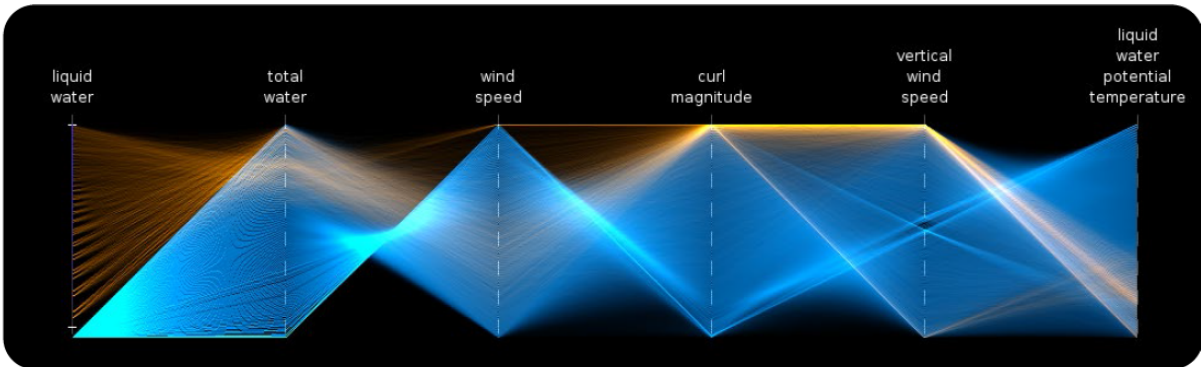


Scalability

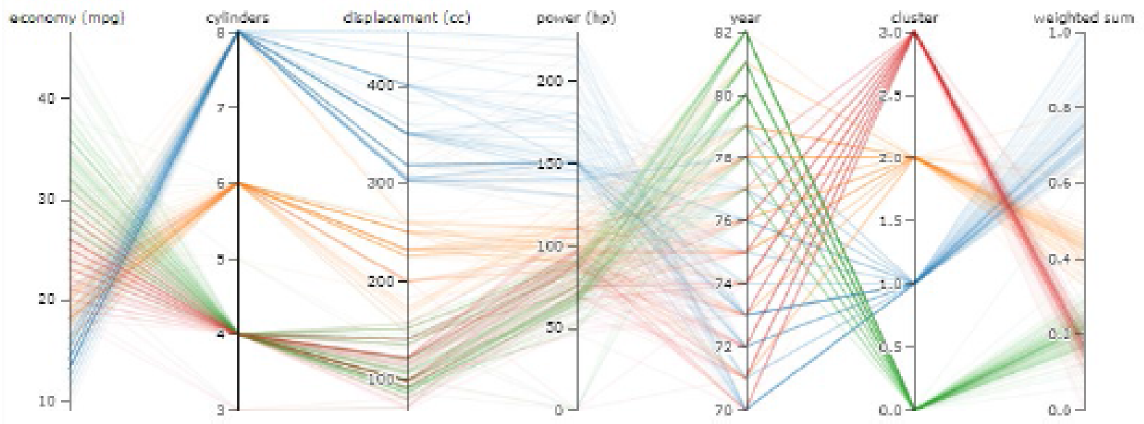
— dozens of attributes, millions of items (with effort)

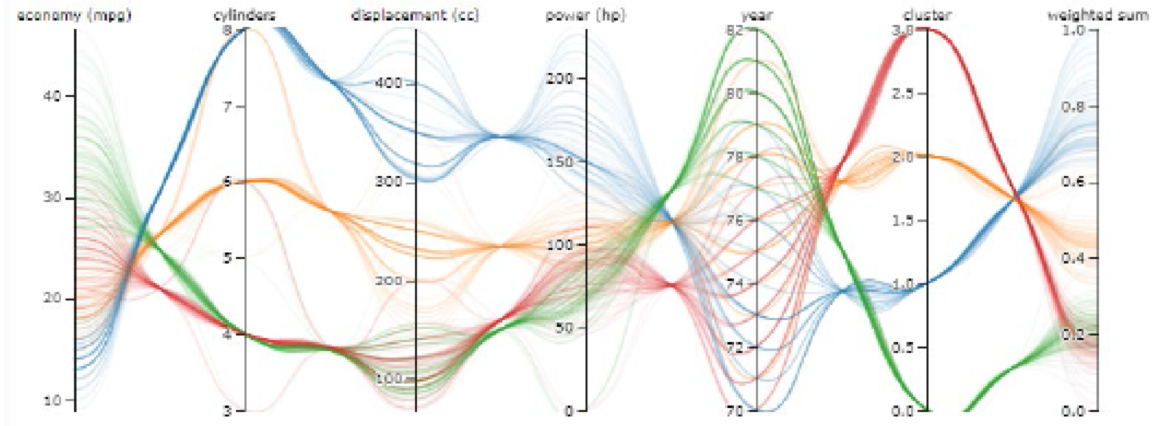
Standard drawing vs. transparency + adaptive blending

- Adaptive blending is when the color value for multiple overlapping lines is added to produce one color. A kind of highlight is produced when coupled with transparency:

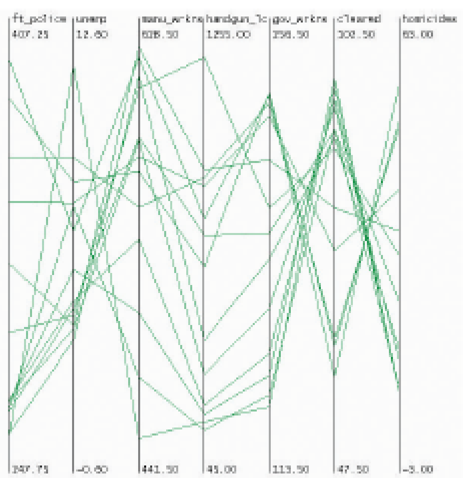


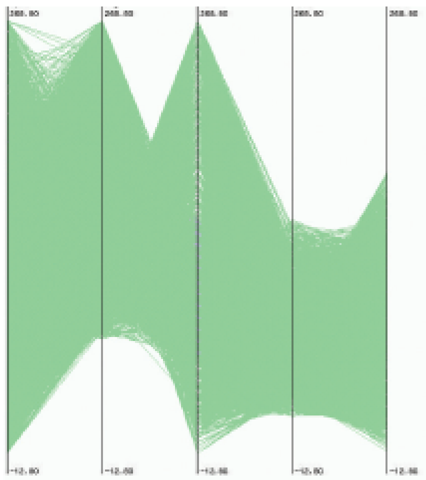
- There are 2 additional ways to improve readability:
  - **Smoothing** — removes sharp boundaries and corners (related to Gestalt's principle of continuity)
  - **Clustering** — groups together the lines using a 'strength' variable. Maximum strength would cluster all the lines into one.
  - Demonstrated below :-



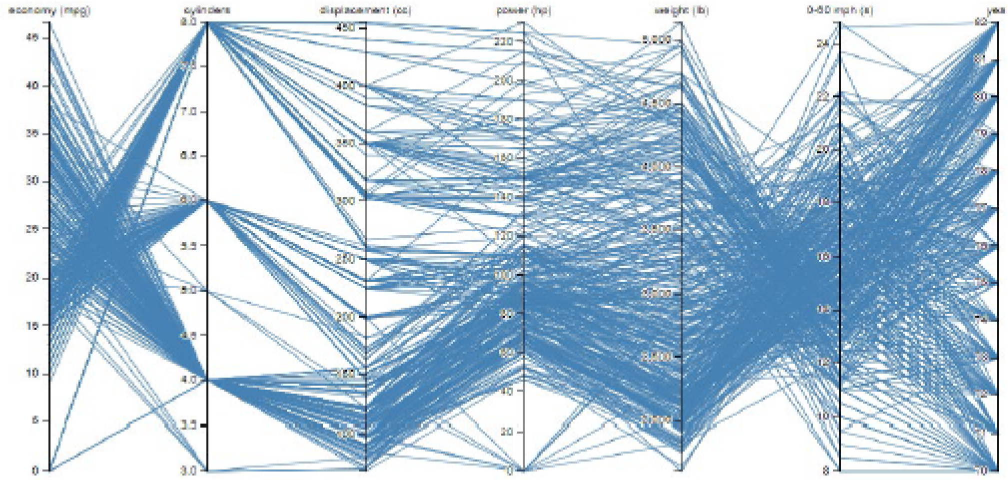


- Standard drawing results in bit of a overdraw, which hinders with interpretation:



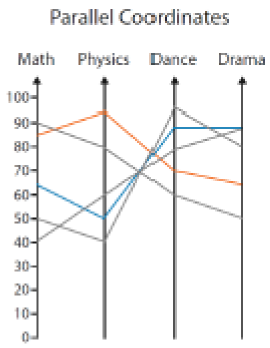


- We can also use interaction to reduce the number of variables (clutter). This can help with interpretation and exploration.

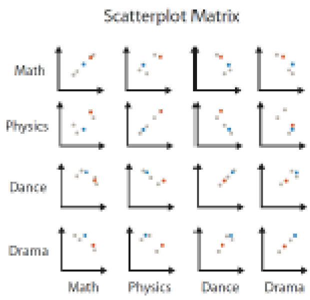


- **The order of the axes in PCP is very important!**
  - changing them largely impacts the interpretation (data can be lost)
- Varying **opacity** of the PCP can help reveal patterns/trends.

High-Dimensional Data Visualization



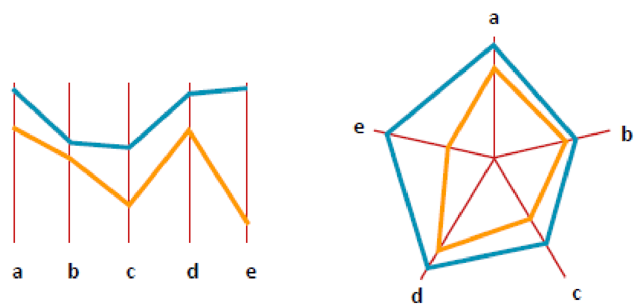
Parallel coordinates **(PCP)**
- Primarily relationships between adjacent axis
- Limited scalability (standard rendering)
  ~ 50 dimensions
  ~ 1-5k samples
- Interaction is crucial
- Axis ordering is important



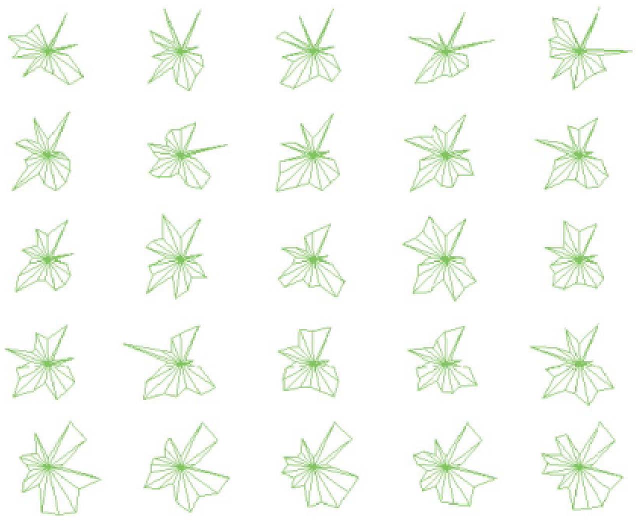
Scatterplot matrix **(SPLOM)**
- Primarily relationships between pairs of axis
- Limited scalability (standard rendering)
  ~ 20 dimensions
  ~ 500~1k samples
- Brushing is crucial

Radar Plots

- Instead of parallel, arrange axes in star shape
- Samples are now **polygons**
- Shapes can be recognized and compared



- This would also have serious scalability issues when the axes increase in number
- However, recognizing shapes is easier here.
- Another advantage is that at a higher level, multiple smaller radar plots can be constructed (**Iconification**)
    - this way, they can help observe shapes and patterns that can’t be seen with PCP

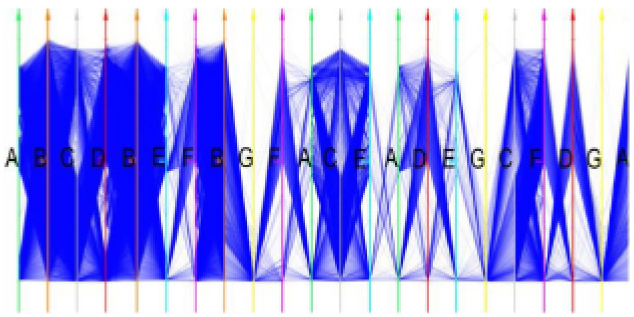


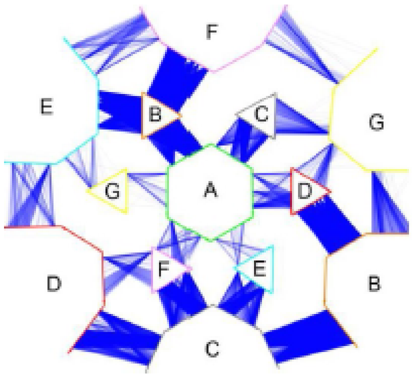
Flexible linked axes

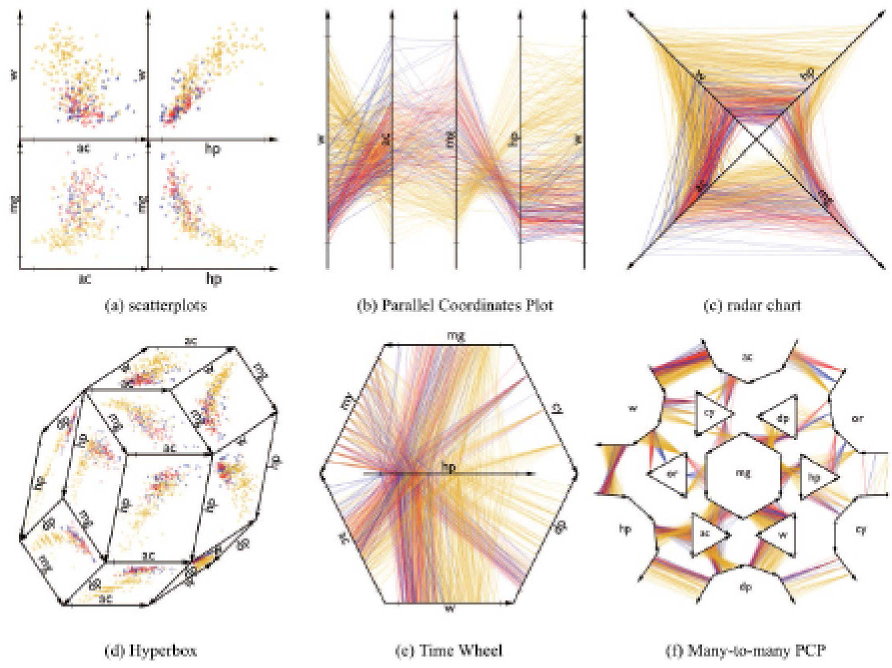
- Instead of parallel, arrange axes in star shape
- Samples are now **polygons**
- Shapes can be recognized and compared

The difference from radar plots is that instead of putting all the axes in a circle kind of shape, you can place you axes anywhere in the 2D space

- The advantage is that you can compare everything with everything (pairwise correlation)







(a) scatterplots

(b) Parallel Coordinates Plot

(c) radar chart

(d) Hyperbox

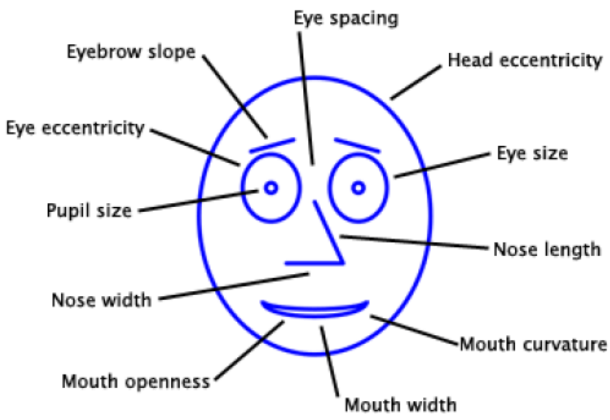
(e) Time Wheel

(f) Many-to-many PCP

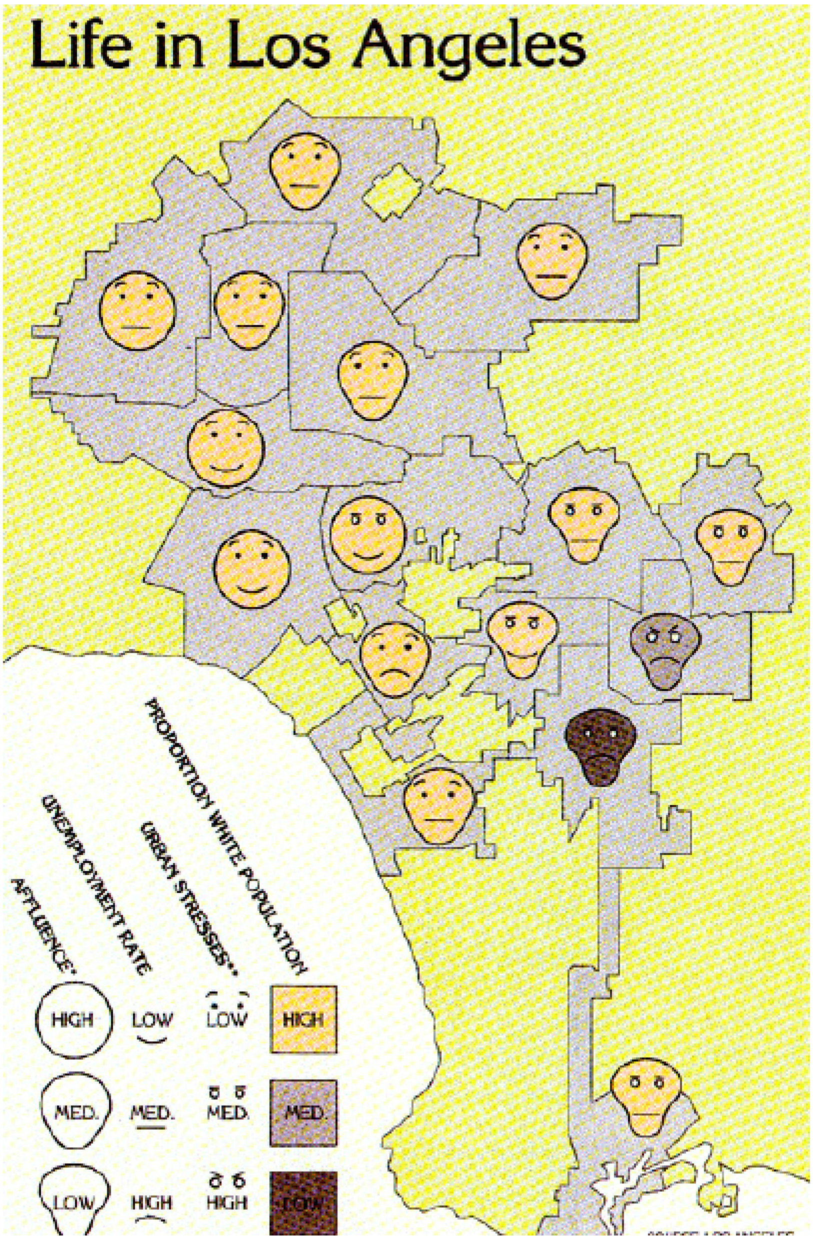
Icons/Glyphs (iconification)

— map multi-dimensional data to properties of graphics object

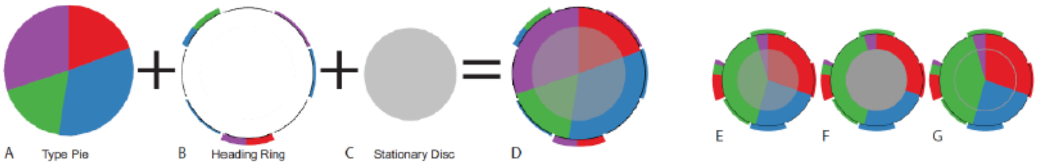
- Famous example: Chernoff faces (1973)
  - Use capabilities of humans to recognize faces to communicate information.

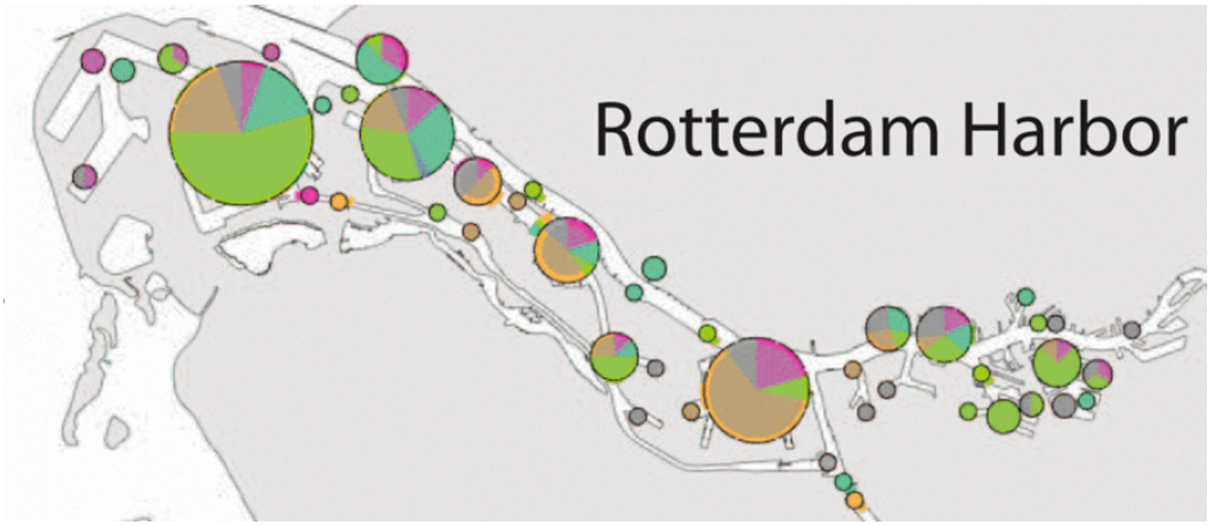


  - be careful as some features are more important than others



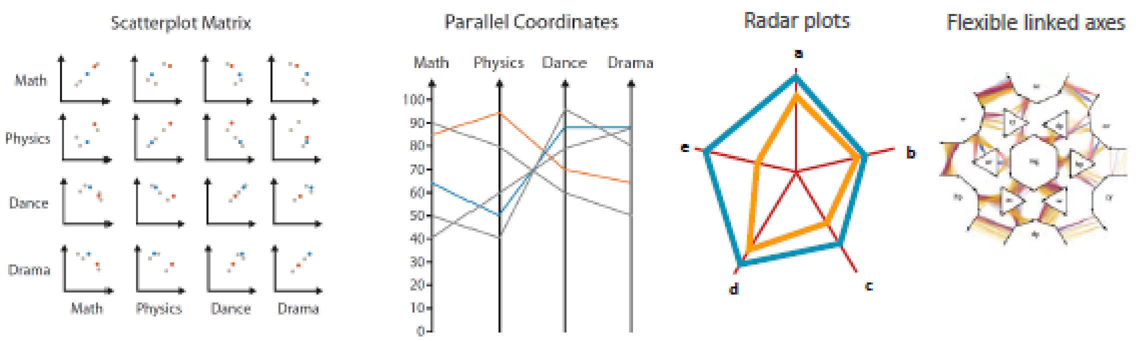
ANOTHER EXAMPLE —

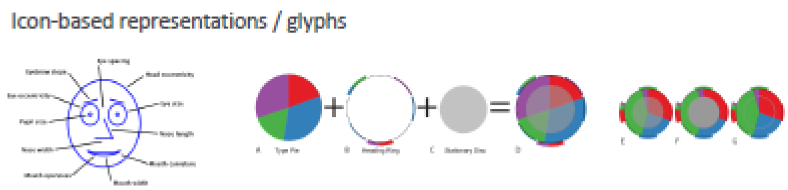




Representation for ships docked at Rotterdam harbor (where they are headed, what type of ship, etc.)

Multivariate idioms



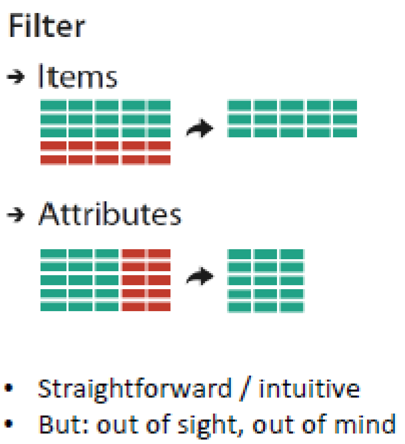


TU/e

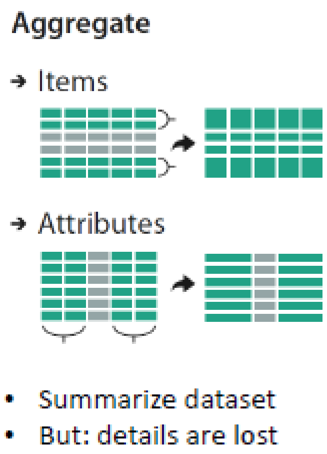
Data Reduction

- there is overdraw when there are a lot of variables, and then we can't see patterns anymore

Reduce items and attributes



We miss the context of those dropped



If two variables are almost the same, we can aggregate them into one

> (Filtering items) **GOAL —** eliminate items based on their values with respect to specific attributes

> **Number of attributes do not change**

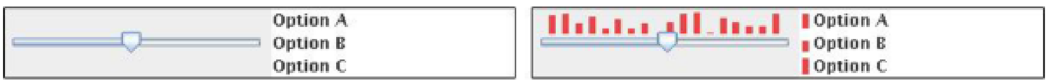
Item filtering

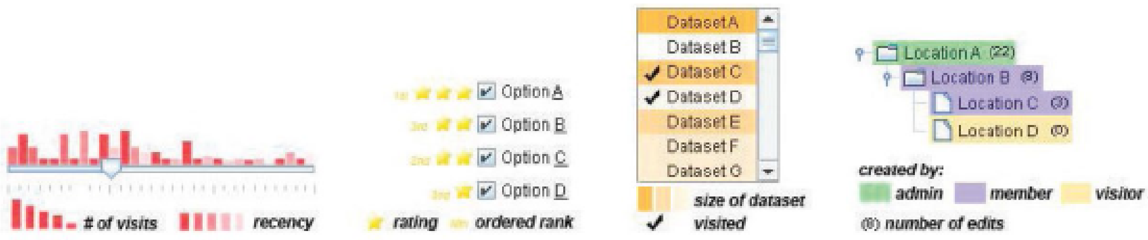
- **Dynamic querying** & **Direct manipulation**
- User is interacting with the system to process/filter data.
- **Immediately** show result.
- **Selection/filtering by pointing** (Visual queries). Not typing for example.

- **Promotes exploration**.

SCENTED WIDGETS —

- show data properties / distribution
- guide user where to look / filter





> (Filtering attributes) **GOAL —** eliminate attributes.

> **Number of items does not change (but fewer attributes for each item)**

Attribute filtering

— reduce the number of attributes

- **Strategy:** create attribute ordering (sort by most important ones)
  - e.g. remove highly correlated attributes
  - determine similarity — computationally (using mathematic formulas) or visually (using scatterplot matrix)

- Pearson correlation coefficient (ratio between covariance & standard deviation products)

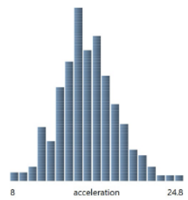
Item Aggregation

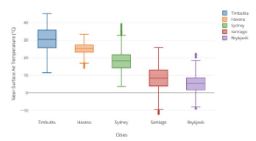
= reduce the number of items

> **GOAL —** reduce items through summarizing / grouping

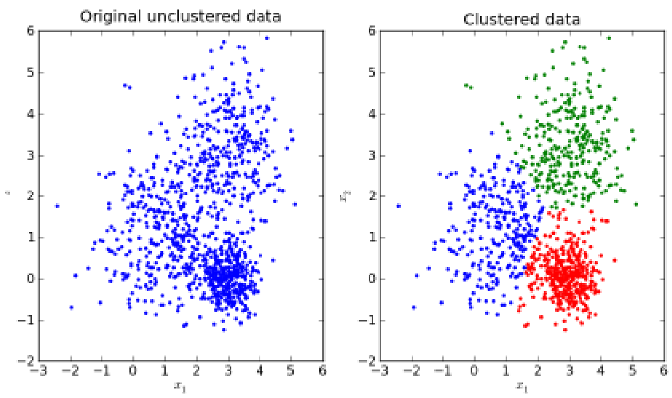
> **Number of attributes for each does not change**

- using visual encoding idioms
- supported by interactive idioms
  - dynamically update changes (fancy animations and visualization tools)

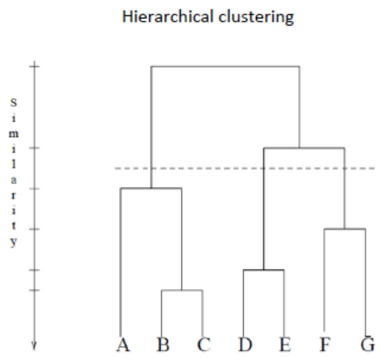


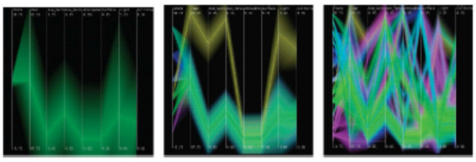


- Clustering
- Navigation between items and (alternative) clusters



- This can be done automatically but there are multiple possibilities (e.g. k-means, hierarchical clustering, etc.)





Attribute Aggregation

= reduce number of attributes

> **GOAL —** summarize attributes

> **Number of items does not change (but fewer attributes for each item)**

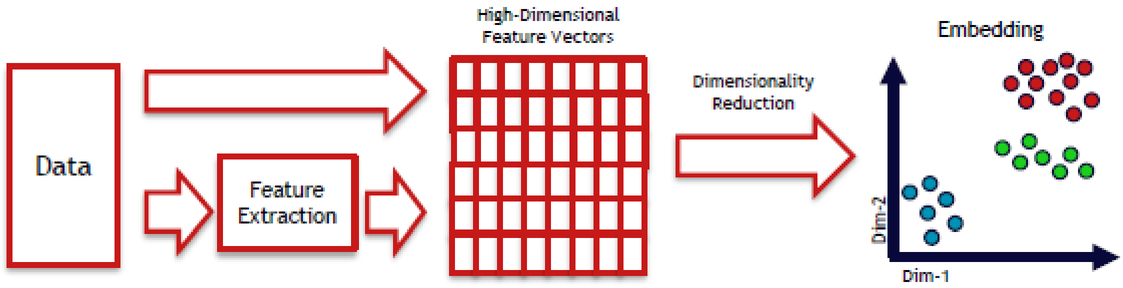
- **Strategy:** group attributes by a similarity measure
- More complex: **Dimensionality Reduction**
    - **Preserve meaningful structure** of a dataset while using fewer attributes to represent the items.
        - E.g., n-dimensional → 2-dimensional
    - **Find best (combination of) attributes that represent the n-dimensional space**
    - However, what is **complexity** of such a task? → $2^n - 1$
        - The complexity corresponds to n being number of dimensions and -1 refers to the empty set that does not need to be checked
    - Automatic (heuristic) methods

Dimensionality Reduction

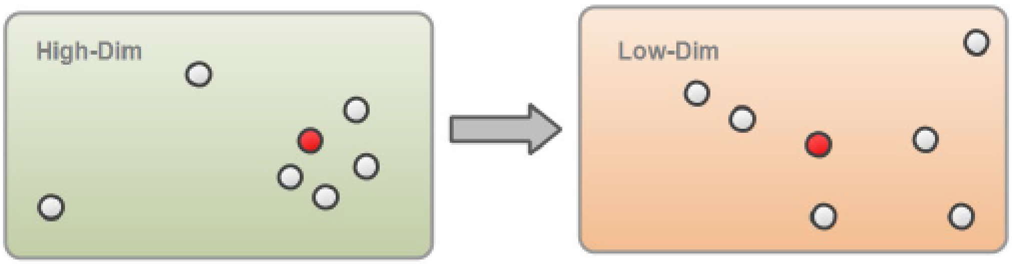
> **ASSUMPTION —** redundancy in the data

> **GOAL —** preserve “structure” of the n-dimensional space

What does structure mean?



- **Build** a **lower dimension** in which distances between points **reflect similarities in the HD data**

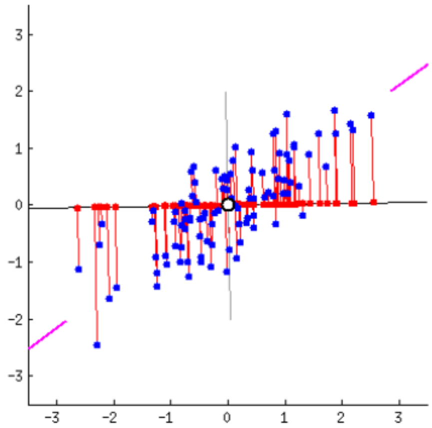


- **Minimize** an **objective function** that measure the **discrepancy between similarities** in the data and similarities in the map

Transformation to a projection/subspace (for Vis, mainly 2D)

<u>Linear method</u> — resulting attributes are linear combination of existing attributes (advantage = it is interpretable)

- **Principal Component Analysis (PCA)** — preserve variation



  - the horizontal red line moves around, each time calculating distance from all points and stops and the best fit

- **Linear Discriminant Analysis (LDA)** — preserve class separation
- ...

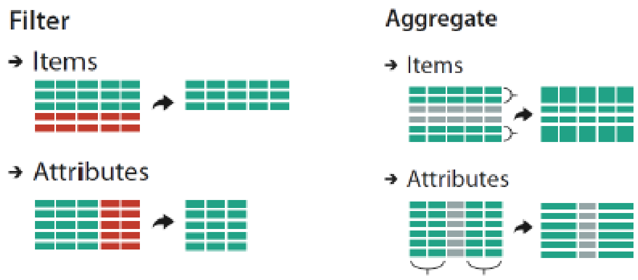
Non-linear method — resulting attributes do not have straightforward relation to original attributes

- **Multi-Dimensional Scaling (MDS)** — preserve distances
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)** — preserve neighborhoods
- **UMAP (Uniform Manifold Approximation and Projection) —** uses Riemannian metric and preserves global space better than the above ones
- ...

Which dimensionality reduction technique to use?

- Depends on the task and the data structure

SUMMARY — data reduction



- Straightforward / intuitive
- But: out of sight, out of mind

- Summarize dataset
- But: details are lost

Filtering and aggregation using

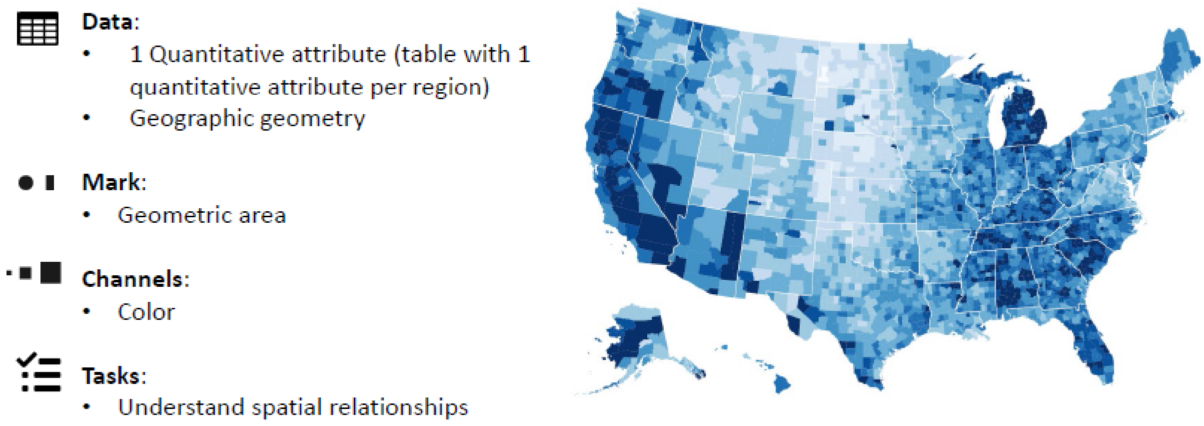
- Dynamic querying
- Direct manipulation

→ promotes exploration

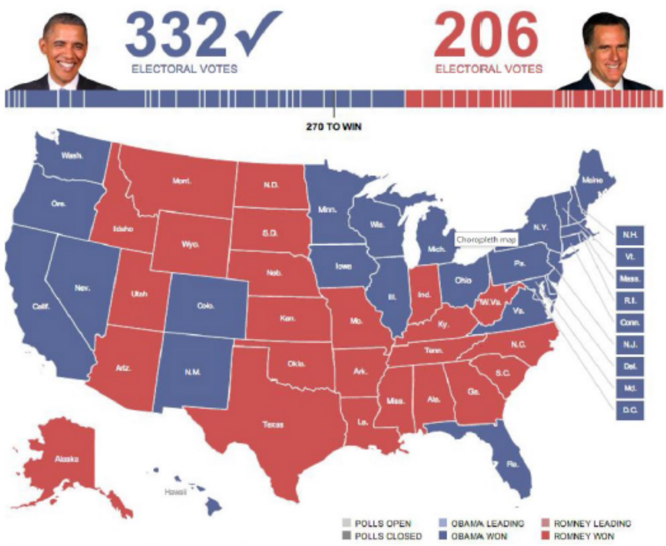
Lecture 9 - Maps, Graphs, Time Varying Data

Date	@December 22, 2021
Type	Extra Lecture
Completed	☑
Book	☑
Notes	

Choropleth Map



- ISSUE → the size of the objects depends on the geometrical geography , not on the value of the attributes:

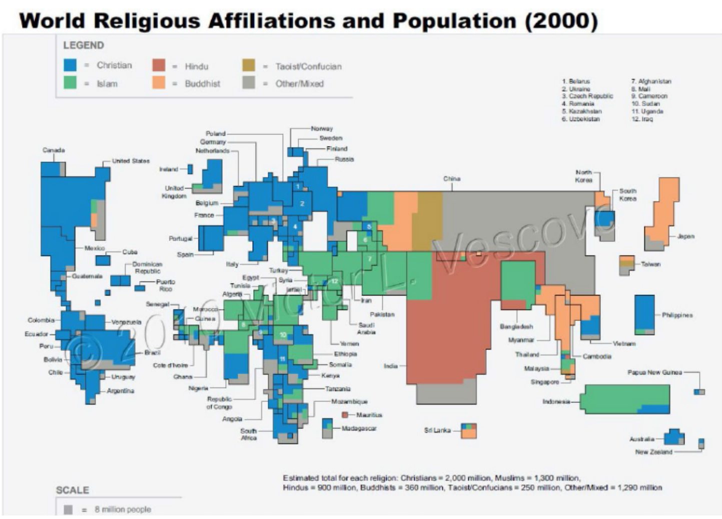


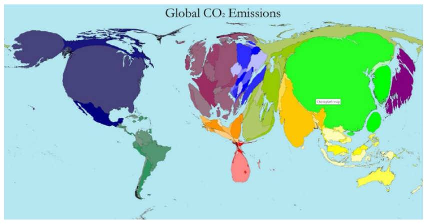
- From the amount of red on the map, one might think Romney won

- We can use cartograms to circumvent this

Cartogram

- We use area to encode the quantitative attribute





- Distortion can be used as well: change the size so that it corresponds to the quantitative value

Caution with Maps

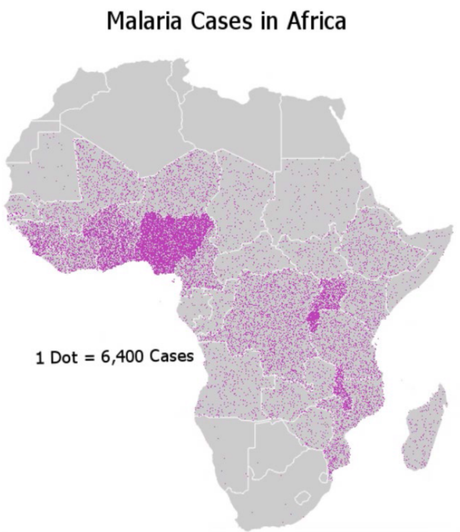
A dataset may contain geographical information

- Yet creating a geographical visualization may not be relevant
- Ask: "Does spatial arrangement matter for my task?"

- Position is the most effective visual channel ⇒ do not waste it if its not relevant
- A map not always the best or only solution

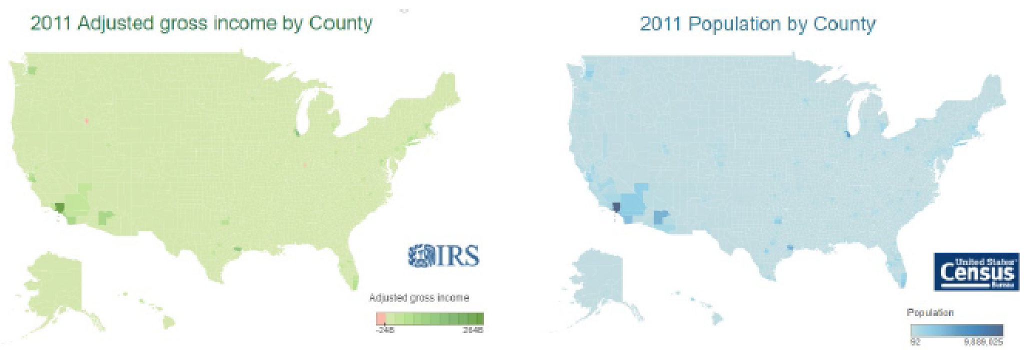
Dot Map

- Dot Maps also have some problems like choropleth maps



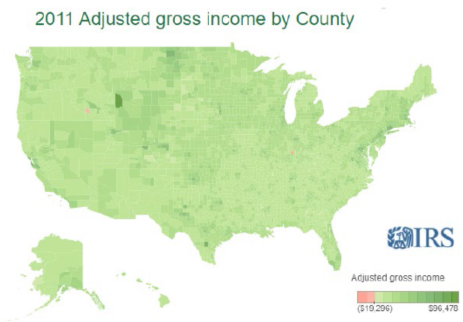
<u>You need to be careful when plotting **absolute vs. relative**</u>

- **Population density vs. Per region**



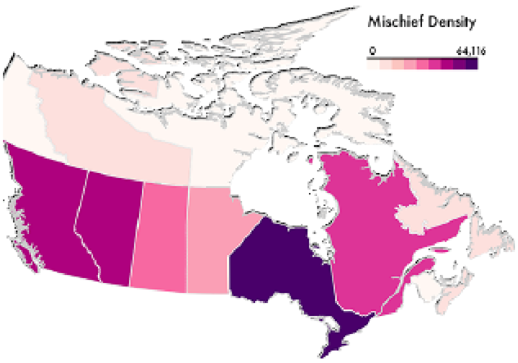
Absolute, with choropleth maps

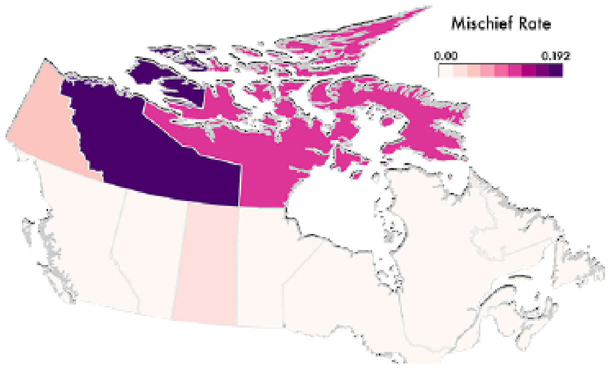
  - The income is higher for the bottom left are because the population is higher as well. So we cannot just look at the absolute values
  - Next, we look at relative values and get a totally different statistic:



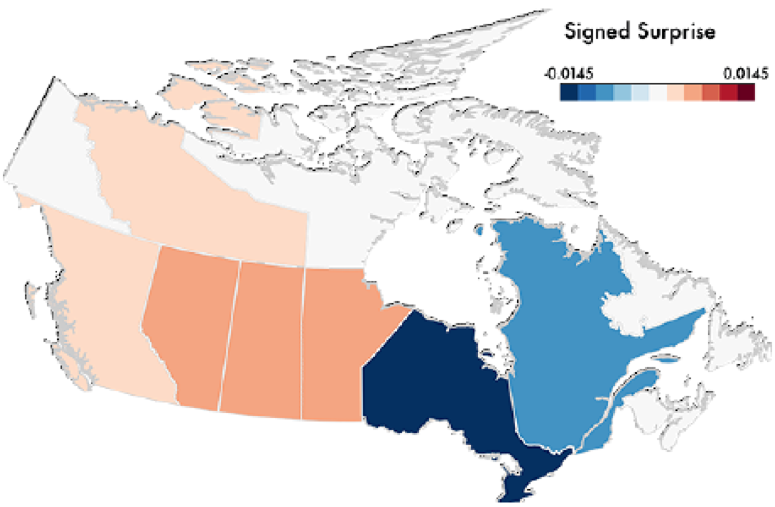
Relative

- **When populations are low, variations tend to be high**

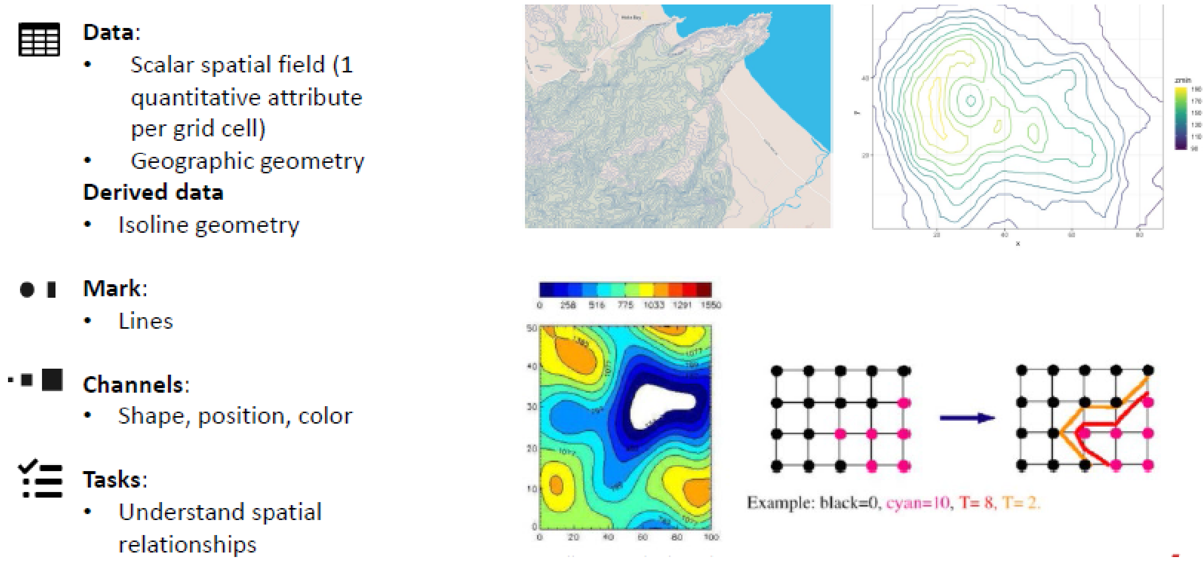




  - First one shows an absolute view for the crime in Canada
  - Second one shows a relative view
  - However a better approach here is by looking at the change in expectations:
    - The dark blue area shows that the crime here is actually less than expected (when population was not accounted for, crime was pretty high)



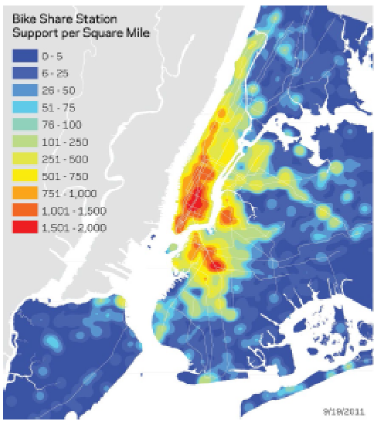
Topographic Map

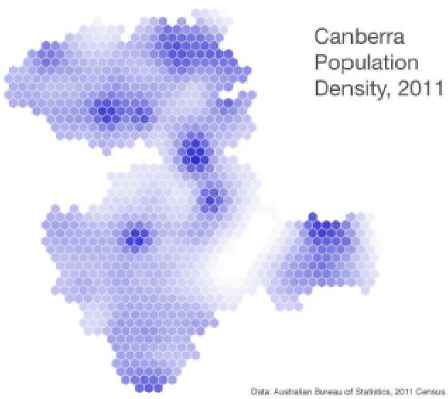


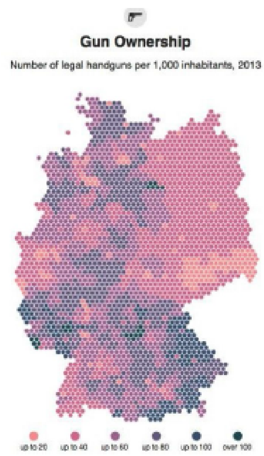
- To make an isoline for a value, you first mark that value (if it exists) in each link of a grid. And then connect all the markings to complete an isoline

Density Maps

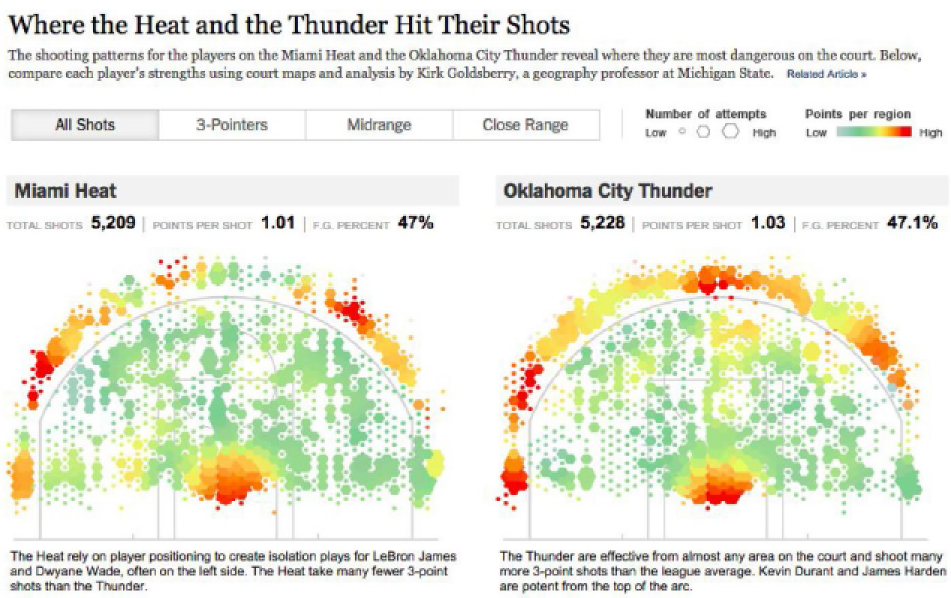
- Need a data transformation to turn discrete data into continuous data. Typically, using *density estimation mechanism (*e.g. KDE)
  - kind of a smoothing function to fill in for the areas for which values are not known







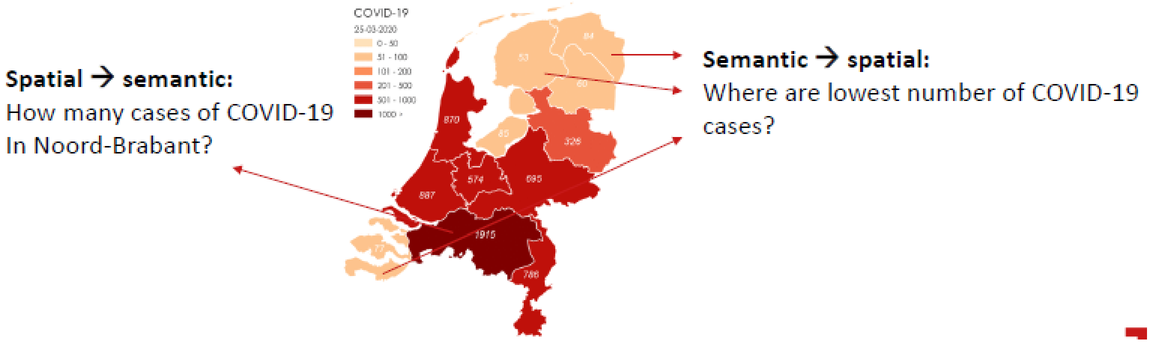
- An advantage of these is that they can be used not only with geographical data but also with other type of space
  - For example, using this basket ball data:



Advantage 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

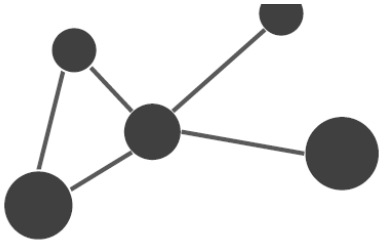


- The data types Networks and Trees can be categorized as Graphs

Network

- Describe relations between objects e.g. mail communication between two people or a transaction between two accounts

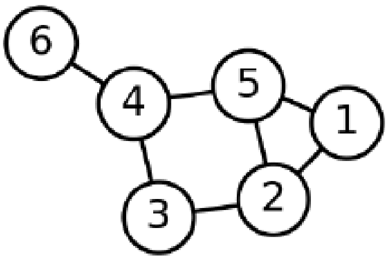
Graph – Network
Vertices – Nodes
Edges – Links



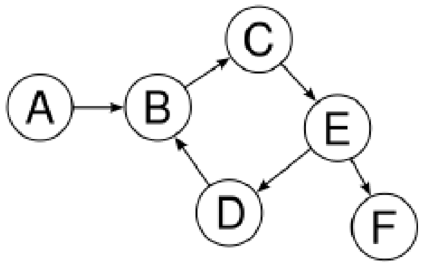
$G = (V, E), E \subseteq V^2$

V set of vertices

E set of edges



Undirected

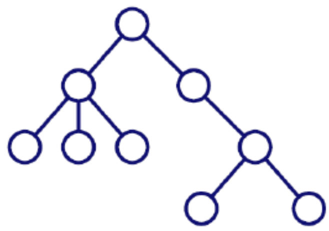


Directed

- In directed, the holds some sort of meaning, for example, an airplane's route

Tree

- a special case of a network
- graph without cycles and one root
    - Every other node has exactly one parent → **acyclic**
    - **Single root** ⇒ all other nodes are reachable from there
    - **E = V-1** (since the root node does not have a parent)
- Hierarchy = "rooted tree"



Network Types

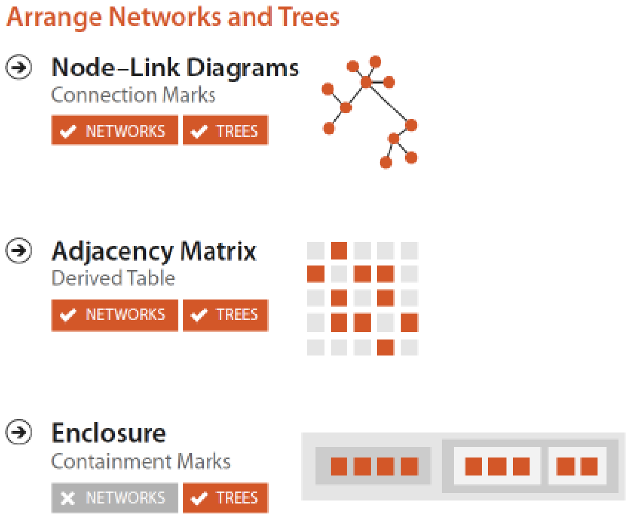
Static → we are interested in the structure

Multivariate Data → extra information

Dynamic → evolving/changing over time

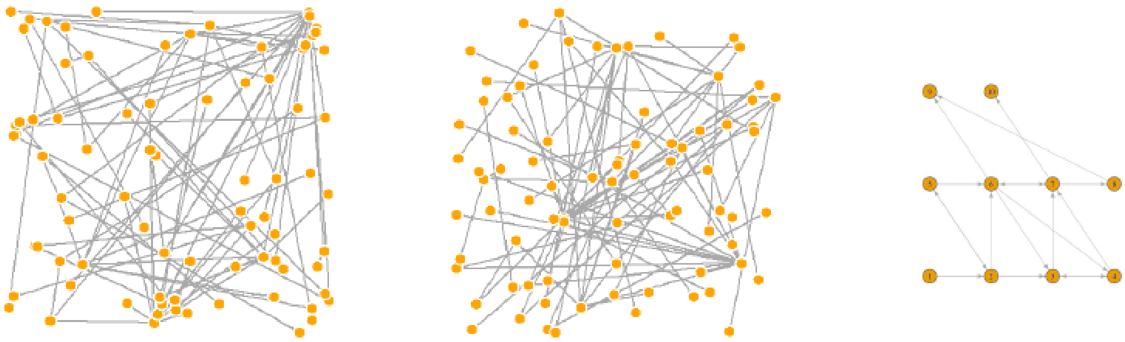
Static Networks

- There are different design choices for arrangement:

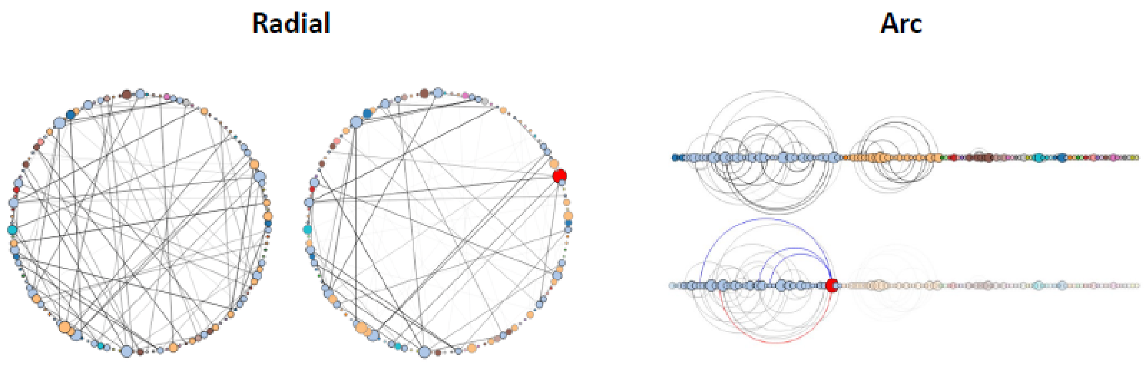


  - Enclosure can not be done for networks since a node may have multiple parents or it could be cyclic → no containment relation

- In the following example of a network, the structure is not quite visible. The third picture shows the arrangement on a grid but that also has more cons than pros



- A way to improve this by putting the nodes in a radial arrangement (ring), and if there is a relation in between two nodes, we connect them. The color and size of the nodes can be used to encode different attributes:
- Another way is to linearly arrange them on a horizontal scale and connect them using arc:



Linearize nodes: ordering plays a very important role

The **advantage of node-link diagrams** is that it is easy to understand by non-experts.

- Positioning of the nodes is called **layout** or **embedding**
- Compute layouts → graph drawing which is a field of its own (force directed methods)
- Criteria (readability and aesthetics):
    - 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
    - Sometimes all are not achievable at the same time. but this conflict can be addressed using **heuristics.** There isn't really an algorithm for this

Computing layouts is based on **force-directed algorithms,** as stated above

- Mechanical Laws
- Model edges (connections between nodes) as springs
    - Also nodes repel each other (to avoid overlapping)
- Numerically simulate until stable state is reached

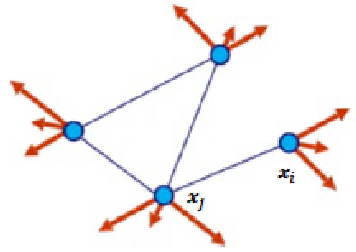
How do we do this? →

Vertex Force

- repelling force between vertices i and j
- prevents them from coming too close to each other
- For each node, we compute the repelling force with respect to another node:

Repulsion strength

$$g_{ij} = \frac{r_{ij}}{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

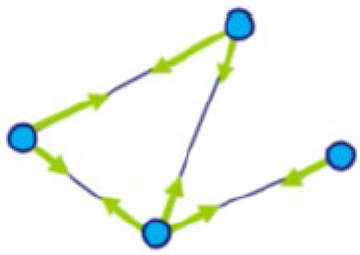


Edge Force

- spring forces on the edge
- attracts the connected nodes together
- prevents connected nodes from being too far away

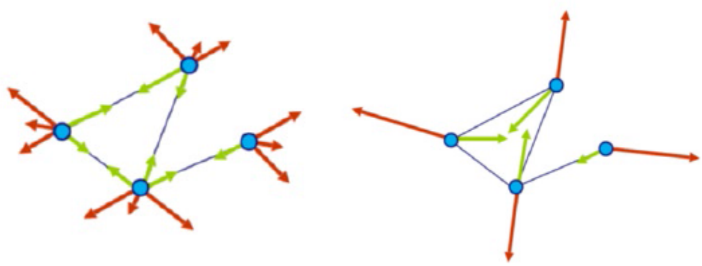
Spring tension　　Spring length

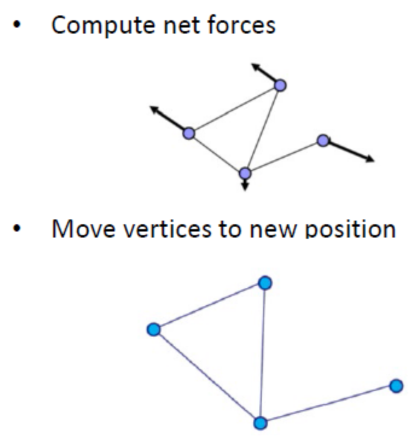
$$f_{ij} = k_{ij}\left(d(\boldsymbol{x}_i, \boldsymbol{x}_j) - s_{ij}\right)\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$



Forces in Action

- We compute repulsion and attraction forces for all nodes → resultant force





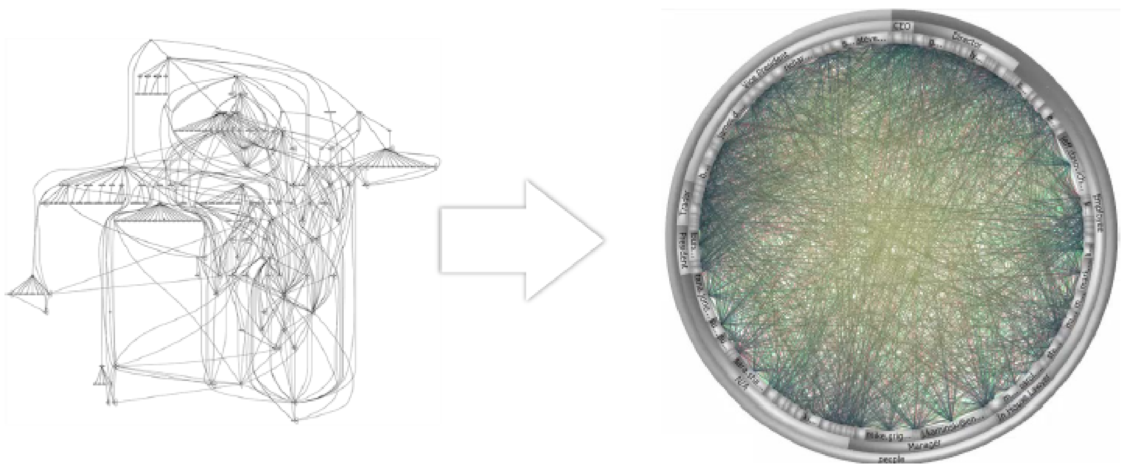
Termination occurs when

- fixed number of iterations
- total energy gets below some threshold
- local minimum reached
- Some user input

Tasks are to explore topology and locate paths

Large Network Visualization

- Traditional (force-directed) networks do not scale



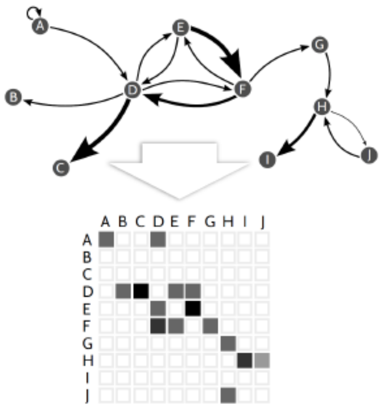
Network + Hierarchy = compound network   Hierarchical Edge Bundling[1]

You can interact with the data in Hierarchical Edge Bundling by selecting bundles and groups that you are interested in

⇒

Visual Adjacency Matrix

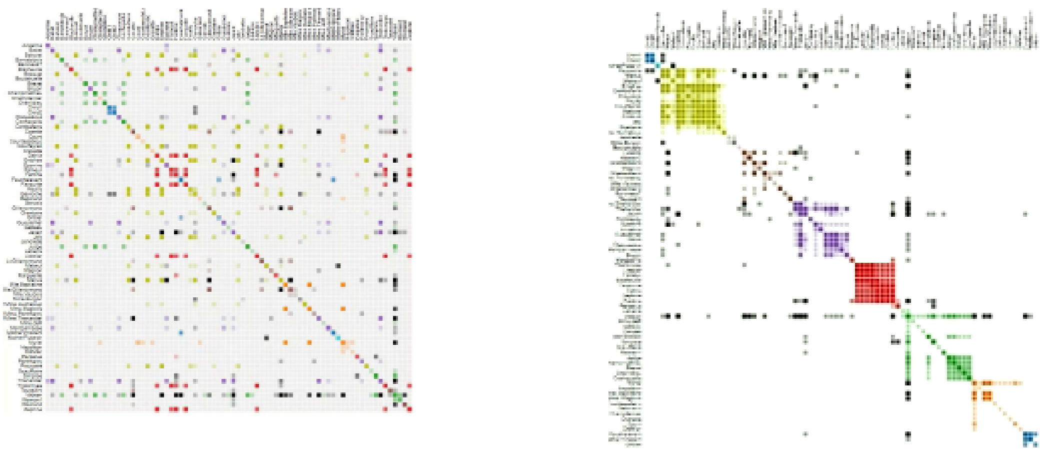
The matrix will be symmetrical if the graph is not directed. If the graph is directed, the matrix will not be symmetrical



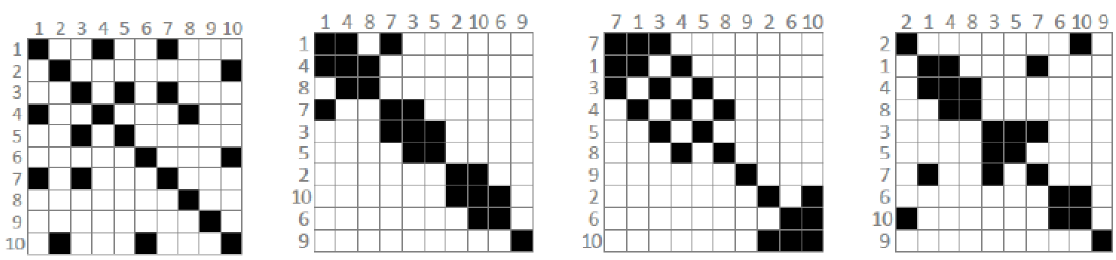
- more scalable than a node-link diagram
  - since there are no overlapping nodes or edges because there are separate positions for edges

- $M_{i,j}$: existance/weight of edge i → j
- Maximize edge visibility, no crossing edges
- Cons: difficult to see communities, paths, motifs
- Note: sorting extremely important[1]

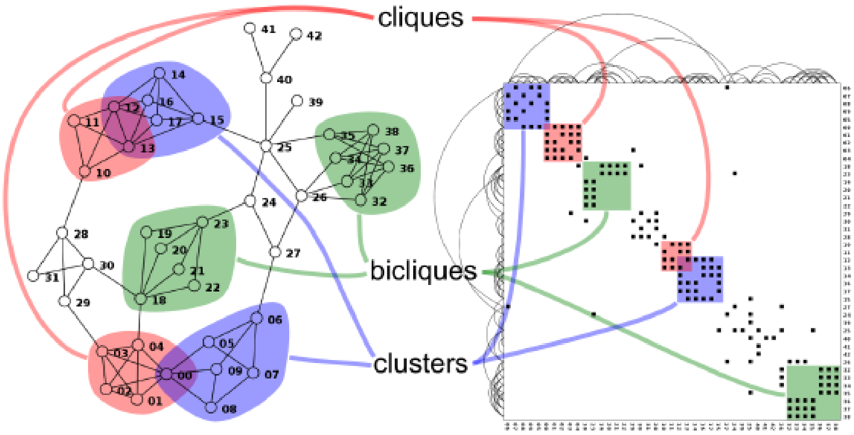
- we can see communities and structure when we use ordering (1 → 2):
  - naive ordering used here



- Choose an ordering that makes sense and suits your task. Different results due to difference in ordering used:



Relation between node-link diagrams and visual adjacency matrix:

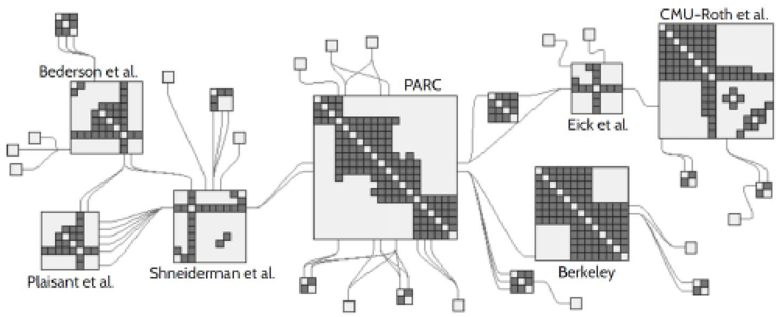


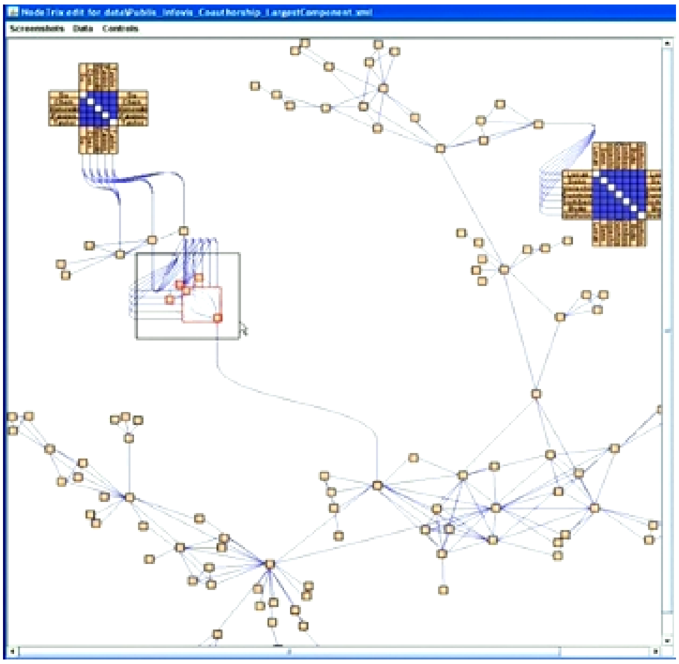
cliques are group of nodes that are communicating everywhere; fully connected
bicliques are two groups of nodes communicating

Hybrid Techniques

NodeTrix

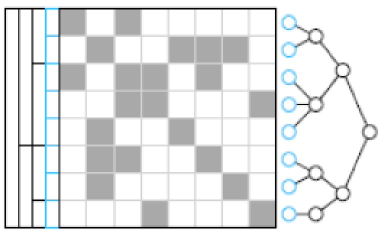
- put all the dense areas in the visual adjacency matrix
- put all outliers with low number of connections as node-link diagram
  - this avoids edge crossings

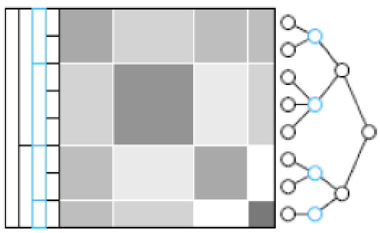


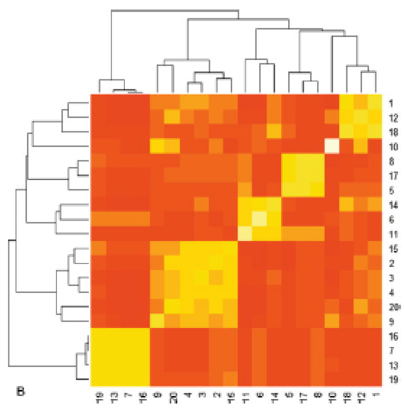


Compound Graph

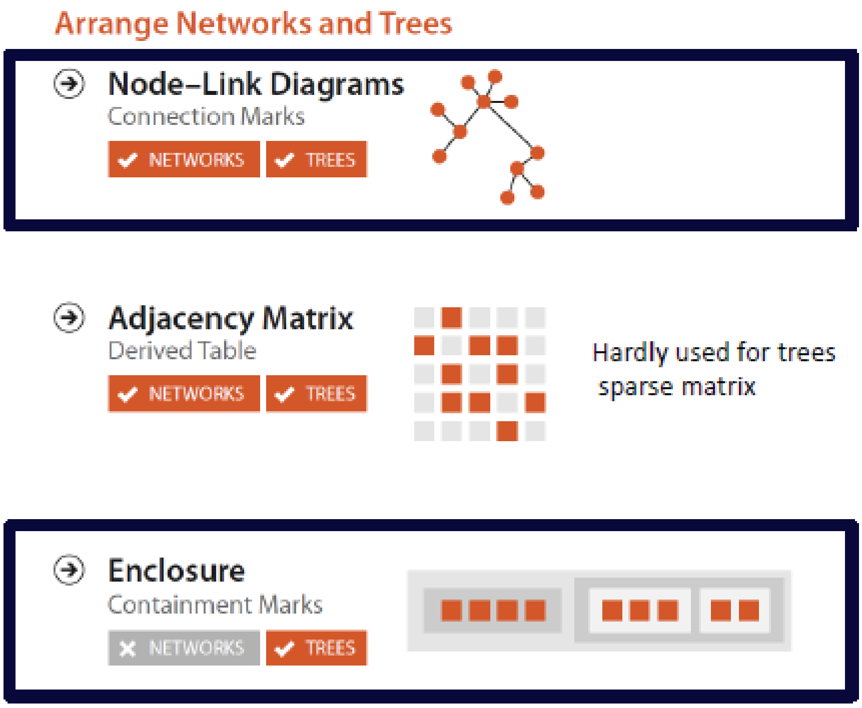
- network with a tree structure on top
- can also combine that with visual adjacency matrix



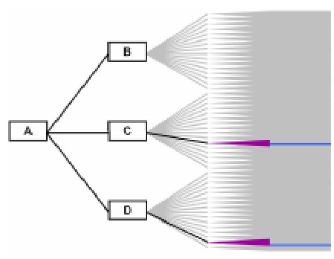




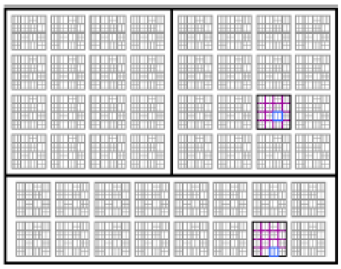
Trees



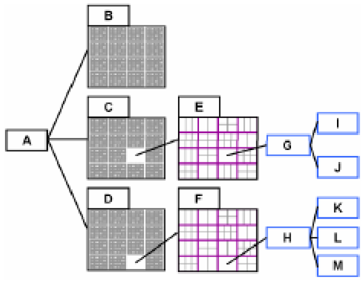
Visualization of Static Trees



Node-link diagram



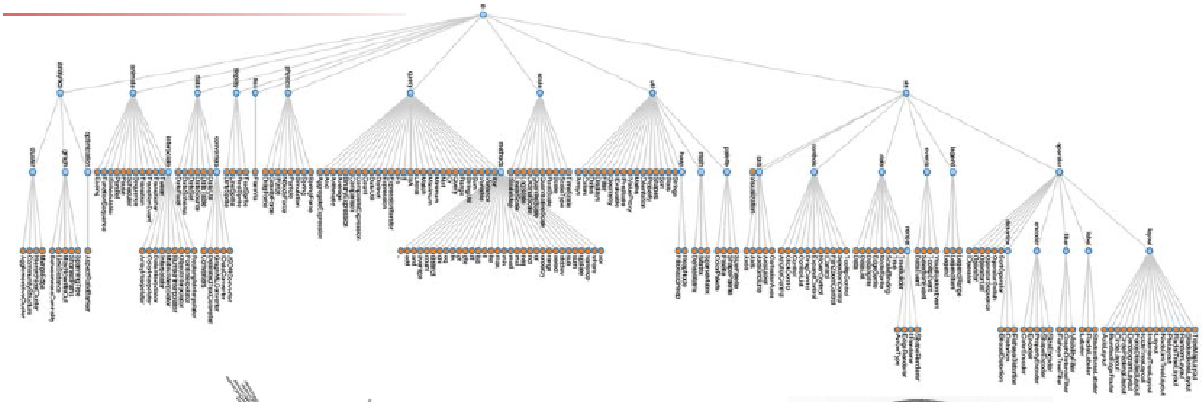
Enclosure: Space-filling diagram

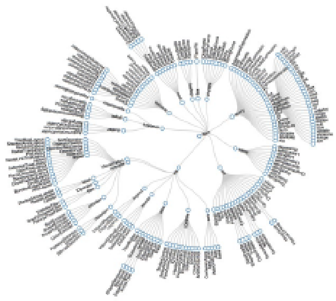


Combined representations

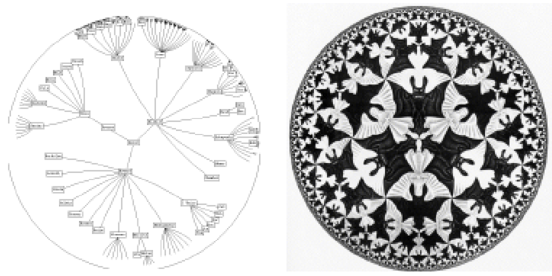
Node-Link Trees

- Usually start from the root node in the middle and work your way out





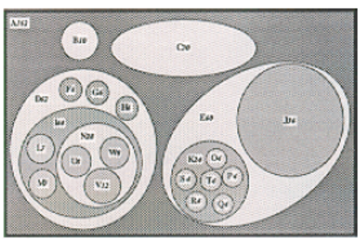
Radial Layouts



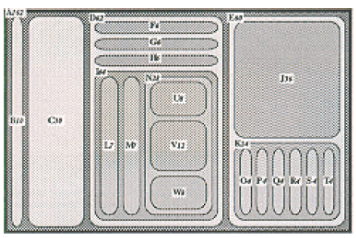
Hyperbolic Layouts

Enclosure

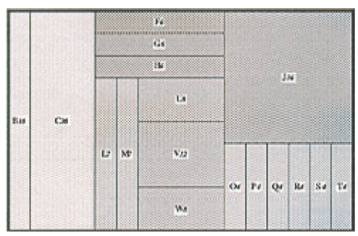
- Treemap ⇒ space-filling techniques



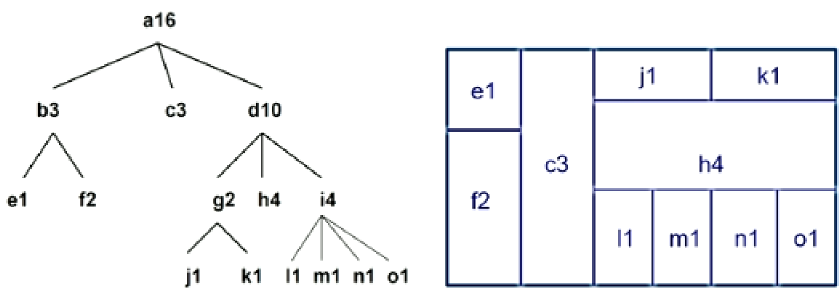
Venn Diagram



Nested Treemap

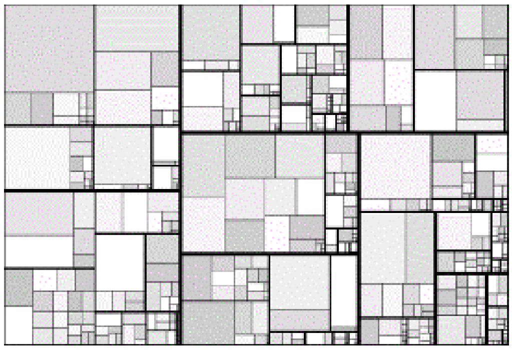


Treemap



- With vertical dissections, we get a lot of long-thin rectangles
- Squarified layout generates nicer rectangles

Slice-and-dice layout

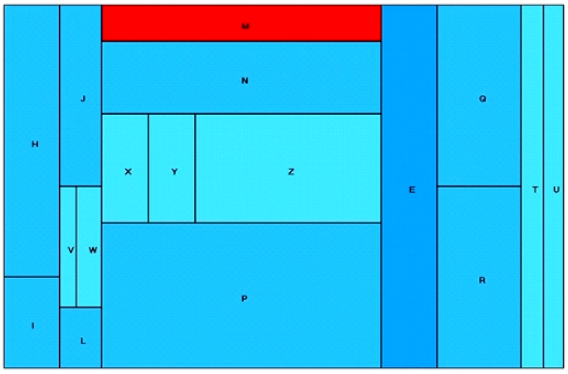


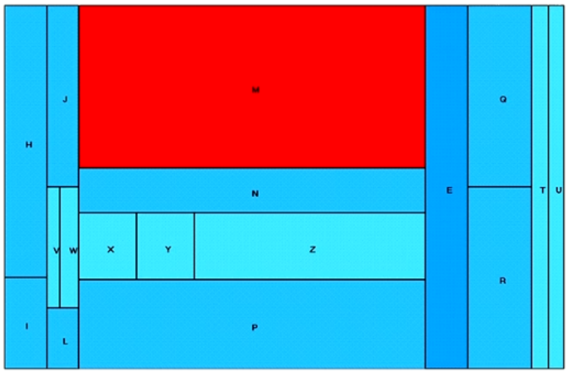
Squarified layout

Treemap + Time Series

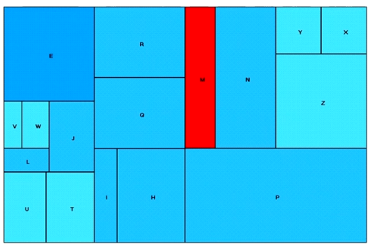
Node changing (not squarified)

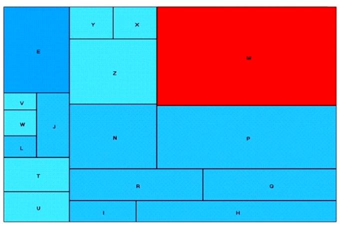
- might become hard to visualize and interpret

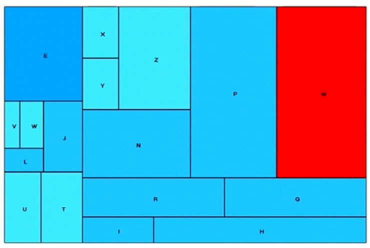




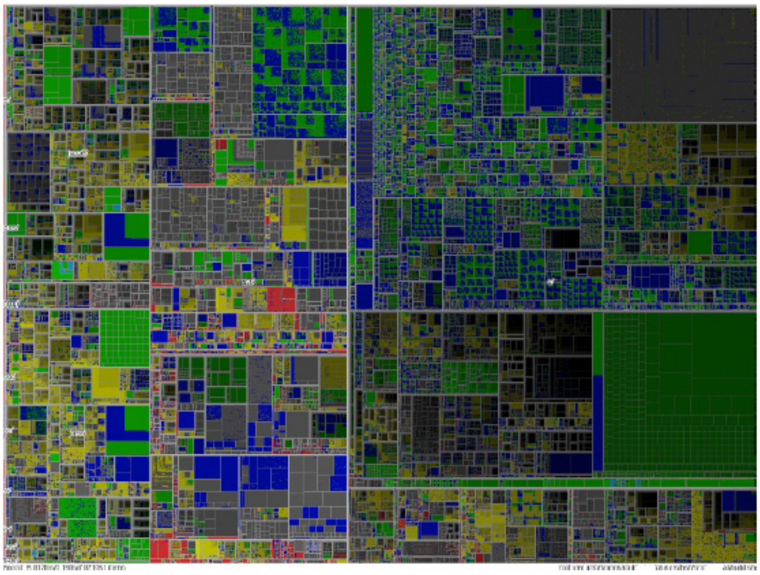
With squarified layout, blocks can move around:







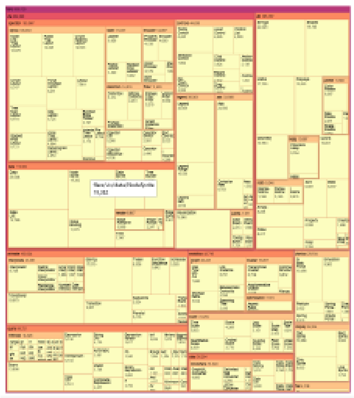
Treemaps scale pretty well

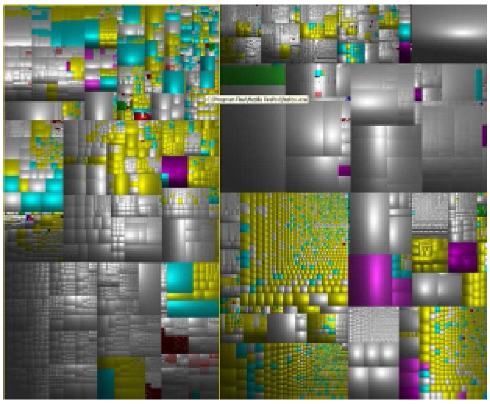


Fekete, J.-D., Plaisant, C. Interactive Information Visualization of a Million Items, Proceedings of IEEE Symposium on Information Visualization 2002 (InfoVis 2002), Boston, USA, Octobre 2002.

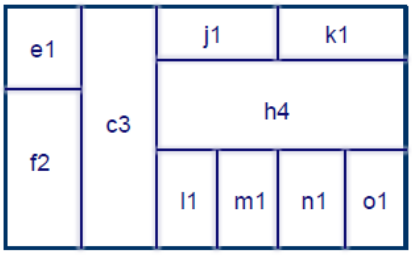
Treemap Applications

- computer systems, stocks, files in a file system, etc.



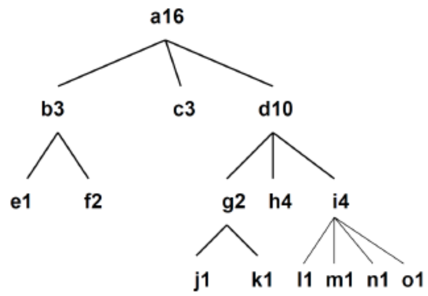


Node-Link vs. Treemap



Treemap – Enclosure techniques

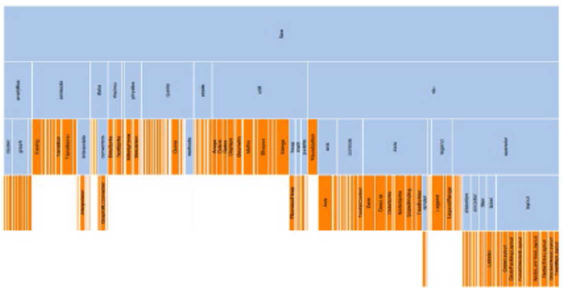
- Scalable: very good usage of available space – show attributes
- Difficulty in distinction of the hierarchy (implicit)



Node-link diagram

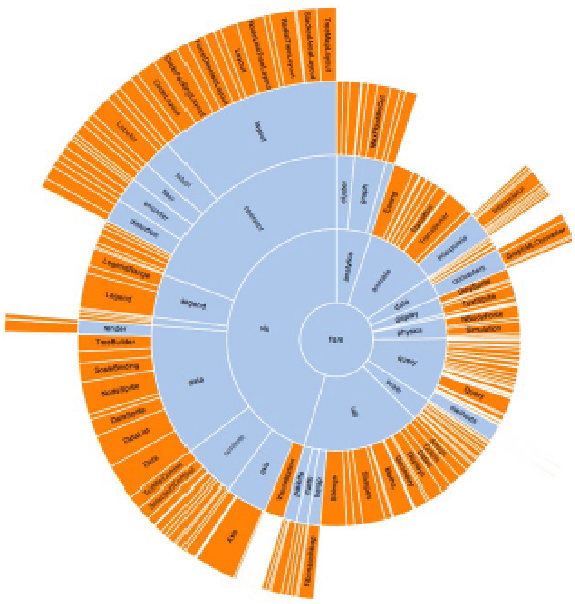
- Intuitive
- Good at exposing structure of information
- A lot of empty space

Examples of 2 closely related visualizations →



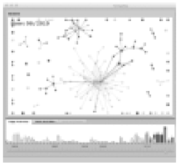
Icicle plot

- no overlapping parent child — attributes displayed easier
- Not as dense as treemaps



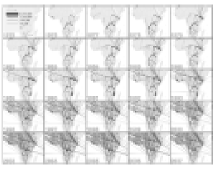
Sunburst diagram

Time Series



Animation

Map time to time



Small Multiples

Split time in intervals

Animation

Pros:
- Simple to implement
- Easy to spot big changes
- Applicable to all methods for which a single timestep can be visualized.

Cons:
- Need to focus on many moving or changing items simultaneously
- Keep track of (multiple) changes over longer time periods.
- Change blindness

Small Multiples

Pros:
- Independent of the visualization method used.
- Eyes beat memory

Cons:
- Decide on the number of multiples to use.
- Limit on the number of multiples.
- Multiples might be far apart → difficult to spot patterns

Encoding time series data

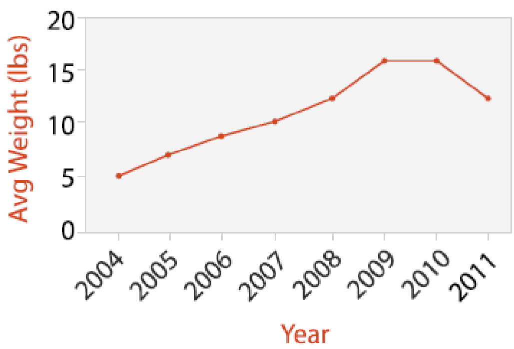
Line chart

Data:
- 2 Quantitative attributes
- One key, one value

Mark:
- 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 attributes dozens to hundreds

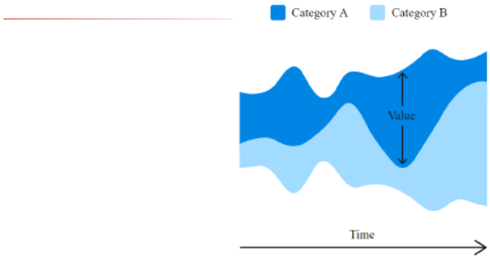
Streamgraph

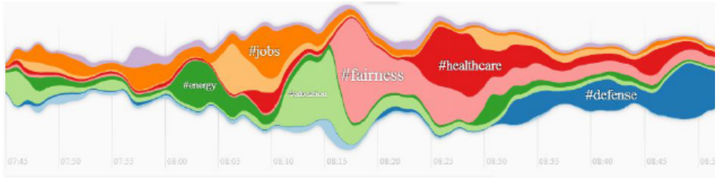
Data:
- 1 Categorical attribute (names)
- 1 Ordered key attribute (time)
- 1 Quantitative value attribute (counts)

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



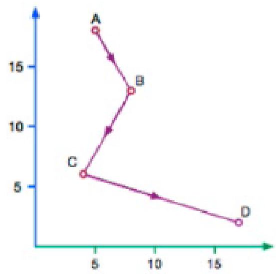


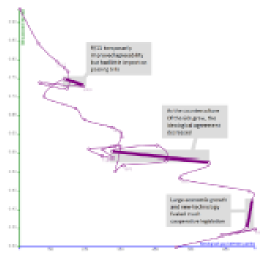
Connected Scatterplot

Scatterplot with line connection marks
- Popular in journalism
- Horizontal + vert axes: value attributes
- Line connection marks: temporal order

Empirical study
- Engaging, but correlation unclear

The Connected Scatterplot for Presenting Paired Time Series. Haroz, Kosara and Franconeri. IEEE TVCG 22(9):2174-86, 2016.]





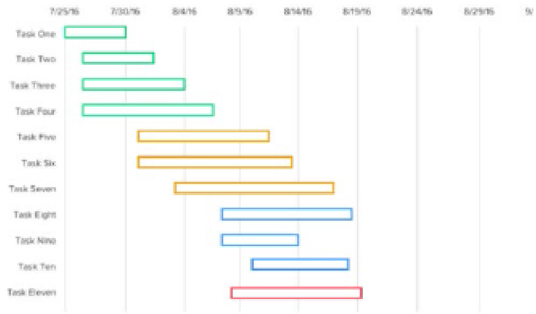
Gantt Chart

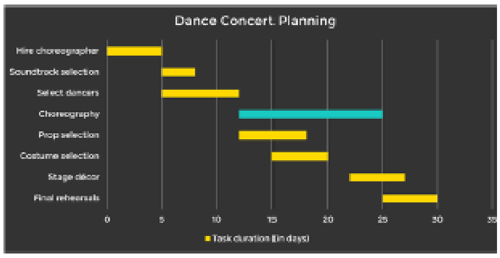
Data:
- 1 Categorical attribute
- 2 Quantitative attributes
- One key, two (related) values

Mark:
- 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

Lecture 10 - Validation

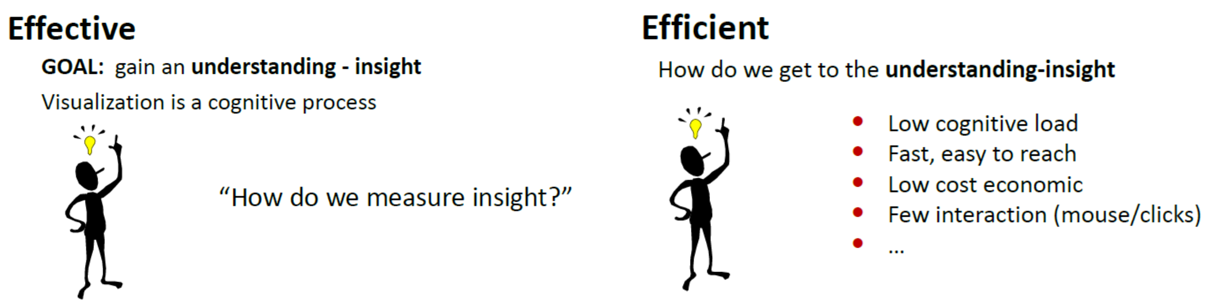
Date	@January 12, 2022
Type	Extra Lecture
Completed	☑
Book	☐
Notes	Chapter 4

How do we know our visualization design/tool is ‘good’?

- Subjective opinions of the developers are not relevant in the way to validation

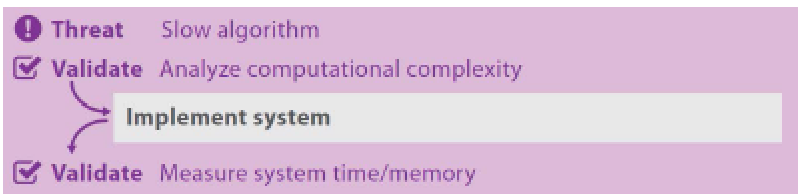
Effective — do the right thing

Efficient — do the thing in the right way (e.g., fast, using fewer resources)



(Design) 4 Nested Levels — Validation Perspective

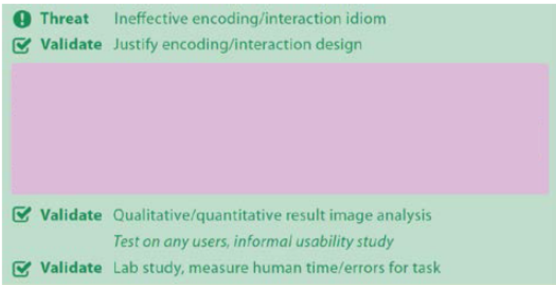
Algorithm Validation (lowest level)



- Complexity can be analyzed before the implementation and the memory/time resources and quality of visual results afterwards

Visual Encoding / Interaction Idiom

Threat → idiom is not effective or efficient for the data/tasks



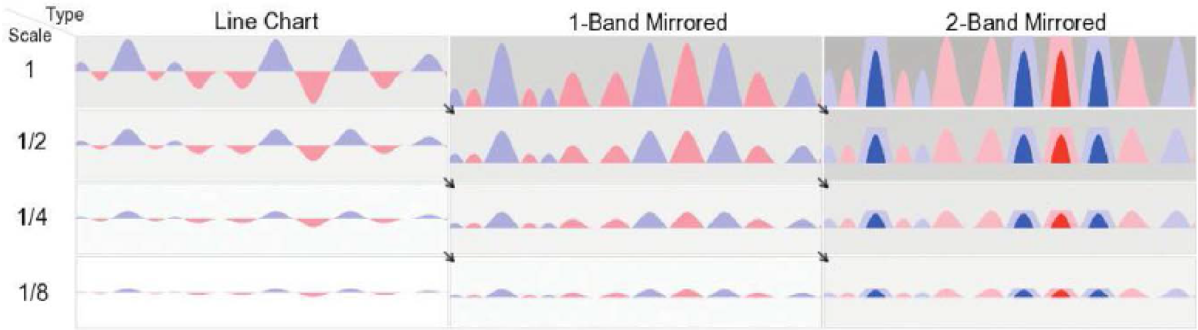
Validation → justification of choices

- Show that the idioms fulfill the guidelines and principles known
- Tasks/data are not domain dependent ⇒ back-up from literature

Lab Study / User study →

- Abstract data/tasks so **any** user (around 50)
- Controlled experiments – limit variables and factors
    - measures very specific aspects
- Specific idiom design
- **Quantitative**: measure times to task, errors made, mouse movements, clicks, eye movement
- Enough users to allow statistical analysis such as significance analysis (e.g., p-values)
- **Qualitative**: preferences for one or another idiom
- Lab studies allow for **stronger conclusions**
- **Limited applicability**
    - Well defined controllable tasks: perception or cognitive tasks
    - Thinking, deciding, and exploring cannot be tested

<u>Example: Encoding Validation</u>



- 2 alternatives proposed for Line Charts → saves space etc.,

- **Quality metrics** specific to some idioms
  - Example — Layout metrics like node-link diagram
- **Quality metrics** relation to human judgement is not always clear
- Comparison of multiple potential encodings

Task/Data Abstraction Level

Threat → misunderstanding the needs; solving the wrong problem

Validation → Measure adoption

- rate of acceptance of the tool — long term
- Well designed is not always what is adopted

Is the user able to solve the domain tasks for which the visualization was intended?

- Usually few experts / target users (no statistical power here)

Field study →

- How people act in the real-world setting using the tool
- Ask **domain task** and see whether it can be solved (effective-efficient)
- Usually qualitative evaluation, semi-structured interviews

Conducting Heuristic Evaluation

Step 1 Be prepared (respect the time of your users)

- *Develop descriptions of a few typical users and* ***tasks*** (specific rather than general)
- *Determine your objectives and choose a set of heuristics*
- *Select a set of experts (around 5)*
  - independent of the development team
  - Including end users, also usability experts give different perspective.
- Establish a good rapport with the evaluators so they will be comfortable talking with you.

Step 2: Conduct the evaluations

- *Have experts work independently (observe let them talk out loud)*
- *Don't place too much emphasis on the heuristics. Qualitative might give you more information.*
- *Remain neutral; don't defend your visualization tool*
- *Take copious notes (taking notes should not disturb, record session)*

Step 3: Analyze the results

Review after the evaluations while memory still clear

Compare responses of the evaluators (common themes and areas of disagreement).

Dangers

- few users and they are involved in the design

- biased responses to favor/please you
- HCI developed multiple sound solutions: **focus groups, field studies,** and **expert reviews**
- Usability difference between GUI (difficult to exit a dialog box) or **visualization** (how well the visualization supports exploration)

Domain — Task/Data Abstraction Level

Basic (minimum) with few user involvement

- Qualitative discussion of result images

Case study:

- Test on target users collect anecdotal evidence of utility:
    - insights / discovery found
    - hypothesis confirmed
    - hypothesis generated
- Test yourself, instead of target users, and identify non-trivial discoveries

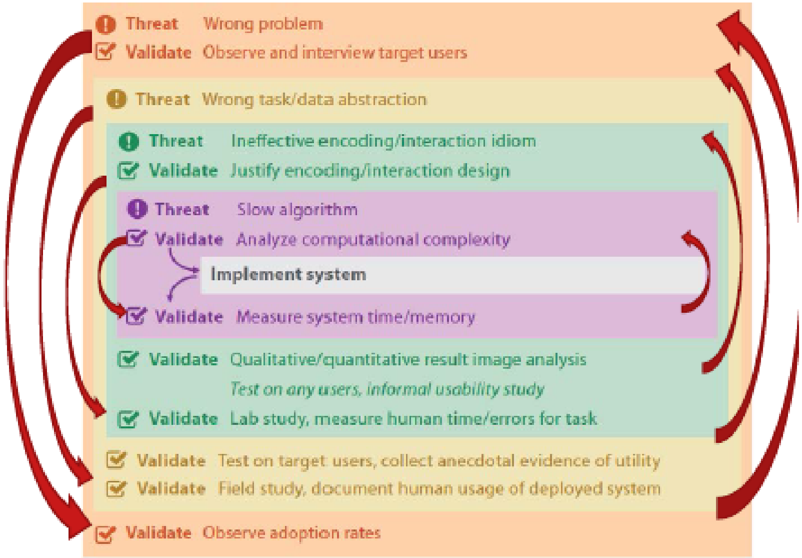
A case/scenario where non-trivial discoveries can be achieved **thanks to the visualization**

Evaluation / Validation

- Not a linear process — rather iterative
- Not necessarily evaluate all levels
- Vis evaluation is in development and

evolving constantly

- finding methods to result in the best possible evaluations



Expected Validation for the final project

- Justification of visual encodings
- Qualitative discussion of result images/videos
- Case studies