

7CCMSDV

Introduction to Data Visualisation

If you spot anything wrong, please let me know!

discord: @insertish

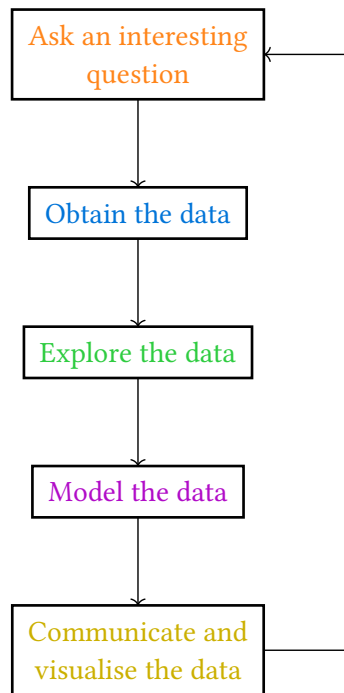
izzy · insrt.uk

Contents

1. Introduction to Visualisation and Exploratory Data Analysis	3
2. Data Types and Visual Encodings	4
2.1. Data Abstraction	4
2.2. Data Types vs. Data Semantics	4
2.3. Data Attributes	7
2.4. Attribute Semantics	8
3. Perception - Visual Encodings & Channels	9
3.1. Visual Encodings	9
3.2. Visual Channels	9
3.3. Perception/Cognition	11
3.4. Perceptual Effectiveness	12
3.4.1. Steven's Power Law	13
3.4.2. Jacques Bertin: Marks as basic units	13
3.4.2.1. Expanded work by John D. Mackinlay	14
3.4.3. Cleveland & McGill: Ranking of graphical encodings	15
3.4.4. Channel Effectiveness	16
3.4.4.1. Discriminability	16
3.4.4.2. Separability	17
4. Visual Perception & Colour	18
4.1. Gestalt	20
4.2. Colour	22
4.3. Cornsweet Illusion	24
4.4. Contrast Sensitivity	25
4.5. Relativity of Perception / Relative Differences	26
4.6. Axis of Alignment	26
4.7. Simultaneous Contrast	27
4.8. Crispning Effect	27
4.9. Interaction of Colour	28
4.10. Colour	30
4.10.1. Colour Spaces	30
4.10.2. Colour Guidance	31
4.11. Colour Maps	33
4.12. Colour Deficiency	34
4.13. Colour Tools	34
5. Spatial Data - Visualisation	35
5.1. Maps	36

5.1.1.	Types of Overlays	36
5.2.	Choropleth Maps	41
5.3.	Global Neighbourhood	44
5.3.1.	Mercator (Conformal Projection)	44
5.3.2.	Latitude/Longitude (Conformal Projection)	45
5.3.3.	Azimuthal Projections	46
5.3.3.1.	Equidistant (Azimuthal Projection)	46
5.3.3.2.	Lambert equal-area (Azimuthal Projection)	47
5.3.3.3.	Winkel Tripel (Azimuthal Projection)	48
5.3.4.	Tissot Indicatrix	48
5.3.5.	Conic Projections	48
5.3.5.1.	Albers - Equal-Area (Conic Projection)	49
5.3.6.	Misc. on Projections	49
5.3.7.	Unfolding the Earth	50
5.4.	Contour (Isopleth) Maps	51
5.4.1.	Line Integral Convolution (LIC)	52
5.5.	Cartograms	53
6.	Spatial Data - Cluster Analysis	56
6.1.	Inferential Statistics	57
6.2.	Hotspot Analysis	58
7.	Spatial and Temporal Data - Visualisation	62
7.1.	Temporal Data Model	62
7.2.	Temporal Scale	63
7.3.	Visualizing Temporal Phenomena	63
7.3.1.	Changing Axis: Radial Layouts	65
7.4.	Visualizing Spatio-Temporal Phenomena	71
7.5.	Interaction	76
8.	Hierarchical and Network Data	84
8.1.	Small World Networks	92
8.2.	Affiliation Networks	94
8.3.	Preferential Attachment	94
9.	Hierarchical Data Visualisation	95
9.1.	Graph Representations	96
9.1.1.	Explicit (Node-Link) Representation	96
9.1.1.1.	Force-directed Layouts	98
9.1.1.2.	Other Issues	100
9.1.2.	Styled Restricted Layouts	101
9.1.2.1.	Explicit Graph Representation: Trees	106
9.1.3.	Matrix Representation	108
9.1.4.	Implicit Representation	110
10.	Dimensionality Reduction Techniques	113
10.1.	Pixel-Oriented Techniques	113
10.2.	Geometric Techniques	116
10.2.1.	Scatterplot Matrices	116
10.2.2.	Space Re-organization: Changing Axis	116
10.3.	Icon-/Glyph-based Techniques	119
10.4.	Dimensionality Reduction	121
10.4.1.	Principal Component Analysis	121
10.4.2.	t-SNE (T Distributed Stochastic Neighbour Embedding)	124

1. Introduction to Visualisation and Exploratory Data Analysis



Important

Never finished week 1, nothing else here :)

2. Data Types and Visual Encodings

2.1. Data Abstraction

Data abstraction is a powerful method to start reasoning about what may be an appropriate visual representation for a given dataset. We provide a language to describe data in a way that is meaningful and useful, e.g. is it a table, network, unstructured? Is it spatial, temporal, spatio-temporal? Data is typically described with a domain language, but in order to search for a suitable visual representation we must translate it into a more abstract structure we know how to encode (narrowing down the design space).

Suppose the following datasets:

- Friendship in Facebook, interactions between proteins, connection between terrorists

network data

- Animal movement, election results in counties, simulation of air turbulence

spatial data

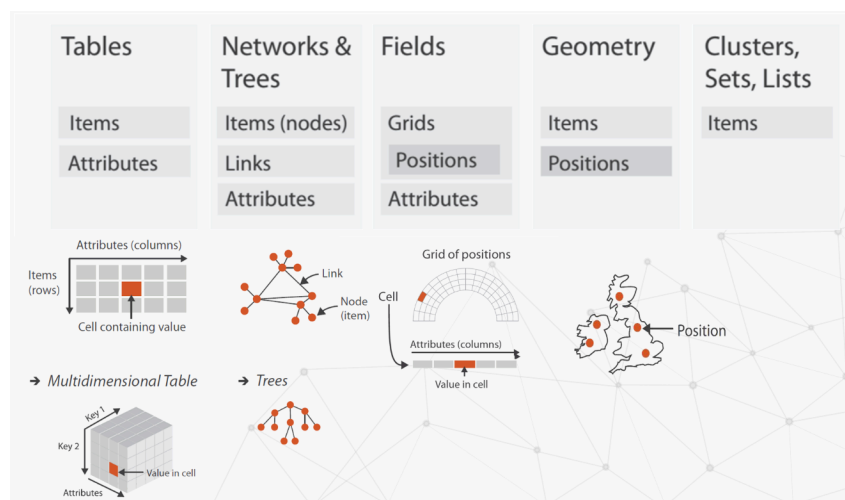
2.2. Data Types vs. Data Semantics

Every dataset can be described as a collection of 5 data types:

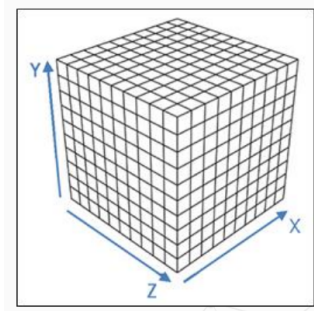
- Items: individual entity, discrete, e.g. row in a table
- Attributes: specific property, can be measured, e.g. temperature
- Links: relationship between items, e.g. a network
- Positions: spatial data providing a location in space, 2D, 3D, e.g. latitude/longitude
- Grids: strategy to sample continuous data, in function of topological or geometrical relations between cells

Basic dataset types:

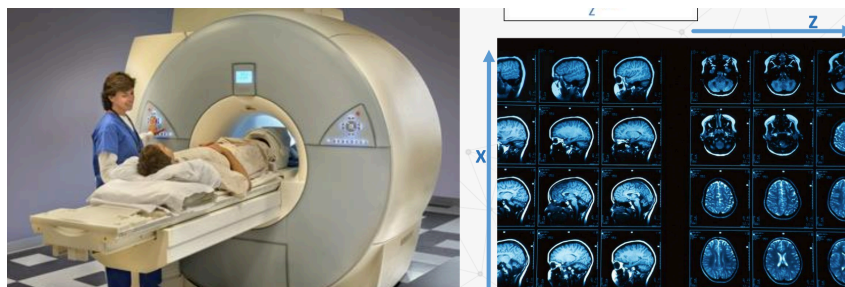
- Tables: items are cells identified uniquely by a vector key; multi-dim
- Networks: items identified as nodes and connected to each other by links
- Fields: fields are best sampled in a grid; grid provides sampling of arbitrary space
- Geometry: group items by position in space; e.g. tessellations or GIS locations
- Sets: where the items characteristics and attributes implicitly define organisation



A **grid** can be multi-dimensional.



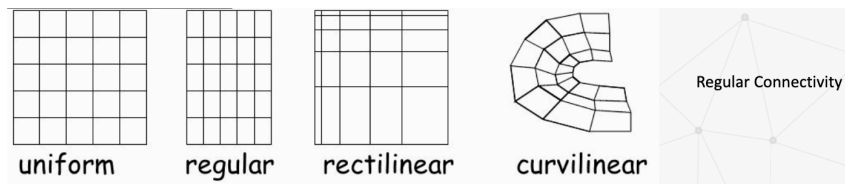
An example is an MRI scan which generates field data that can be embedded in a 3D space, it can then be sliced to obtain 2D projections.



Different types of grid can provide different types of spatial sampling.

A **structured grid** is identified by regular connectivity.

The possible element choices are quadrilateral (2D), hexahedra (3D). This model is highly space efficient as neighbourhood relationships are defined by storage arrangement. Other advantages include better convergence and higher resolution.



An **unstructured grid** is identified by irregular connectivity. It cannot easily be expressed as two-/three-dimension array in computer memory. This allows for any possible element that a solver might be able to use.

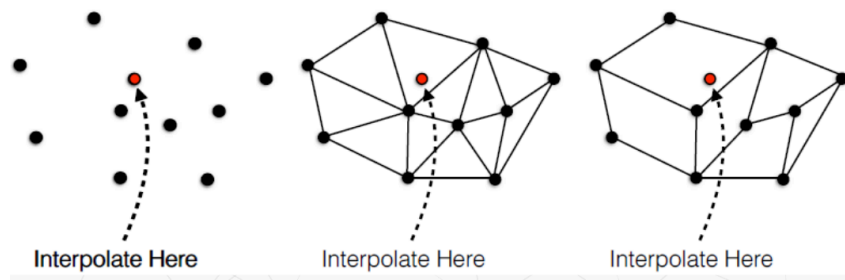
Often highly space inefficient. Grid employs triangles in 2D and tetrahedra in 3D.



A **hybrid grid** contains a mixture of structure and unstructured portions. Integrating structured and unstructured meshes in an efficient manner.

Grid choices impact how continuous data is interpreted, two key considerations are:

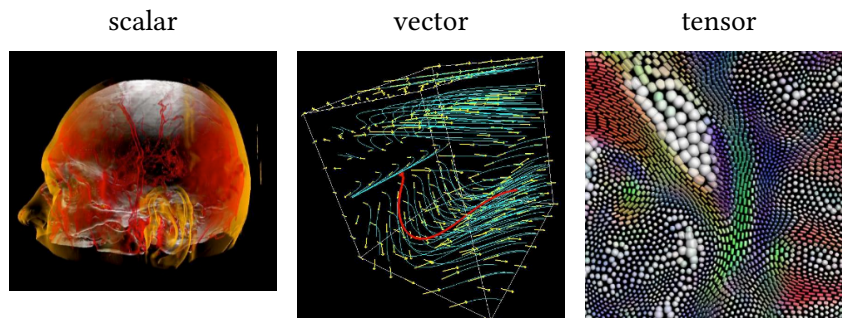
- sampling: choice of where attributes are measured
- interpolation: how to model attributes embedded in space



A **field** can be:

- scalar: univariate with a single value attribute at each point in space
- vector: vector field is multi-variate with a list of multiple attribute values at each point
geometrically, each point in the field has a direction and magnitude (encoded with an arrow)
- tensor: tensor field has an array of attributes at each point representing a more complex multi-variate structure than the list of numbers in a vector; geometrically arrows are not enough, more complex shapes are used

e.g. stress; represented by 9 variables which represent forces acting in three orthogonal directions



2.3. Data Attributes

Data attributes can be categorised:

- Categorical: no implicit ordering
- Ordered: implicit ordering exists
 - Ordinal: non quantitative but can be orderable
 - Quantitative: you can also perform operations

Ordering can be sequential (0 to something), diverging (diverging in both directions from a zero point in the middle of a range), or just a loop onto itself rather than increasing.

Some attributes may have an internal hierarchical structure:

- e.g. counties, states, animal families, etc

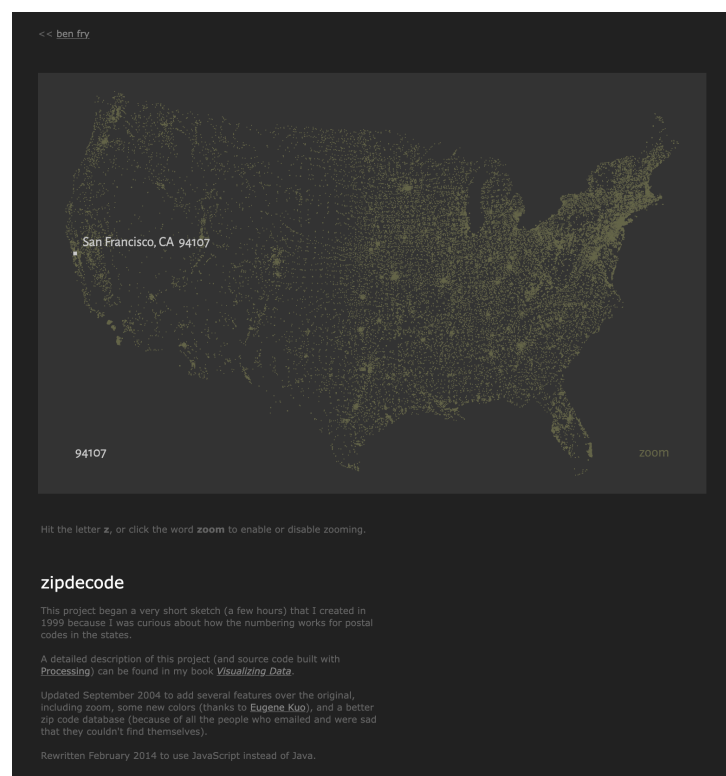
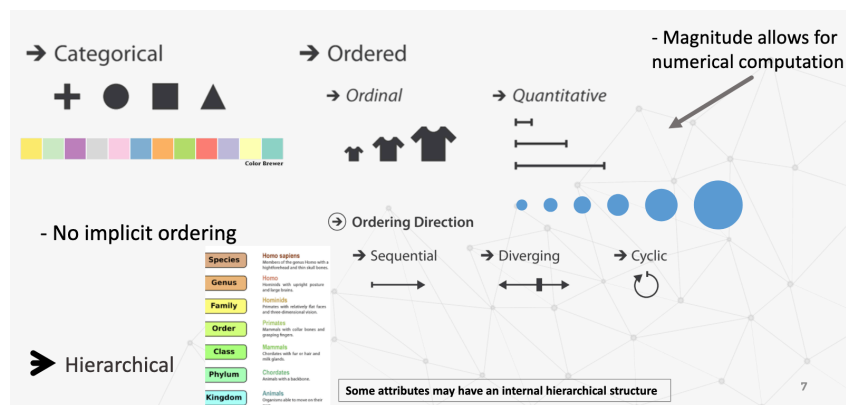


Figure 1: ben fry's zipcode decoder¹ hierarchical data structure

¹<http://benfry.com/zipcode>

2.4. Attribute Semantics

A **semantic** is 'of or relating to meaning'.

- **Spatial semantics**: when attributes describes some spatial characteristics (often geographical)
- **Temporal semantics**: when attributes describe something related to time
- **Continuous vs. Discrete**: influences different questions esp. when dealing with exceptions such as spatial&continuous vs. non-spatial&discrete.

A dataset often needs to be transformed beyond its original state to create a visual encoding that can solve the desired problem, **derived attributes** are computed from the originals (but extend). It could be a simple change of type (e.g. temperature to floating point/hot-cold), acquiring additional data (city name to long-lat), or complex transformation (e.g. through arithmetic).

3. Perception - Visual Encodings & Channels

3.1. Visual Encodings

A **visual encoding** is the way in which data is mapped to visual structures; typically mapping data items to visual marks and data attributes to visual channels in a principled way.

Every visualisation can be described as a set of mappings:

- from data items to visual marks
- from data attributes to visual channels

3.2. Visual Channels

A **mark** is a basic virtual object/unit that represents data visually.

Visual channels are visual variables we can use to represent characteristics of these objects.

Example of visualisation mappings:



Data Items: Sales

Data Attributes: Price, Profit, Product Type

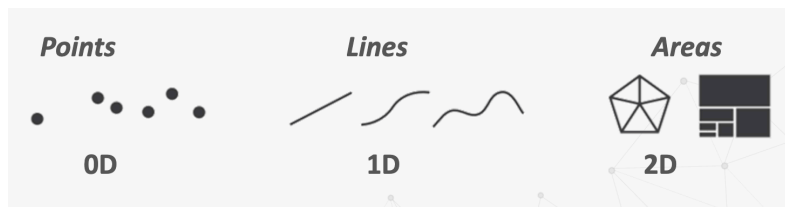
Visual Mark: Dot

Visual Channel: x-position, y-position, colour

Encoding Rules:

- sale = dot
- product price and profit = x-/y-positions
- product type = colour

Marks are a graphical element in an image and are classified according to their spatial dimensions.



Parameters control the appearance of marks.

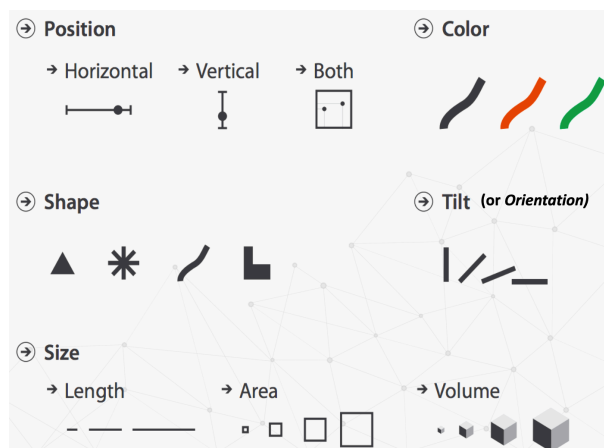




Figure 2: Example of interesting marks from Project Ukko²

Further examples of using marks and channels:

1 CATEGORICAL
1 QUANTITATIVE



Bar chart encodes 2 attributes:

- line with vertical spatial position channel for quantitative attribute
- horizontal spatial channel for categorical attribute

2 QUANTITATIVE



Scatter-plot encodes 1 quantitative attributes:

- spatial position

1 CATEGORICAL
2 QUANTITATIVE



Scatter-plot encodes 2 quantitative attributes:

- spatial position
- additional categorical attribute added using hue channel

1 CATEGORICAL
3 QUANTITATIVE



Scatter-plot encodes 3 quantitative attributes:

- spatial position
- additional categorical attribute added using hue channel
- size to add further quantitative attribute

Channels can be combined to enhance the same information (redundancy) but use of more channels means our design space becomes smaller.

²<http://www.project-ukko.net/map.html>

3.3. Perception/Cognition

The human-perceptual system has 2 kinds of sensory modalities:

- what-where (metathetic) – **identity channel**: information about what, who, where something is
- how much (prothetic) – **magnitude channel**: information about how much

Marks can be items/nodes or they can also represent links:



Containment Connection

A **treemap** is an example of a parent/child containment hierarchy:

- It typically uses size/colour encoding in addition.
- Alternative to node-link tree design.

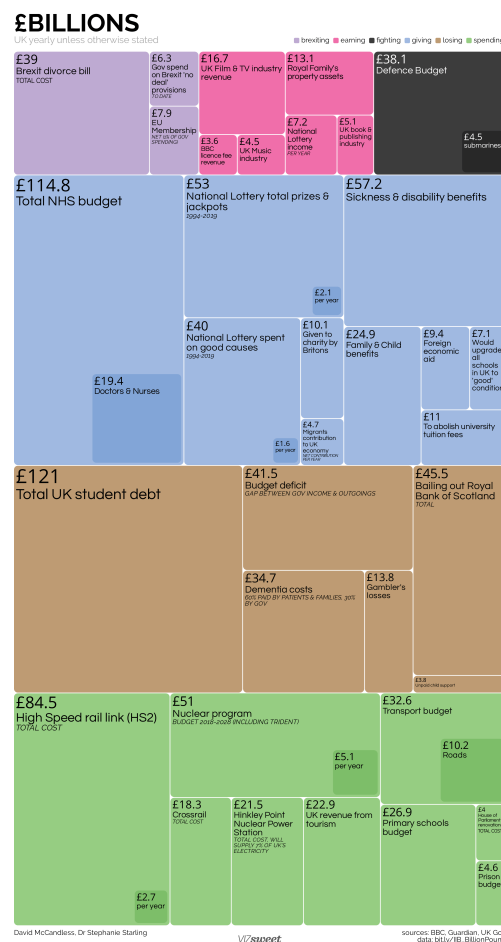


Figure 3: The Billion Pound O Gram³

“Huge sums of money are being bandied about and no-one knows what they are. It’s time to put them into perspective. 289 billion spent on this. 400 billion spent on that. When money reaches this level it literally becomes mind-boggling. Yet these figures are regularly issued by the government – and the media – as if they are self-evident facts that everyone understands. Frustrated by this, I created The Billion Pound-O-Gram. I’ve mixed up of 2008/09 figures from the Treasury and the Guardian. Visualising the numbers like this puts them in visual context, making them easier to relate to. I was pretty shocked by the size of the UK budget deficit – essentially the country’s overdraft. It’s more than an entire year’s worth of income tax.” — David McCandless

³<https://informationisbeautiful.net/visualizations/the-billion-pound-o-gram/>

3.4. Perceptual Effectiveness

Two principles guide the use of visual channels:

- **Expressiveness Principle:** visual information should express all and only the information in the data; ordered data should not appear as unordered and vice-versa
- **Effectiveness Principle:** importance of the information should match the **salience** of the channel — the quality of being particularly noticeable or important

Visualisations have a large design space, so we want to understand which are more effective than other ones and which are more expressive than others?

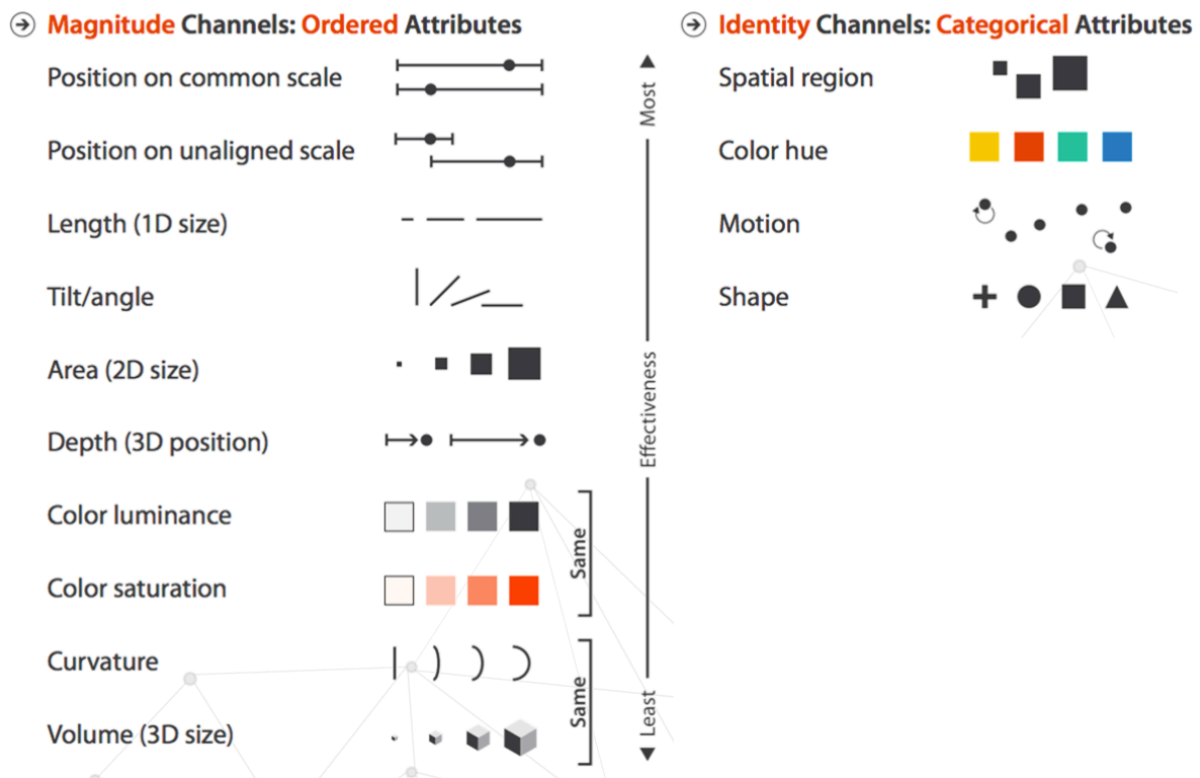


Figure 4: Expressiveness and effectiveness of different channels

Case study of Al Gore documentary: former president shares his concerns on pressing issue of global warming; there are some interesting elements such as colours as visual channel (red to encode alarming data) and most importantly leveraging position to increase tension, pushing it outside the graph. However, they do not show the scale of reference of the new data, assumptions are made.



Figure 5: Al Gore presentation⁴

⁴<https://youtube.com/watch?v=9tkDK2mZlOo>

3.4.1. Steven's Power Law

The given plot shows the intensity of sensation versus the sensation reported.

We find length is linear, we under-estimate volume/loudness, and we over-estimate electric shock and heaviness. I is the magnitude of stimulus, N is an exponent that depends on type of stimulation.

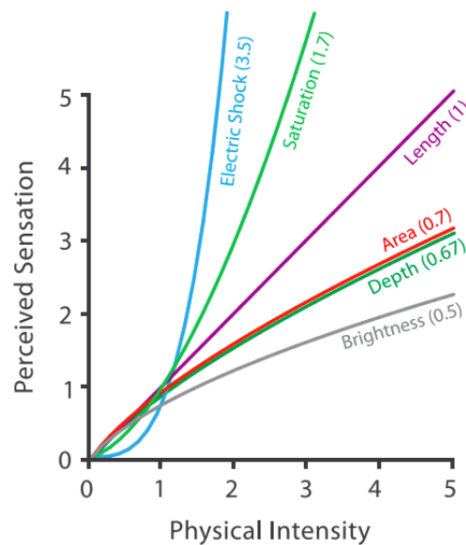


Figure 6: Steven's Psychophysical Power Law: $S = I^N$

3.4.2. Jacques Bertin: Marks as basic units

Bertin proposed marks as basic units and methods through which units can be modified including position, size, shape, or colour. These predefined modifications are called visual variables, each variable (sometimes called visual attributes) can have certain characteristics.

Marks can be points (dimensionless locations represented by signs that need size/shape/colour), lines (that represent information with certain length, visualised by signs of some thickness), areas (that have length and width therefore 2D size, can be visualised by signs of texture, etc), surfaces (areas in 3D space with no thickness), and volumes (truly three-dimensional).

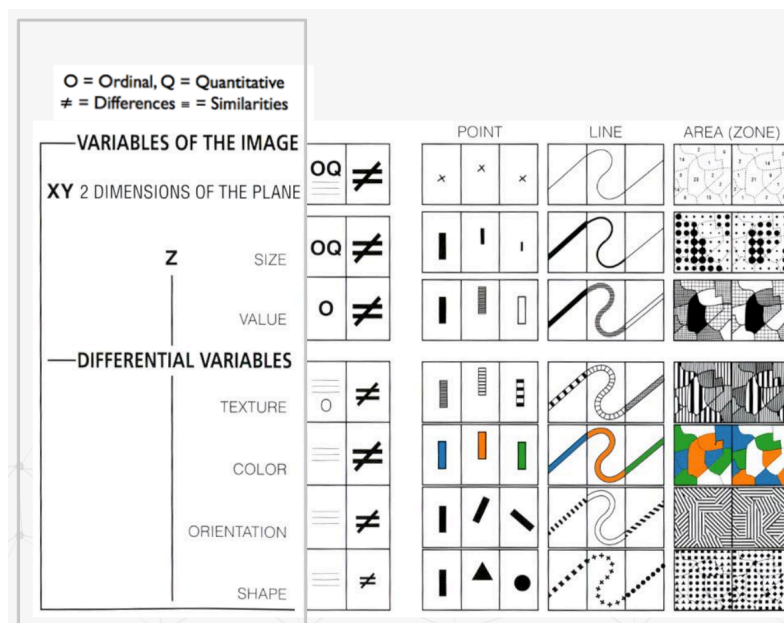


Figure 7: Variables of the image

Highlights from Bertin's work:

- Relative size is the most widely useful and easily perceived visual variable, no matter what information is visualised.
- Size and planar position are only variables that can accurately communicate quantitative information.
- For communicating order, value (light to dark) is far superior to colour (one colour to another).
- Colour, orientation, and shape function best when creating visual associations and groups.

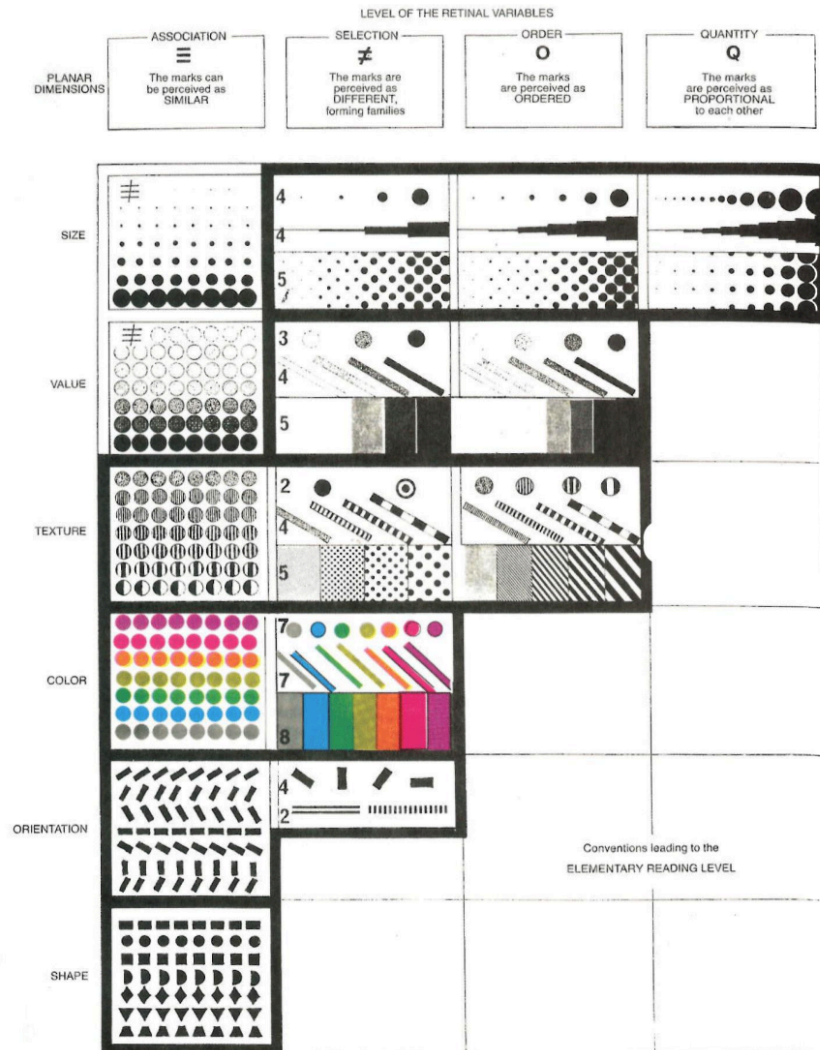


Figure 8: Level of Retinal Variables

3.4.2.1. Expanded work by John D. Mackinlay

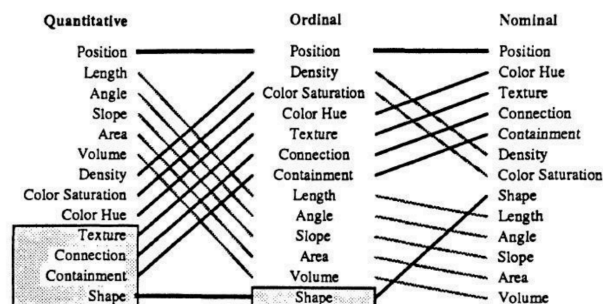


Figure 9: Ranking of accuracy based on type of variable

3.4.3. Cleveland & McGill: Ranking of graphical encodings

Cleveland & McGill carried out extensive study of graphical encodings to obtain a ranking of different encodings we most accurately understand:

1. Position along a common scale; bar chart, dot plot
2. Positions along non-aligned, identical scales; small multiples, stacked bar chart
3. Length, direction, angle; pie chart
4. Area; treemap
5. Volume, curvature; 3D bar charts, area charts
6. Shading, colour saturation, colour properties; heat maps, choropleth maps

Take-away messages:

- Use highest possible encoding on scale
- Be careful when using angles and slopes to encode numerical data
- Don't use area/volume to encode numerical data
- Don't use colour to encode numerical values

3.4.4. Channel Effectiveness

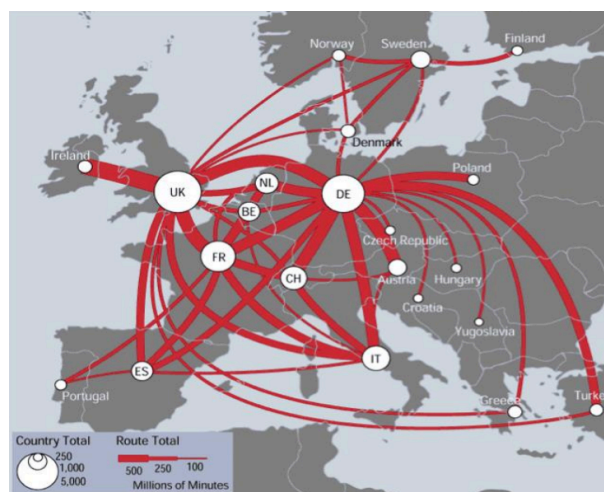
There are a few criteria (beyond accuracy) that we can look at:

- Accuracy: how accurately values are estimated
- Discriminability: how many different values can be perceived
- Separability: how much interaction there is with multiple encodings
- Popout: how easy it is to spot some values from the rest
- Grouping: how good a channel is in conveying groups

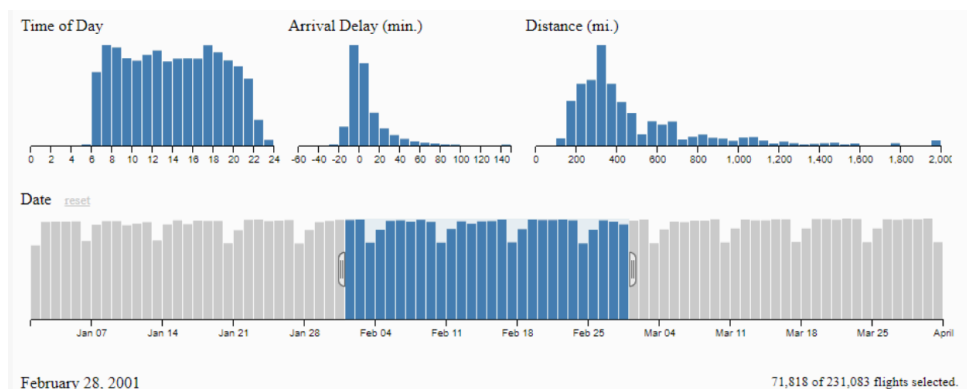
3.4.4.1. Discriminability

Bins are steps/levels of a visual variable which are distinguishable, e.g. line width: changing line size only works for a small number of steps before we perceive it as a shape/polygon area.

Rule of thumb is to match the number of values that need to be encoded with the available bins of the visual attribute we want to use (# of values should not be greater than # of bins).



If there is too much data, we may want to **aggregate** by replacing individual data points with representation of subsets:



The take-away message is that: we should not overestimate the no. of available bins. Most visual channels have a very low number. When you have too many categories, find a way to group/bin the data further or change the channel altogether.

3.4.4.2. Separability

separable: can judge each channel individually

integral: two channels are viewed holistically

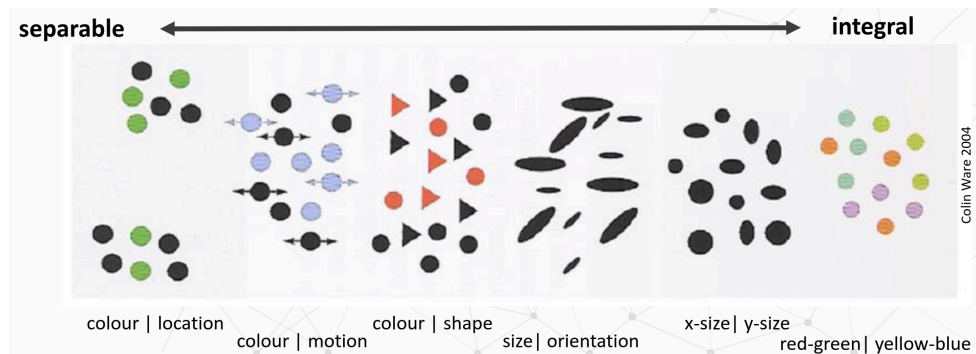
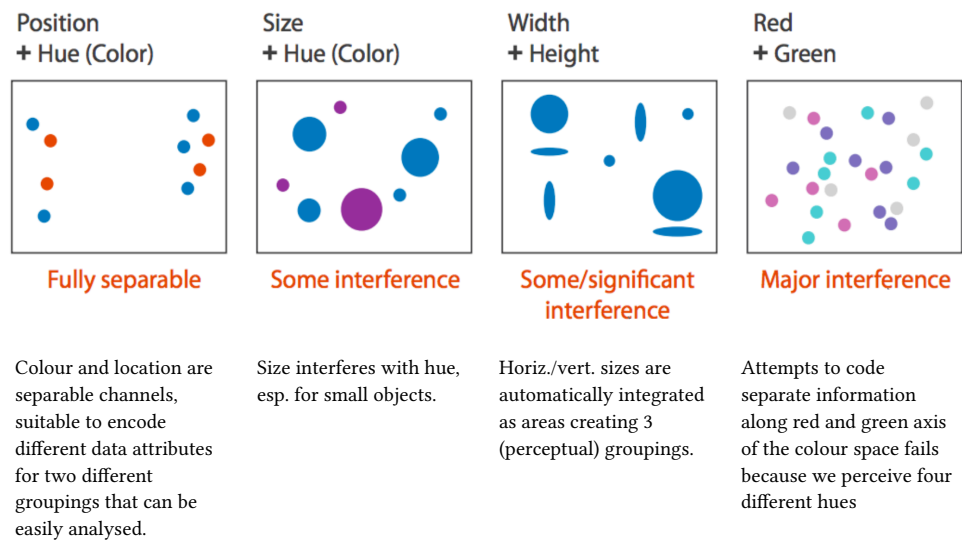


Figure 10: Pairs of visual channels fall along a continuum

4. Visual Perception & Colour

Further reading for this section & diagram source: <https://www.csc2.ncsu.edu/faculty/healey/PP>

The **pop-out effect** is a process involving pre-attentive processing where-by our brains can spot an outlier from a group of distractors (anything which is not our target object).

Attentive processing: conscious processing of information

Pre-attentive processing: subconscious processing of information, pre-processed by brain quickly

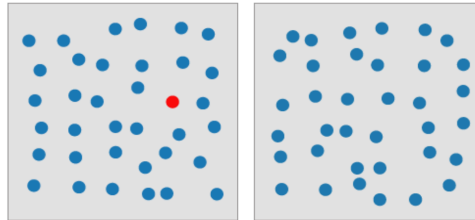


Figure 11: Example Image 1 (strong)

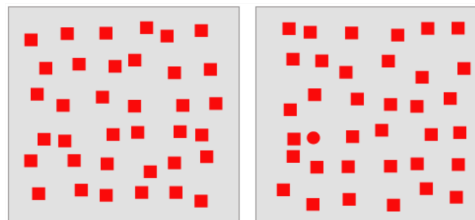


Figure 12: Example Image 2 (weak)

The pop-out is:

- not dependent on the number of distractors but dependent on the number of visual channels & distance between visual channels e.g. how different is the target from its surroundings
- the change from Figure 11 to Figure 12 is in the 'shape' channel

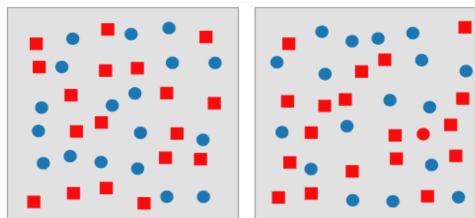


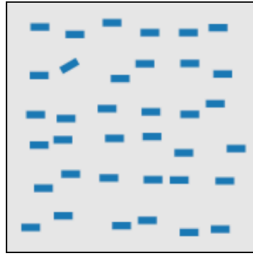
Figure 13: Example Image 3 (no pop-out)

Conjunction: too many channels hinder our ability to process pre-attentively

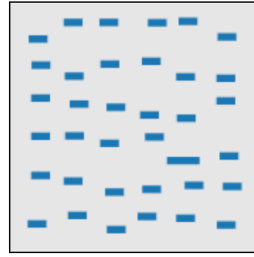
Many channels support visual pop-out which can be leveraged to improve perf. in analytical tasks:

- target and pattern detection: to rapidly and accurately detect the presence or absence of a 'target' or set of 'target' element(s) with a common visual feature within a field of distractor elements
- boundary detection: to rapidly and accurately detect a texture boundary between two groups of elements, where all of the elements in each group have a common visual property region tracking: to identify and track one or a set of elements with a common visual feature as they move in time and space
- counting and estimation: to estimate the number of elements sharing a common visual feature

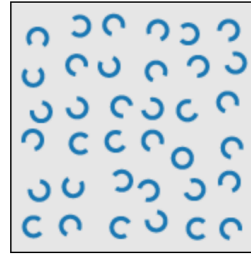
Takeaway: To direct attention (e.g. selection), use pre-attentive features. We can easily see objects that are different in colour and shape, or that are in motion. If you want to encode data 'holistically' use integral dimensions, otherwise use separable dimensions.



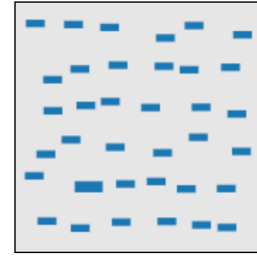
line (blob) orientation
Júlész & Bergen 83; Sagi & Júlész 85a, Wolfe et al. 92; Weigle et al. 2000



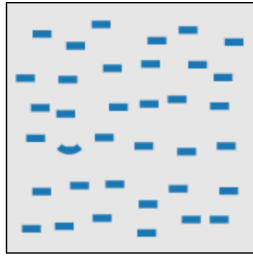
length, width
Sagi & Júlész 85b; Treisman & Gormican 88



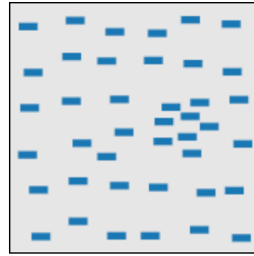
closure
Júlész & Bergen 83



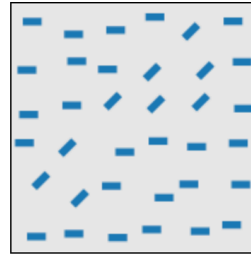
size
Treisman & Gelade 80; Healey & Enns 98; Healey & Enns 99



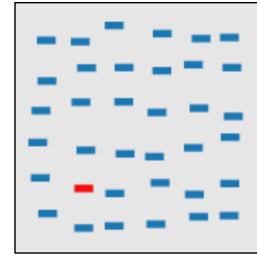
curvature
Treisman & Gormican 88



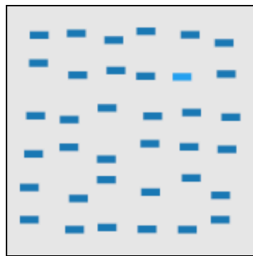
density, contrast
Healey & Enns 98; Healey & Enns 99



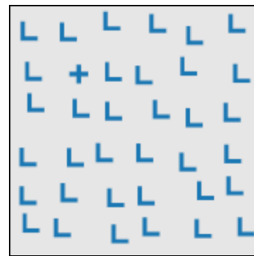
number, estimation
Sagi & Júlész 85b; Healey et al. 93; Trick & Pylyshyn 94



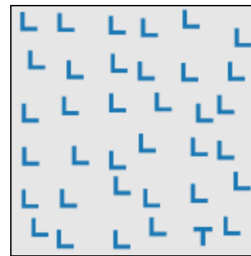
colour (hue)
Nagy & Sanchez 90; Nagy et al. 90; D'Zmura 91; Kawai et al. 95; Bauer et al. 96; Healey 96; Bauer et al. 98; Healey & Enns 99



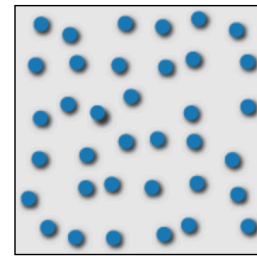
intensity, binocular lustre
Beck et al. 83; Treisman & Gormican 88; Wolfe & Franzel 88



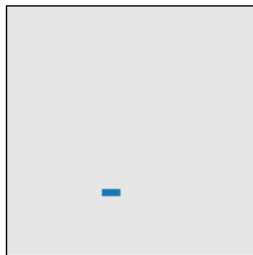
intersection
Júlész & Bergen 83



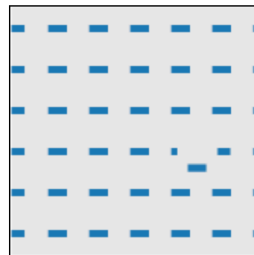
terminators
Júlész & Bergen 83



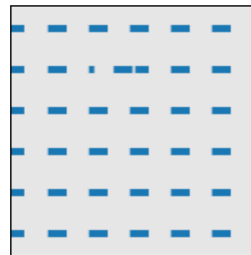
3D depth cues
Enns 90b; Nakayama & Silverman 86



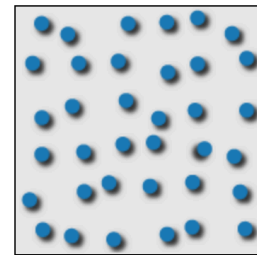
flicker
Gebb et al. 55; Mowbray & Gebhard 55; Brown 65; Júlész 71; Huber & Healey 2005



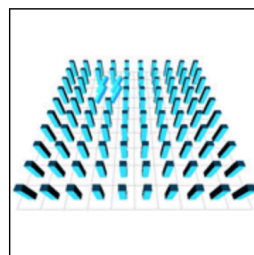
direction of motion
Nakayama & Silverman 86; Driver & McLeod 92; Huber & Healey 2005



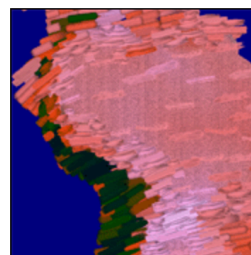
velocity of motion
Tynan & Sekuler 82; Nakayama & Silverman 86; Driver & McLeod 92; Hohnsbein & Mateeff 98; Huber & Healey 2005



lighting direction
Enns 90a



3D orientation
Enns & Rensink 90a; Enns & Rensink 90b; Liu et al. 2003

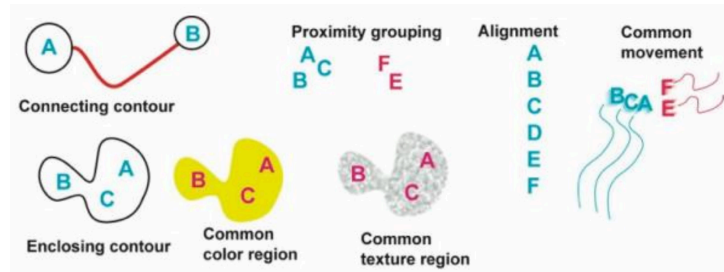


artistic properties
Healey 2001; Healey & Enns 2002; Healey et al. 2004

4.1. Gestalt

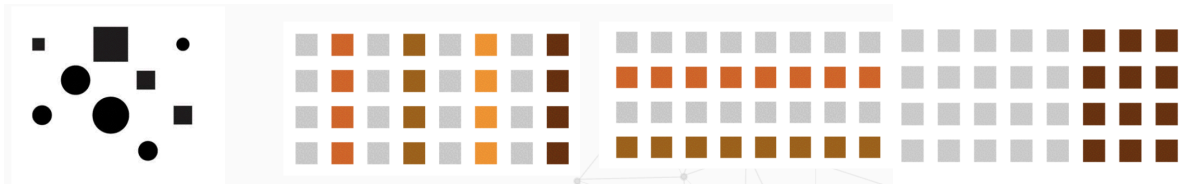
Gestalt is a movement in experimental psychology which began prior to WWI, based on concept of 'grouping', patterns transcend the visual stimuli that produced them.

We perceive objects as well-organized patterns rather than separate components. "The whole is greater than the sum of it's parts."

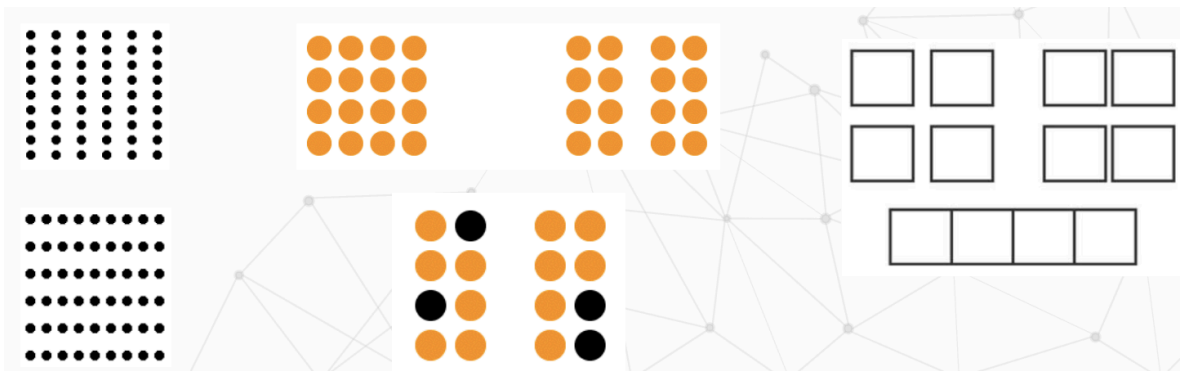


Gestalt Principles of Visual Perception

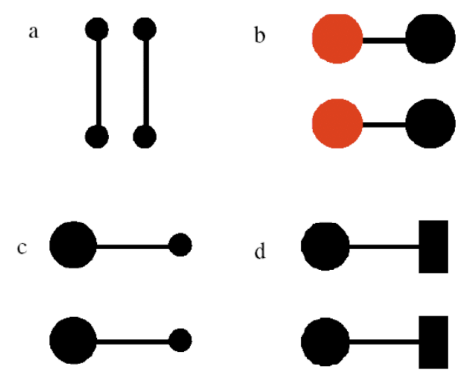
- similarity: things that look like each other (size, color, shape, texture, orientation) are related



- proximity: things that are visually close to each other are related; even if shapes and objects may be unrelated, we perceive them as the same group if they are within close proximity of each other.



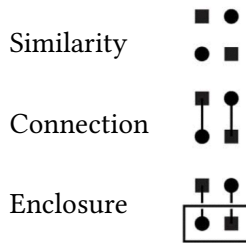
- connection: things that are visually connected are related (when compared with other elements that are not linked in the same manner; works to contradict other principles, e.g. prox./sim.)



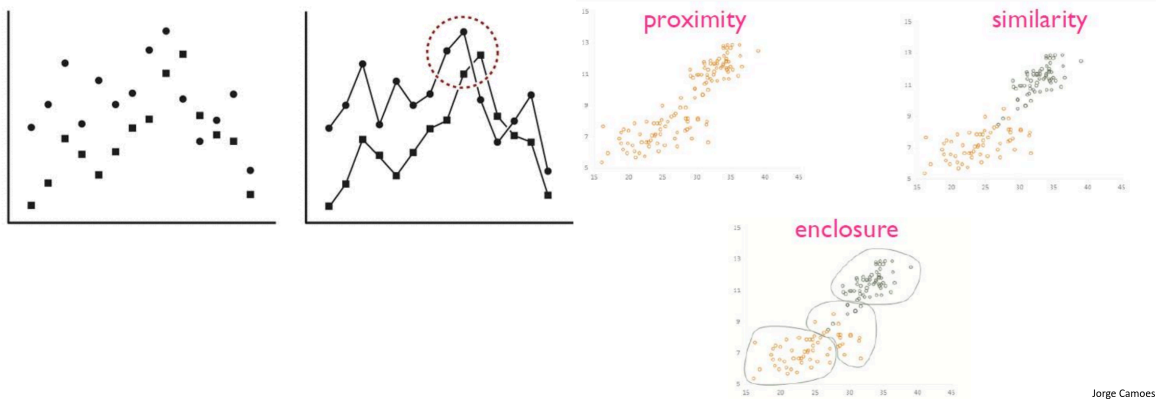
- *cont.*

Note

Further expanding on contradictions with connections:

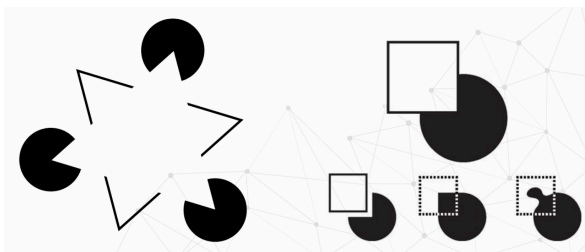


Two examples of this in practice:



Gestalt Principles of Visual Perception (cont.)

- continuity: we complete hidden objects into simple, familiar shapes

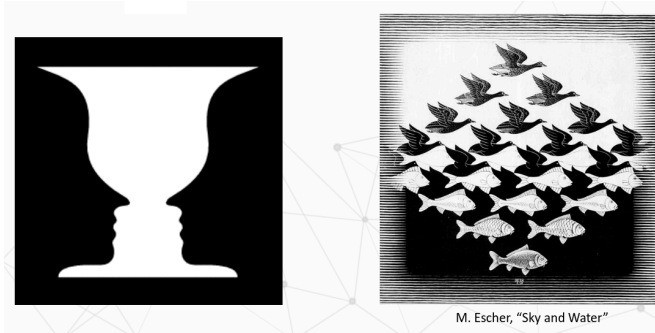


- closure: we see incomplete shapes as complete



Gestalt Principles of Visual Perception (cont.)

- figure / ground: elements are perceived as either figures or background



- common fate: elements with the same moving direction are perceived as a unit

Try Bio Motion Lab: <https://www.biomotionlab.ca/Demos/bmlwalker/>

4.2. Colour

Colour is a very powerful channel and its use is pervasive in data visualization. The image shows a summary of the Beatles working schedule from 1963 to 1966.

Strengths of the visualisation below:

- use of multifunctioning elements
- layered information
- maximizes the data density
- maximizes the amount of data shown

In a single layout the authors have been able to pack several layers of information. Color plays a central role here. It is important to own the basic principles behind Color properties and how they interact and affect our perceptual system, how we perceive what is in front of us.

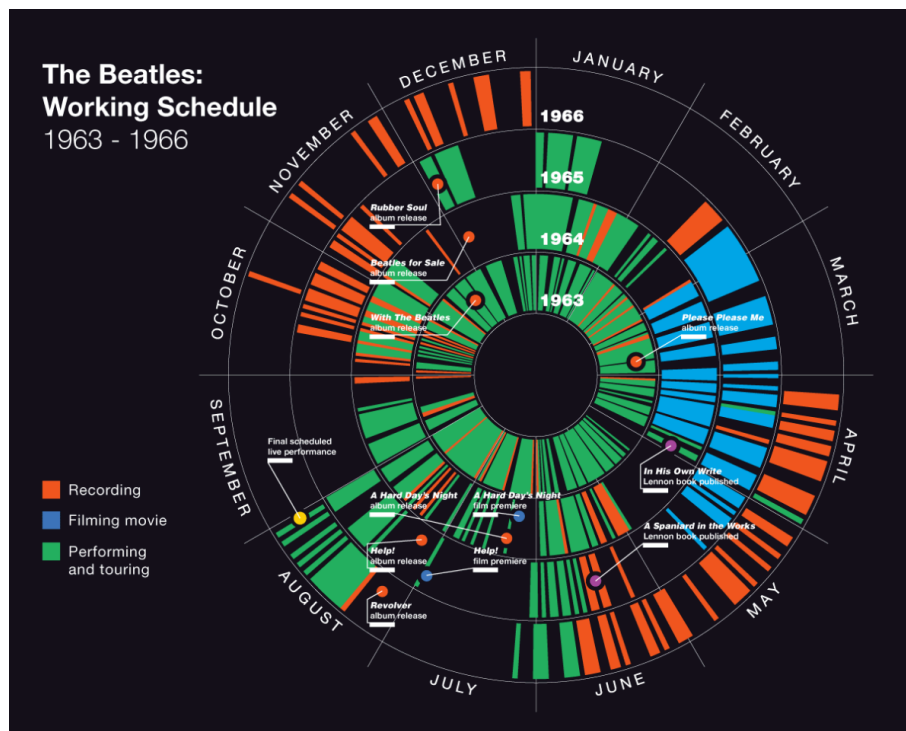
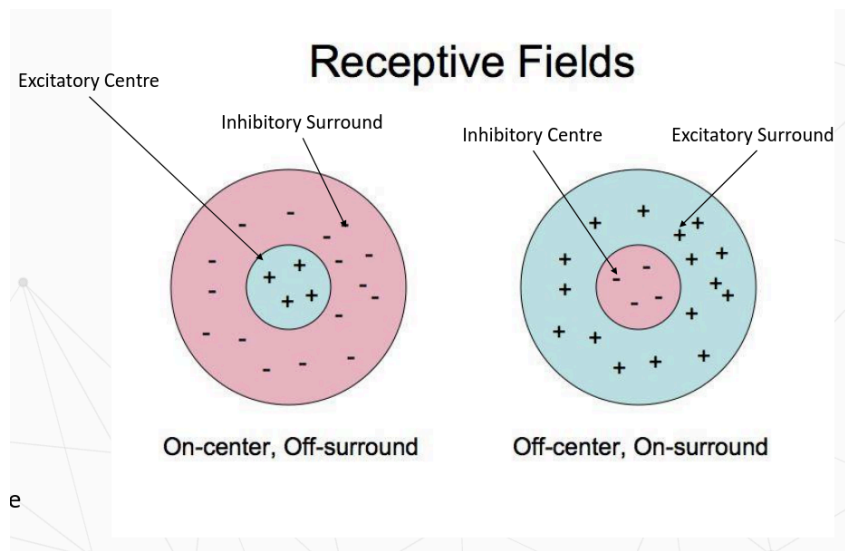


Figure 14: The Beatles: Working Schedule

The **receptive field** is identified as the region of the retina where the action of light alters the firing of the neuron.

receptive field = centre + surround



The receptive field in the eyes retinal ganglion cells is an area (volume in space) of the retina which encompass all the photoreceptors (e.g. rods and cones in our eyes)

A retinal ganglion cell is a type of neuron in the retina of the eye. It receives visual information from photoreceptors. These come in two types:

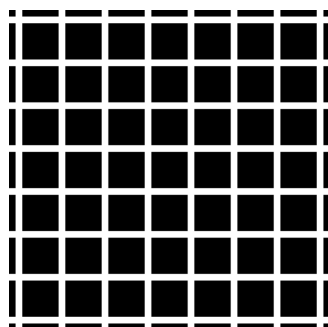
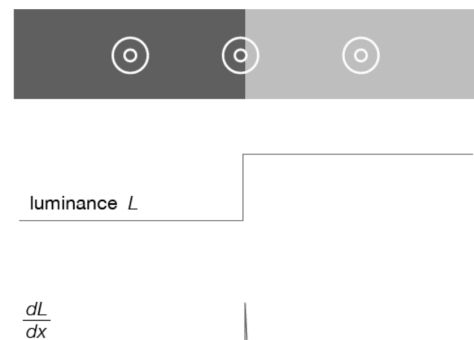
- On-centre: stimulated when the centre of its receptive field is exposed to light and is inhibited when the surround is exposed to light
- Off-centre: have opposite reaction

Centre-surround antagonism

Ganglion cells respond to differences in intensity. Changes in luminance trigger reactions in our receptive field.

The Center-surround antagonism phenomena refers to antagonistic interactions between center and surround regions of the receptive fields of photoreceptor cells in the retina.

This “reaction” enables edge detection and contrast enhancement within the visual cortex.

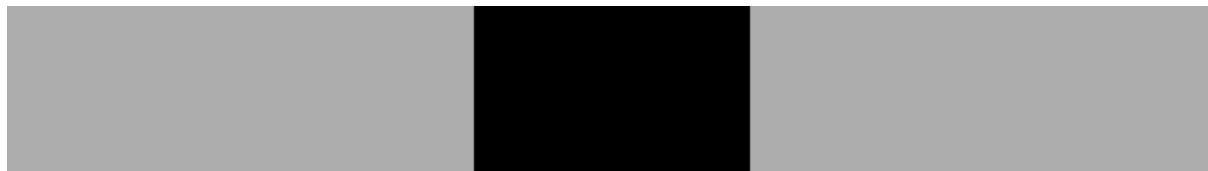


Lateral inhibition refers to the capacity of excited neurons to reduce the activity of their neighbors. Neurons that are firing inhibit the stimulation of surrounding. Accordingly, only the neurons that are most stimulated and least inhibited respond. Lateral inhibition increases the contrast and sharpness in visual response. Lateral inhibition plays an important role in visual perception by increasing the contrast and resolution of visual stimuli.

This can lead to the ghosting effect seen on the left.

Looking at pictures of data means looking at lines, shapes, and colors. Our visual system works in a way that makes some things easier for us to see than others. Perception is not a simple matter of direct visual inputs producing straightforward mental representations of their content. Rather, our visual system is tuned to accomplish some tasks very well, and this comes at a cost in other ways.

4.3. Cornsweet Illusion



Consider the perceived contrast above, and now look at the actual contrast below:



Actual luminance distribution

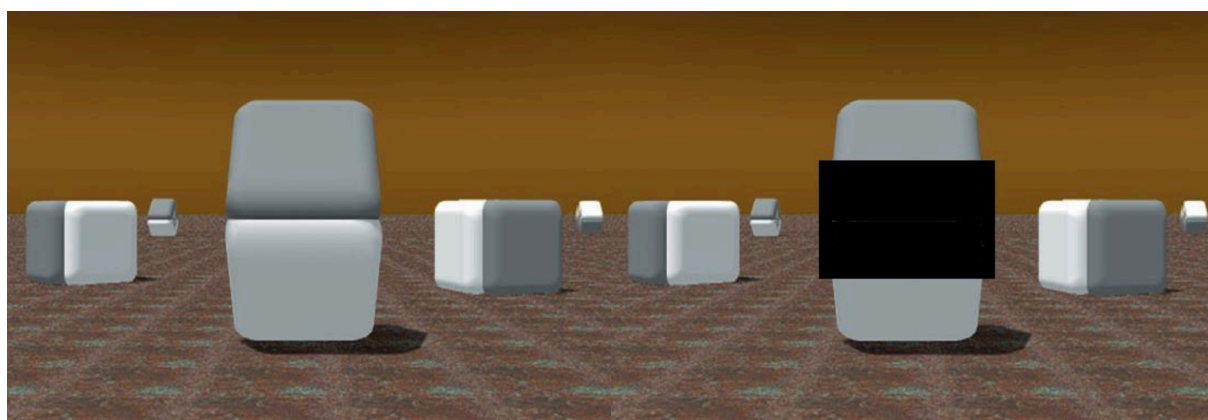


Perceived luminance distribution



In the Cornsweet illusion, the region adjacent to the light part of the edge appears lighter, and the region adjacent to the dark part of the edge appears darker, just the opposite of the usual contrast effects.

The Cornsweet illusion is an example of the way the brain uses the edges of shapes to impart information to the surrounding areas, and takes place because of the way nerve impulses travel through the visual cortex. The retina interprets what it sees using certain luminance profiles. The Cornsweet image is processed with nearly identical codes, which the cortex then integrates and arrives at the same perceptual result. In other words, the brain is used to seeing things a certain way, and often interprets information within those parameters.



4.4. Contrast Sensitivity

Eyes are mostly sensitive to abrupt changes in luminance. Our visual system is attracted to edges.

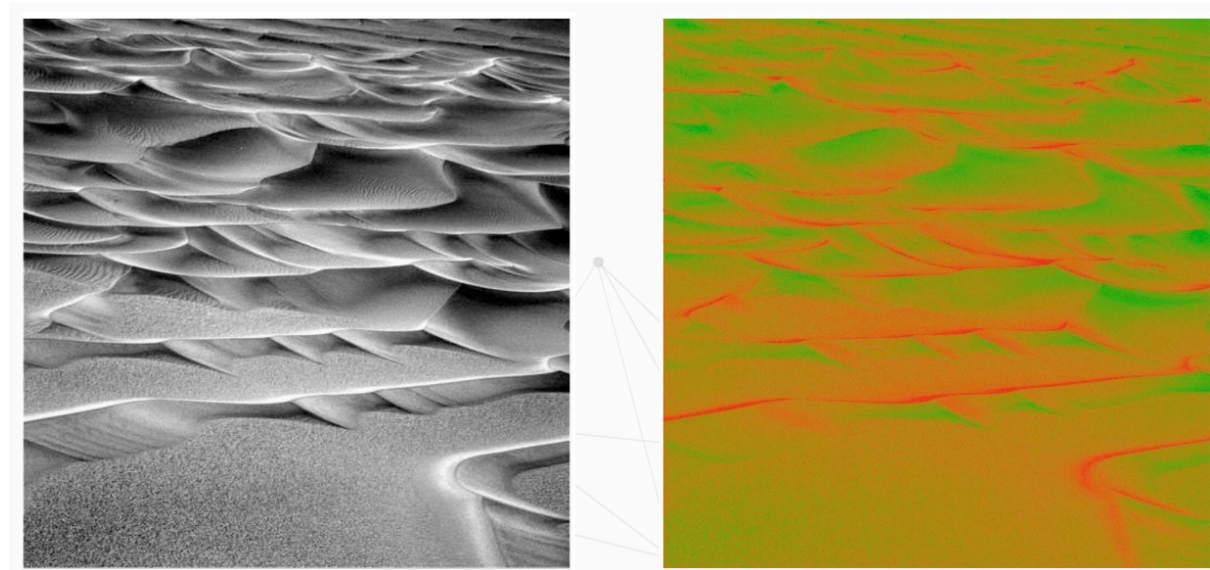


Figure 15: Higher contrast sensitivity in luminance than in chrominance

The brain constructs surface colour based largely on edge contrast information too, but we have higher contrast sensitivity in the luminance than in the chrominance channel.

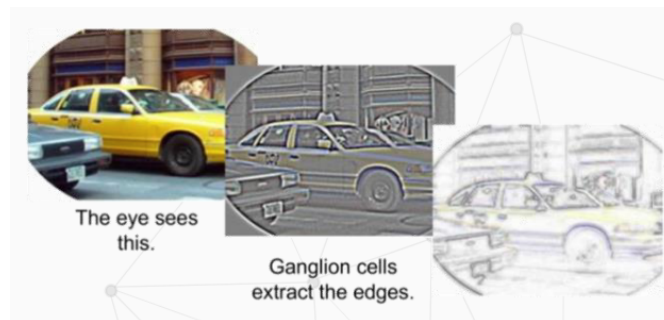


Figure 16: Features extracted at various stages

Our visual system seems differences not absolute values.

Takeaway: Maximize the contrast with the background if the outlines of shapes are important.

4.5. Relativity of Perception / Relative Differences

We judge on relative, not absolute, differences.



Weber's Law

The ratio of the increment threshold to the background intensity is a constant. So when you are in a noisy environment you must shout to be heard while a whisper works in a quiet room. And when you measure increment thresholds on various intensity backgrounds, the thresholds increase in proportion to the background.

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

When we measure increment thresholds on various intensity backgrounds, the thresholds increase in proportion to the background.



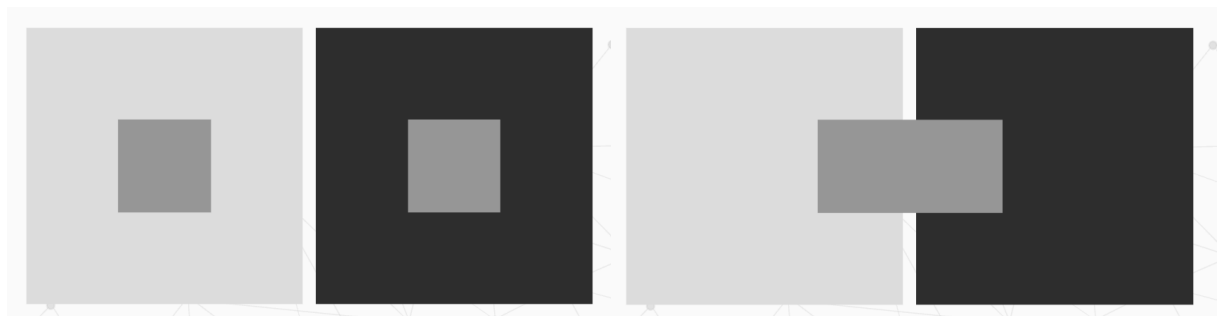
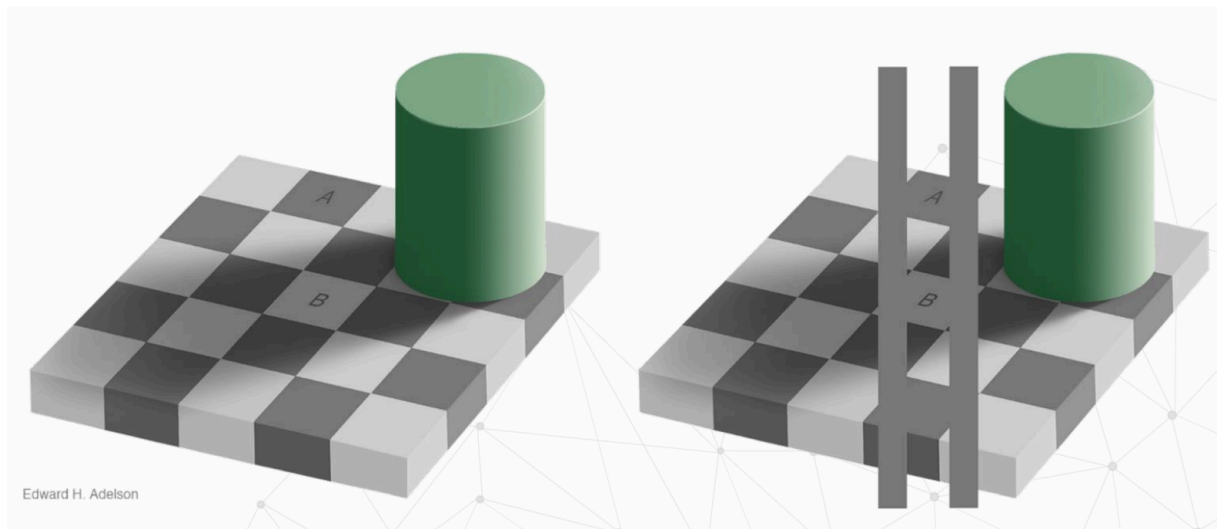
4.6. Axis of Alignment

Consider the following lines, do they look the same length? What about if we flip them on their side?



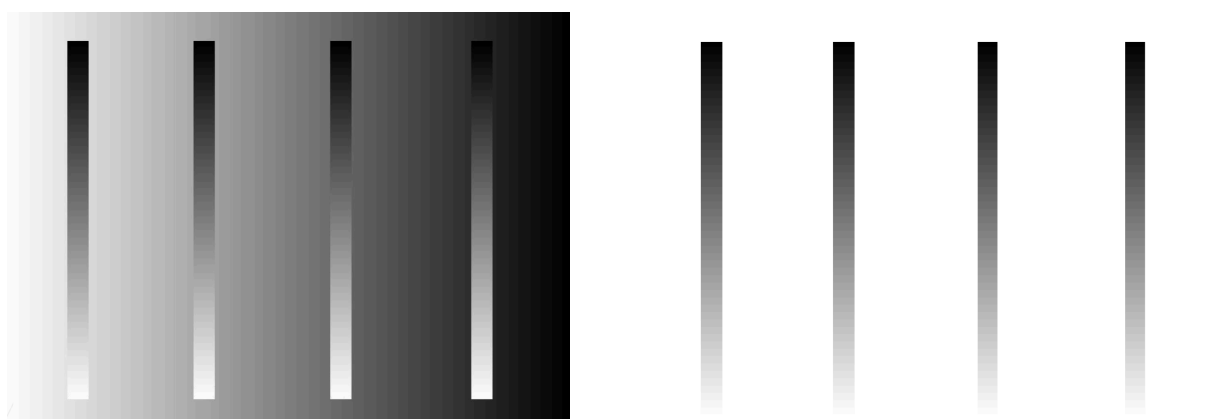
4.7. Simultaneous Contrast

Perceived colour is highly context dependent, variable lighting and background conditions affect what we see. Pushed to the logical extremes:



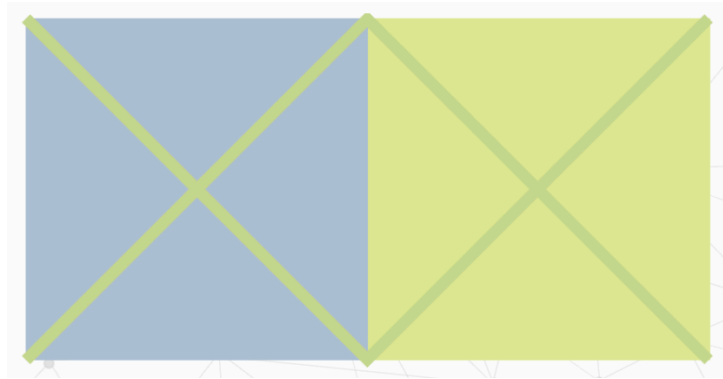
4.8. Crispning Effect

Perceived differences between grayscale values are enhanced where the values are close to the background grey value.



4.9. Interaction of Colour

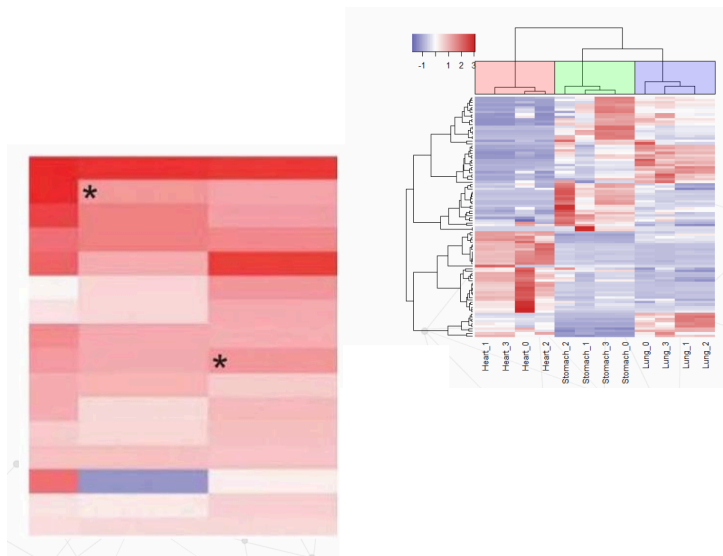
The same colour can look different:



Different colours can appear to be nearly the same by changing the background colour



A heat map requires us to judge the relative brightness of colours in a matrix. The interaction of colour can cause a profound effect that makes this graphical representation suffer.



We judge on relative, not absolute, differences.

- Specifically, ratio of intensity increment to background is constant.
- The perceived minimum increment is also referred to as the: **just noticeable difference (JND)**
- JNDs are different for each visual channel
- Change the background, change the increment.

However, we can exploit perception, e.g. using closure, continuity, and cornsweet contours.



Figure 6.15 Both contour- and color-defined regions have been added to make clear the distribution of hotels (orange), subway stations (brown), and medical clinics (purple). (From Collins et al. (2009). Reproduced with permission.)

4.10. Colour

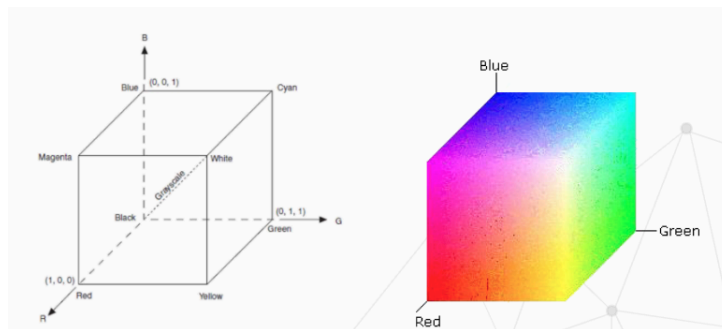
Purpose of Colour:

- to label (colour as a noun)
- to measure (colour as a quantity)
- to represent and imitate (colour as a symbol)
- to enliven and decorate (colour as beautification)

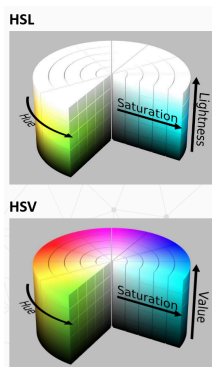
4.10.1. Colour Spaces

RGB

- Most common
- Additive (useful for monitors)
- Not perceptually uniform

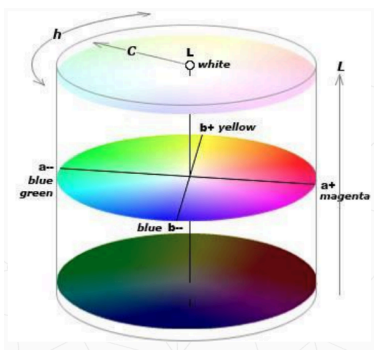


Perceptual uniformity is the change in amount of colour value that should produce an equivalent visual change.



HSL/HSV

- common cylindrical-coordinate representations of points in RGB space
- rearrange geometry of RGB in attempt to be more intuitive and perceptually relevant
 - hue: what people think of as colour
 - saturation: intensity of the colour (or amount of white mixed in)
 - Luminance or value: how bright or dark the colour is (or amount of black mixed in)



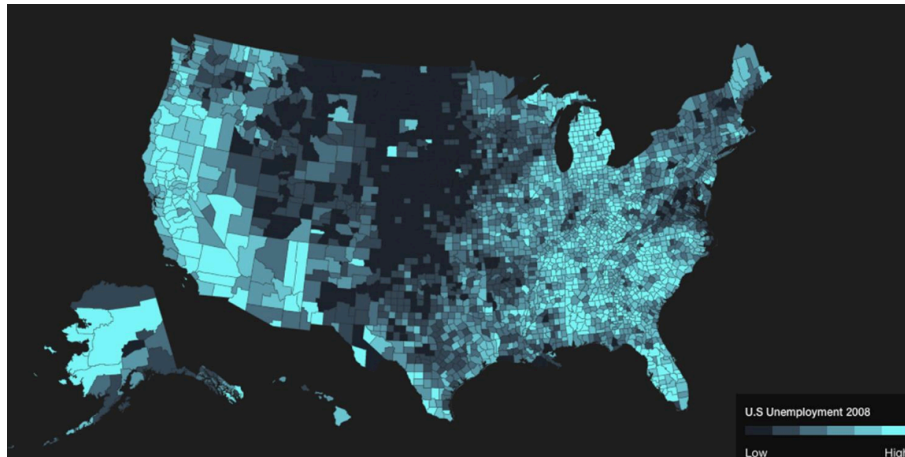
CIE L*A*B*

- Designed to approx. human vision
- Describes all colours known to human eye
- Uses + 've and - 've values
 - L*: lightness
 - a*: red/magenta to green
 - b*: yellow to blue
- Relative to a point of white (D50)
- Supersedes RGB and CMYK and is device independent

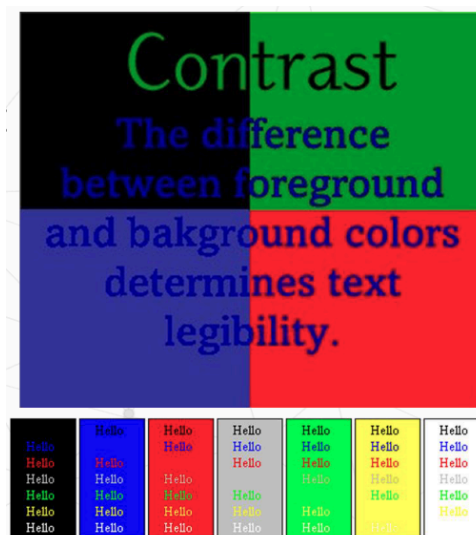
4.10.2. Colour Guidance

Luminance is suitable for ordered/quantitative data.

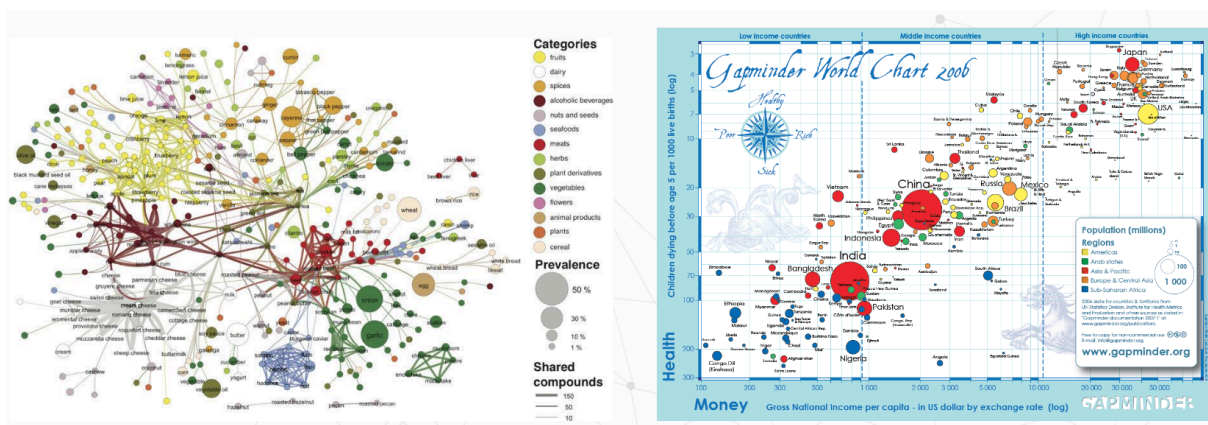
But we can only easily distinguish about 5 different non-contiguous values!



Also, contrast with background is crucial! You cannot visualize data with high luminance on a background with high luminance. Even more so with text (and text in vis is crucial as labels give meaning to objects)



Hue is a very effective channel for categorical attributes and to show groupings.



4.10.2 Colour > Colour Guidance

saturation and hue are not separable in small regions

- in small regions use bright, highly saturated colours

saturation interacts strongly with size

- more difficult to perceive in small regions
- for points and lines use just two saturation levels

higher saturation makes large areas look bigger

- use low saturation pastel colours for large regions and backgrounds

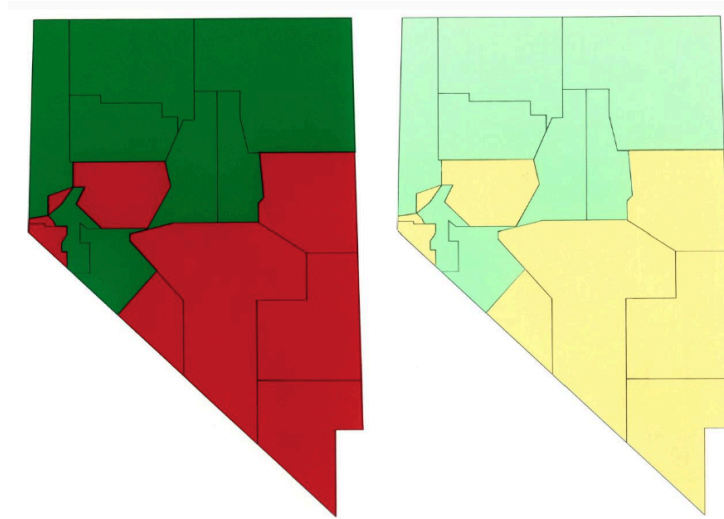


Figure 1. Stimulus From the High-Saturation Group

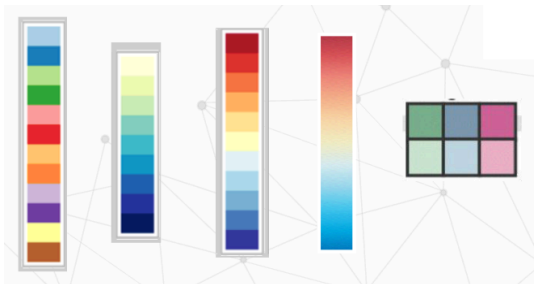
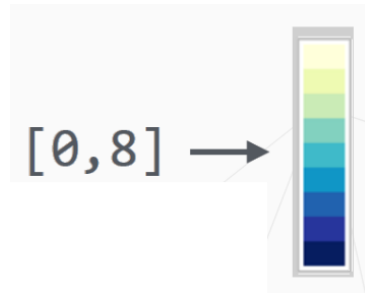
Figure 2. Stimulus From the Low-Saturation Group

tl;dr

- Hue has no implicit ordering: use for categorical data (but limited hues, do not use more than 5 to 8 at a time!).
- Saturation and luminance do: use for ordered/ordinal data.
- Luminance is suitable for ordered/quantitative data

4.11. Colour Maps

A **colour map** specifies a mapping between colour and values (sometimes called a transfer function).



It can be:

- categorical vs. ordered
- sequential vs. diverging
- segmented vs. continuous
- uni-variate vs. bi-variate

expressiveness: match colormap to attribute type characteristics

Bivariate colormap shows the quantitative relationship between two variables



Used in Geography to emphasize the highest and lowest values in a dataset or to find correlations within a dataset (example bivariate choropleth maps). Difficult to interpret when one of the dimension is not a binary value

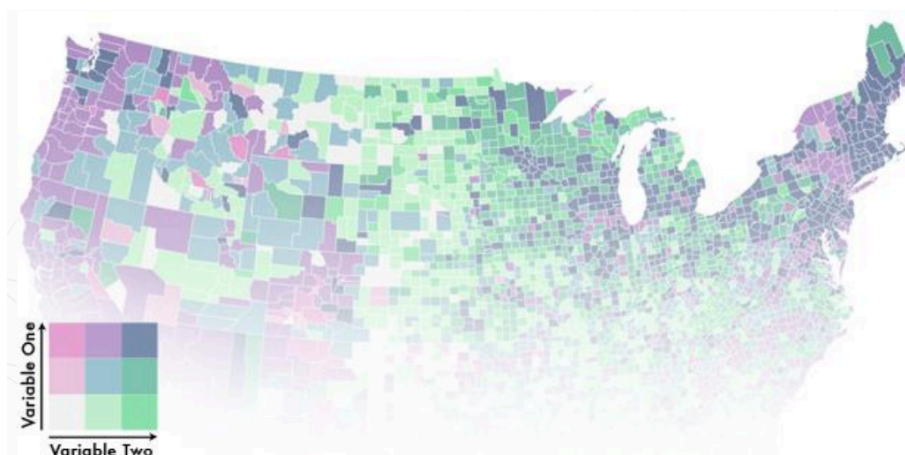


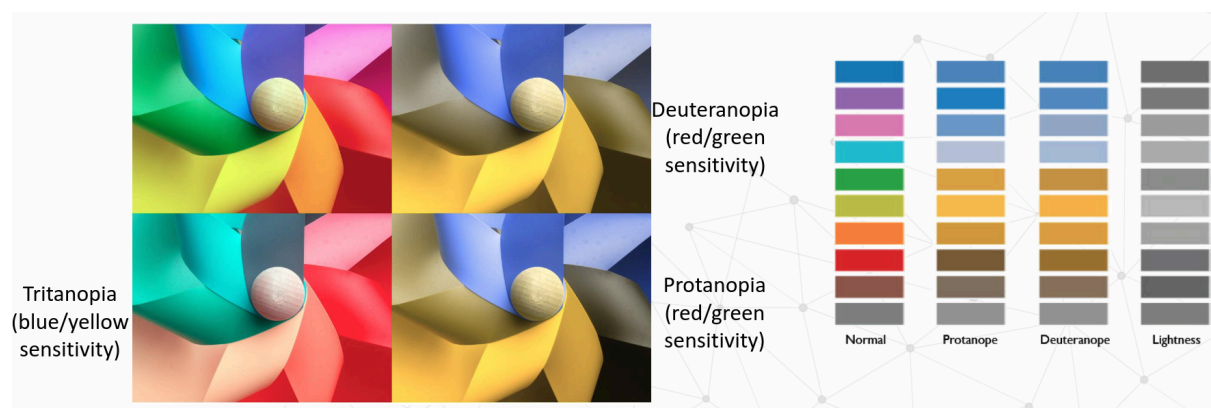
Figure 17: Bivariate choropleth map

tl;dr guidelines

- Ordered colormaps should vary along saturation or luminance
- Bivariate colormaps are difficult to interpret if at least one variable is not binary
- Categorical colours are easier to remember if they are nameable
- Number of hues, and distribution on the colormap, should be related to which, and how many structures in the data we wish to emphasize -min or max, ends or middle, etc...

4.12. Colour Deficiency

Often forgotten aspect, affects 1 in 12 men, much more rare in women.



	NORMAL VISION	DEUTERANOPIA common (6%)	PROTANOPIA rare (2%)	TRITANOPIA very rare (<1%)
RGB	1 0 0 0 6 73 0 146 11 146 0 0 2 0 73 73 7 0 109 219 12 146 73 0 3 0 146 146 8 182 109 255 13 219 209 0 4 255 109 182 9 109 182 255 14 36 255 36 5 255 182 119 10 182 219 255 15 255 255 109			

Figure 18: 15-colour palette for colour blindness

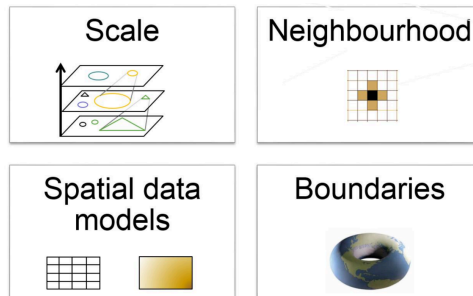
4.13. Colour Tools

- <https://colororacle.org>
- <http://colorbrewer2.org/>
- <https://colorsphere.app/#dot=false&harmony=Neutral&hue=0&saturation=100&lightness=50&quantize=0&space=RYB&theme=dark&vision=Normal>
- <http://tristen.ca/hcl-picker/#/hlc/6/1/20313E/EFEE68>

5. Spatial Data - Visualisation

In spatial modelling, there are four important concepts:

- **Scale:** multiresolution, aggregation, data on demand
- **Neighbourhood:** neighbourhood has influences and shapes of emerging patterns
- **Spatial Data Models:** discrete/continuous, static/dynamic
- **Boundaries:** the world 'ends' at the boundary (static), the world wraps (periodic), infinite space



Maps are the default approach to spatial data; pragmatic approach usually applied for real-world study areas.

There are specific tasks related to maps, most obvious find a route, also enrich your map with features and attributes. It is a special type of spatial data; we use them **when spatial relationships are paramount**.

Map tasks:

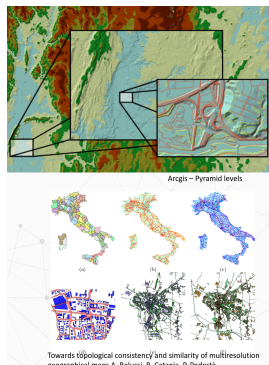
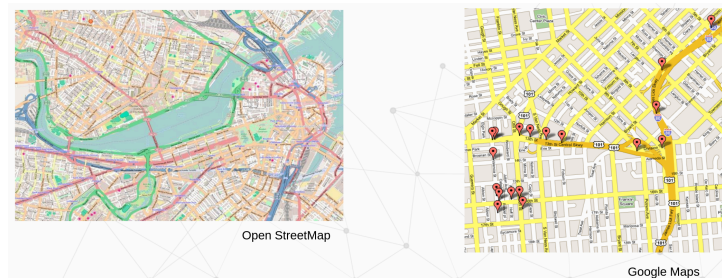
- find location / feature (county, country, city, street)
- find route
- identify attribute associated with location (elevation, land/water, GDP)
- compare attributes between areas

5.1. Maps

The problem is that depending on the scale of the phenomena to visualise spatially we need to project data from an almost spherical object in nature (that is the earth) onto a flat surface. There is not a "unique" way to do this but several.

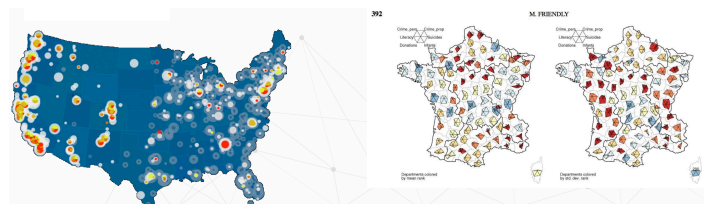
Local neighbourhood

Our data might be restricted to a sufficiently small neighbourhood which allows us to approximate the space with a plane without much noticeable distortion which does not influence emergence of patterns.



Data can also be 'different' at different scales, we could deal with this by:

- overlays: overlay one or more data layers (as simple as one data point per pixel; as complex as glyph based representations) (default in most GIS)



- Aggregation: e.g. Choropleth maps, covered in Section 5.2!

5.1.1. Types of Overlays

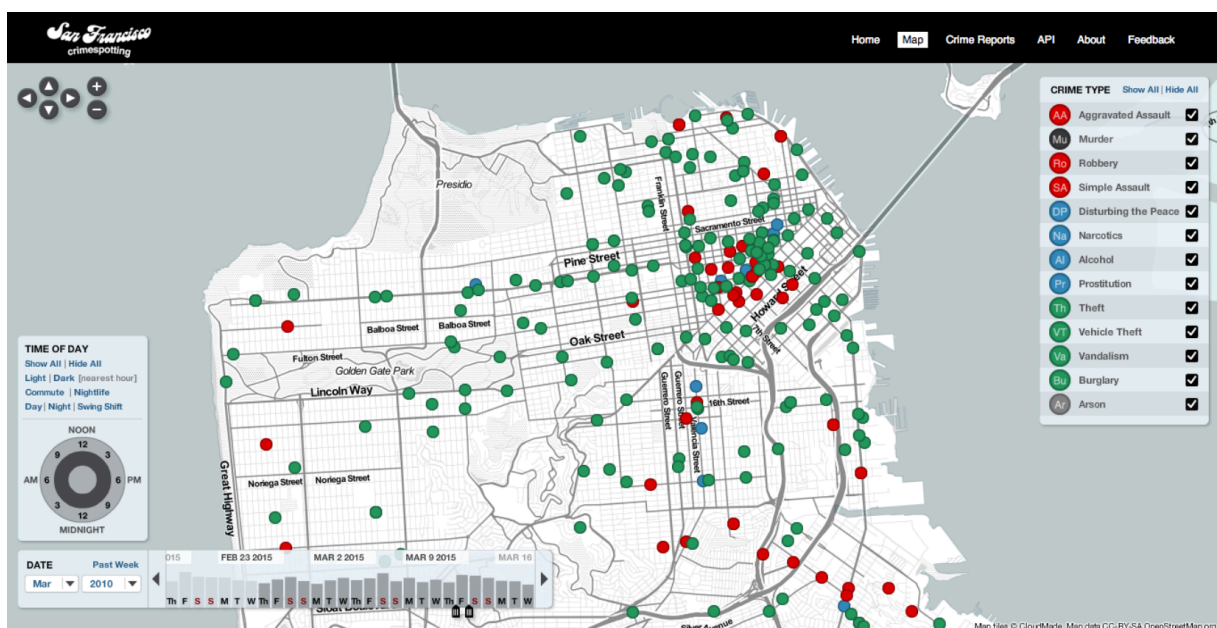


Figure 19: Dots (one data point per pixel); can embed data at the same time
Application Stamen Design, visualizes urban crimes in Oakland, California as small, coloured circles on an interactive city map. Can be filtered by crime type, time of day, and a per-day bar chart.

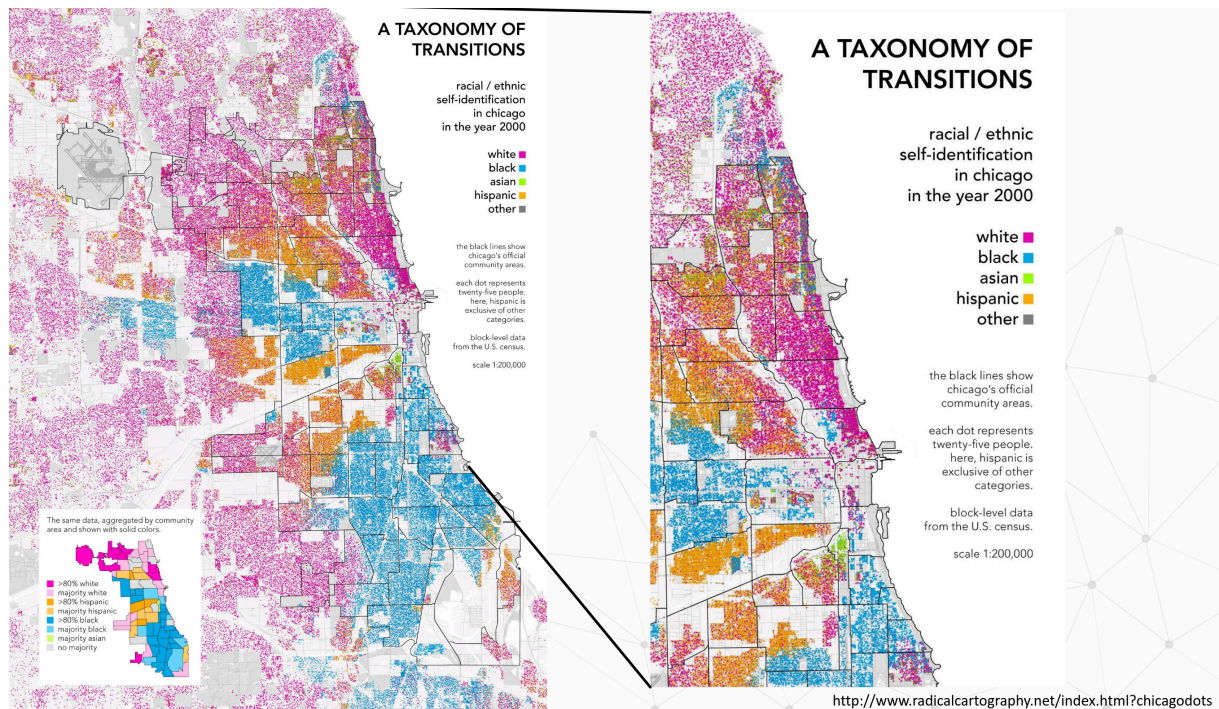


Figure 20: 200k dots, each dots represents 25 people; geographically referenced

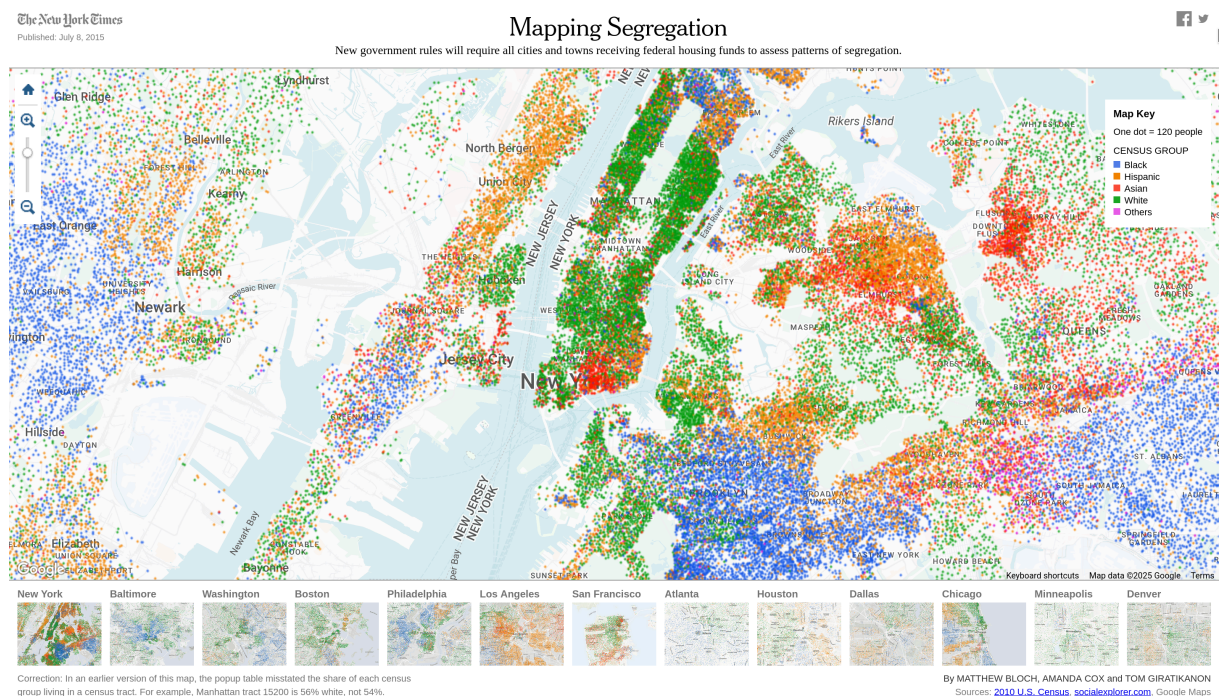
[“A taxonomy of transition”](#), (2009) by radical cartographer [Bill Rankin](#) is a visual essay on how racial boundaries mark the neighborhoods of a city like Chicago, “where the delimitation of (...) official “community areas” in the 1920s was one of the hallmarks of the famous Chicago School of urban sociology.”

This work uses dot mapping to show populations (Red/Purple is White, Blue is Black, Green is Asian, Orange is Hispanic, Gray is Other, and each dot is 25 people), hence describing three kinds of urban transitions: stark and precise boundaries, transitions and gradients, gaps.

This project was originally published as an essay in the Spring 2010 issue of [Perspecta](#), the journal of the Yale School of Architecture.

In 2011 it won the MiniMax mapping contest at the “[Moving Maps](#)” cartographic biennale in Lausanne, Switzerland.

Re-created by [NYT](#):



Clustering, grouping, binning

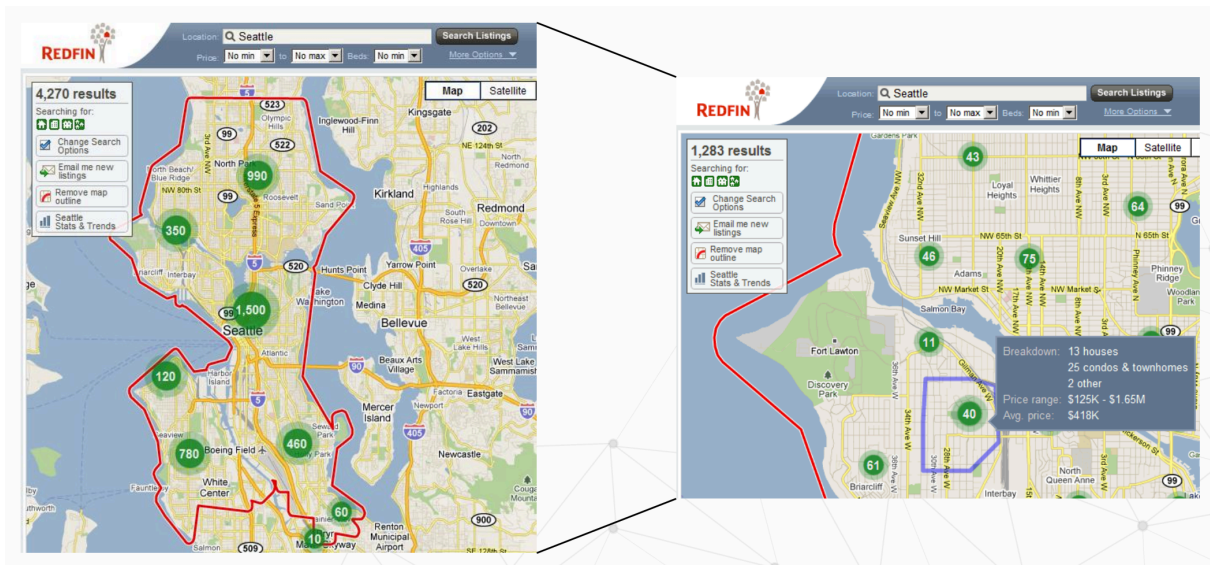


Figure 21: clustering used by redfin⁵ ; clustering listing search result, dynamically sizes and locates the clusters based on a real-time survey of listings in the area



Figure 22: Sometimes we can just let patterns emerge.

Map of pictures taken in SF.

Blue pictures are by locals. Red pictures are by tourists. Yellow pictures may be either.⁶

⁵https://www.redfin.com/news/cluster_buck_rogers/

⁶<https://www.flickr.com/photos/walkingsf/4671581511/in/set-72157624209158632>

Patterns and Context

Heat maps are a graphical representation of point data on a map through the use of colors that indicate the density of some variable such as crime incidents or traffic accidents. Heat maps let users quickly visualize the density of locations.

Being able to understand the density of point locations makes it much easier to see patterns in your data, especially when using colors.

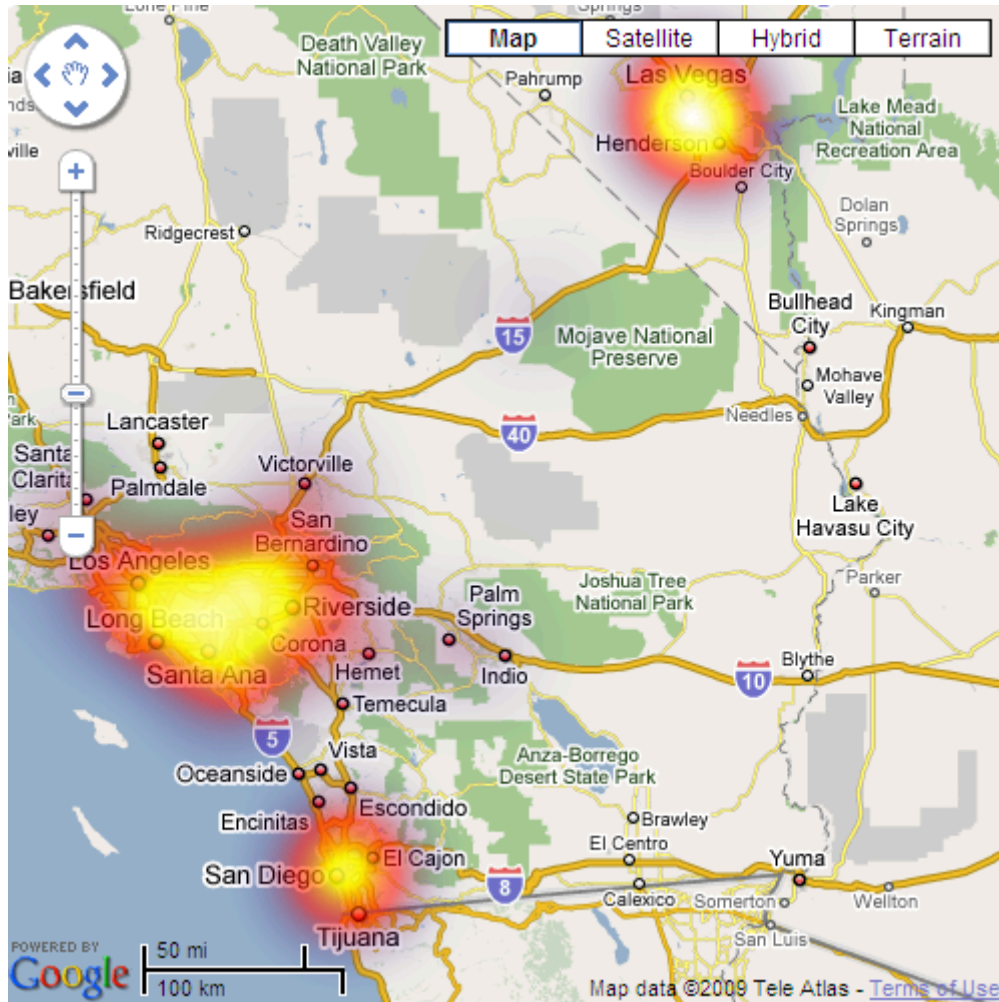


Figure 23: Density Mapping in Google Maps with HeatMapAPI

Overlays: Navigation

Another common way of information overlay are navigation maps. Route maps, which depict a path from one location to another.

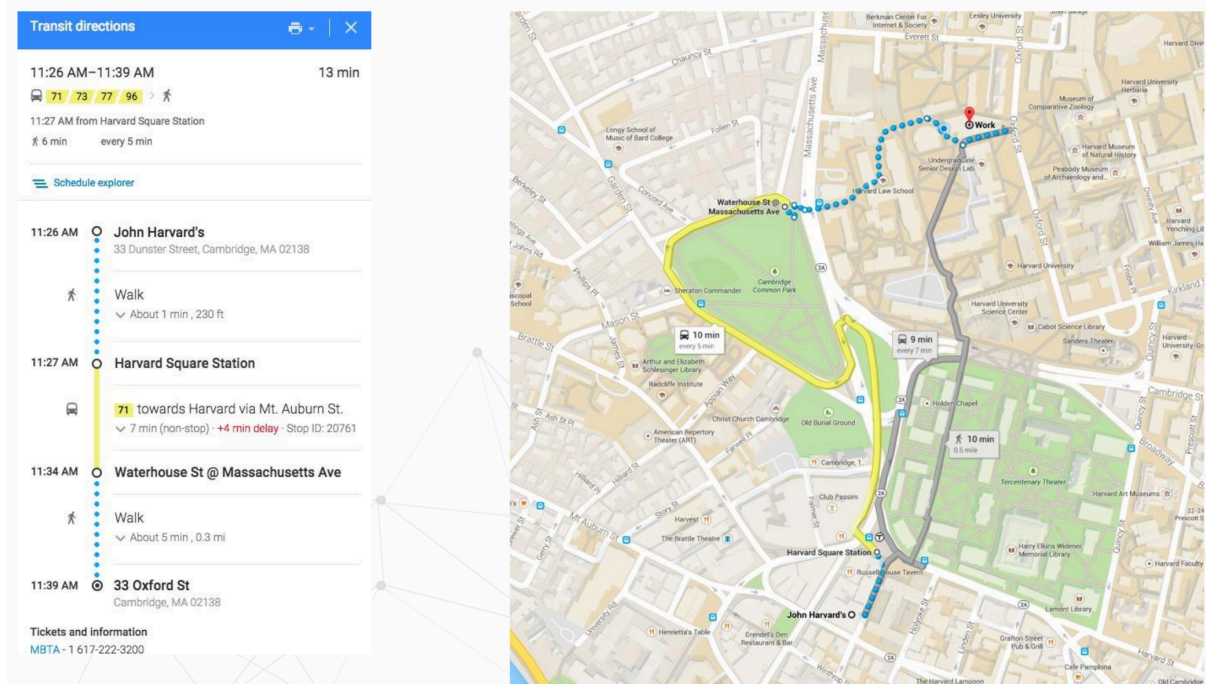


Figure 24: Abstract vs. specific navigation overlays

Mapping Software

OSM gives you access to all data, while Google Maps only allows you to create overlays.

Generally, there are two main tasks:

- Abstract maps so for example political context or continents and resources
- Overlays: when you want to display more local information, e.g. at council level, borough etc.

5.2. Choropleth Maps

Choropleth maps are a way to visualize aggregate data.

- Area are shaded or patterned in proportion to measurement
- Each spatial unit is filled with a uniform colour or pattern

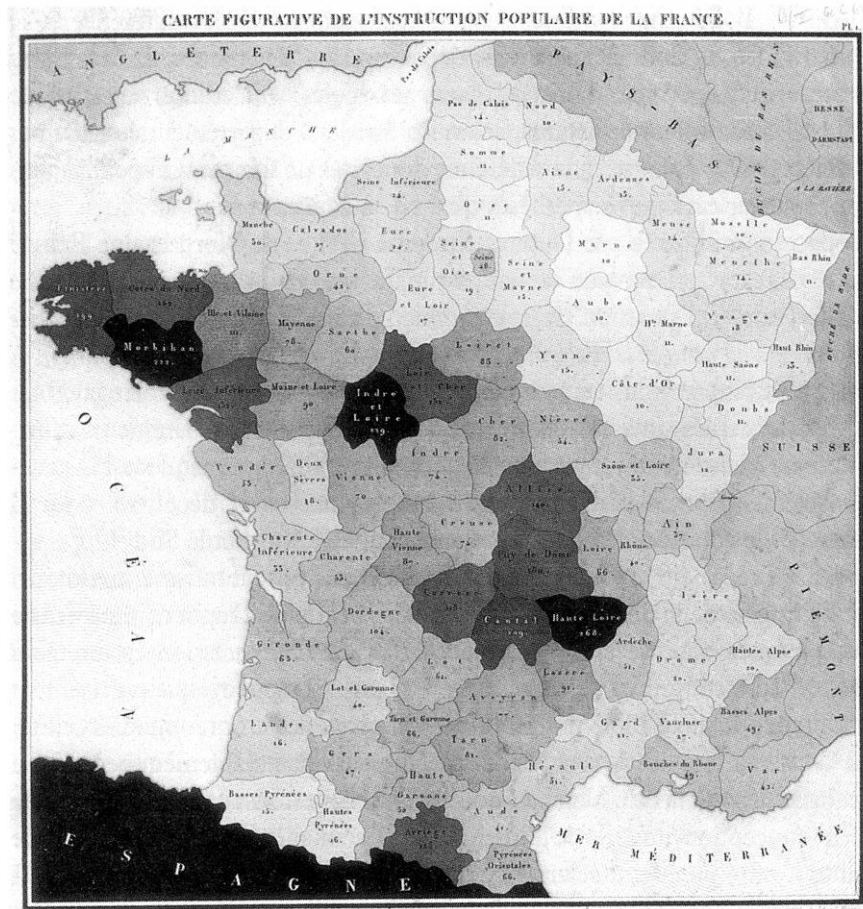


Figure 25: First known instance; Illiteracy in France (black to white distribution)

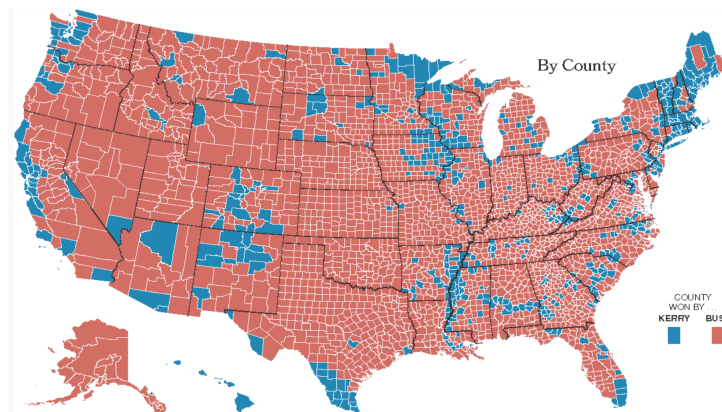
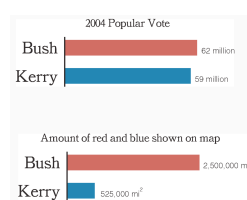


Figure 26: An example misleading map: Kerry vs Bush, 2004



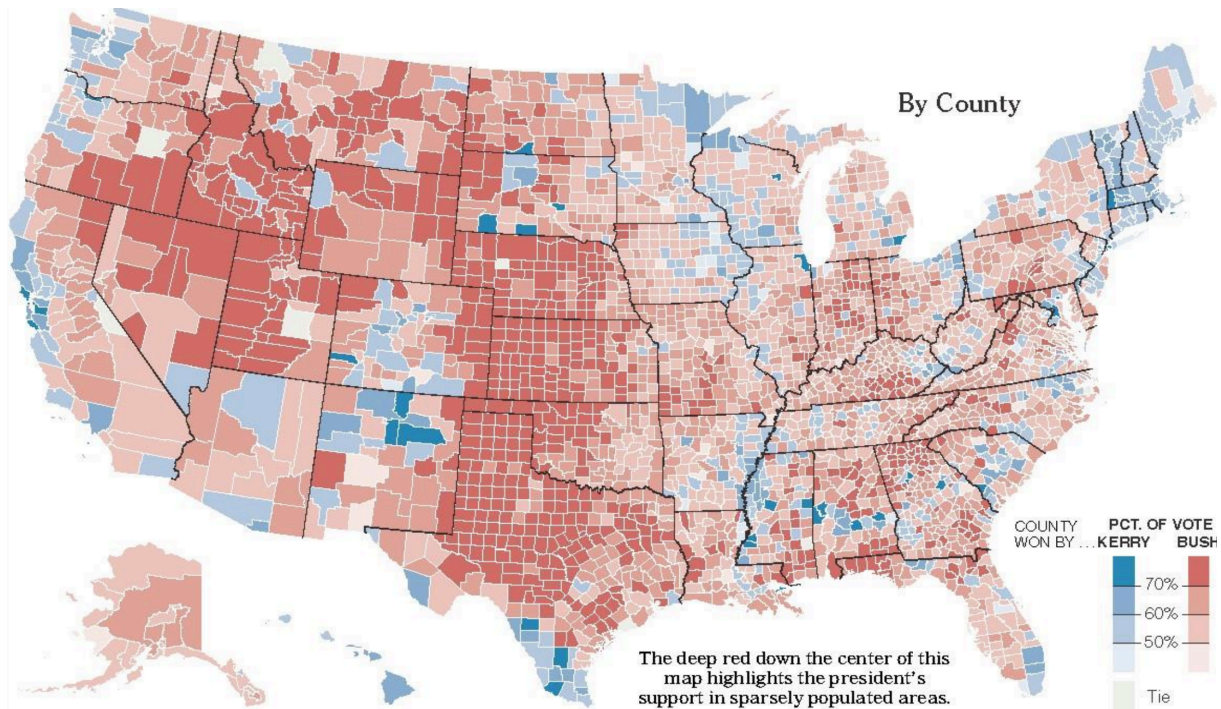


Figure 27: Fairer map with continuous shading

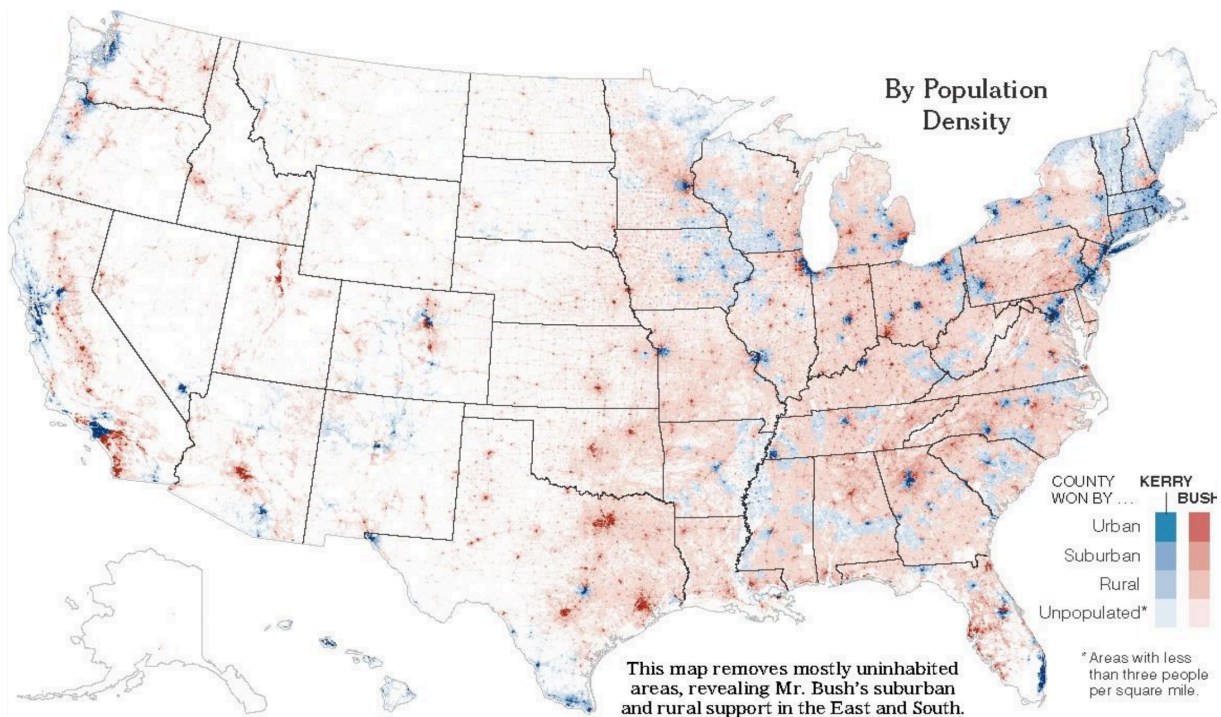


Figure 28: Further improved by population density; we cannot see who won a state however

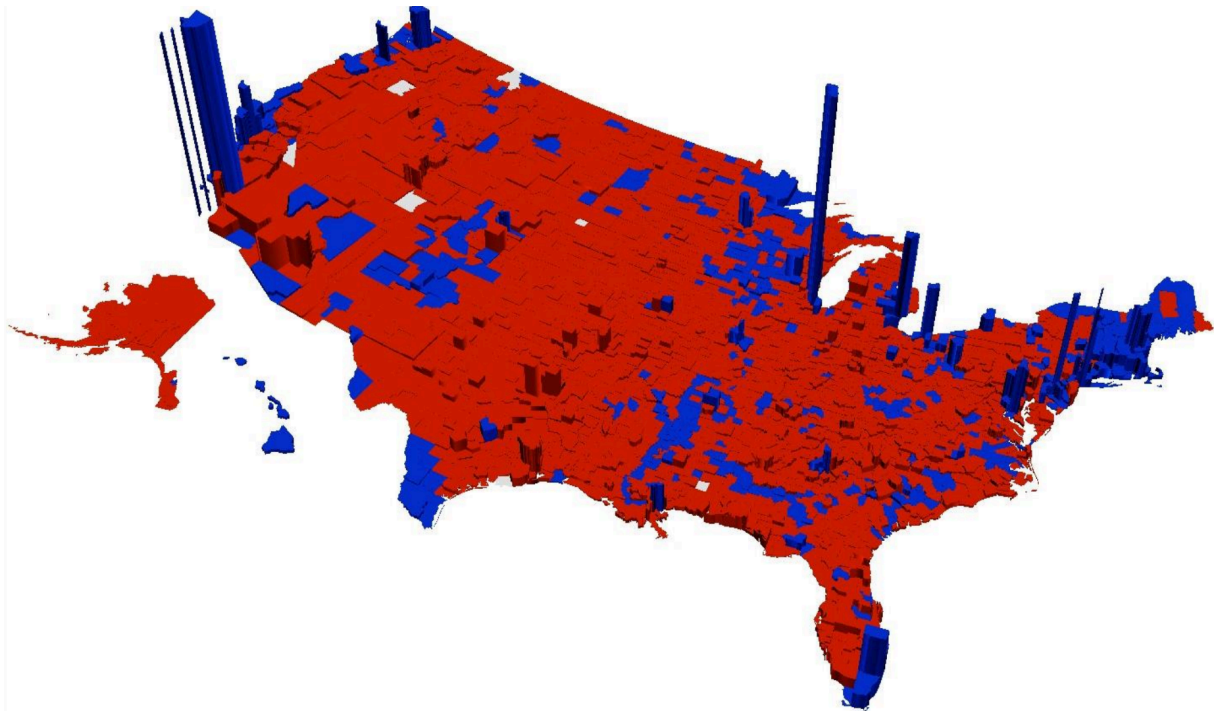


Figure 29: We can also visualise it in 3D!

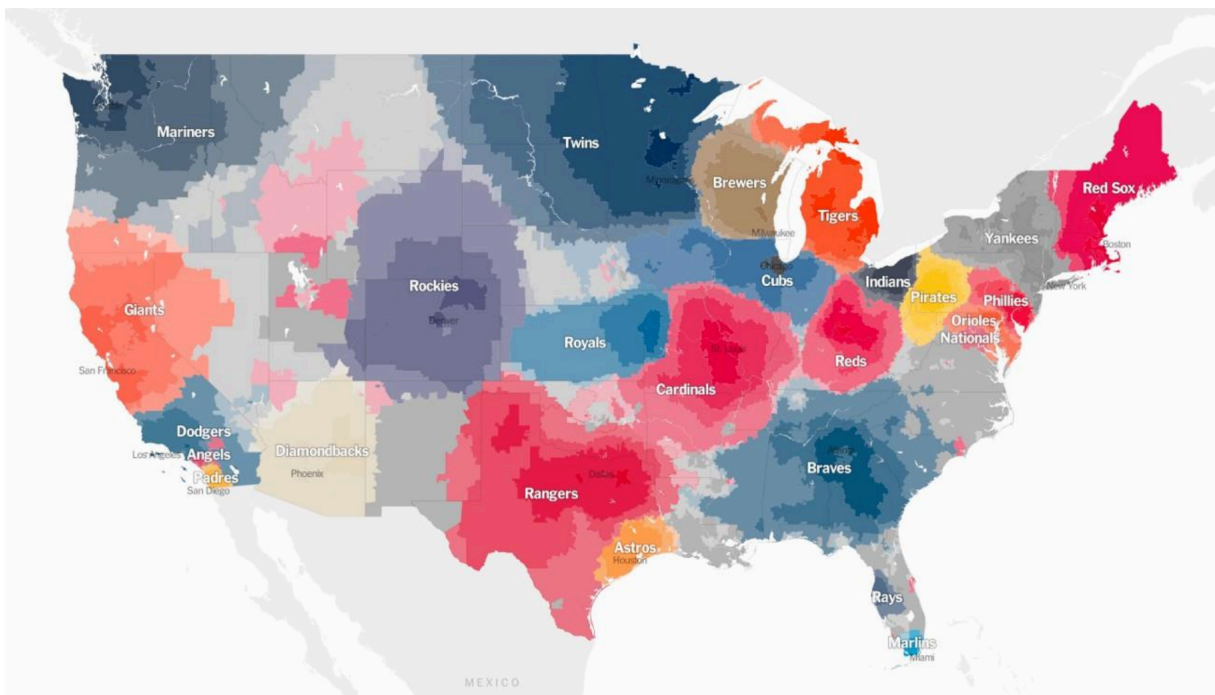


Figure 30: Choropleth can be done on top of regular maps⁷
Additional examples linked in footnotes^{8,9,10}

⁷<https://www.nytimes.com/interactive/2014/04/24/upshot/facebook-baseball-map.html>

⁸<https://www.nytimes.com/interactive/2014/05/12/upshot/basketball-map.html#5,33,547,-97,690>

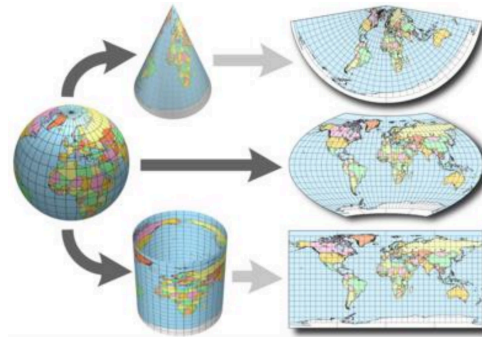
⁹<https://archive.nytimes.com/www.nytimes.com/interactive/2011/03/06/weekinreview/20110306-happiness.html>

¹⁰<https://www.dailymail.co.uk/sciencetech/article-3118671/Mapping-90-years-UFO-sightings-Fireballs-commonly-seen-egg-shaped-craft-rarer-Maine-hotspot.html>

5.3. Global Neighbourhood

For a **global neighbourhood**:

- Boundaries: global neighborhoods. World “wraps” at boundaries. Projection causes distortion.
- Neighbourhood: may or may not influence the emergence of patterns.
- Scale: data can be at different resolution.



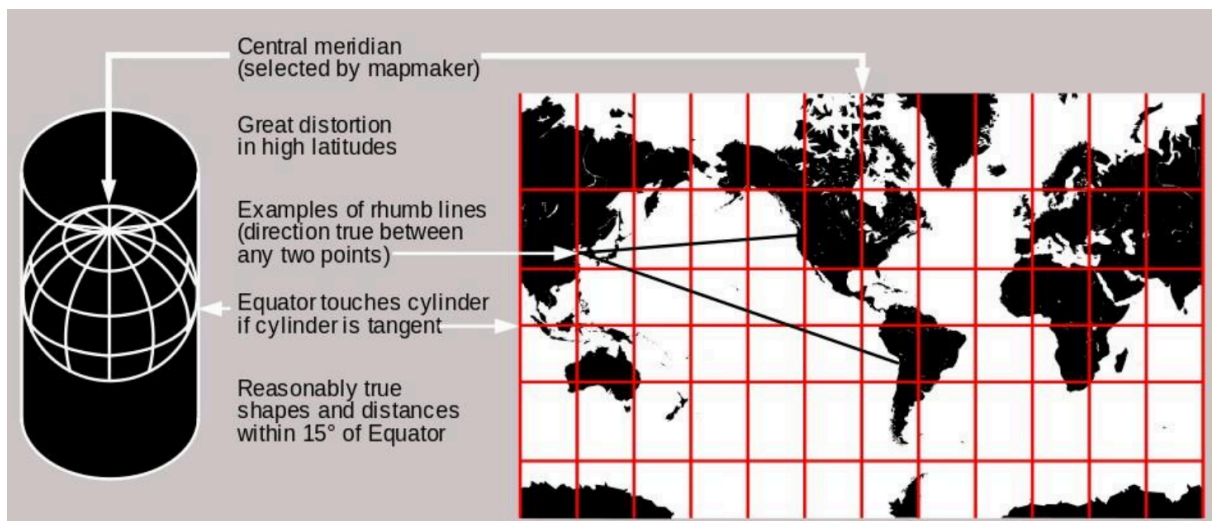
Earth is a (flattened) Sphere so we need to project or “unfold” the hull of the sphere to fit onto paper/screens.

Relevant attributes are Area, Shape, Direction, Bearing, Distance, Scale
Not all attributes can be preserved in projections.

Classes of Projections

- Azimuthal: Preserves direction
- Equal-area: Preserves area
- Conformal: Preserves local shapes, Preserves local angles

5.3.1. Mercator (Conformal Projection)



- Projection onto a cylinder wrapped around the globe
- Conformal map projection: angles are preserved
- All lines of constant bearing are straight lines.
- Constant bearing means constant compass heading.
- Developed for sailors.

Try <https://vega.github.io/vega/examples/world-map/>

“Mercator works really great if you’re, say, Ferdinand Magellan looking for a compass bearing that will take you around Cape Horn, because all of the latitude and longitude lines and angles in between lay out nice and straight on the map like we experience them in real life. It also works well if you’re Google and you want a map image that you can neatly slice up into little squares that your server sends to a customer’s browser. North is always up, your hometown doesn’t look squished or slanted when you zoom in to it, and everybody’s happy.”

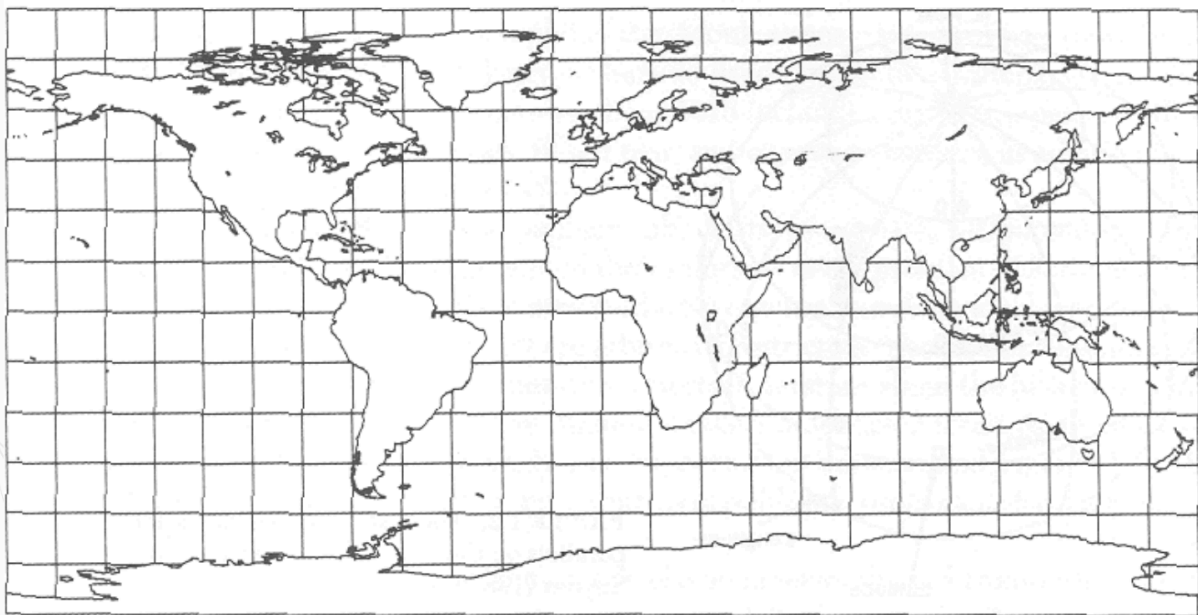
Problems with Mercator:

- Traditional map, used to teach geography
- Massive distortion of area distant from equator
- “unfair to the Global South, making places that are mostly trees, snow, and better-off people look huge, and the places where most of the world’s population lives look puny”

Constant bearing, decreasing range (CBDR) is a term in navigation which means that some object, usually another ship viewed from the deck or bridge of one’s own ship, is getting closer but maintaining the same relative bearing. If this continues, the objects will collide.

5.3.2. Latitude/Longitude (Conformal Projection)

Latitude and longitude are a convention commonly used, we can plot coordinates according to a lat/long reference system and convert to x and y system.

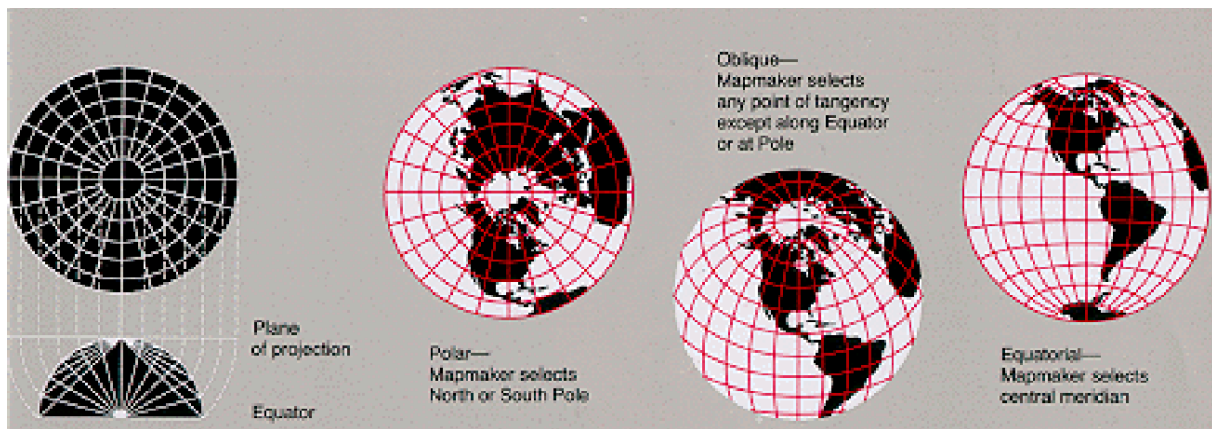


However this has some drawbacks:

- Angles are not preserved
- Areas are not preserved
- Deformations at both top and bottom

5.3.3. Azimuthal Projections

Azimuthal is projection onto a plane tangent to the Earth, present the whole Earth in a single map. Area is accurately represented in all regions of the sphere. Angles are correct around the centre point but not far away.



Constant radial scale (distances increase linearly from the centre of projection). (with exception of Lambert azimuthal projection)

Great circles through the centre are straight lines on a plane. Radii (away from the center point) correspond to true distances. Common in airline magazines centred around the hub.

5.3.3.1. Equidistant (Azimuthal Projection)

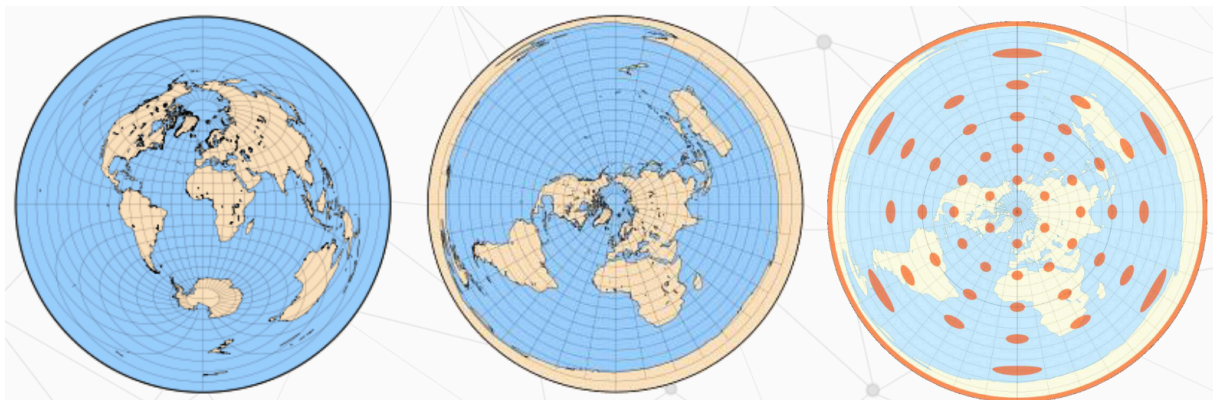


Figure 31: Azimuthal equidistant projection with Tissot's indicatrix of deformation

Useful properties:

- all points on the map are at proportionately correct distances from the center point,
- all points on the map are at the correct azimuth (direction) from the center point.

Useful application is polar projection where it shows all meridians (lines of longitude) as straight, with distances from the pole represented correctly.

The flag of the United Nations contains has a polar azimuthal equidistant projection.

5.3.3.2. Lambert equal-area (Azimuthal Projection)



Figure 32: Lambert azimuthal equal-area projection with Tissot's indicatrix of deformation

- Special mapping from a sphere to a disk (that is, a region bounded by a circle).
- Accurately represents area in all regions of the sphere.
- Does NOT accurately represent angles.
- Strongly distorts shapes in the boundary of a worldwide map (similar to azimuthal equidistant).
- Radial scale is NOT constant: in the polar aspect, parallels get closer together towards the border, just enough to preserve areas.

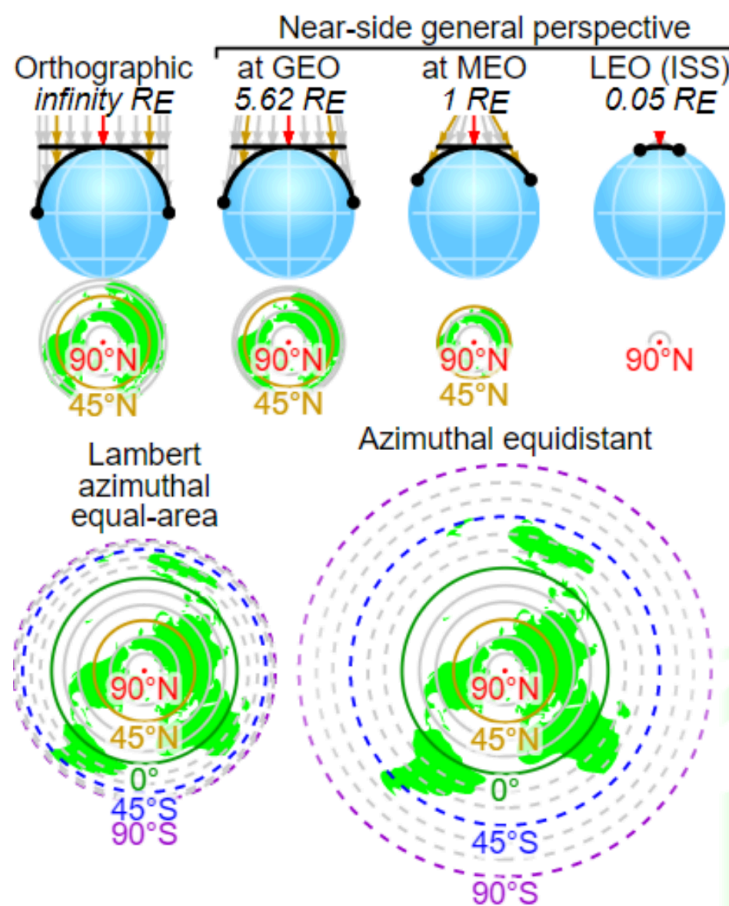
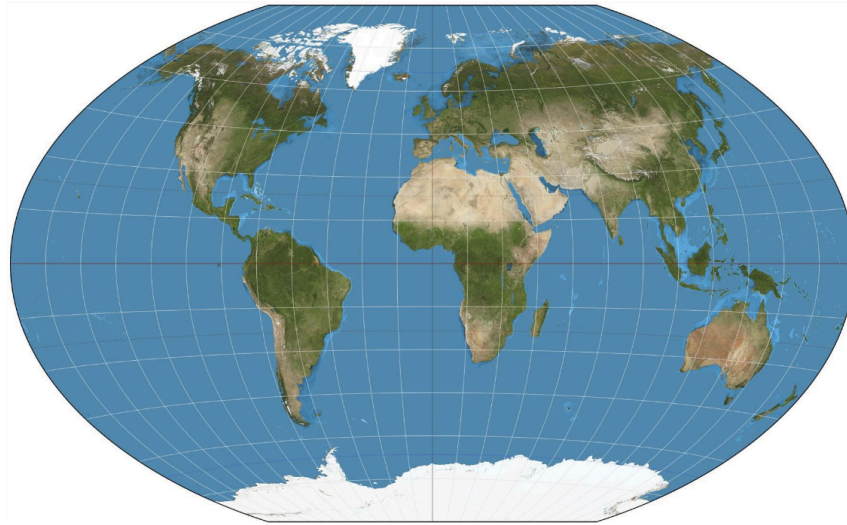


Figure 33: Compared with equidistant

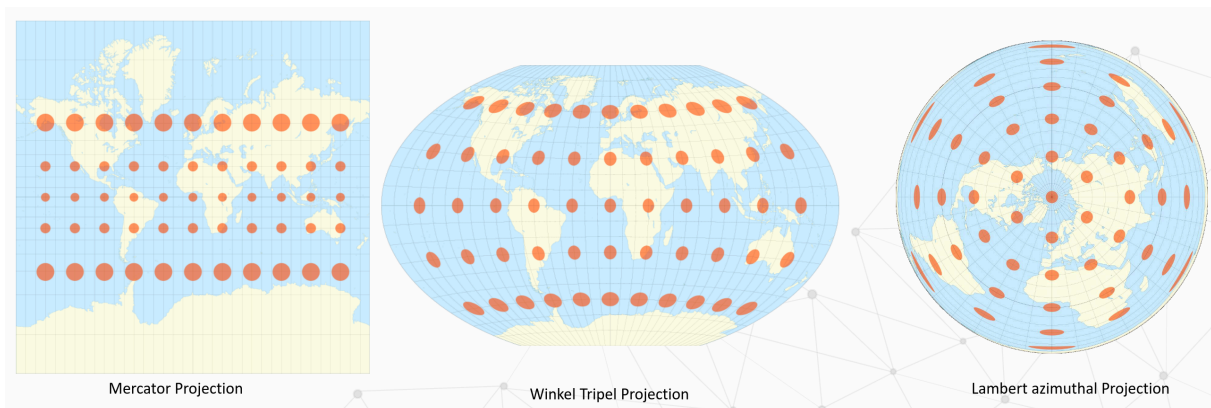
5.3.3.3. Winkel Tripel (Azimuthal Projection)



Modified azimuthal map projection

- Averaged to cylindrical projection
- Minimizes three kinds of distortion: Area, Direction, Distance
- Considered good projection for world maps, endorsed by National Geographic Society, used in Textbooks today.

5.3.4. Tissot Indicatrix

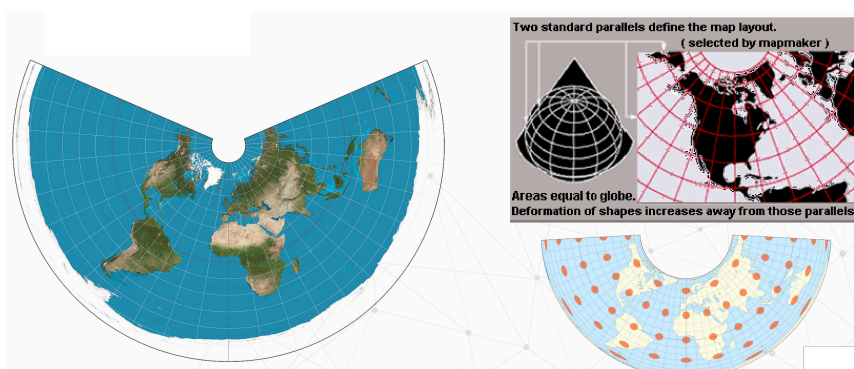


Single indicatrix describes the distortion at a single point.

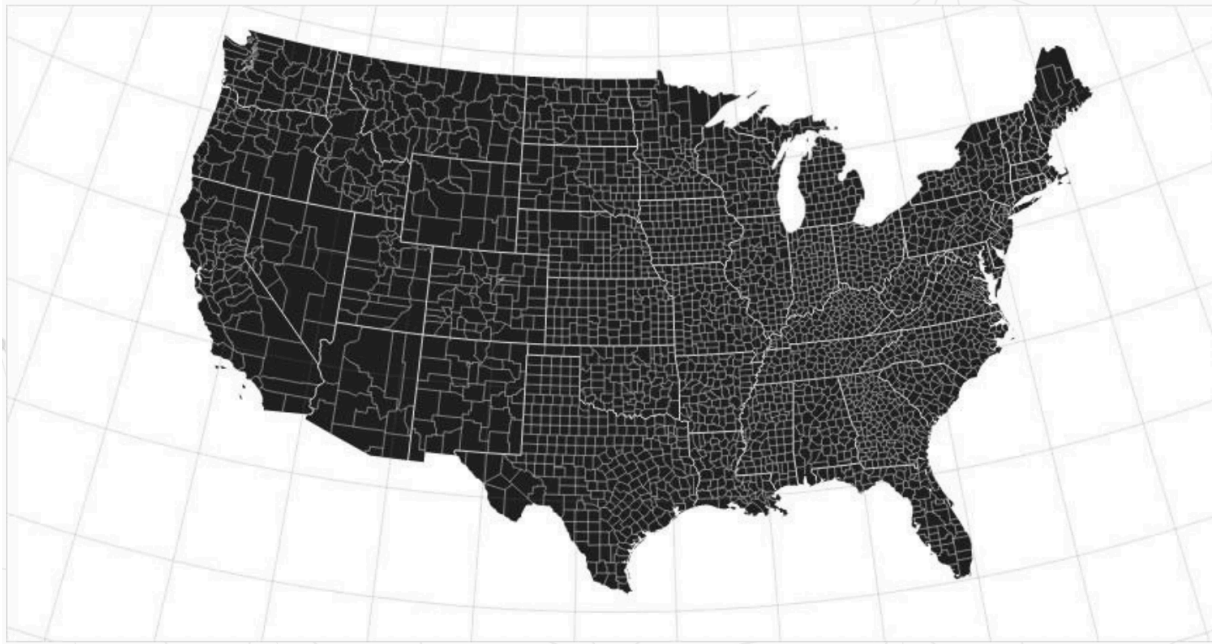
Overall, single indicatrix describes the distortion at a single point.

5.3.5. Conic Projections

Projection onto a cone that slices through the globe, intersecting the Earth twice.



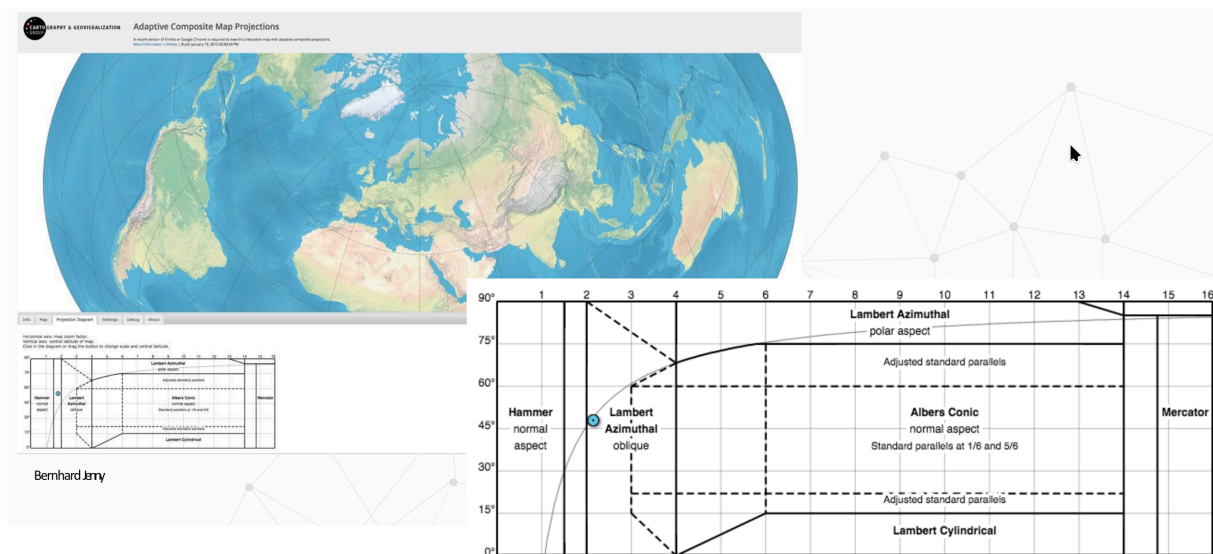
5.3.5.1. Albers - Equal-Area (Conic Projection)



- Shows areas correctly
- Distorts distances and shapes
- Uses two standard parallels to reduce some of the distortion .
- Shape and linear scale are not preserved however ...
 - The distortion of these properties is minimized in the region between the standard parallels.
- Best suited for land masses extending in an east-to-west orientation rather than those lying north to south. It is used by the USGS for maps showing the conterminous United States (48 states) or large areas of the United States.

5.3.6. Misc. on Projections

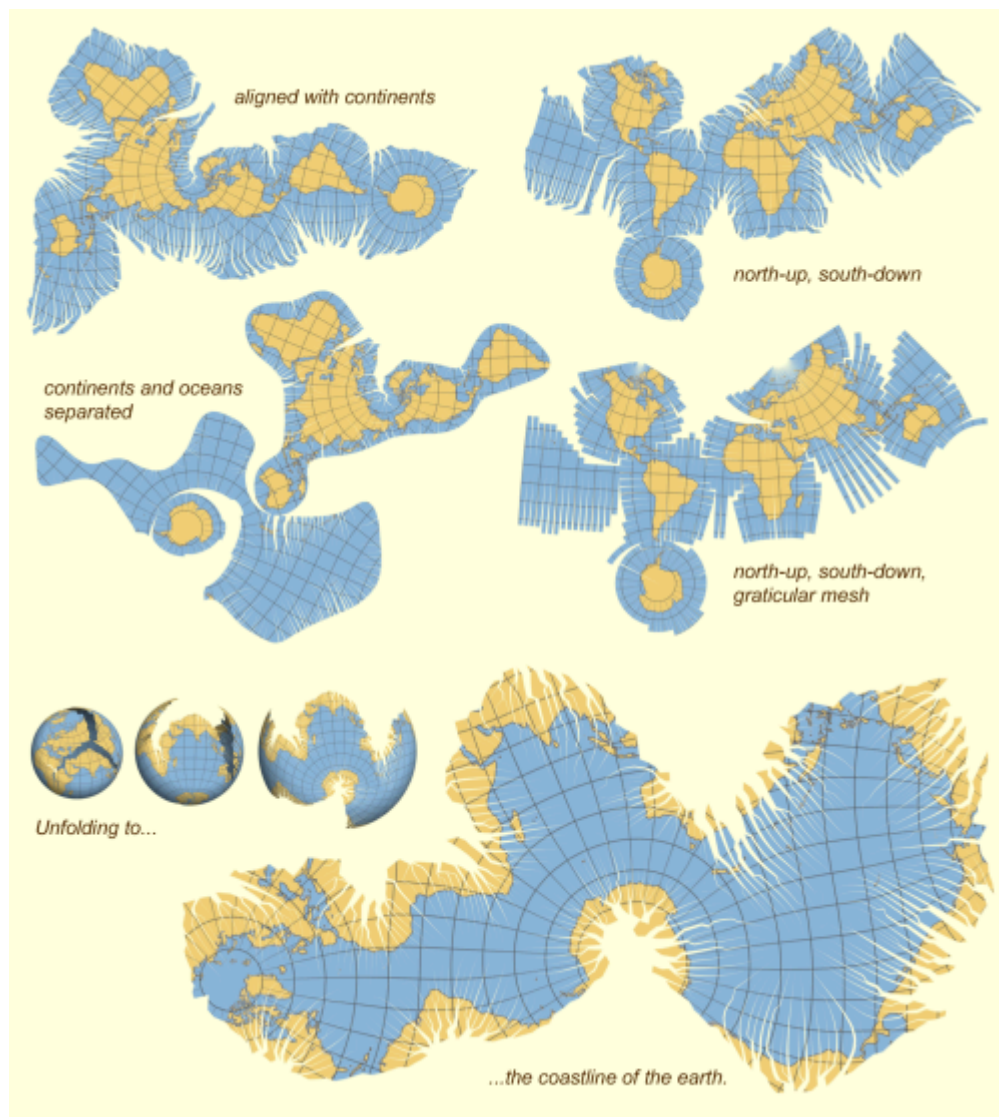
- Can use distortion in shapes.
- Can create composite projections.
- Can use D3 to create these projections.
- As well as Python: matplotlib, cartopy



5.3.7. Unfolding the Earth

We can also fundamentally avoid projection and tear up the surface.

We can use small patches and flatten them out.



You can preserve land area or coast lines.

Check out for more: <https://vanwijk.win.tue.nl/myriahedral/>

5.4. Contour (Isopleth) Maps

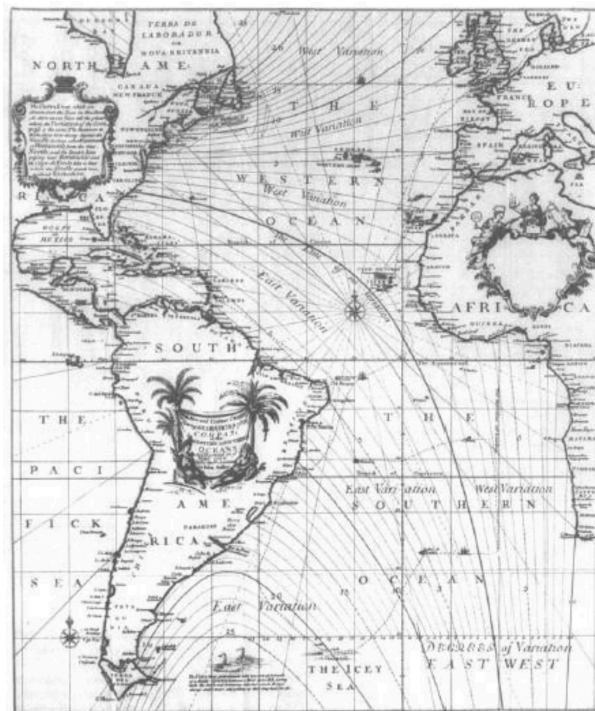


Figure 34: Very early map of contours, Magnetic declination used for navigation

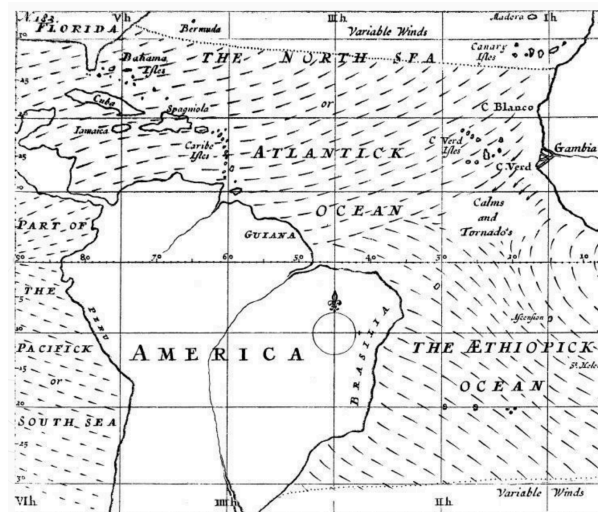


Figure 35: Early Weather Map, Halley's wind map, 1686

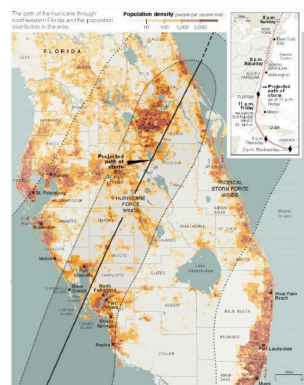


Figure 36: Projected path of hurricane in the center and areas affected, Contours show possible paths

5.4.1. Line Integral Convolution (LIC)

In scientific visualization, line integral convolution (LIC) is a technique to visualize a vector field, like a fluid motion, such as the wind movement in a tornado. LIC has been proposed by Brian Cabral and Leith Leedom. Compared to other integration-based techniques that compute field lines of the input vector field, LIC has the advantage that all structural features of the vector field are displayed, without the need to adapt the start and end points of field lines to the specific vector field. LIC is a method from the texture advection family.

Intuitively, the flow of a vector field in some domain is visualized by adding a static random pattern of dark and light paint sources. As the flow passes by the sources, each parcel of fluid picks up some of the source color, averaging it with the color it has already acquired in a manner similar to throwing paint in a river. The result is a random striped texture where points along the same streamline tend to have similar color.

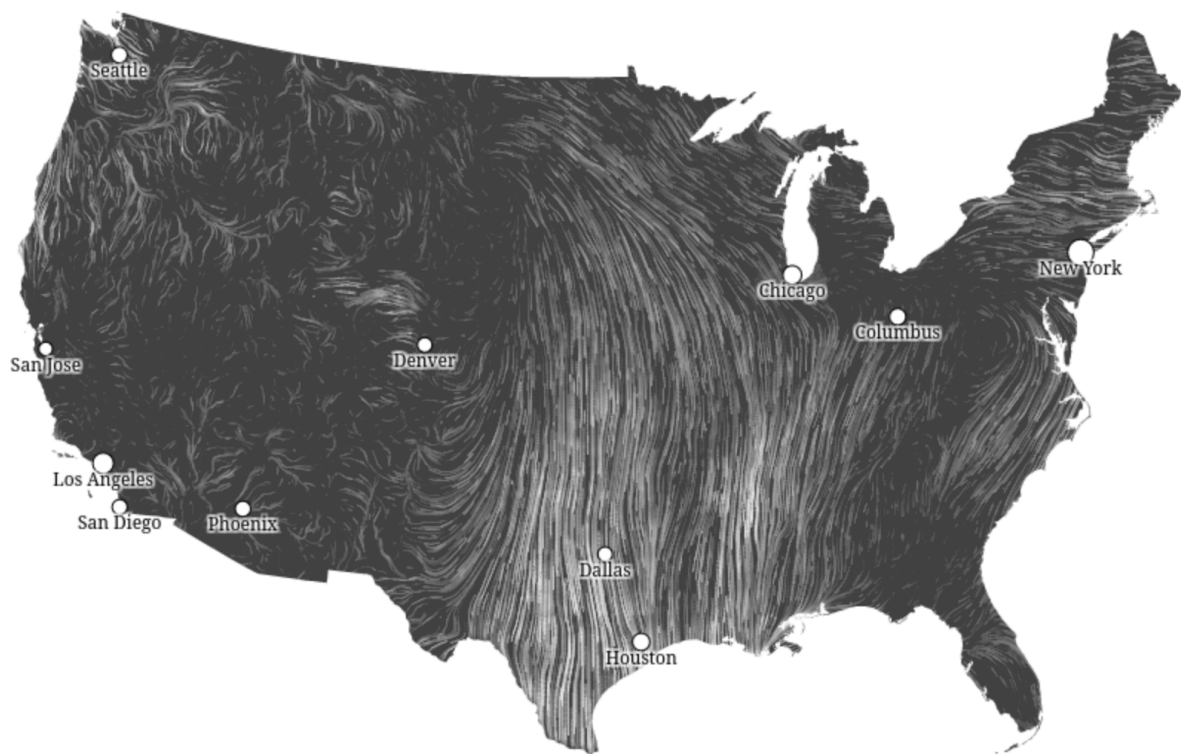


Figure 37: LIC wind map¹¹

Algorithm

- Generate a random gray level image.
- For every pixel in this image, the forward/backward streamline of a fixed arc length is calculated.
- The value assigned to the current pixel is computed by a convolution of a suitable convolution kernel with the gray levels of all the pixels lying on a segment of this streamline.

This creates a gray level LIC image.

¹¹<http://hint.fm/wind/>

5.5. Cartograms

Cartograms compromise between showing relative position and data — by scaling distance by data.

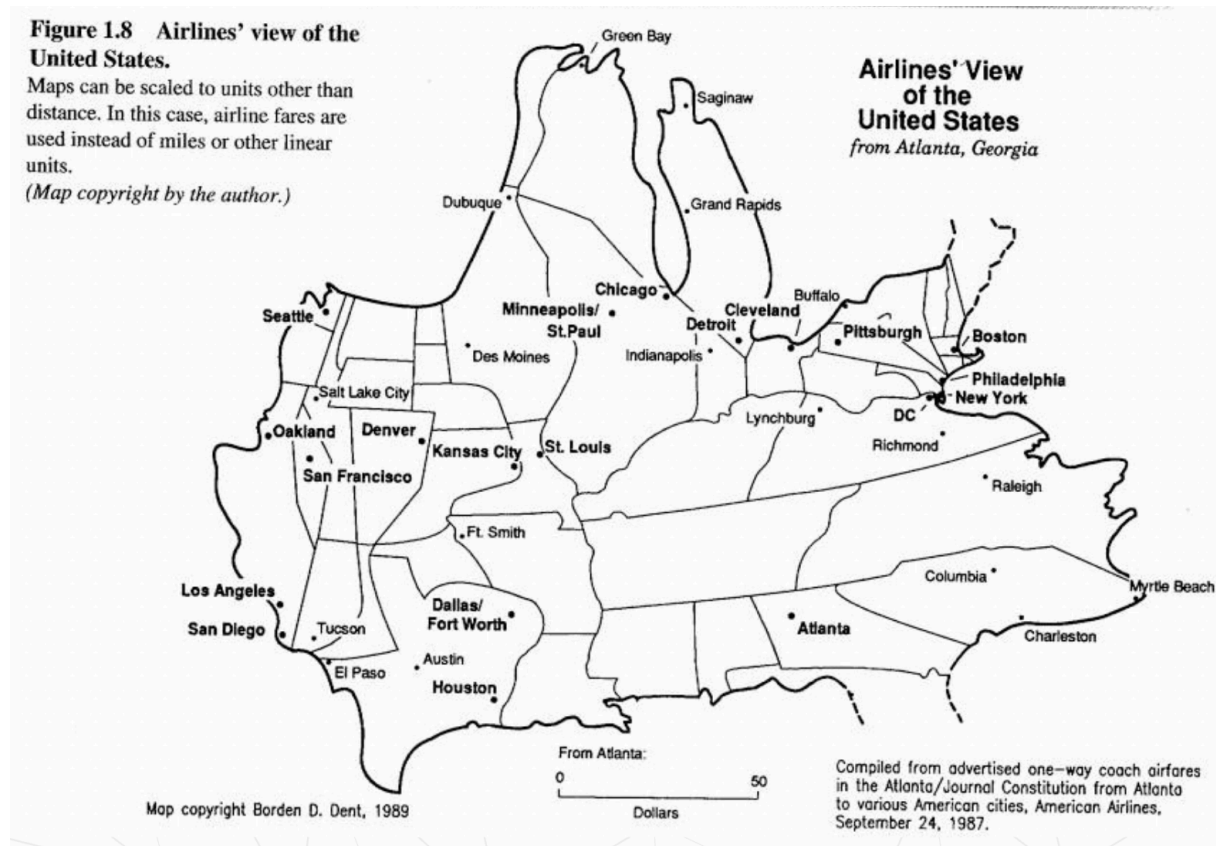
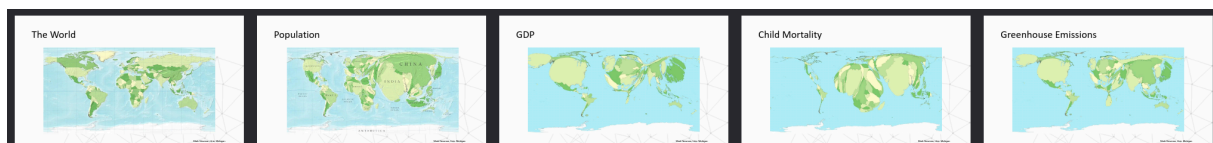
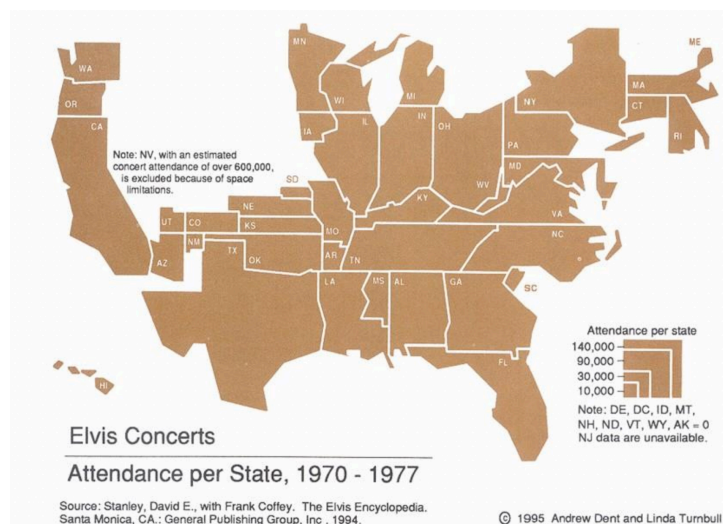


Figure 38: Cost of journey instead of actual distance



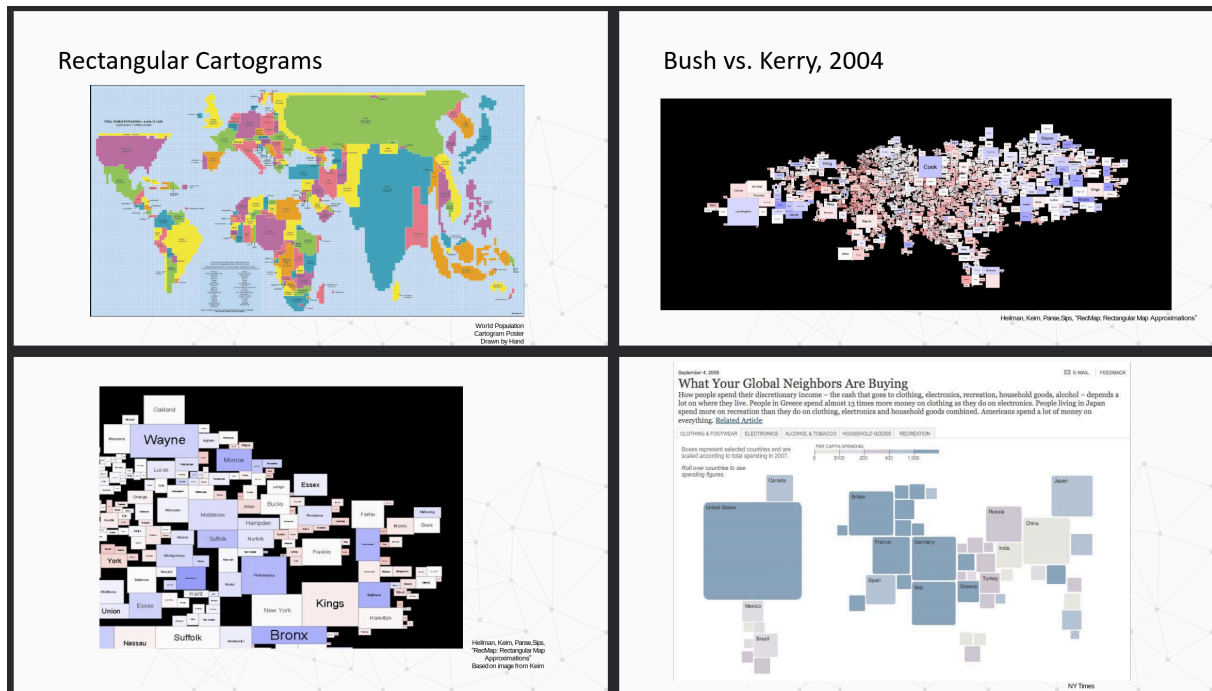
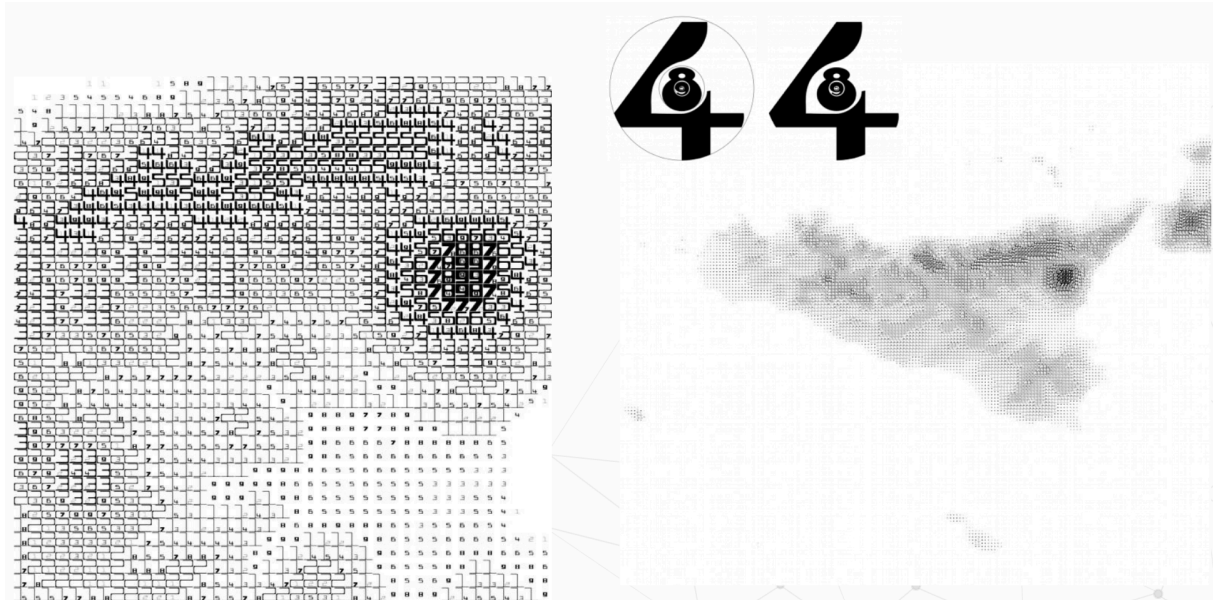


Figure 41: Rectangular Cartograms

Fatfonts¹² are designed so that the amount of dark pixels in a numeral character is proportional to the number it represents. For example, “2” has twice the ink than “1”, “8” has two times the amount of dark ink than “4” etc. You can see this easily in the set of characters below: This proportionality of ink is the main property of FatFonts. It allows us to create images of data where you can read the numbers, and represent tables that can be read as images (like the example below)



Data-Driven Maps

- <https://benfry.com/zipdecode/>
- Census Dotmap
- <https://eagereyes.org/zipscribble-maps/united-states>
- <http://realtime.waag.org/>
- https://www.youtube.com/watch?v=ha0PO_ONPQ

Thematic Maps

Maps that are hand drawn and annotated, this is where we move more towards infographics.

- <https://xkcd.com/256/>
- <https://xkcd.com/802/>
- https://www.flickr.com/photos/the_bumblebee/2229041742

¹²<http://fatfonts.org/>

6. Spatial Data - Cluster Analysis

Consider the following information:

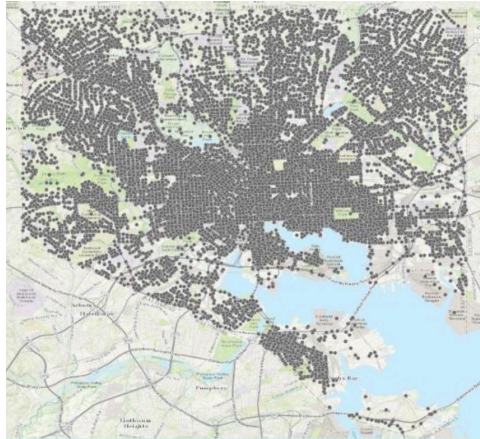
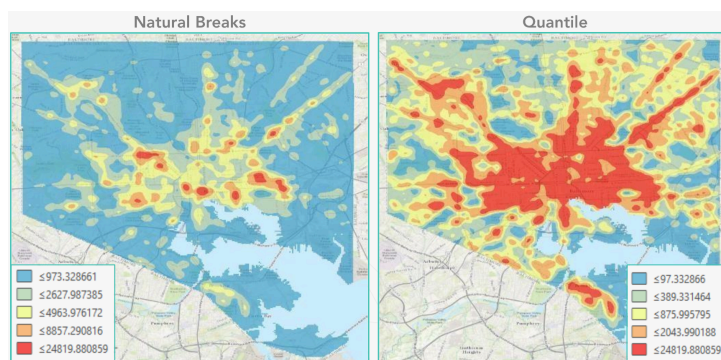
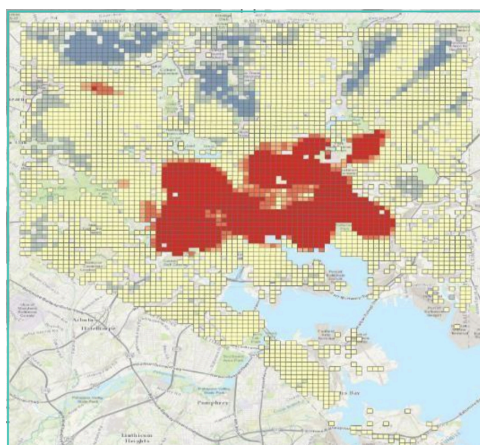


Figure 42: High Priority 911 Calls in Baltimore

There are subjective ways in which we can analyse the visual patterns:
We may want to ask questions about hot spots and variation.



One way to minimise subjectivity is to turn a map into an information product that is as faithful as possible: This type in particular is **hotspot analysis**.



The red areas on this map are a set of **statistically significant hotspots**.

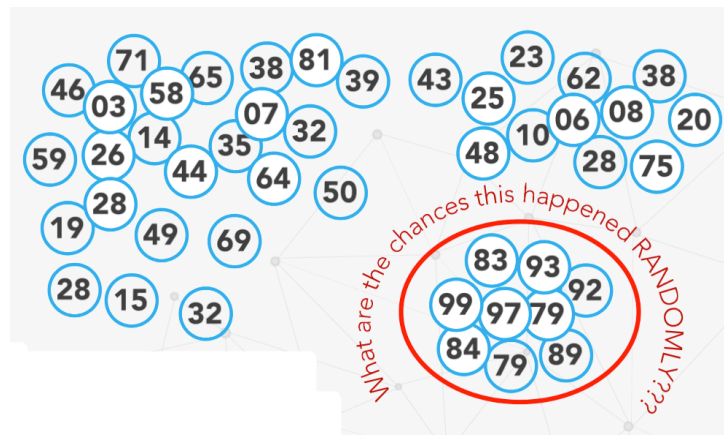
There is more clustering than we would expect from higher values, and similarly blue areas are where there is less than we would expect. This is all computed through statistics.

6.1. Inferential Statistics

Null hypothesis is the probability data pattern that we see as a result of a random process.

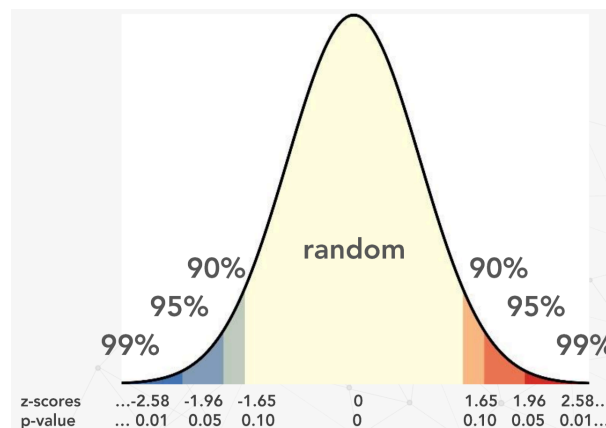


So we question how likely it is that we would get such a pattern based on random chance:



So if we saw our spread of numbers to start appearing to create clusters which are presumed to have a meaning, then what is the probability that this happened by chance?

p-values are themselves just probabilities; so a *p*-value of 0.01 represents a 1% chance that you are looking at the facts and that is the result of a random special process. (i.e. 99% confidence that clustering is statistically significant and meaningful)



z-scores correspond to colours in the graph given above.

Positive values tell us we have a hotspot, negative values means we have a cold spot.

The **First Law of Geography** states that everything is related to everything else.

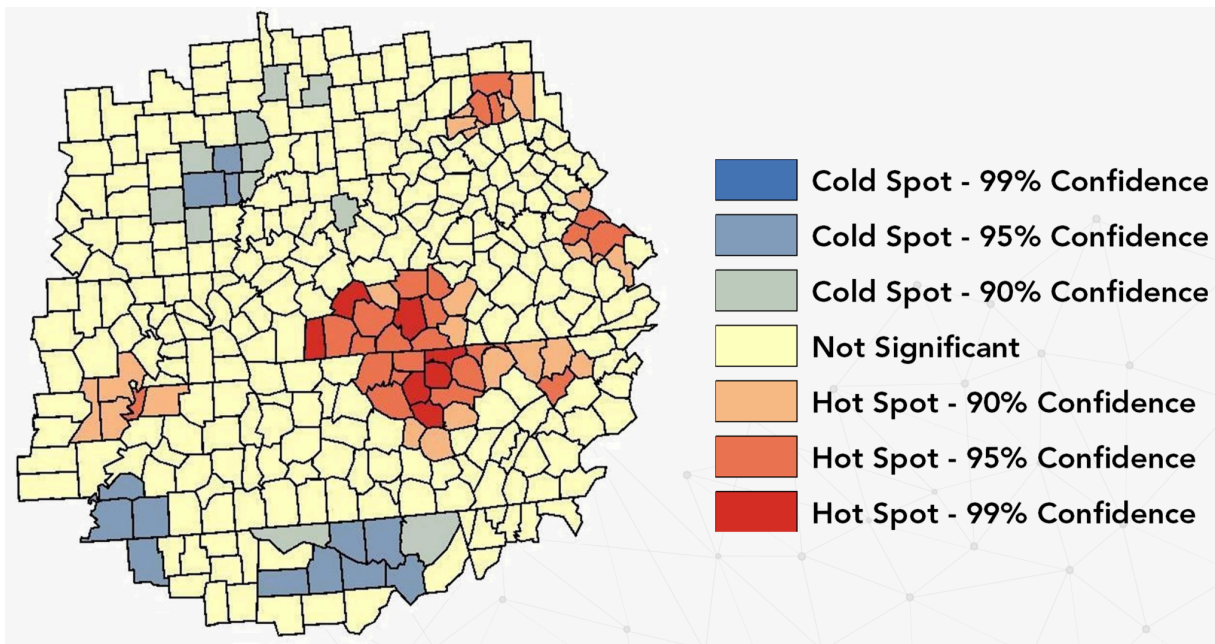
Hot spot analysis is probably the most popular spatial statistical tool, given a set of weighted features it identifies statistically significant hot spots and cold spots. It is also important to differentiate it from heatmaps.

6.2. Hotspot Analysis



Each feature has a value.

We consider how different the neighbourhood is from the study area, if we find a feature which is significantly higher, it is marked as a hot spot!



There are many different grid systems we can do:

- **Tessellation:** cover a surface in a pattern of flat shapes so there are no overlaps or gaps

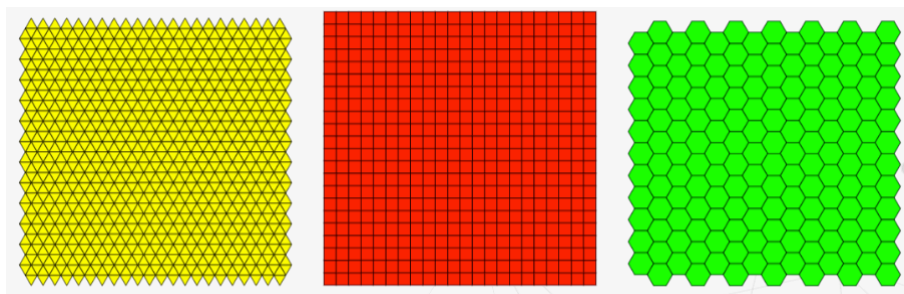


Figure 48: Regular grids

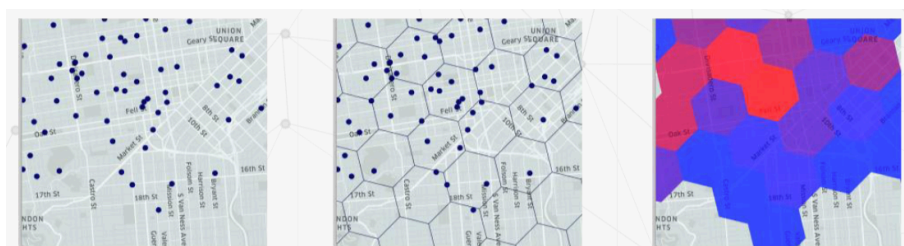


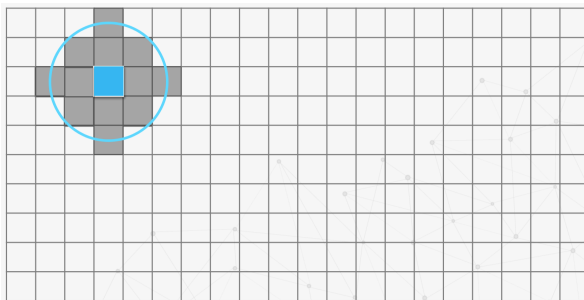
Figure 49: Example Hex/Honeycomb Grid

Additional points of interest include:

“where are the hot spots?” is not necessarily the same as “where are the highest values?”

How are neighbourhood sizes determined?

- Simplest way is to give them a fixed size/define neighbour distance from cell of reference for feat.



- **Spatial Autocorrelation** measures presence of systematic spatial variation in a mapped variable where adjacent observations have similar data values the map shows positive spatial autocorrelation and where adjacent observations tend to have very contrasting values then the map shows negative spatial autocorrelation
 - Useful to indicate that there is something of interest in the distribution of map values that calls for further investigation
 - the presence of spatial autocorrelation implies information redundancy and has important implications for the methodology of spatial data analysis
 - Common technique for detecting its presence: Moran's I test

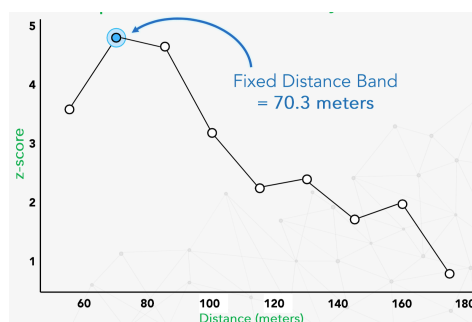
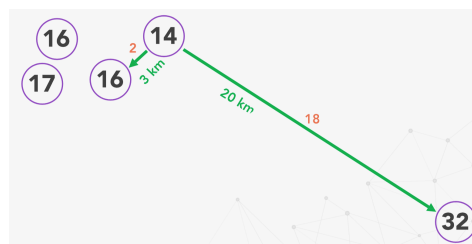
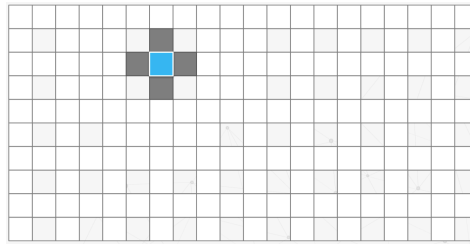


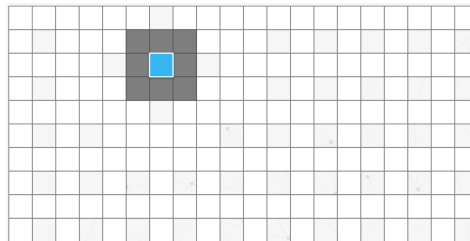
Figure 50: Spatial Autocorrelation by Distance



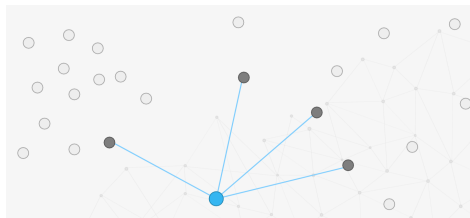
- **Contiguity Edges (Rooks case):** anything with an edge touching is a neighbour



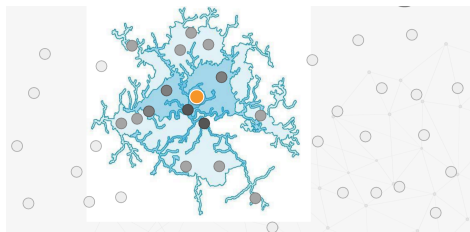
- **Queens case:** anything with an edge or corner touching is a neighbour



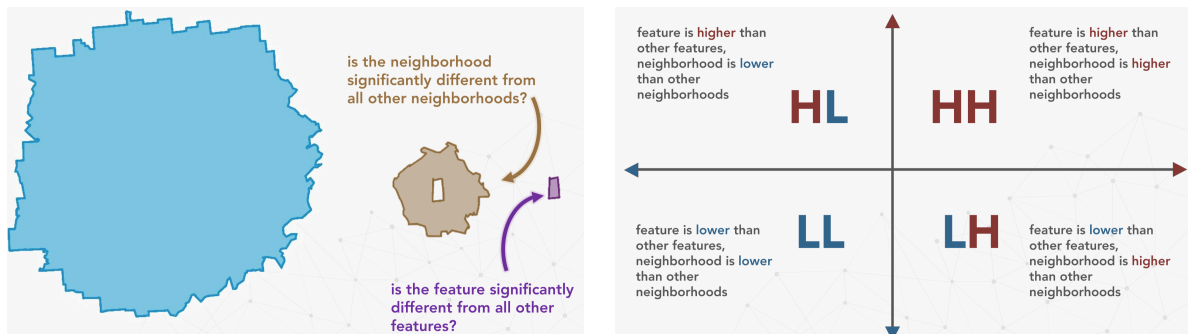
- **K-nearest neighbour**

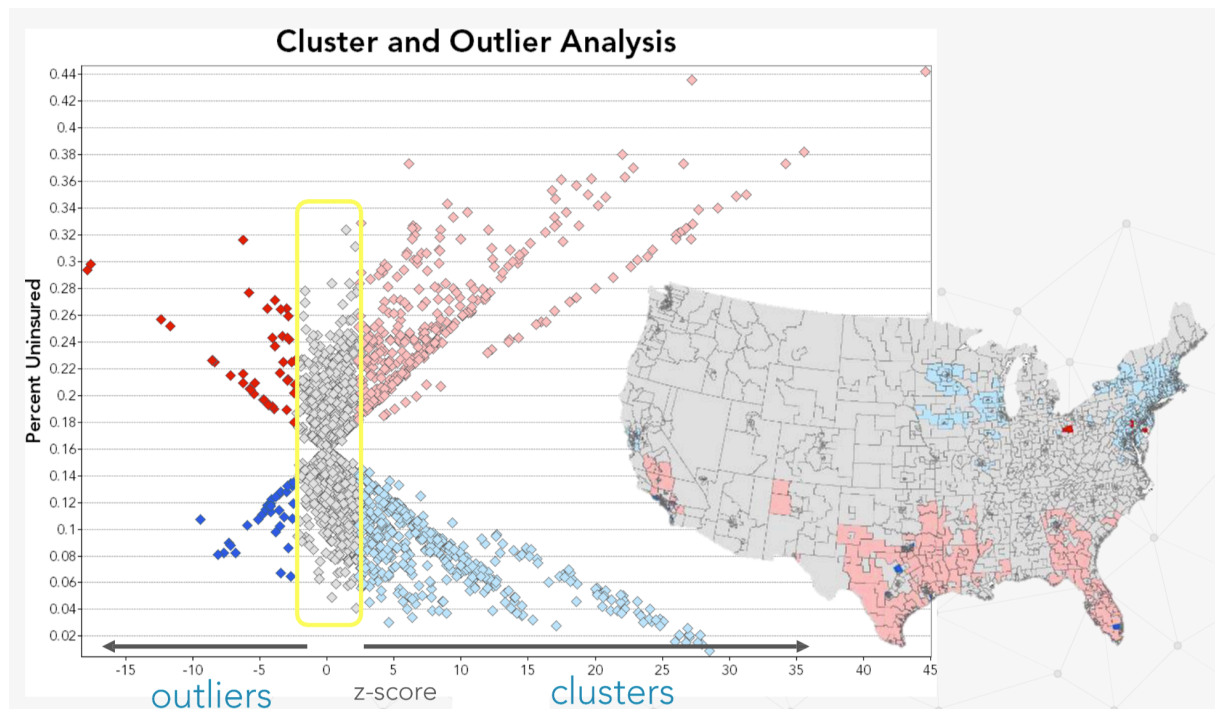


- **Network spatial weights**



- **Cluster and Outlier Analysis:** given a set of weighted features we can use it to identify statistically significant hot spots, cold spots, and spatial outliers
One method is Anselin Local Moran's I statistics





7. Spatial and Temporal Data - Visualisation

Time is often approached as simply ‘another dimension’, it has a direction and only one-dimensional, however it can be cyclic (24h day cycle, seasons). Cyclic behaviour has an influence on which samples are more significant and correlated. For example, the state of the system a year ago may be more relevant in predicting the upcoming change, than the state two months ago.

Temporal data model - choice of how to represent time e.g., time is continuous but needs to be discretized to be used ‘digitally’

Temporal Scale - scale in space refers to the extent of the area while scale in time refers to the duration in time; a temporal scale is used to measure the change in a variable over time. Different phenomena may be measured using different scales

Example, the change in temperature as late spring turns into summer might be measured in “degrees per day” while the changes in temperature from global warming might be measured in “degrees per year.”

7.1. Temporal Data Model

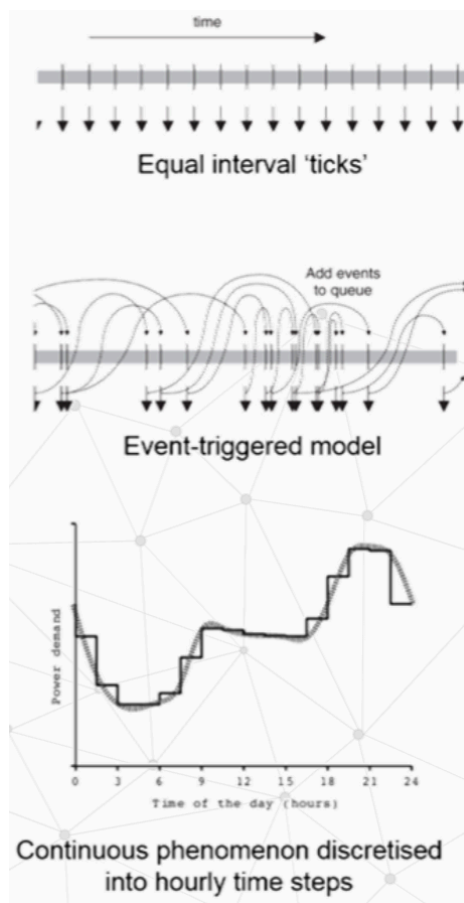
Different choices available to represent time

- Can be imposed by data itself
- May need to be chosen by us

First step is to start by looking at the temporal nature of the phenomena:

- **Continuous temporal phenomena:** describe an ongoing processing (e.g. temp. change/tree growth)

The continuous process can be broken down into small enough time steps, to sequentially represent the dynamic behaviour of the process.



- **Discrete temporal phenomena:** (also referred to as ‘events’); events can happen regularly (e.g. each morning the sun rises thus triggers flowers opening, birds become active, etc); events can be regular or irregularly paced (e.g. natural catastrophes)
Irregularity of events can be addressed by choosing a fine resolution of discrete steps, where there is only a certain probability that an event happens.

7.2. Temporal Scale

Temporal scale can also be an indicator of the data resolution through the duration of a time step that is: the smallest detectable event or property at a particular scale.

Self-similarity is common aspect in the temporal domain.

The cyclic nature of time (day-night oscillations, seasonal changes), leads to strongly fractal properties of temporal phenomena. This property allows us to decompose spatio-temporal patterns into multiple scales → very useful in temporal data analysis.

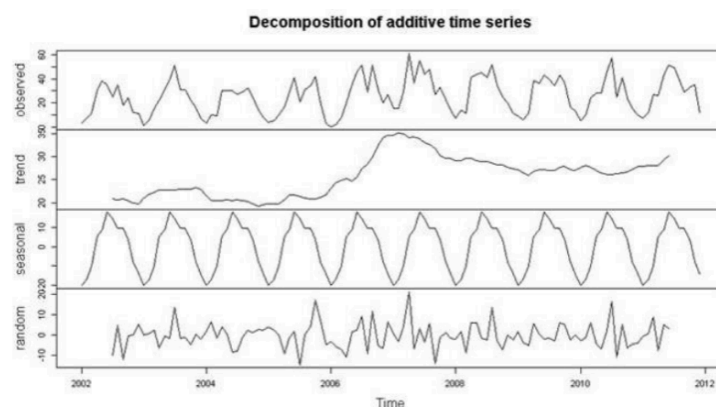


Figure 58: Decomposition of bicycle accidents in the city of Salzburg over 12 years (top row)

Two temporal scales:

- 12-year trend (2nd row)
- A repetitive seasonal pattern (3rd row)
- Bottom row shows a random noise as remainder

7.3. Visualizing Temporal Phenomena

Temporal data or time series data refers to data with time attributes that change over time. They may or may not have a spatial component (financial data, power consumption of a household).

There are three main ways to characterize time attributes:

- Linear time and cycle time: linear time means that the time from the past to the future is linear. The cycle time refers to the cycle of the time point domain, such as the change of seasons.
- Time points and time intervals: the time point corresponds to a discrete time point, and the time interval is a small linear time domain.
- Sequential time, branch time and multi-angle time: sequential time refers to events occurring in chronological order, and branch time refers to multiple time branches, such as different main lines in the novel. Multi-angle time can describe different points of view of a fact.

Advantages:

- The size of each individual stream shape is proportional to the values in each category.
- The axis that a Stream Graph flows parallel to, is used for the timescale.
- Colour can be used to:
 - distinguish each category or
 - Use shade variation to plot each category's additional quantitative values.

Stream Graphs are ideal for displaying high-volume datasets, in order to discover trends and patterns over time across a wide range of categories.

Disadvantages:

- Suffer from legibility issues
- Categories with smaller values are often drowned out to make way for categories with much larger values, making it impossible to see all the data
- Impossible to read the exact values visualised in a Stream Graph, as there is no axis to use as a reference.

Hint: Stream Graphs are better for giving a more general view of the data. They also tend to work significantly better as an interactive piece rather than a static or printed graphics.

7.3.1. Changing Axis: Radial Layouts

Radial Layouts: items are distributed around a circle taking advantage of the angle channel.

Coordinate system uses Polar coordinates.

There are several configurations, simple examples include:

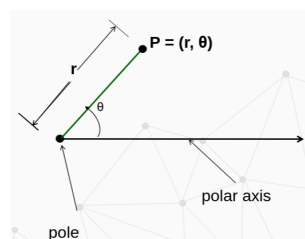
- Radial bar chart: radial axes meet at central ring, line mark
- Star plot: radial axes, meet at central point, line mark
- Accuracy: length unaligned with radial, less accurate than aligned with rectilinear

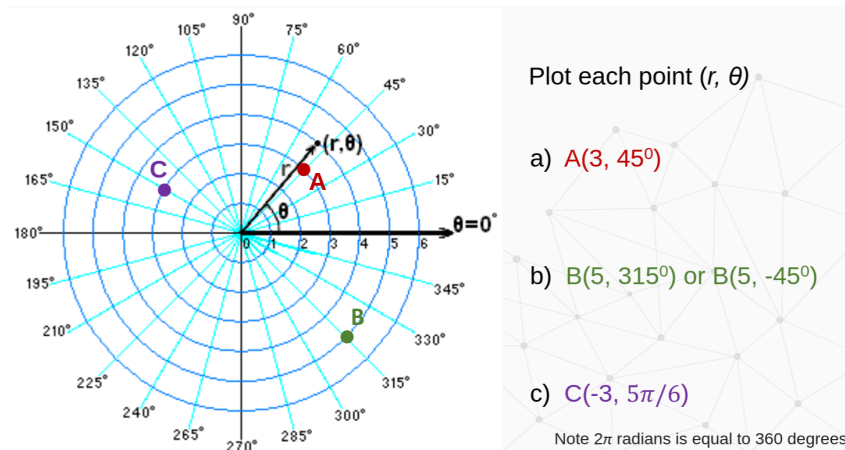


The foundation of the polar coordinate system is a horizontal ray that extends to the right.

This ray is called the polar axis. The endpoint of the polar axis is called the pole.

A point P in the polar coordinate system is designated by an ordered pair of numbers (r, θ) .



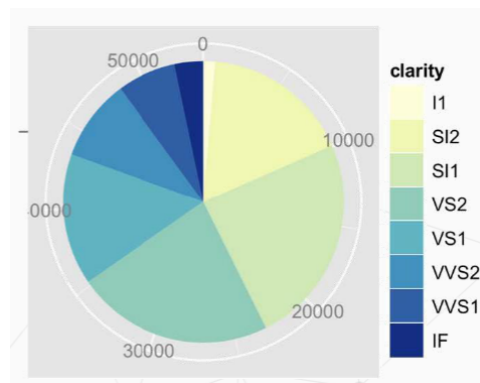


To convert the polar coordinates (r, θ) of a point to rectangular coordinates (x, y) , use the equations:

$$x = r \cos \theta \quad y = r \sin \theta$$

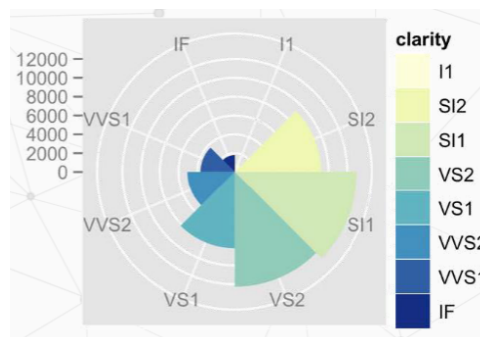
Pie chart

- Area marks, what varies is the angle (channel)
- Accuracy: angle/area less accurate than line length
- Arc length also less accurate than line length



Polar area chart

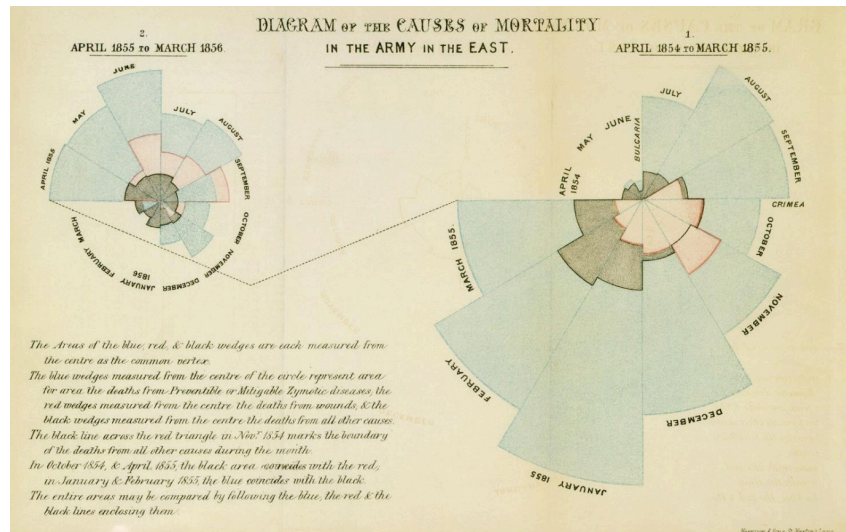
- Area marks, what varies is the length of a wedge (length is channel)
- More direct analog to bar charts



- Data has 1 categorical key attribute, 1 quantitative value attribute
- Task is part-to-whole judgements
- Scales to about 12 categories

Coxcomb Chart (F. Nightingale)

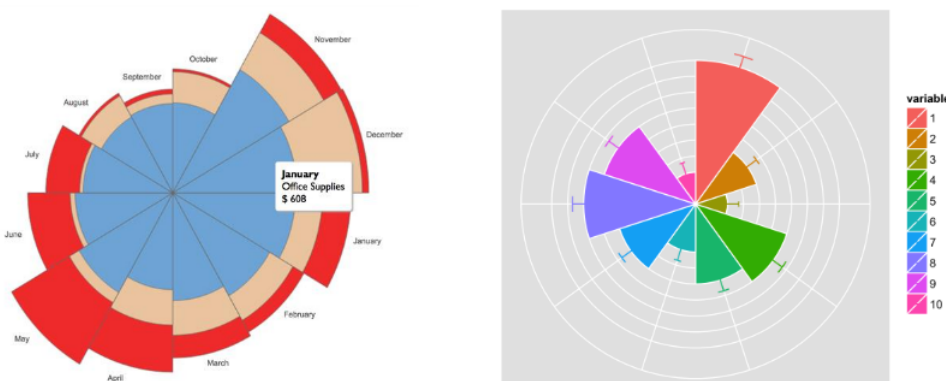
In a **Coxcomb chart**, each category is represented by a section of the disc, and each section has the same angle. The area of a section represents the value of the corresponding category.



The invention of the Coxcomb chart is attributed to Florence Nightingale, a nurse during the Crimean War. (Interestingly, Nightingale doesn't seem to have ever referred to them as Coxcomb charts. Instead, she called the books that contained the charts Coxcombs, and the name stuck to the charts themselves).

Florence Nightingale was one of the first to establish the importance of sanitation in hospitals, and she collected meticulous data about the relationship of cleanliness to death tolls. When she presented her data in 1858, she developed a Coxcomb chart, a variation of a pie chart that represented the number of soldiers using the area of the circle segments instead of the radius. Her message — that more soldiers died from infection than from wounds — could be easily seen and resulted in policy changes that helped save lives.

One problem with Coxcomb charts is that subtle differences are hard to see, so they are best used when trying to demonstrate patterns rather than delivering exact numbers. But because Coxcomb charts make for a nice image without any decorative elements, Coxcomb charts can be a good choice for presentations to viewers who don't have much technical knowledge about the information being presented.



7.3.1 Visualizing Temporal Phenomena > Changing Axis: Radial Layouts

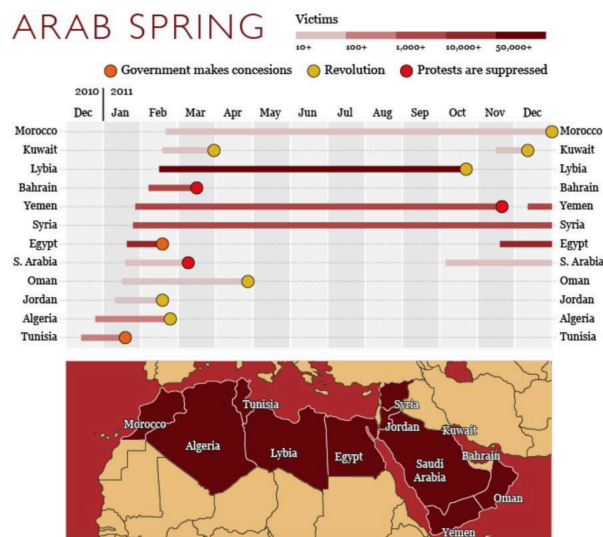
If we compare Radial layouts versus Rectilinear layout we know that they are geometrically equivalent, however perceptually:

- Angle channel less accurate than rectilinear spatial position channel
- Angle channel inherently cyclic (start point = point)
- Position channel inherently linear
- Radial layout good for periodic patterns

This is an example of an Infographics showing the events taking place in North Africa in the spring of 2011, which became the starting point of the protests in the region:



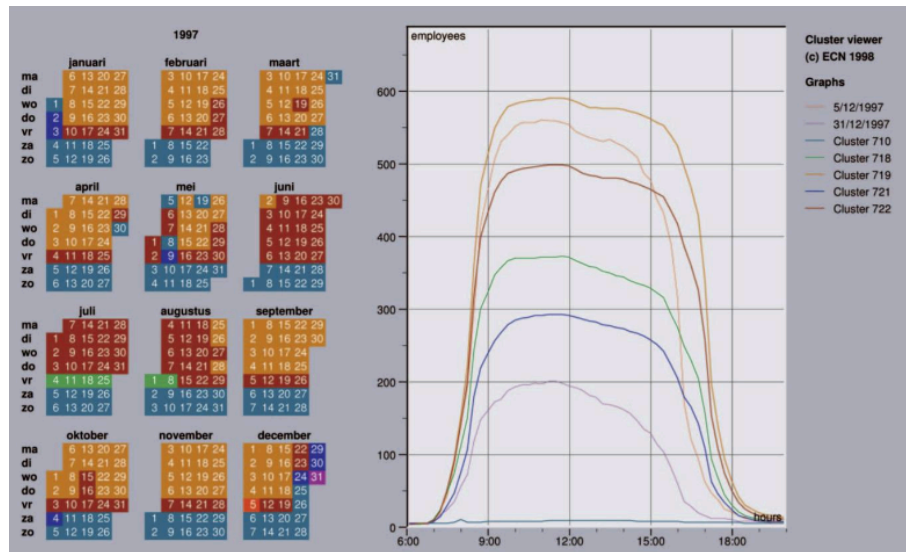
Same layout but in rectilinear form:



- Timelines shaped as circles are usually more difficult to interpret.
- It also suffers from distortions, lines in the outer regions look longer than those in the regions closer to the center.
- Sometimes it is difficult to decide which one is best, it really depends on your objectives.
- If it is accuracy in decoding information perhaps linear in this case is the way to go.
- It is true also that distortion may help emphasizing importance of areas with higher density like January in this case.

Calendar Views

Calendar views are layouts where the time attribute can be displayed according to such granularity



Proposed by van Wijk in 2006 as a method for clustering everyday data; image shows the number of employees on the job from 6:00 to 18:00 after 5 types of aggregation, and the comparison with the two types of average charts

Animated Transitions

Visual chart that plays each time point frame by frame continuously showing the changing trend of the data.

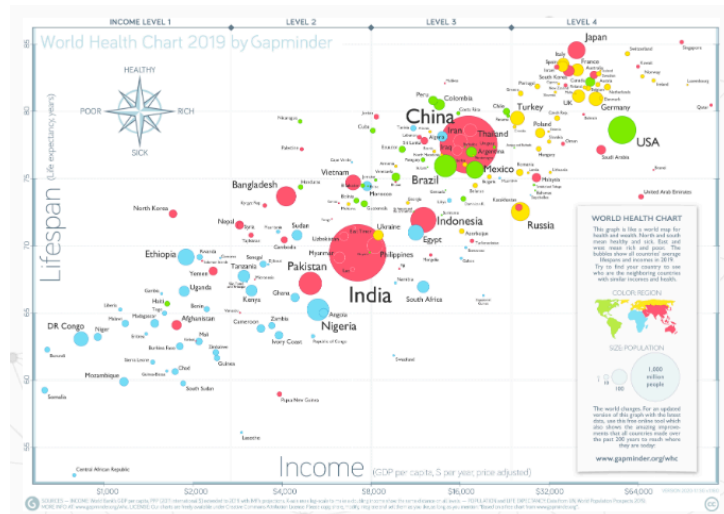


Figure 71: Gapminder Trendalyzer – Hans Rosling 2005¹³

Gapminder uses the position information (x- axis and y-axis) of the bubble chart, the size of the shape to display the three dimensions of the data, and the animation of the fourth dimension (time).

¹³<https://www.gapminder.org/tools/?from=world#&chart-type=bubbles&url=v2>

Small Multiples Line Charts

Basic idea: rather than using one single line chart and lines overlap, make smaller graphs, repeat the graph multiple times (ideally as many times as the number of lines that you have or objects or categories that you want to represent).

Note: Small Multiples are a “general purpose” technique



Effective when you have a large number of timelines.

You can construct small multiples for almost any type of chart: bar charts, pie charts, maps, etc. Small multiple organization often produces a more effective layout than coerce all the data into a single plot.

Advantages	Disadvantages: comparison is harder
Less clutter because the lines don't overlap.	Data are no longer in the same chart
	Individual charts tend to become smaller
	For line charts using area charts makes comparison “easier”: filled areas make perceiving shapes easier.

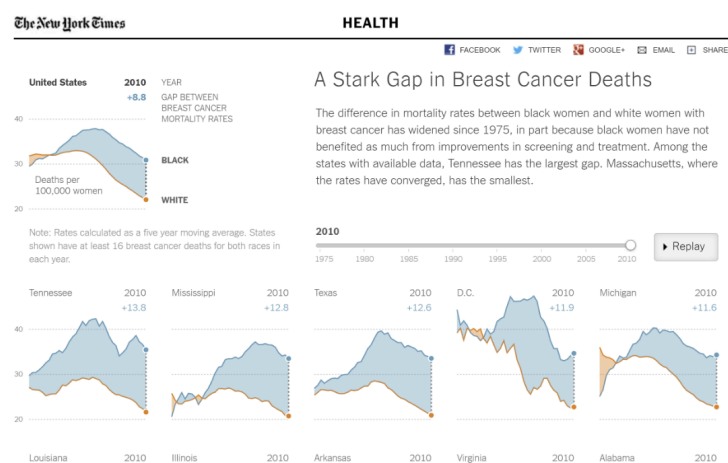


Figure 73: Animated transitions in small multiples¹⁴

¹⁴<https://archive.nytimes.com/www.nytimes.com/interactive/2013/12/20/health/a-racial-gap-in-breast-cancer-deaths.html>

7.4. Visualizing Spatio-Temporal Phenomena

Spatio-Temporal Data

- Difficult to decouple space and time
- Difficult to integrate them visually in a way that is also effective.
- Time can be encoded: implicitly (2.5D; flow maps, small multiples), explicitly (2D+time; animated transitions)

Beware: not all Temporal processes are spatial (e.g. financial data)

Trajectories - Flow Maps

By modifying its visual attributes lines can give different amounts of information.

- The most basic form is a line between two points which gives information about the movement of an object.
- Temporal component: timeline with a start and an end point + points in between (intermediate timestamps)
- Spatial information: locations recorded at these time stamps.
- Trajectory data can help the understanding of human behavior during an unusual event (sports event, crisis or disaster such as a hurricane or terrorist attack or a pandemic ...), traffic flow, city planning.

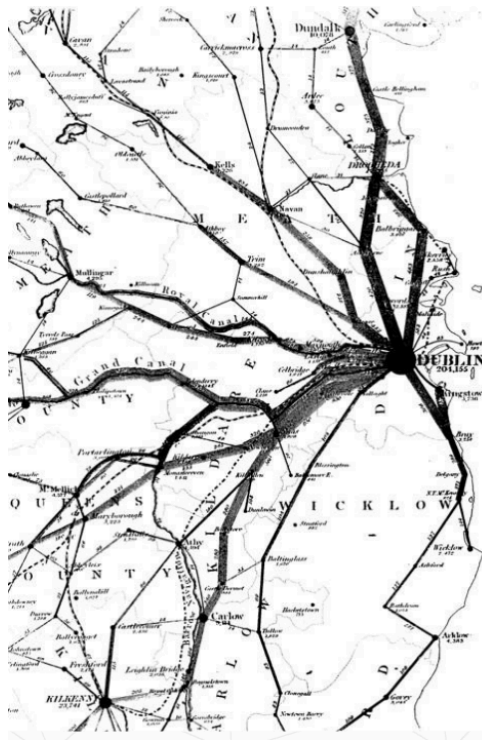


Figure 74: Transportation of Passengers in Ireland, Henry Drury Harness, 1837

Effective means to visualize spatio temporal interactions when datasets imply flow between spatially different regions. Show you how something is transported from A to B.

Structure of Flow Maps

- Flow Tree: Root source of the flow is the root, Leaves distribution target nodes
 - Width of the flows are scaled relative to the quantities of items that are being transported.
 - Flows going from the root to the same node are merged (edge merging) -> trajectory grows in width + reduction in visual clutter
 - Edge merging is a type of spatial aggregation and a strength of flow maps.
 - Trajectories can be enriched with arrows to show flow of the objects.
- arrows to show flow of the objects.



Figure 75: A flow map of the traffic in Milan. The trajectories are aggregated and only flows representing 50 or more trajectory segments are shown.

Three aspects to consider:

- Merging of the edges that share the same goal,
- Smart distortion of positions and
- Optimal routing of the edges.

Routing is important since to avoid crossings and overlaps of different trajectories (likely to happen if trajectories are drawn as straight lines) when destinations are close to each other.

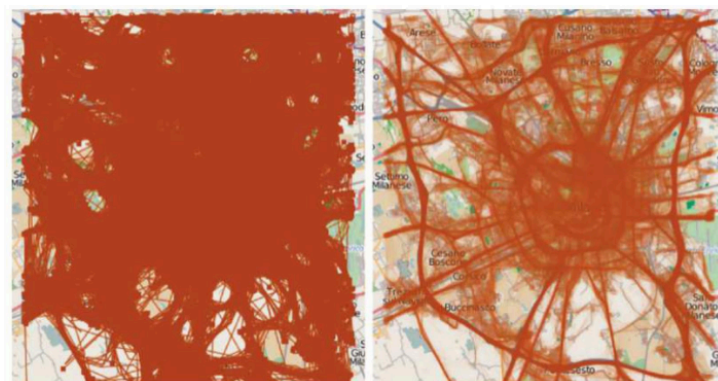


Image shows a flow map of traffic in Milan

- Left image has trajectories drawn fully opaque and
- Right has trajectories drawn with a 5% opacity
- You can see effects of routing and edge bundling

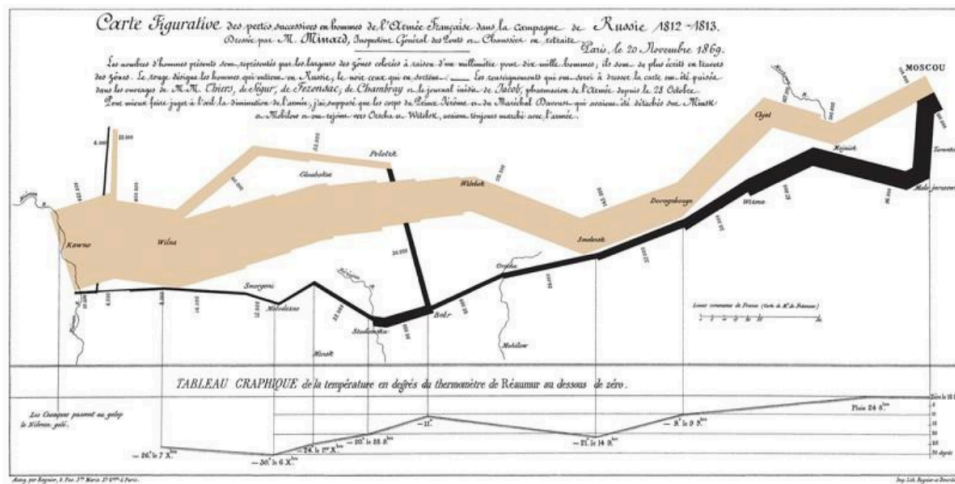
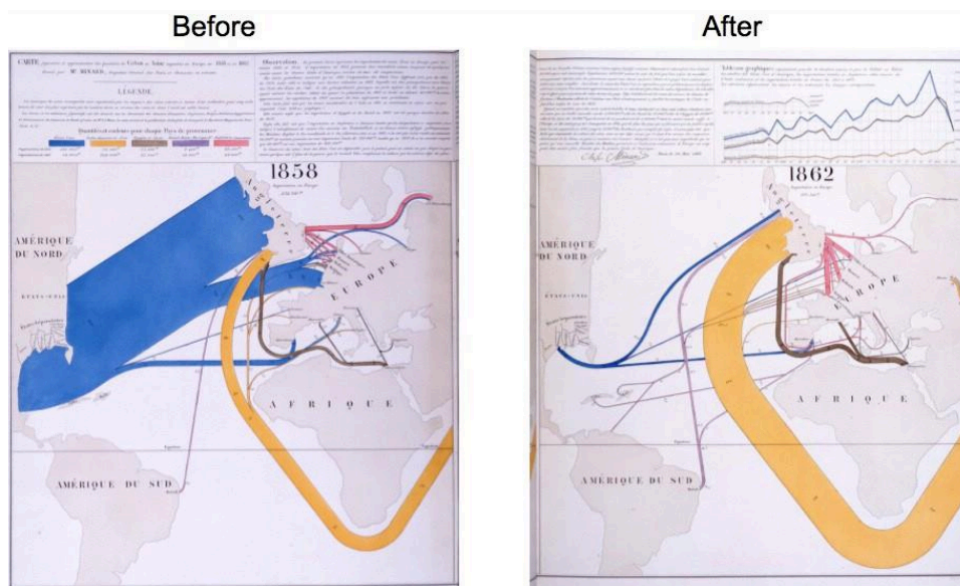
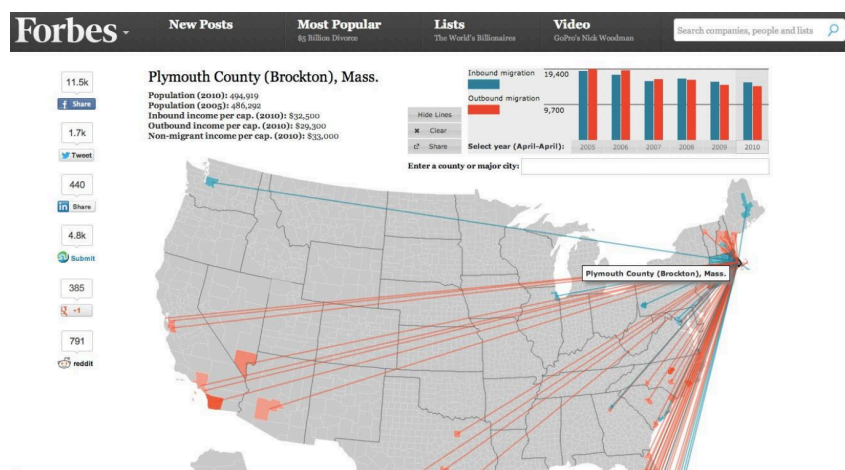


Figure 77: Napoleon Russian Campaign

Figure 78: Effect of U.S. Civil War on Cotton Trade¹⁵Figure 79: Example of interactive flow map¹⁶

¹⁵<https://www.datavis.ca/milestones/>

¹⁶<https://www.forbes.com/consent/ketch/?toURL=https://www.forbes.com/special-report/2011/migration.html>

Non-spatial Representation

You can do the same without the spatial context.

- Removing the spatial constraint allows us to show other data such as Time
- Relies on user knowledge -> you can add more information

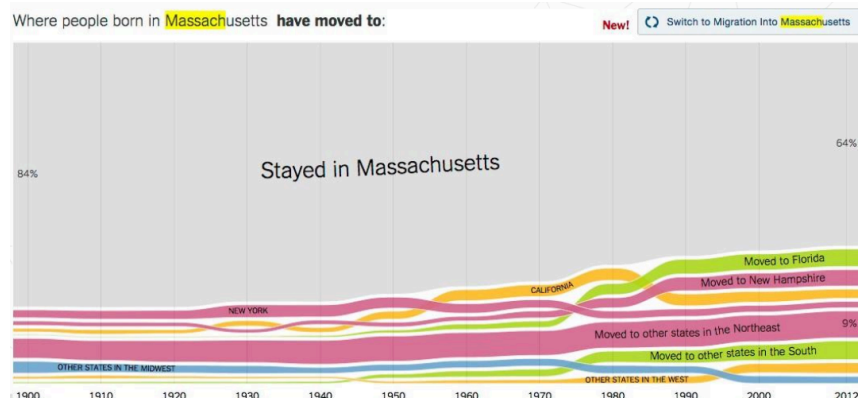


Figure 80: Where people born in Massachusetts¹⁷

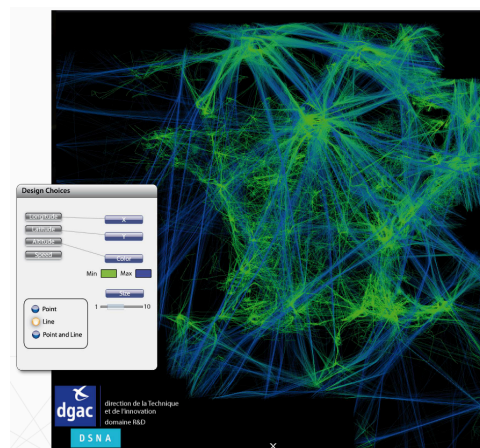


Figure 81: Example shows: Air traffic over France.

Interesting feature: Shows a very flexible mapping of attributes to visual channels

¹⁷<https://www.nytimes.com/interactive/2014/08/13/upshot/where-people-in-each-state-were-born.html>

Space-Time Cube There are many more visual representations for times series. We have surveyed the most fundamental ones. 3D worth mentioning is the Space-time cube.



We can partition the space, allowing us to do the same analysis that we did in 2D (hotspot, cluster, and outlier analysis).

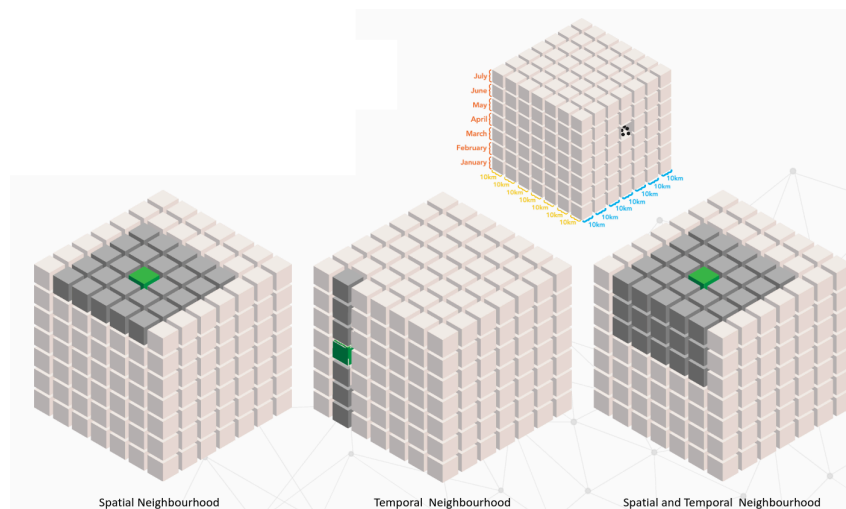


Figure 84: As applied to the Earth

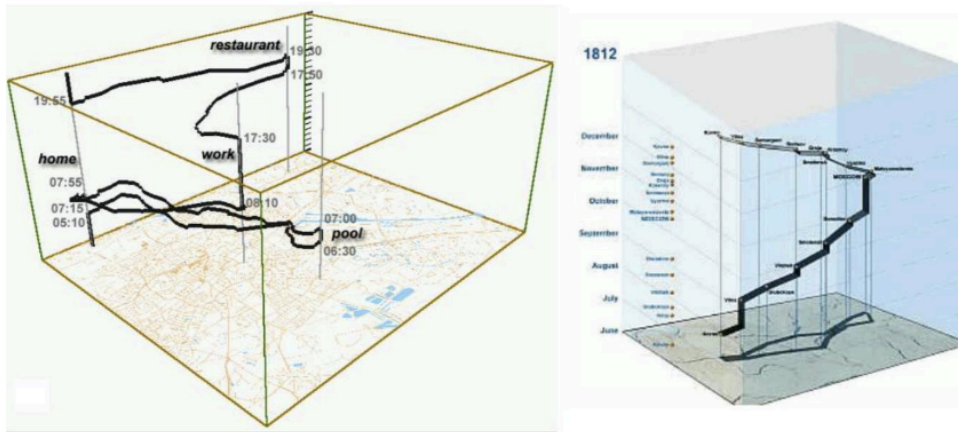


Figure 85: Left represents the travel of one person on an average Thursday. Right represents Napoleon's Russian campaign in 1812.

Space-time cube suffer from same limitations as other 3D visualizations: perspective distortion, occlusion. Interaction in this case is key.

7.5. Interaction

We can use **interaction** to support inspection of very large datasets when a single self-contained visualisation is not enough and there is more than one way to look at the data.

To amplify engagement with data:

- Building relationships
- Exploring and testing causality
- etc.

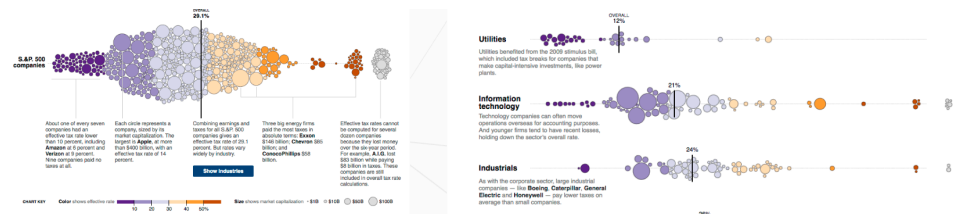


Figure 86: Good examples of interaction¹⁸

Interaction is not “unique”!

- Design dependent: touch, gesture, speech
- Platform Dependent: Mouse, keyboard, touchscreen, mobile platform

Indirect Manipulation

- Algorithmic transformations
- Software providing algorithms as automatic tools acting as black boxes
- You may be able to set parameters

Single View

- Change over time, Transition
- Navigation, Semantic zooming
- Filtering and Querying
- Focus + Context

Direct Manipulation

- Can be acted upon via physical, reversible, incremental actions that receive immediate feedback

Multiple Views

- Selection & Details on Demand
- Linking & Brushing

¹⁸<https://archive.nytimes.com/www.nytimes.com/interactive/2013/05/25/sunday-review/corporate-taxes.html>

Change over time

Use, e.g., slider to see view with data at different times, very simple interaction; Sometimes it is however better to show difference explicitly rather than through a slider

Chord flow diagram with time slider

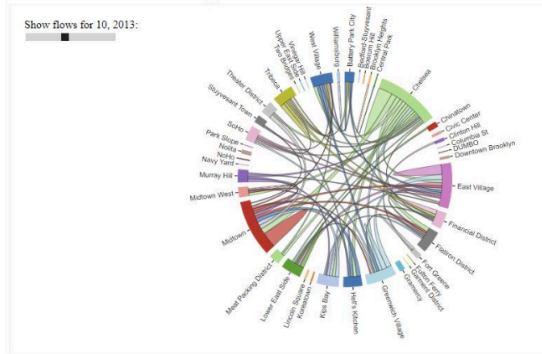


Figure 87: Chord flow diagram¹⁹

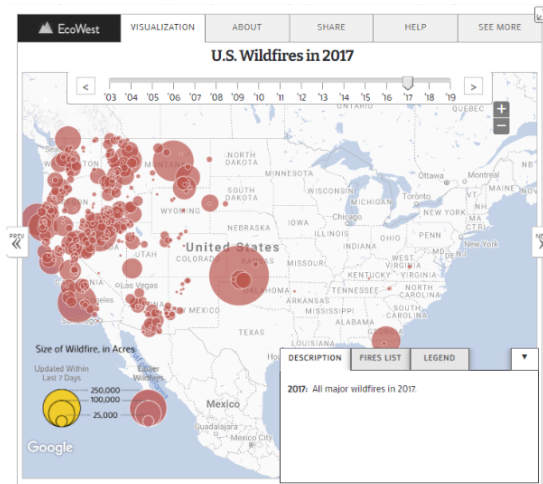


Figure 88: Wildfires²⁰

Transitions

Change over time is a type of Transition; Not all Transitions are “over time”

Example: use slider to see view with data at different times

Transition make it possible for users to track what is going on across different views of the data

Animated Transition

Smooth interpolation between states or visualization techniques (e.g. GapMinder application)

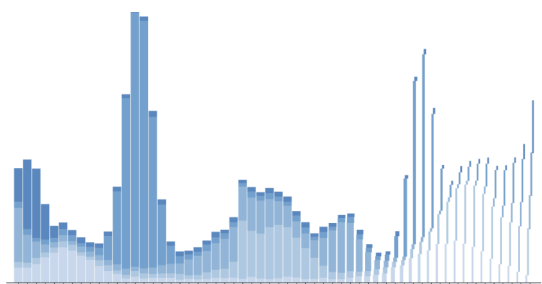


Figure 89: Stacked-to-group bars²¹

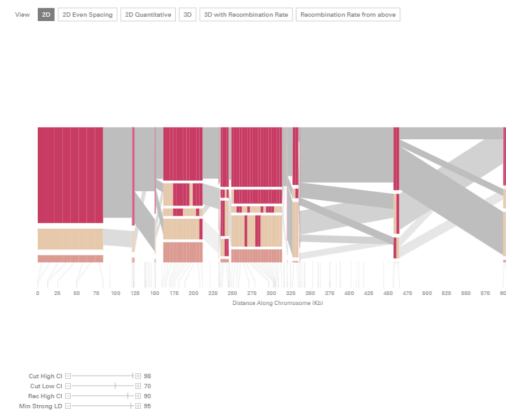


Figure 90: isometric blocks²²

¹⁹<https://gist.github.com/carsonfarmer/11478345>

²⁰<http://vis.ecowest.org/wildfire.html?start=0>

²¹<https://observablehq.com/@d3/stacked-to-grouped-bars>

²²<https://benfry.com/isometricblocks/>

Transitions as in Sorting

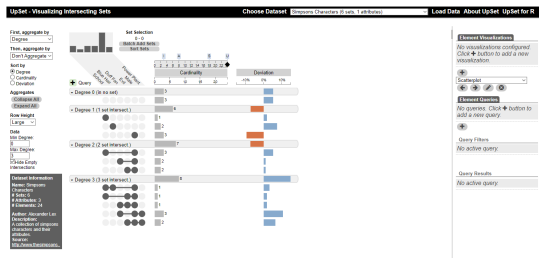


Figure 91: Visualizing Intersecting Sets²³

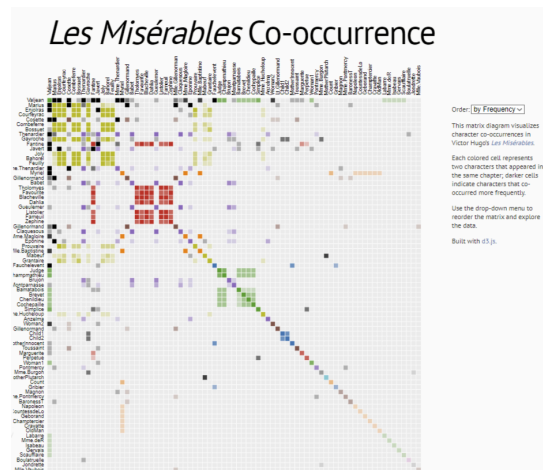


Figure 92: Les Misérables Co-occurrence²⁴

Limitations of Change over Time

Changes can be hard to track

- For animation: if too fast then it is not really useful
- Too much data is difficult to compare in an animation

You may not need animation

- Show all states in multiple views
- Show data in different linked charts

Navigation

Traditional means of interaction

- Pan: brings different parts of the model into view (similar to moving the model around)
- Zoom: enlarge/ make smaller (move camera), happens in the direction underneath the cursor
- Rotate: rotate a virtual camera around the model

Zooming

- Geometric (standard) zooming: get close in to see information in more detail; view depends on physical properties of what is being viewed
- Fisheye zooming: similar to the geometric zoom; outside information is not lost from view only distorted
- Semantic zooming: get close in to see information in more detail; zooming can be conceptual as opposed to simply reducing pixels, e.g., when zooming away, instead of seeing a scaled-down version of an object, see a different representation; the representation shown depends on the meaning to be imparted.

e.g. Google Maps, Windows 8

²³<http://vcg.github.io/upset/>

²⁴<https://bost.ocks.org/mike/miserables/>

Focus + Context

Object of primary interest is presented in full detail while at the same time an overview–impression of all the surrounding information – or context – is also available.

Pragmatically

- User selects region of interest (focus) through navigation or selection
- Systems provides context through: Aggregation, Reduction, Layering

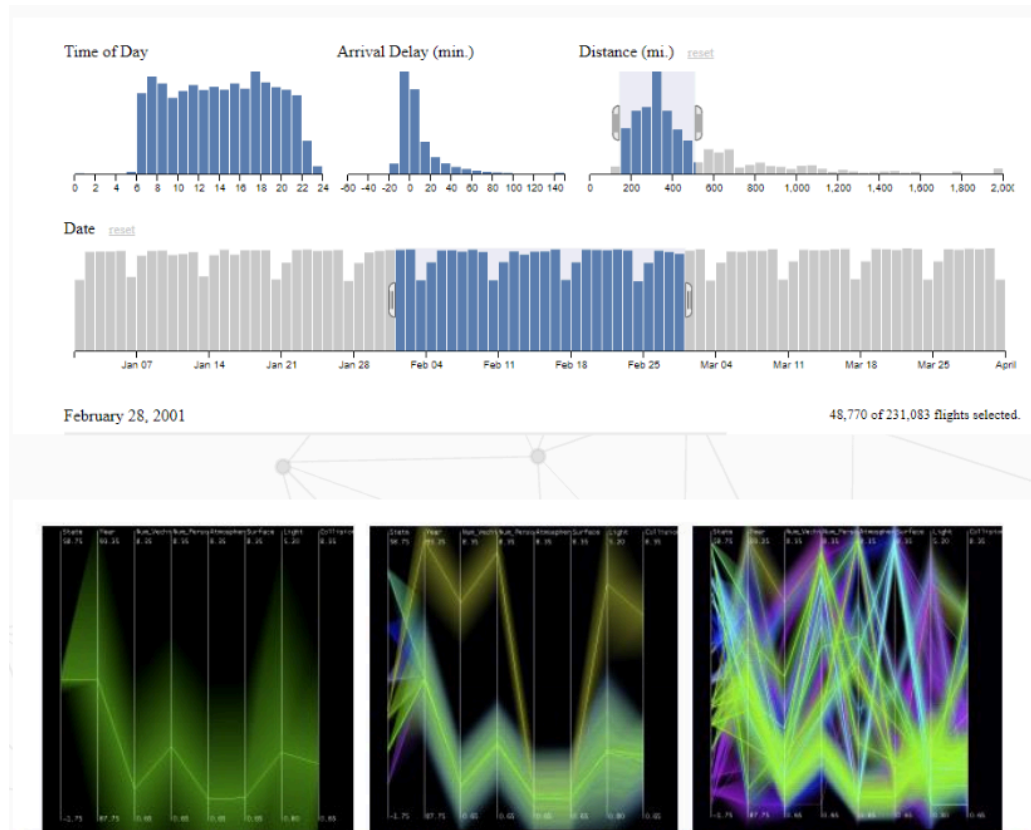


Figure 4: This image sequence shows a Fatal Accident data set of 230,000 data elements at different level of details. The first image shows a cut across the root node. The last image shows the cut chaining all the leaf nodes. The middle image shows an intermediate cut.

Use geometric distortion of the contextual regions to make room for the details in the focus region(s)

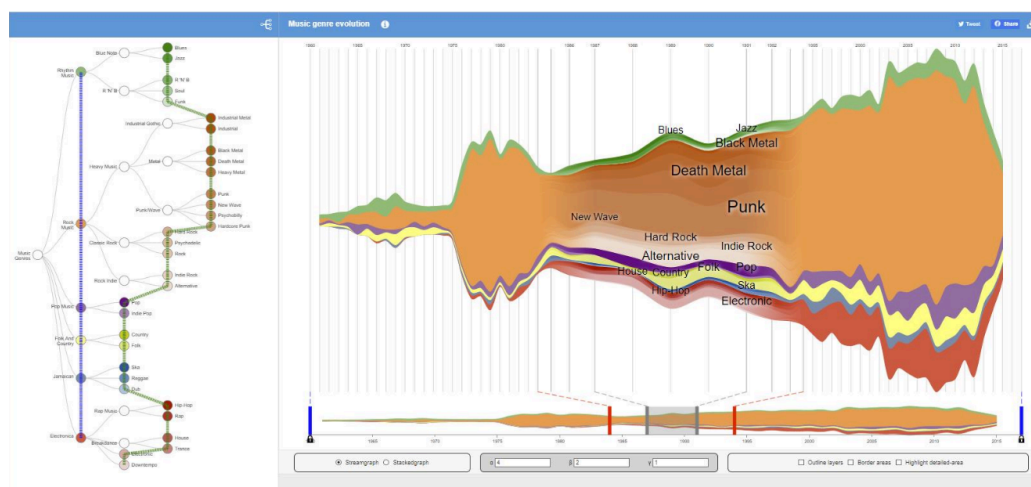


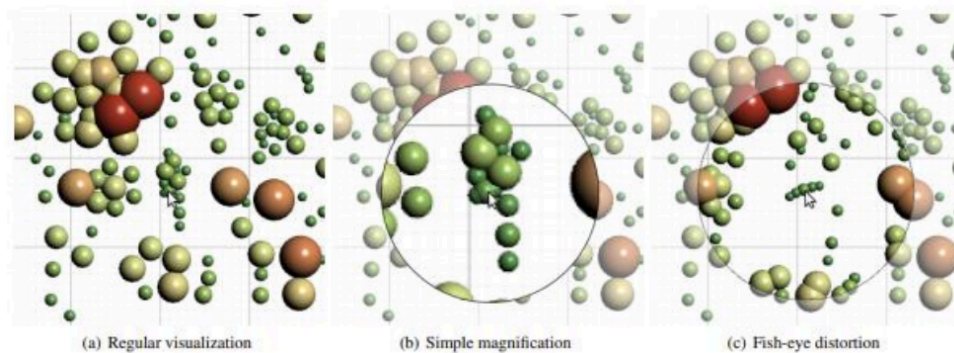
Figure 94: MultiStream visualisation²⁵

²⁵<http://advanse.lirmm.fr/multistream/visualize.php>

Use geometric distortion of the contextual regions to make room for the details in the focus region(s)

Lenses are a type of Distortion:

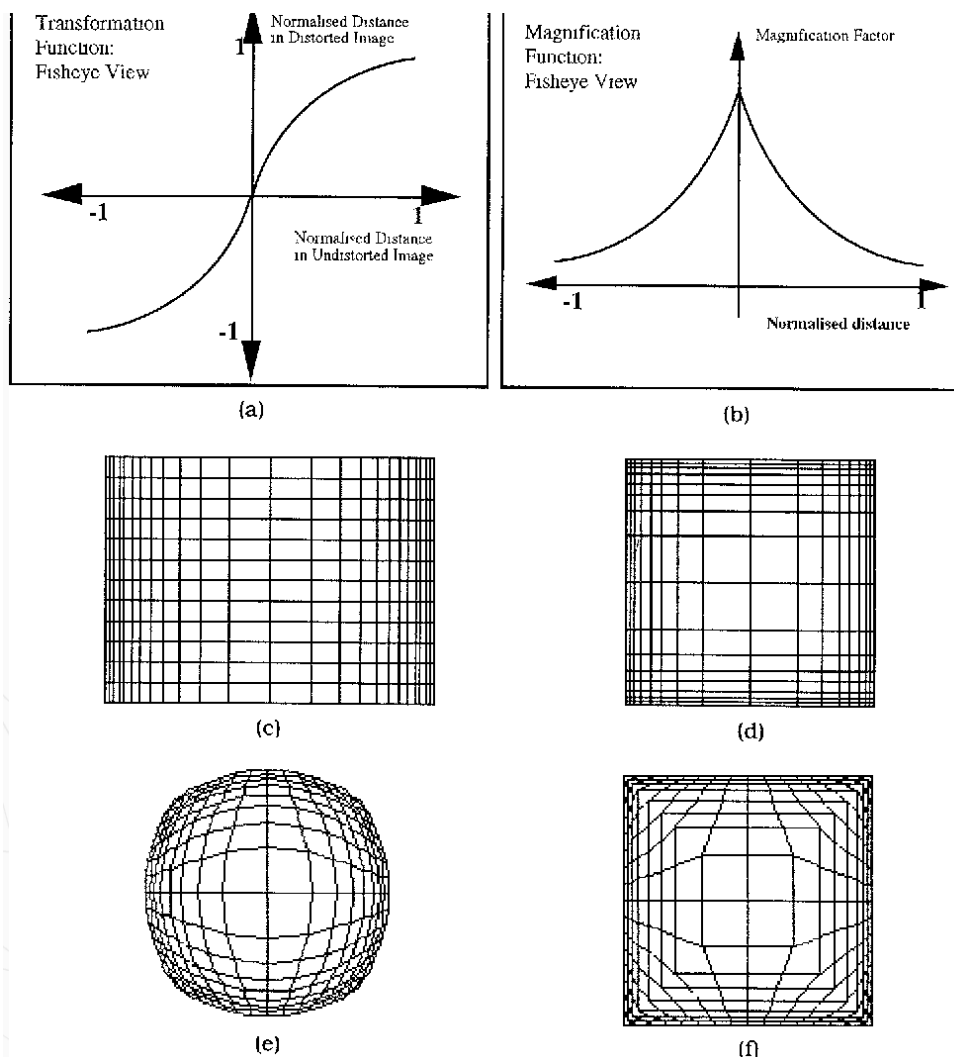
- Traditional lenses: magnify region, maintain context but loose connection
- Fisheye: allow to see both local detail and global context without losing connectivity



Fisheye View

It works through geometrical distortion

Applications in medicine, chemistry, biology, large graph exploration, areas where maintaining connectivity is important



Aggregation as Alternative to Distortion

A similar idea uses aggregation instead of distortion, the focus is the central grey area, the rest is context. The chart below are not distorted but rather aggregated data, in this way the central window shows the details while the neighbours show the aggregated data.

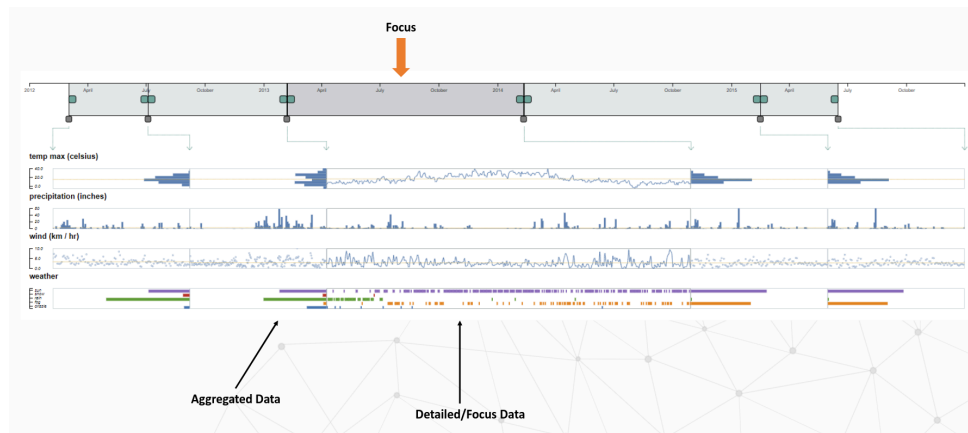


Figure 97: Periphery Plots Demo²⁶

Limitations of Distortions

- Unsuitable for relative spatial judgements
- Adds overhead of tracking distortion
- Requires visual communication of distortion: gridlines, shading
- Displacement makes target acquisition problem: lens displacing items away from screen location
- Mixed results compared to separate views and temporal navigation

Overview & Detail

Two images are used for presentation.

- One shows a rough overview of the complete information space and neglects details.
- The other one shows a small portion of the information space and visualizes details. - Both images are either shown sequentially or in parallel.
- Overview + detail provides multiple views with the first being an overview providing overall patterns to users



Visual Information Seeking Mantra (Shneiderman, 1996)

- Overview first,
- Zoom and Filter,
- Then Details on demand
- Then relate, history, extract

²⁶<https://precisionvissta.github.io/PeripheryPlots/>

Multiple Views

Single, monolithic view has strong limits on the number of attributes that can be shown simultaneously different views support different tasks

Eyes over Memory: Trade-off of display space and working memory

Allow for **Multiform** that is: use different visual encodings are used between the views

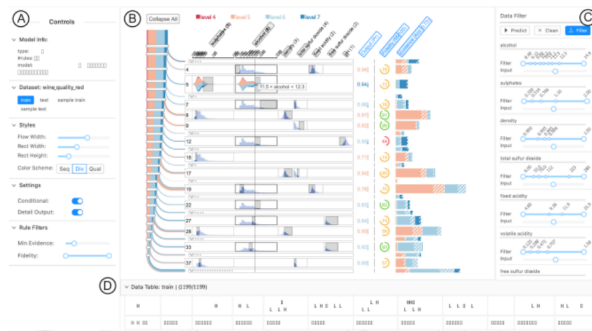


Figure 99: Multiview

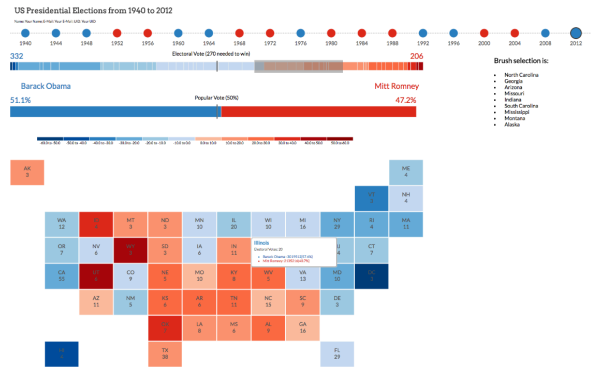


Figure 100: Multiform

Multiple view & multiform representations enable the user to better understand the information.

- Different views = multiple viewpoints
- Alternative representations
- Seeing information from different angles, perspectives, additional insight, adding context, overcome misrepresentations, alternate viewpoints.

Linked Views are multiple Views that are simultaneously visible and linked together such that actions in one view affect the others.

One view shows (often summarized) information about entire dataset, while additional view(s) shows more detailed information about a subset of the data

- Rational: for large or complex data, a single view of the entire dataset cannot capture fine details
- Example: TimeNotes

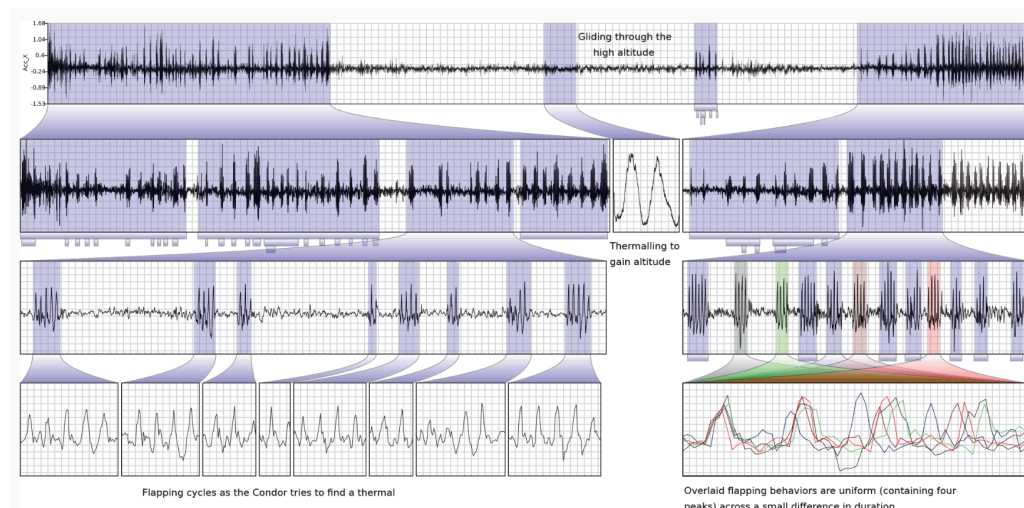
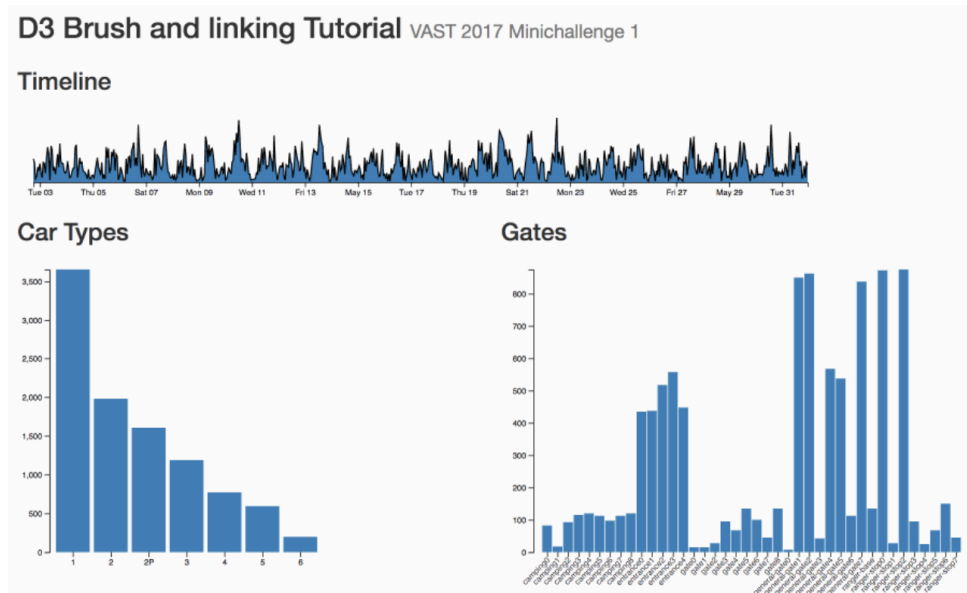


Figure 101: TimeNotes

The idea of **linking and brushing** is to combine different visualization methods to overcome the shortcomings of single techniques.

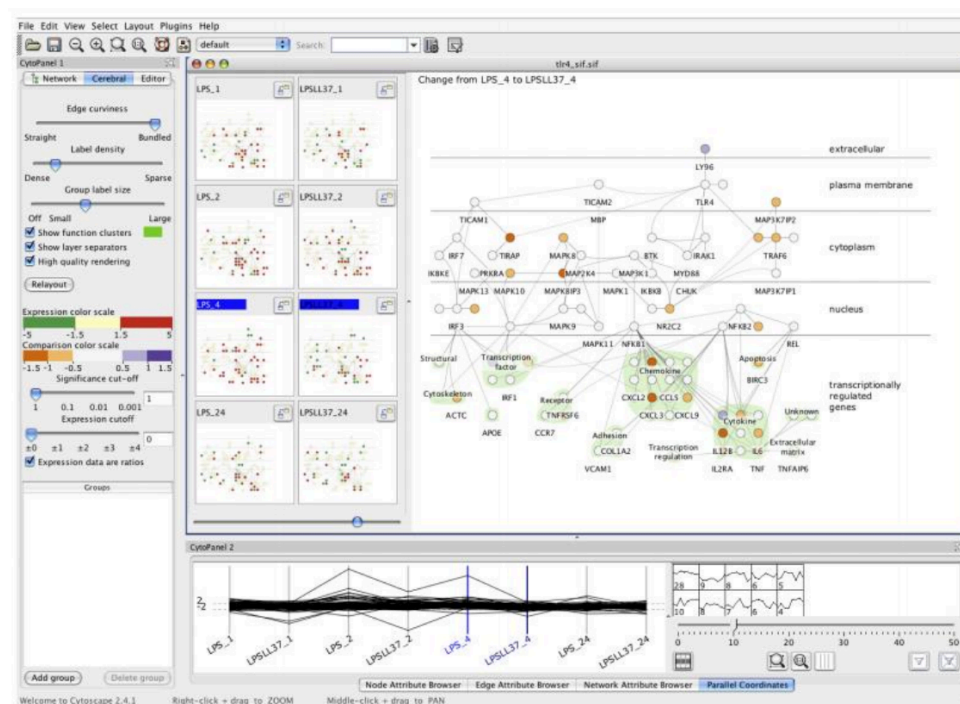
- Interactive changes made in one visualization are automatically reflected in the other visualizations.
- Connecting multiple visualizations through interactive linking and brushing provides more information than considering the component visualizations independently.
- Brushing and Linking support dynamic visual queries: selection of subsets of data on which to enact analytics



Multiple Views: Small Multiples

Each view uses the same visual encoding, but shows a different subset of the data

- Rational: quickly compare different parts of a data set
- Eyes instead of memory



8. Hierarchical and Network Data

Networks are the framework we use to model (temporal) interactions between static entities; they explicit model interactions between individual agents (networks of relationships change over time)

Can be used to model any phenomena where we have interaction between entities such as:

- social interaction
- economic Interaction
- financial interaction/contagion

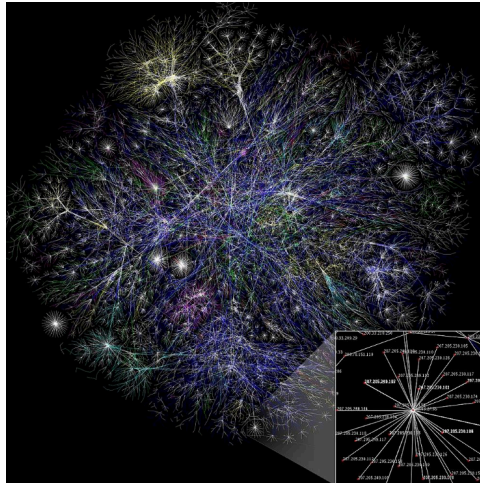


Figure 104: Internet Mapping Project²⁷

Collected and preserved traceroute-style paths to some hundreds of thousands of networks almost daily since 1998

Networks are also called **graphs**; consist of set of vertices (nodes) and set of edges (links); edges connect pairs of vertices (however not every pair of vertices are connected).

Formally, $E \subseteq \{(u, v) | \exists u \in V \wedge \exists v \in V\}$

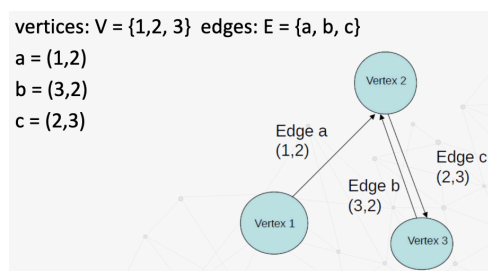


Figure 105: Directed graphs may contain two edges between one pair of nodes.

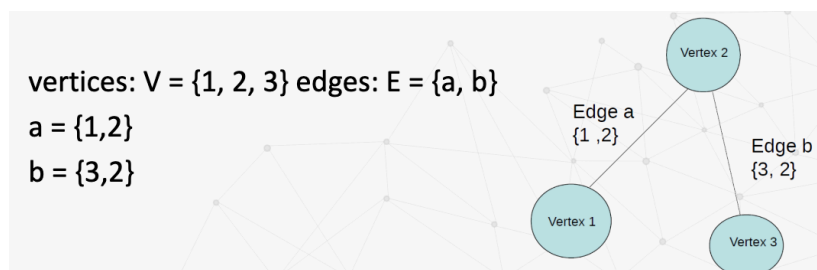


Figure 106: Undirected graphs do not have a direction, ordering of nodes in edge is not important.

²⁷https://en.wikipedia.org/wiki/Internet_Mapping_Project

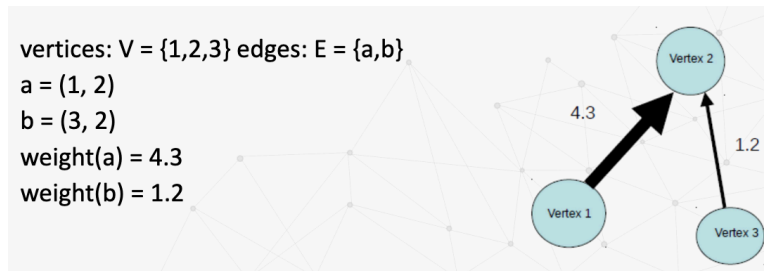


Figure 107: We can weight the edges to represent the strength of the connection.

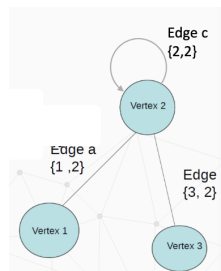
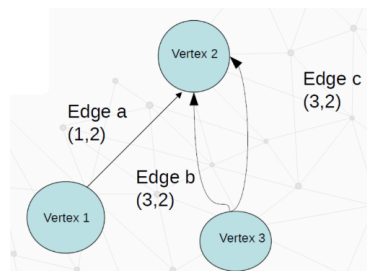
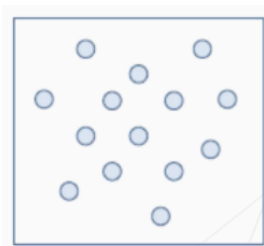
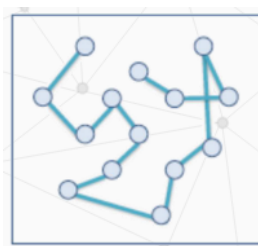
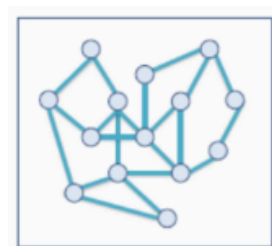
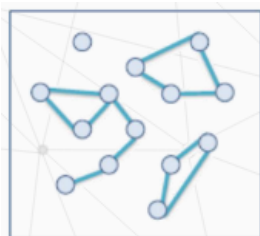
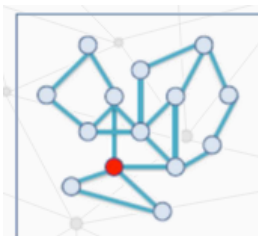
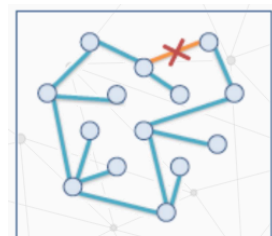
Figure 108: Acyclic graphs/DAGs do not allow vertices to connect to themselves.
Have no closed loop; can be topologically ordered

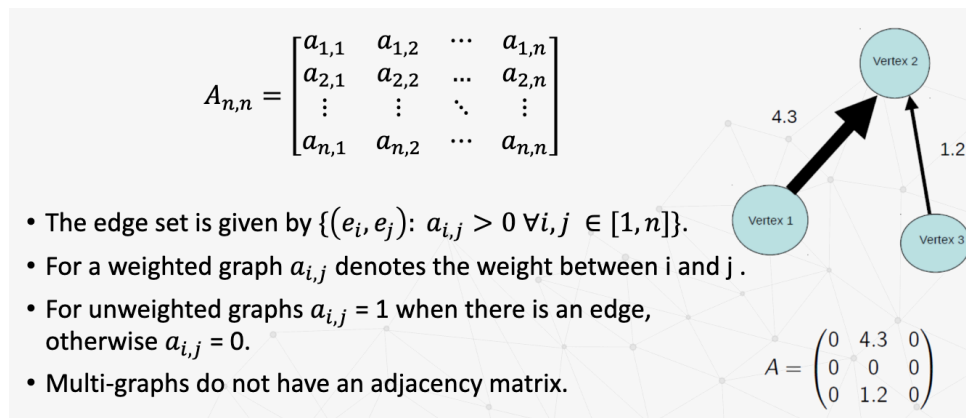
Figure 109: In a multi-graph we allow multiple edges between the same pair of nodes

independent set
(no edges)clique
(all possible edges)path (only
consecutive traversal)biconnected graph (no
articulation points)bipartite graph (can
be partitioned in two
independent sets)

unconnected graph

articulation point
(points that break up
the graph if removed)tree: graph that
contains no cycles w/
parent-child relations

An **adjacency matrix**; **edge set**; **weighted graph**; **unweighted graph**; **multi-graphs**:



Graph Theory and Network Science are related academic fields. Graph and network often interchangeable terms. Graph theory has developed algorithms that can be applied to any graphical structure irrespective of the domain:

- Planar graphs: Can a graph be laid onto a 2D surface such that no edges cross.
- Application: circuit board design where no two wires can overlap.
- Shortest paths: What is the minimum number of hops required to get from vertex A to vertex B in a graph? Moreover, what is the path that was taken?
- Applications: routing, automated reasoning, and AI planning.
- Energy flows: If a continuous “energy field” is diffused out from a particular vertex (or set of vertices), how much energy do the other vertices in the graph receive?
- Application: recommendation engines, knowledge discovery, artificial cognition/intelligence, ranking, and natural language processing.

Network science

Real life networks tend to be large-scale, we need to study from statistical perspective:

- Degree distribution: Given a node randomly selected from a network, what is the probability that it has X number of edges emanating from it?
- Applications: understanding how disease spreads through a social network, identify weak spots in a communication network.
- Assortative mixing: Do nodes with characteristic A tend to connect to nodes with characteristic B?
- Application: infer future connectivity patterns in the network.
- Growth models: Do all real-world networks grow according to a similar rule?
- Application: design of learning systems and understanding the future statistics of a developing network

Network metrics

- **Network size**: the total number of vertices N
- **Maximum possible number of edges**:

$$\frac{N(N-1)}{2} \approx \frac{N^2}{2}$$

- **Distance between vertices u and v** : either number of edges on the shortest path from u to v , or ∞ if there is no path.
- **Diameter of the network**: the longest shortest path within the network
 - worst-case diameter: largest distance between any pair of nodes
 - average-case diameter: average distance of all pairs of nodes

- If the distance between all pairs is finite, we say the network is connected; otherwise it has multiple components.
- The degree of vertex v is the number of edges connected to v (same as in graph theory).
- The degree distribution is the statistical distribution of the degrees of all vertices in the network
- **Pagerank**: count number & quality of links

Centrality

There are different metrics use to measure centrality, which identify the most important vertices within a graph. (note: not the same as node influence)

- Degree Centrality: nodes with higher degree are more susceptible (e.g. to viruses or receiving some information, etc.)
- Closeness Centrality: measures the farness of a node from other nodes
- etc.

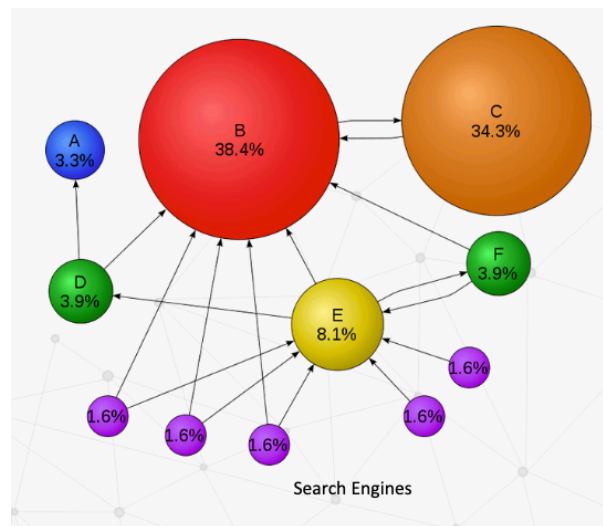
Remark: Nodes (and edges) in networks are of different importance levels!

e.g. identify most influential person(s) in social network, key infrastructure nodes in internet/urban network, identify super-spreaders of a disease, etc.

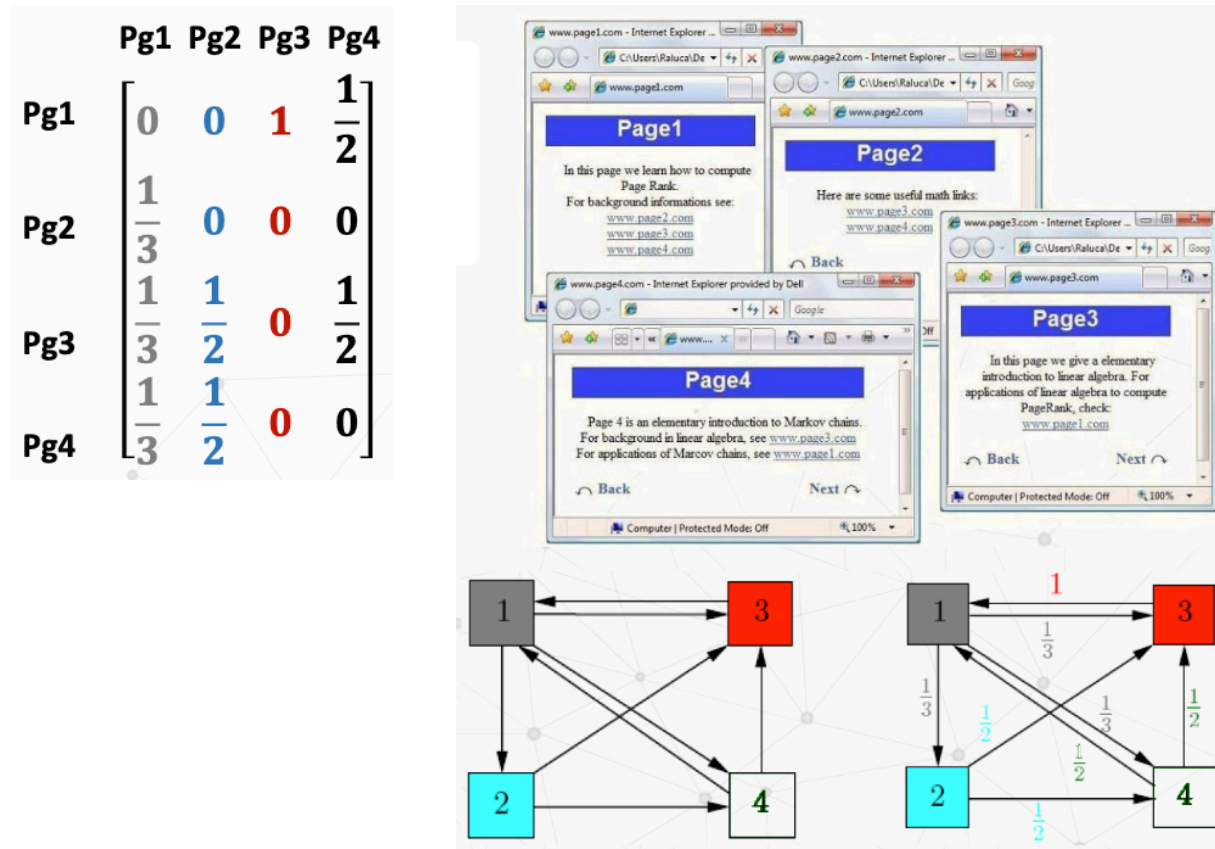
PageRank is a centrality measure used by Google to sort the search results based on their importance. Algorithm invented by Larry Page and Sergey Brin (co-founders of Google) in 1998. A web page is considered to be important if many other pages or a few but important pages link to it.

PageRank is a link analysis algorithm that ranks the relative importance of all web pages within a network. It does this by looking at three web page features:

1. Outgoing Links - the number of links found in a page
2. Incoming Links - the number of times other pages have visited this page
3. Rank - A value representing the page's relative importance in the network



Each page divides its importance equally among pages it links and transfers it to them.
Relationship is represented by transition matrix A :



- Suppose that initially the importance is uniformly distributed among the 4 nodes, each getting $\frac{1}{4}$.
- Denote by v the initial rank vector, having all entries equal to $\frac{1}{4}$.
- Each incoming link increases the importance of a web page, so:
 - Step 1: we update the rank of each page by adding to the current value the importance of the incoming links.
 - This is the same as multiplying the matrix A with v .
 - The new importance vector is therefore $v_1 = Av$.
 - Step 2 [iterate]: the updated importance vector is $v_2 = A(Av) = A^2v$.
- Numeric computations show that:
 - the sequences of iterates $v, Av, \dots, A^k v$ tends to the equilibrium value
 - We call this the PageRank vector of our web graph.

$$v = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \quad Av = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}, \quad A^2v = A(Av) = A \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

$$A^3v = \begin{pmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{pmatrix}, \quad A^4v = \begin{pmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^5v = \begin{pmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{pmatrix}$$

$$A^6v = \begin{pmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^7v = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^8v = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

The PageRank formula also has a “**damping factor**”: simulates the probability of a random user continuing to click on links as they browse the web; perceived to decrease with each link click

Assumption: “The probability of you clicking a link on the first page you visit is reasonably high. But the likelihood of you then clicking a link on the next page is slightly lower, and so on and so forth.”



The **clustering coefficient** measures the proportion of neighbours of a node that are connected with each other (e.g. the average proportion of neighbours of a vertex that are themselves neighbours)

Define the neighbourhood of node i as

$$N_i = \{v_j : e_{ij} \in E \vee e_{ji} \in E\}$$

Degree of node $i = k_i$.

For a directed graph, the local clustering coefficient of node i is

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

For an undirected graph:

$$C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

Global clustering coefficient of the entire network is $\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i$

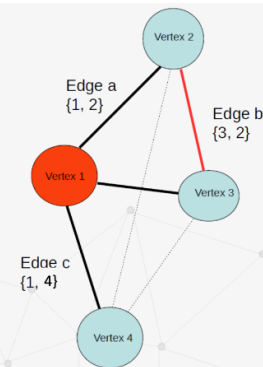
Proportion of links between the vertices within its neighbourhood divided by the number of links that could possibly exist between them.

The clustering coefficient - Example

- Node $i = 1$
- Neighbourhood $N_i = \{2, 3, 4\}$
- Degree of i is $k_i = 3$
- All actual edges $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 2\}\}$
- Possible connections between neighbours

$$P_i = \{(i, j) : \forall i \neq j \in N_i \times N_j\} = \{\{2, 4\}, \{2, 3\}, \{3, 4\}\}$$

- Number of possible connections $|P_i| = \frac{k_i(k_i-1)}{2}$
- Actual connections between neighbours $A_i = P_i \cap E = \{\{3, 2\}\}$
- Clustering coefficient of node $i = \frac{|A_i|}{|P_i|} = \frac{1}{3}$



Hard network (graph) algorithms - NP-complete

- Longest path
- Largest clique
- Maximum independent set (set of vertices in a graph, no two of which are adjacent)
- Maximum cut (separation of vertices in two sets that cuts most edges)
- Hamiltonian path/cycle (path that visits all vertexes once)
- Colouring / chromatic number (colours for vertices where no adjacent vertices have same colour)
- Minimum degree spanning tree

Properties of real life network run counter to conventional wisdom and are inconsistent with conventional graph models; real life networks tend to be:

- scale-free, that is most nodes have only a small number of neighbors (degree), but there are some nodes with very high degree (power-law degree distribution)
- have a high clustering coefficient, that is if a node x is connected to y and z , then y and z are likely to be connected
- follow a small-world network structure, that is most nodes are just a few edges away on average

Empirical networks typically have:

- few connected components (often only 1 or small number given the network size)
- small diameter: often a constant independent of network size, or perhaps growing only very slowly with network size, typically look at average; exclude infinite distances
- high degree of edge clustering: when compared with a random network, despite having small diameter
- heavy-tailed degree distribution: small number of high-degree vertices (hubs), often follows a power law

Modelling network formation

- Real-world networks are the result of complex underlying process.
- Networks from very diverse areas (from internet to biological networks) have similar properties
 - Is it possible that there is a unifying underlying generative process?
- Are there simple statistical models that capture commonalities across lots of different networks?
 - Analogous to random number generation, is there a useful abstraction to generate random networks?

Network Topology looks at structural properties of the network such as metrics: size, diameter, connectivity, degree distribution which define the topology of the network; structure can reveal community, important nodes, robustness, constraints on dynamics; does not analyse what actually occurs on the network

Network dynamics is a research field for the study of networks whose status changes in time

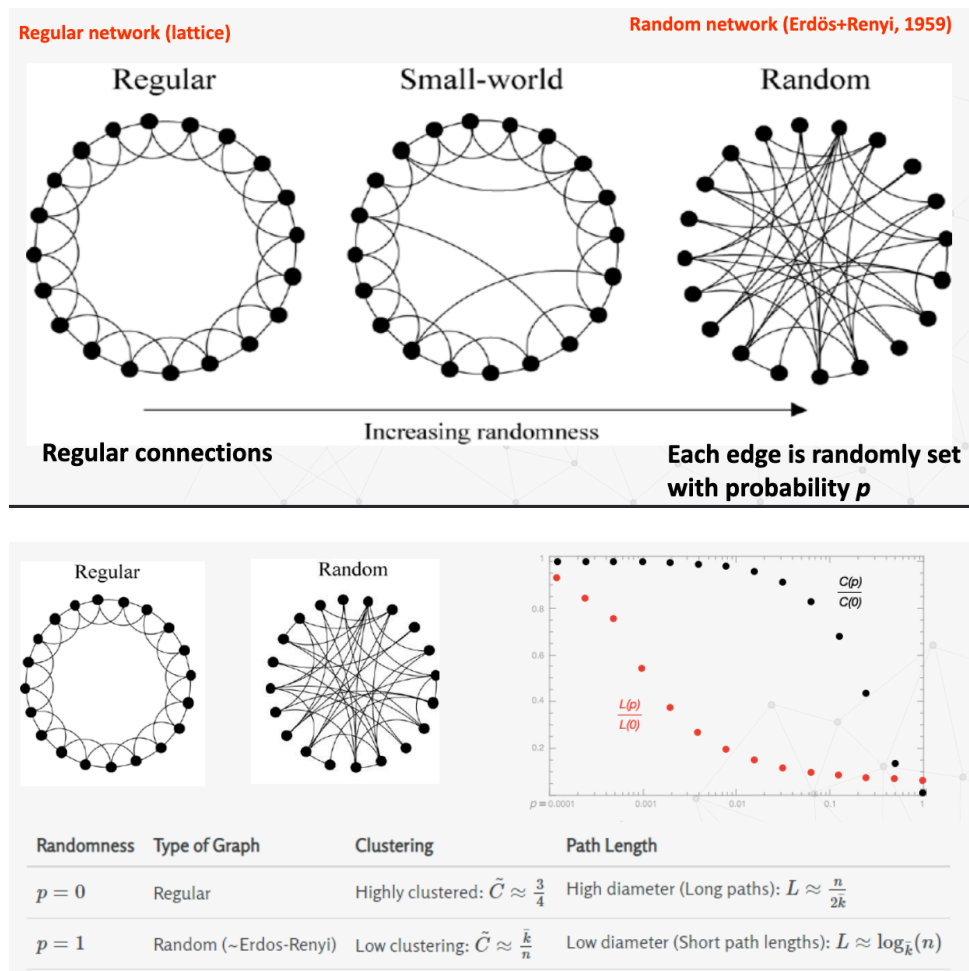
What happens on networks, e.g.:

- spread of disease over social network
- mapping spread of a fad
- computation in the brain
- traffic flows on the internet
- asset or debt owing in an economic network

Topology (static elements of a network) and dynamic are often coupled in a feedback loop:

- distribution of wealth depends on network topology
- in an epidemic, the rate of infection (dynamics) depends on the network connectivity (statics)

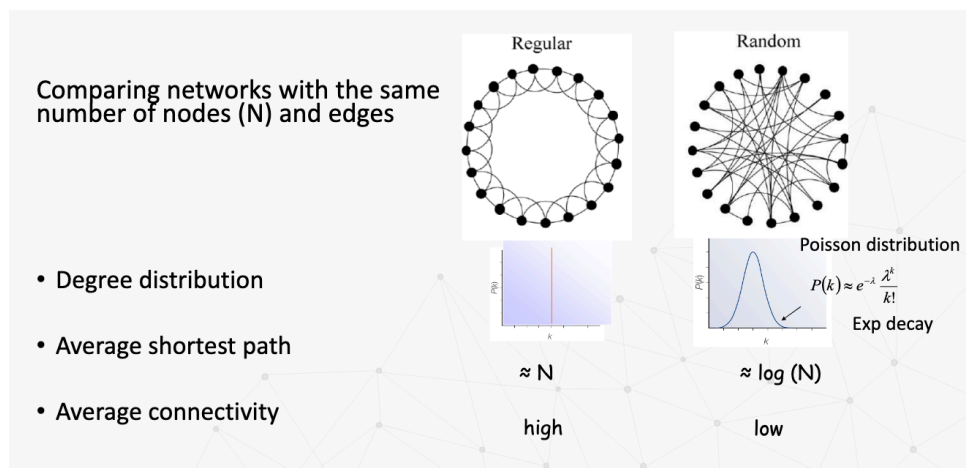
Two reference models for networks



When $p \approx 0.01$, local clustering of the graph is still high but the path lengths in the graph become very short. (characteristic of a real-world network)

Structural properties of regular and random graphs quantified in terms of: path length $L(p)$ and clustering coefficient $C(p)$, p grows from zero to one.

- $L(p)$ is a global property that measures the typical separation between two vertices.
- $C(p)$ is a local property that measures the cliquishness of a typical neighbourhood.



Random graphs: repeatedly connect pairs of nodes randomly chosen at uniform distribution; often used as a null model (compare measurements on real networks)

- Structurally very few components and small diameter, no heavy-tailed degree distributions, and low clustering coefficient
- Well studied model but most vertices have about the same degree; probability of two nodes being linked is independent of if they share a neighbour; average paths are short

Classic graph theory model (Erdős-Renyi): each edge is generated independently with prob. p

The basic $G_{n,p}$ (Erdős-Renyi) random graph model:

- n : the number of vertices
- $0 \leq p \leq 1$
- for each pair (i, j) , generate the edge (i, j) independently with probability p

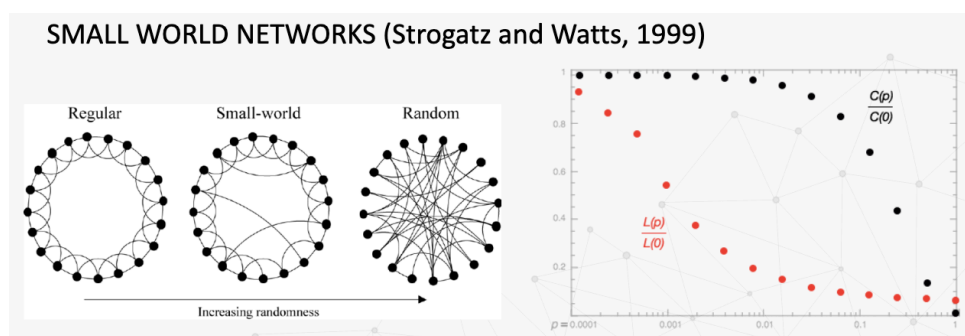
Network	size	vertex degree	shortest path	Shortest path in fitted random graph	Clustering	Clustering in random graph
Film actors	225,226	61	3.65	2.99	0.79	0.00027
MEDLINE coauthorship	1,520,251	18.1	4.6	4.91	0.43	1.8×10^{-4}
E.Coli substrate graph	282	7.35	2.9	3.04	0.32	0.026
C.Elegans neuron network	282	14	2.65	2.25	0.28	0.05

Figure 119: Examples for real networks

- Real networks are not regular (low shortest path)
- Real networks are not random (high clustering)

8.1. Small World Networks

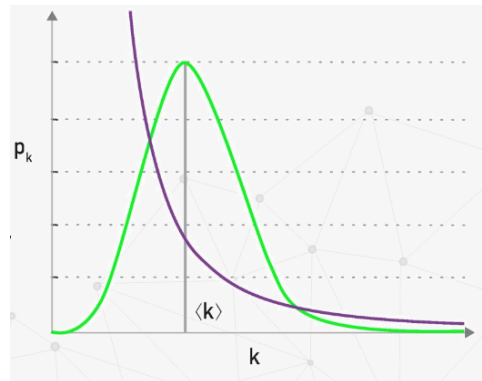
Networks with high clustering (like regular ones) and low path length (like random ones) can be obtained:



A small-world network is a type of mathematical graph in which most nodes are not neighbors of one another, but neighbors of any given node are likely to be neighbors of each other and most nodes can be reached from every other node by a small number of hops or steps.

- Networks with high clustering (like regular ones) and low path length (like random ones)
- A small amount of random shortcuts can decrease the path length, still maintaining a high clustering: this model “explains” the 6-degrees of separations in human friendship network

(derived from small-world networks) **Scale-free property** states that the degree distribution of nodes in the network is a power-law instead of a Poisson distribution, as would be expected for random networks.



The degree distribution is the distribution of the number of edges for each node.

- They showed that several of the networks investigated by Watts and Strogatz actually exhibit the scale-free property, such as the actor collaboration graph, the US power grid, a citation graph, the Internet etc.
- Emergence of this property due by two assumptions
 - First: Nodes being added to the network sequentially. This makes sense in the above mentioned cases where new computers are added to the Internet over time.
 - Second: A new node is more likely to be connected to a node which is already well-connected, i.e. has a high degree. (this is the well-known 'rich-get-richer' phenomenon.)

Example: Social Networks

A social network is a social structure of people, related (directly or indirectly) to each other through a common relation or interest

Social network analysis (SNA) is the study of social networks to understand their structure and behaviour through the use of networks and graph theory.

Metrics of social importance in a social network:

- **Degree centrality** assigns an importance score based simply on the number of links held by each node.
 - **What it tells us:** How many direct, 'one hop' connections each node has to other nodes in the network.
 - **When to use it:** For finding very connected individuals, popular individuals, individuals who are likely to hold most information or individuals who can quickly connect with the wider network.
- **Betweenness centrality** measures the number of times a node lies on the shortest path between other nodes.
 - **What it tells us:** This measure shows which nodes are 'bridges' between nodes in a network. It does this by identifying all the shortest paths and then counting how many times each node falls on one.
 - **When to use it:** For finding the individuals who influence the flow around a system.

Metrics of social importance in a social network:

- **Closeness centrality** scores each node based on their 'closeness' to all other nodes in the network.
 - **What it tells us:** This measure calculates the shortest paths between all nodes, then assigns each node a score based on its sum of shortest paths.
 - **When to use it:** For finding the individuals who are best placed to influence the entire network most quickly.
- **PageRank**
- **Clustering Coefficient**

Metrics of social importance in a social network:

- degree, closeness, between-ness, clustering, pagerank, etc.

Features both local and long-distance connections

Characteristics, "universals"

- Small diameter
- High clustering
- Heavy-tailed distributions

Models of network formation

- random graph models, small world networks
- affiliation networks
- preferential attachment: processes in which the more someone has of something, the more likely they are to get more of it

8.2. Affiliation Networks

An **affiliation network** is a network where the members are affiliated with one another based on co-membership in a group, or co-participation in some type of event.

8.3. Preferential Attachment

Probability of connecting a new node to existing node i :

$$p_i = \frac{k_i}{\sum_j k_j}$$

Structural properties: few components and small diameter, heavy-tailed degree distributions, low clustering coefficient

9. Hierarchical Data Visualisation

Networks and graphs are pervasive in our life. From sciences to medicine to transportations, navigation down to Social networks which have become a topic of great interest.

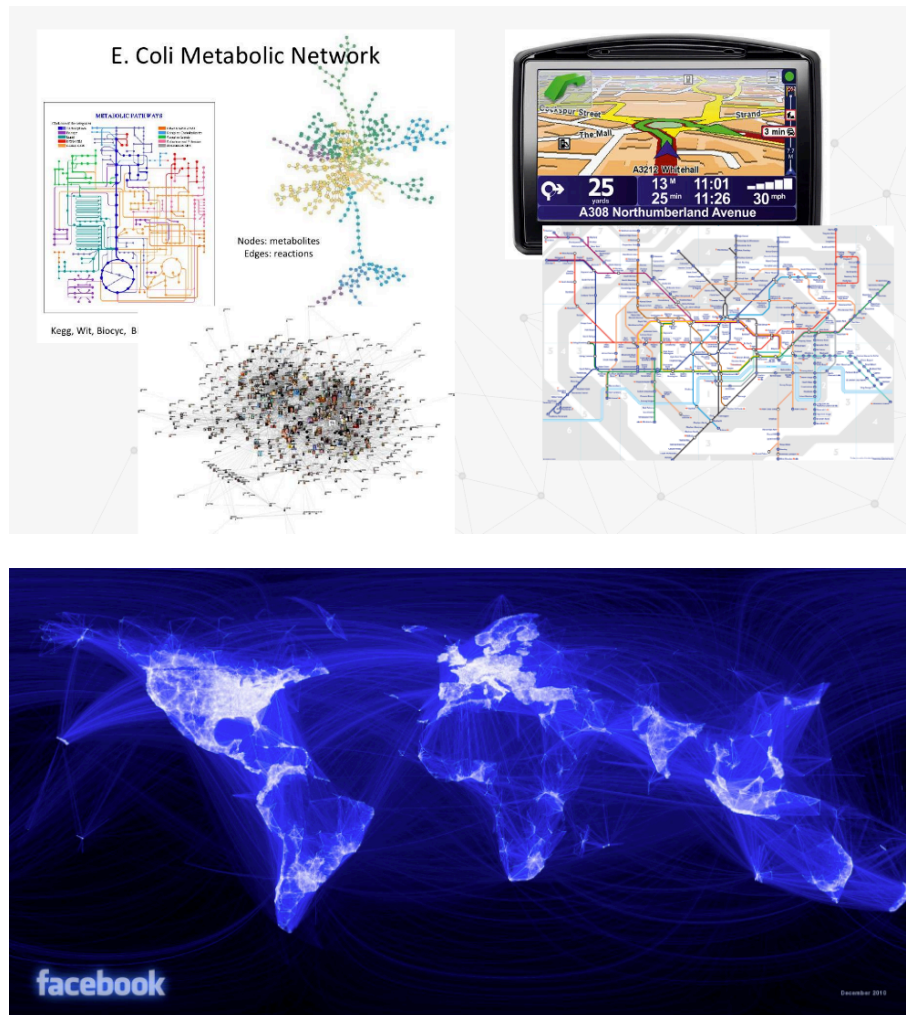
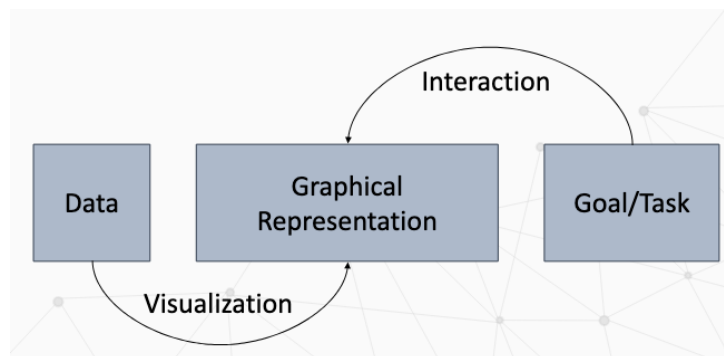


Figure 126: Visualisation of Facebook friendship

Nicholas Christakis explores how the large-scale, face-to-face social networks in which we are embedded affect our lives, and what we can do to take advantage of this fact. How social network are important in your life.²⁸

How to decide which representation to use for which type of graph in order to achieve which kind of goal?



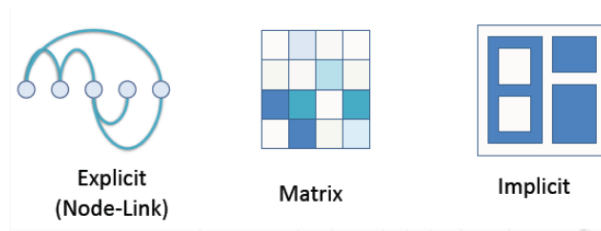
²⁸http://www.ted.com/talks/nicholas_christakis_the_hidden_influence_of_social_networks

Two principal types of tasks: **attribute-based (ABT)** and **topology-based (TBT)**

- Localise — find single or multiple nodes/edges that fulfil a given property
 - ABT: find edge(s) with maximum edge weight
 - TBT: find all adjacent nodes of a given node
- Quantify — count or estimate a numerical property of the graph
 - ABT: give the number of all nodes
 - TBT: give the in-degree (no. of incoming edges) of a node
- Sort/Order — enumerate nodes/edges according to a given criterion
 - ABT: sort all edges according to their weight
 - TBT: traverse the graph starting from a given node

9.1. Graph Representations

Different visual representation support attribute based task or topology based task differently. There are three main types of graph representations that we use:

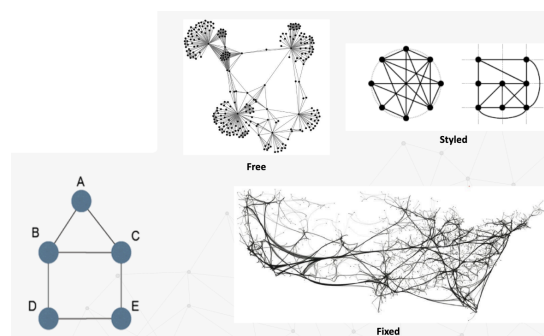


9.1.1. Explicit (Node-Link) Representation

Node-link diagrams use vertex = point and edge = line/arc.

Schulz identifies 3 cases of degrees of freedom:

- free layout: node layout is not restricted and can thus be generated with a wide variety of methods, e.g. force directed graphs
- fixed layout: only trajectory/path of edges can be determined freely
- styled layout: node layout satisfies a predefined scheme; has advantage that overall style of graph visualisation is known beforehand



Criteria for good node-link layouts

- Minimized edge crossings
- Minimized distance of neighboring nodes
- Minimized drawing area
- Uniform edge length
- Minimized edge bends
- Maximized angular distance between different edges
- Aspect ratio about 1 (not too long and not too wide)
- Symmetry: similar graph structures should look similar

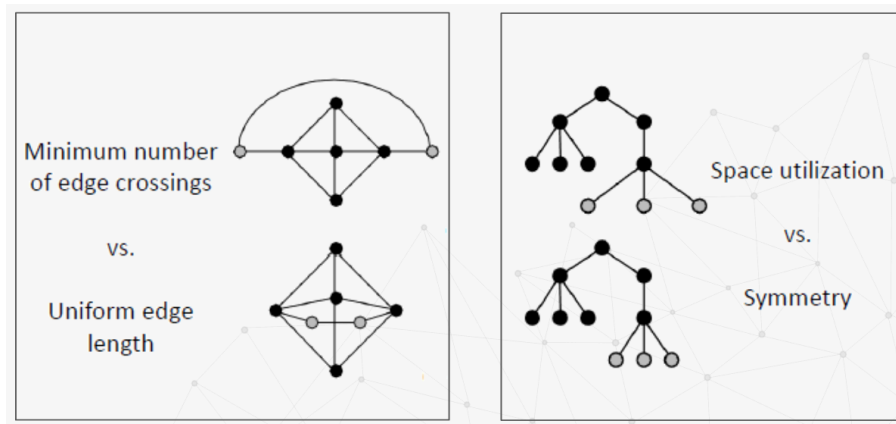


Figure 130: Often times this criteria are conflicting and we cannot respect them all.

Pros of Explicit	Cons of Explicit
Can depict all graph classes	Computation of an optimal graph layout is in NP (Even just achieving minimal edge crossings is already in NP)
Can be customised by weighing the layout constraints	Even heuristics based approaches are still slow-complex (naive spring embedder is in $O(n^2)$)
Good fit to support topology based tasks (TBTs)	Tendency to clutter (edge clutter, 'hairball' effect)

Addressing Layouts in ER

Problem is computing an optimal layout lies in NP.

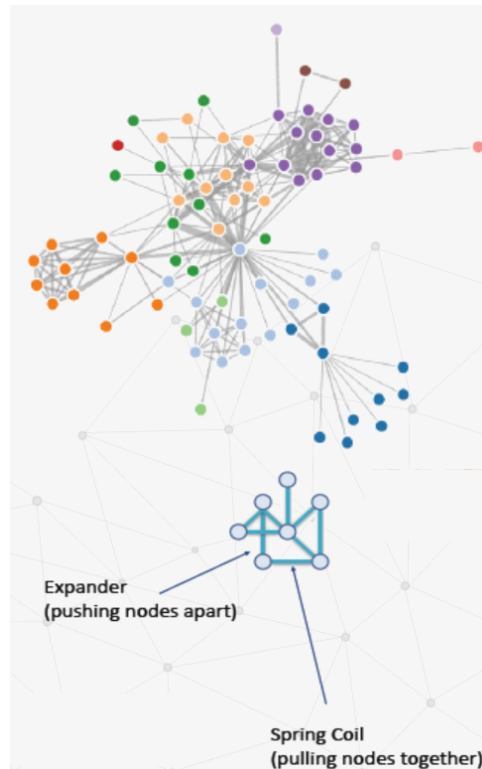
Solution approach is to formulate the layout problem as an optimisation problem.

1. Convert layout criteria to weighted cost function

$$F(\text{layout}) = a \times |\text{edge crossings}| + \dots + f \times |\text{used drawing space}|$$

2. Use standard optimisation technique to find a layout that minimises the cost function. Common approach is to use **force-directed layouts**.

9.1.1.1. Force-directed Layouts



In **Force-directed Layouts** the graph drawing algorithm strategy is that to assign forces among the set of edges and the set of nodes of a graph. In this layouts there are two main forces at play: an attractive and a repulsive force.

- The attractive forces are spring-like attractive forces behaving according to Hooke's law and are used to attract pairs of endpoints of the graph's edges towards each other.

$$F = kX$$

- Repulsive forces act simultaneously to attractive forces but follow repulsive laws similar to those of forces of electrically charged particles based on Coulomb's law and are used to separate all pairs of nodes.
- In practice, we also employ damping; employed to reach an equilibrium state faster.
In physics, damping is any effect that tends to reduce the amplitude of vibrations. In mechanics, the internal friction may be one of the causes of such damping effect.
- Edge attraction and vertex repulsion forces may be defined using functions that are not based on the physical behavior of springs and particles; for instance, some force-directed systems use springs whose attractive force is logarithmic rather than linear.

Algorithm

1. Define forces on nodes and edges of the graph
 2. Apply forces to the nodes by pulling them closer together/pushing them further apart
- Repeat until mechanical equilibrium
 - Positions of the nodes in equilibrium are used to draw the graph.
 - Spring length: proportional to the graph-theoretic distance.

There are other possible approaches. Search for minima options:

- Stress minima: stress majorization. Ideal when springs ideal length is proportional to the graph-theoretic distance
- Energy minima: global optimization methods like simulated annealing and genetic algorithms.

For forces defined from springs whose ideal length is proportional to the graph-theoretic distance, stress majorization gives a stable (i.e., monotonically convergent) and mathematically elegant way to minimise these differences and, hence, find a good layout for the graph. It is also possible to employ mechanisms that search more directly for energy minima, either instead of or in conjunction with physical simulation. Such mechanisms, which are examples of general global optimization methods, include simulated annealing and genetic algorithms.

Note: Stress majorization is an optimization strategy used in multidimensional scaling (MDS) where, for a set of n m -dimensional data items, a configuration X of n points in $r(<m)$ -dimensional space is sought that minimizes a so-called stress function.

Benefits of Force-Directed Layouts

Pros	Cons
Generally good quality results for graphs with nodes $\ll 1000$, uniform edge length, uniform vertex distribution, good to identify clusters, preserve symmetry	Computationally expensive: $O(n^3)$ in general, can be optimized to $O(n^2)$
Intuitive by being based on physical analogies of common objects, like springs, the behavior of the algorithms is relatively easy to predict and understand.	Interactive limit is about 1000 nodes.
	• Careful: Proximity does not mean actual grouping • Careful: Non deterministic • Careful: Spatial position does not encode an attribute

9.1.1.2. Other Issues

Scalability is another limit of node link diagrams in general, I mentioned before the Hairball effect. For large graphs, which are quite frequent, the result can be a giant hairball Where we really cannot see much structure anymore.

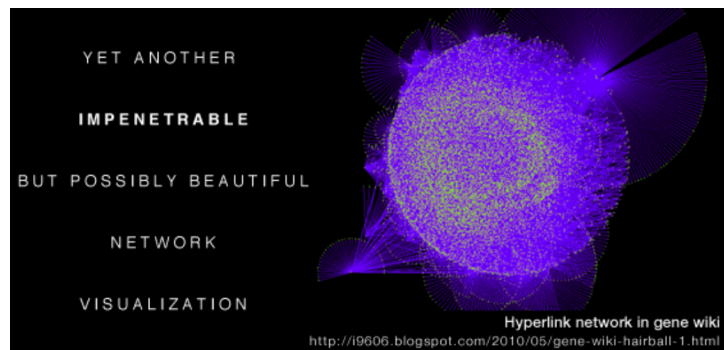
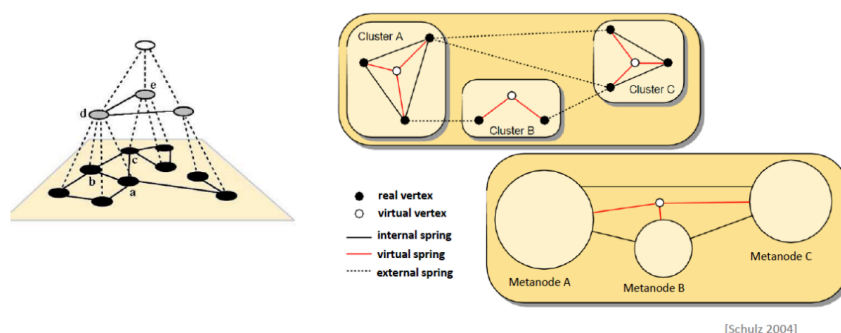


Figure 132: Hairballs – HivePlot.com

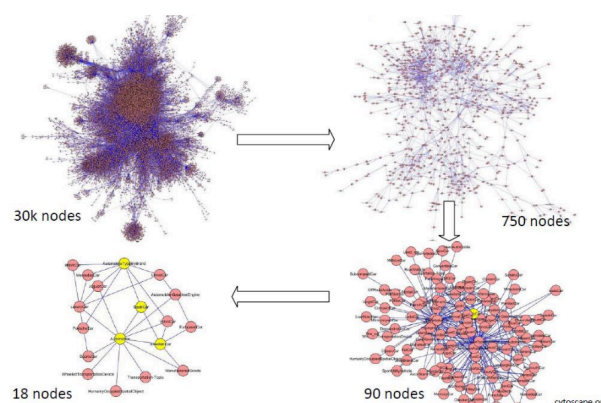
Addressing Computational Scalability: Multilevel Approaches

One way to address scalability is to use multi-level approaches. The idea here is to do a pre-processing step where you look at nodes that are in your neighborhood and start creating clusters based on proximity or other similarity criteria. In the context of force directed layout this implies that we first run a clustering algorithm on the network to create clusters and then we only perform pairwise force simulation within the nodes in a cluster. This is a strategy to reduce the complexity of each iteration of the force directed graph layout algorithm down to $O(n \log n)$.



Addressing Computational Scalability: Abstraction/Aggregation

Another way of addressing Scalability issues is by thinking in terms of what we call “super nodes”; To be more precise: super nodes and aggregation; When dealing with large networks it makes sense to think at nodes not as separate entities but rather in terms of their connections.



Example: Collapsible Force Layout

- Supernodes: aggregate of nodes
- Manual or algorithmic clustering

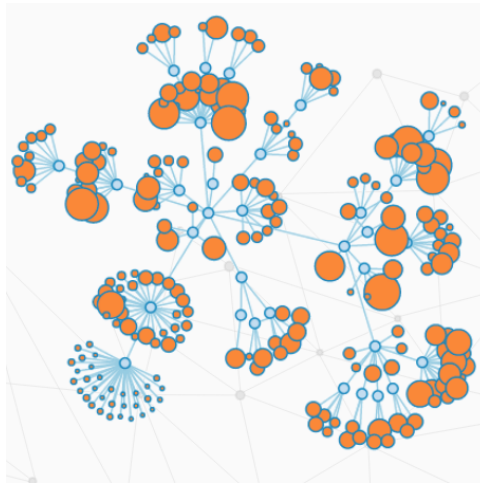
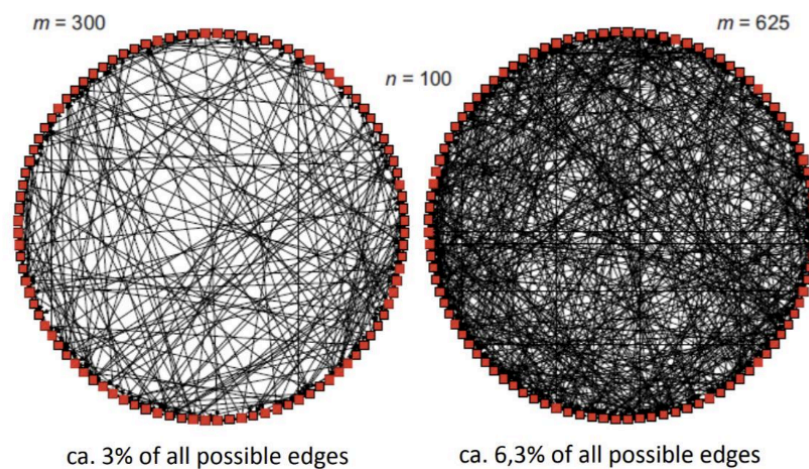


Figure 135: ²⁹

9.1.2. Styled Restricted Layouts

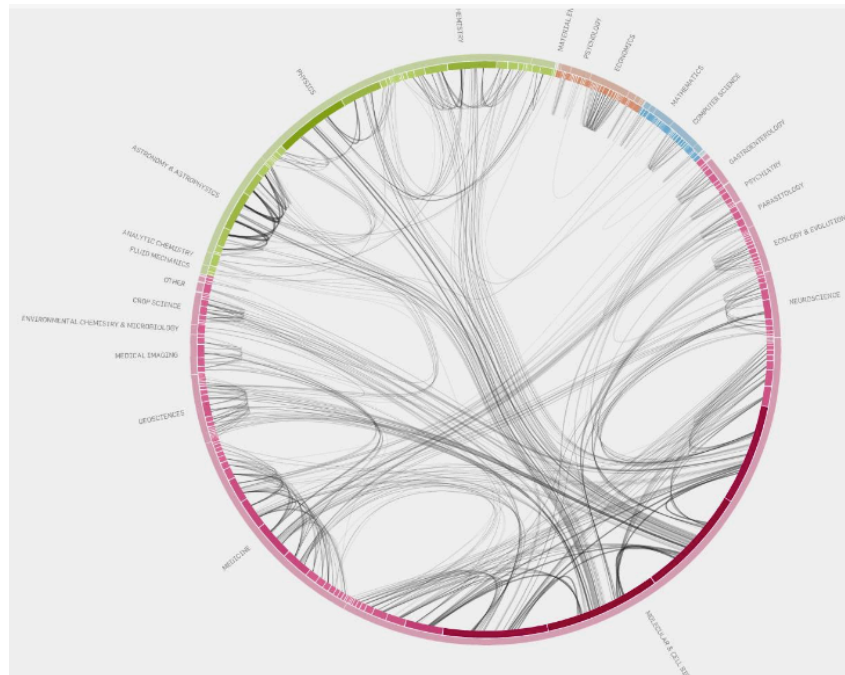
- Circular Layout
- Node ordering
- Edge clutter



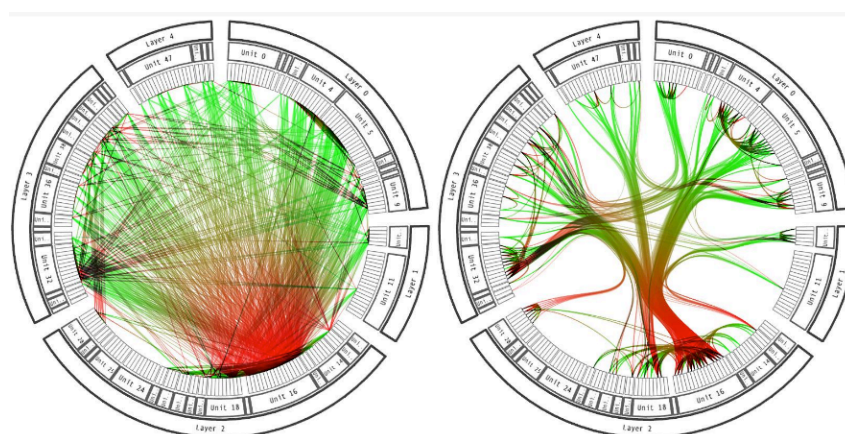
When radial/circular layouts are used for network/graph representations a common problem is that the layout tends to clutter fairly quickly. The image on the left here shows three percent of all possible edges while the one on the right figure shows about six percent of all possible edges. We see easily that things get fairly cluttered quickly. One way to address clutter in this case is to try to optimize node ordering so that nodes that are connected are close to each other, think at small-world network layout we have seen in our previous lecture.

²⁹<https://gist.github.com/mbostock/1062288>

Edge Bundling / attribute-driven layout is generated according to any numerical edge attributes such as directions, timestamps or weights.



Edge bundling allows to visualize adjacency relations between entities organized in a hierarchy. The idea is to bundle the adjacency edges together to decrease the clutter usually observed in complex networks. This visualization provides an overview of the whole citation network data from Thomson Reuters' Journal Citation Reports 1997–2005. The colors represent the four main groups of journals, which are further subdivided into fields in the outer ring. The segments of the inner ring represent the individual journals, scaled by Eigenfactor Score. The initial view plots the top 1000 citation links. Line size and opacity represents connection strength. The Bezier curves follow the hierarchical cluster structure, using the edge bundling technique known as hierarchical edge bundling. Selecting a single journal (inner ring) or whole field (outer ring) displays all citation flow coming in or out of the selection. The color is based on the cluster color of the origin node.



Holten proposes “Hierarchical edge bundling” as a method of bundling those adjacent edges together. He draws an analogy to bundling your electrical wires together in order to reduce clutter, while fanning them out at their terminus in order to plug them in.

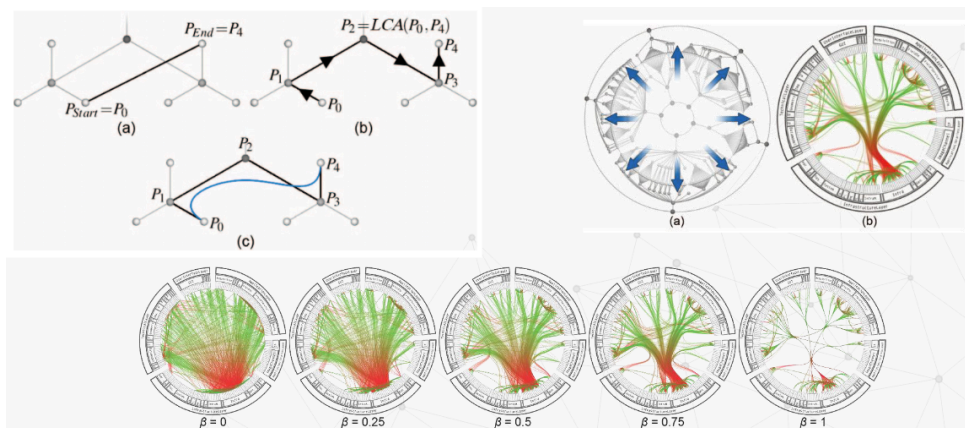
How do they work?

- Radial layout construction.
 - A radial tree layout is used for the inner circle and subsequently mirrored to the outside;
 - the inner layout is hidden and its structure is used to guide the adjacency edges.
- An icicle plot based on the mirrored layout is used to show the hierarchy.

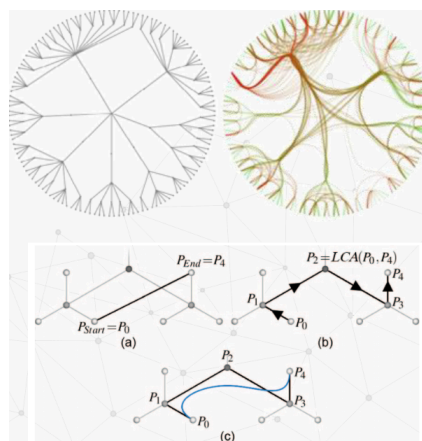
The idea is of a tree structure guiding the bundling:

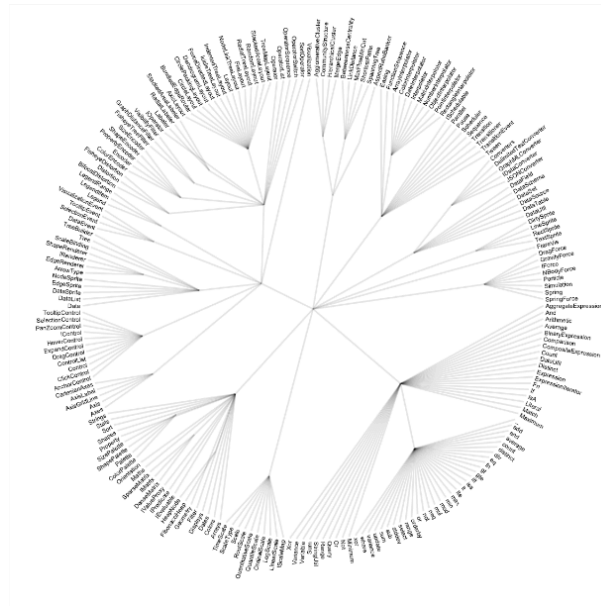
- we have an edge between two nodes: P_{start} and P_{end} .
- We virtually build the tree structure here in the center and we reroute the edges through this tree structure. Using a spline produces this nice bundles here.

You can adapt the bundling strength dynamically, here we have a value of 0 but we can increase getting more and more abstract.

**Algorithm in Detail**

- Given a graph G
- First determine the graph Hierarchy
 - Given/explicit
 - To be Computed - e.g. community detection algorithms, clustering tree algorithm
 - Leaves are nodes of original graph G
- Layout the hierarchy radially
- Use the tree as a skeleton to route the edges of G between its nodes
- The nodes of the tree act as B-spline control points
- Each edge in the original graph is modelled as a single B-spline using the control points along the shortest path in the hierarchy from one end node to the other in the hierarchy

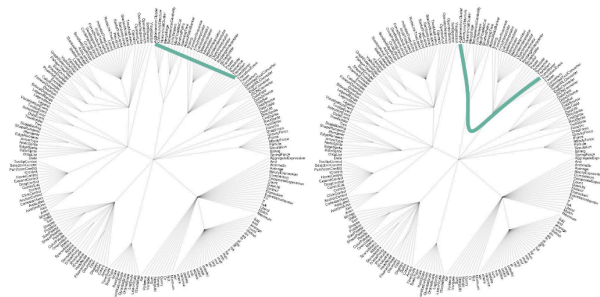


Figure 141: Example Flare Dataset from Rproject³⁰

Hierarchy of Flare (could also use a dendrogram)

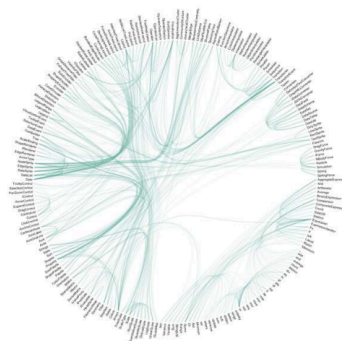
- Folders: query, data etc.
- Subfolders

There are dependencies across elements e.g. some elements call other elements when they are used.



A naive approach to represent this connection would be to draw a straight line (left). Instead, hierarchical edge bundling uses a curve that follows the hierarchy link between the 2 elements (right).

Apply HEB to every adjacency connection of the dataset:



³⁰the thing: <https://blocks.org/micahstubbs/50c3189d7>

Bundling Strength

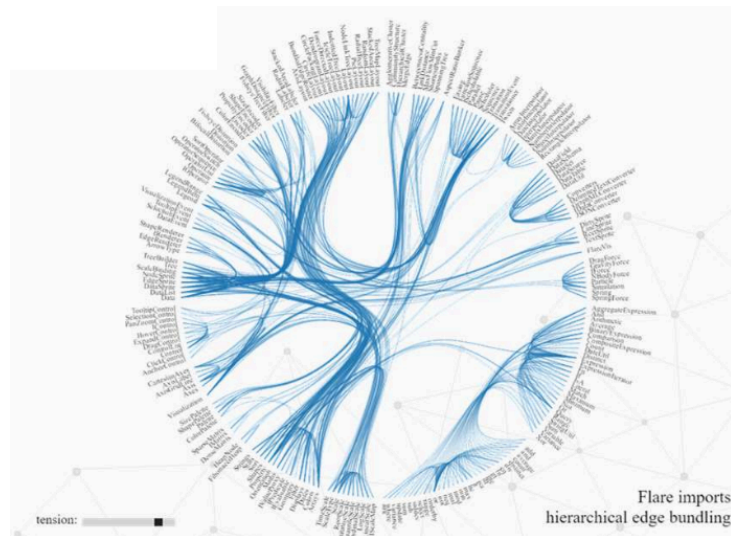
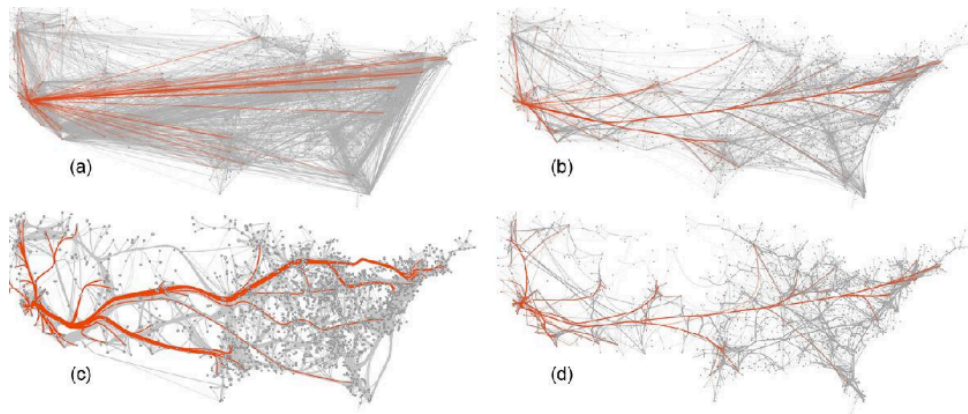


Figure 144: Example where you can interactively change the bundling strength³¹

Bundling is effective when:

- Can't vary position of nodes
- Edge routing important



³¹<https://mbostock.github.com/d3/talk/20111116/bundle.html>

9.1.2.1. Explicit Graph Representation: Trees

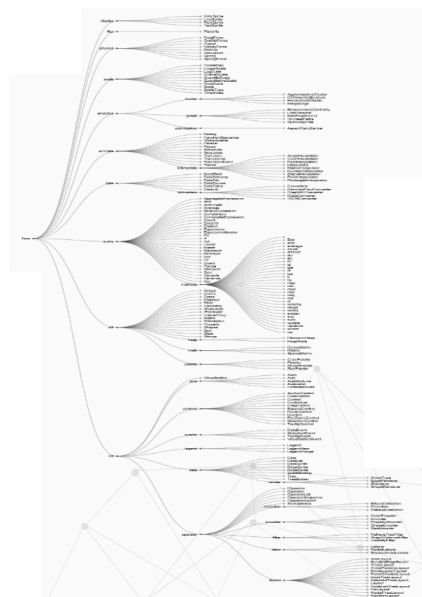
Radial Node-Link diagrams

- Data: Tree
- Encoding
 - link connection marks
 - point node marks
 - radial axis orientation
 - angular proximity: siblings
 - distance from centre: depth in tree
- Tasks: understanding topology, following paths
- Scalability: 1K - 10K nodes



Explicit tree visualization: Reingold-Tilford layout

In the linear layout we go left to right; In the circular we go from the center radially outwards; The radial layout is more compact but the linear is easier to follow.



For further reading: Reingold-Tilford Algorithm

- Goal: make smarter use of space, maximize density and symmetry.
- Originally for binary trees, extended to cover general case
- Linear algorithm
- Starts with bottom-up pass of the tree Y-coord by depth, arbitrary starting X-coord
- Merge left and right subtrees
 - Shift right as close as possible to left
 - “Shifts” in position saved for each node as visited
 - Parent nodes are centred above their children
- Top-down pass for assignment of final positions
 - Sum of initial layout and aggregated shifts

Design suggestions for explicit trees

- Clearly encode depth level
- No edge crossings
- Isomorphic subtrees drawn identically
- Ordering and symmetry preserved
- Compact layout (don’t waste space)

Limitations

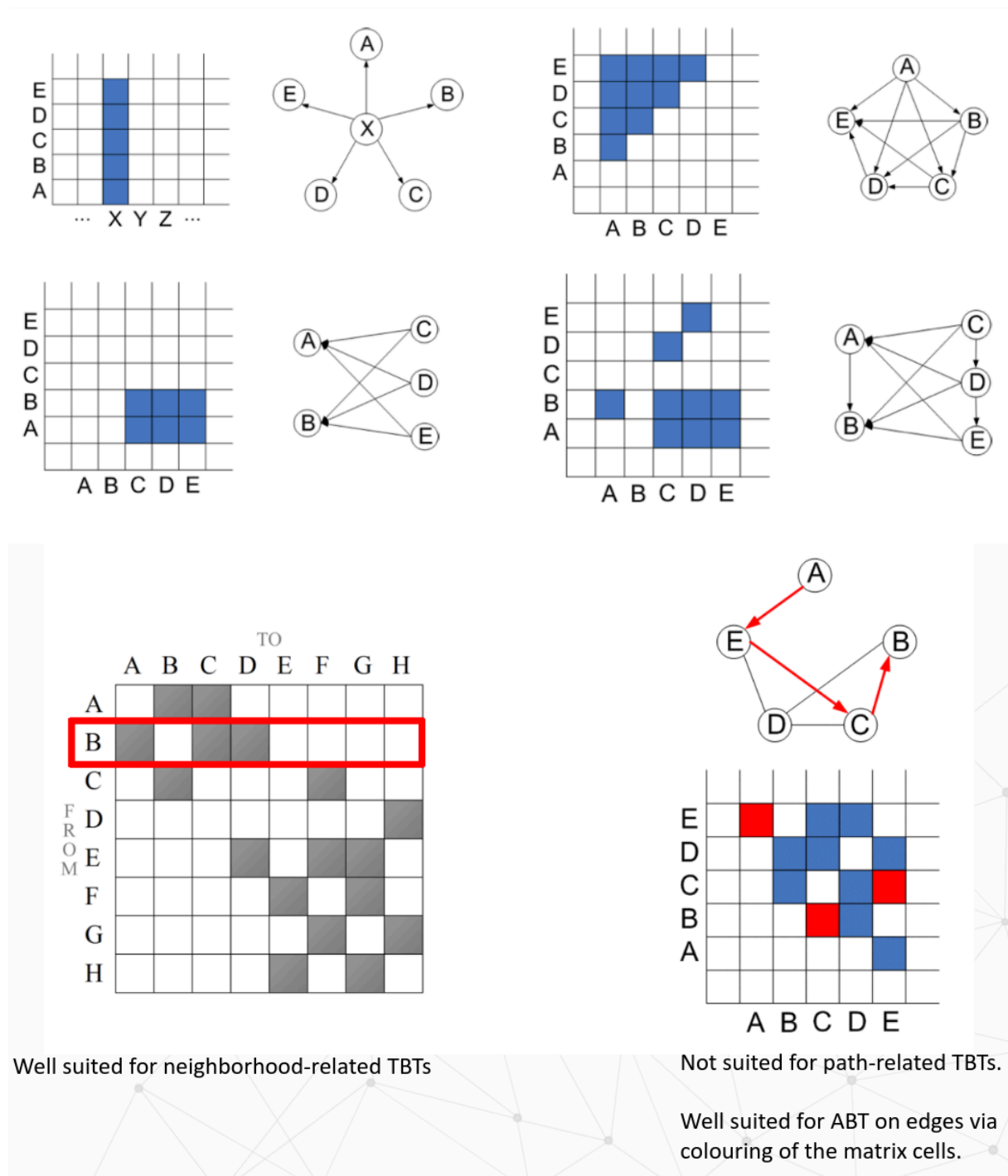
- Tree breadth often grows exponentially
- Even with tidier layout, quickly run out of space

Pros and Cons of Explicit Representations (ER)

- Pros:
 - Can depict all graph classes
 - Can be customized by weighing the layout constraints
 - Good fit to support TBTs,
- Cons:
 - computation of an optimal graph layout is in NP (even just achieving minimal edge crossings is already in NP)
 - even heuristics based approaches are still slow/complex (e.g., naïve spring embedder is in $O(n^2)$)
 - tendency to clutter (edge clutter, “hairball” effect)

9.1.3. Matrix Representation

Instead of node link diagram, use adjacency matrix; visual encoding: cell shows presence/absence of edge; scalability: 1K nodes, 1M edges



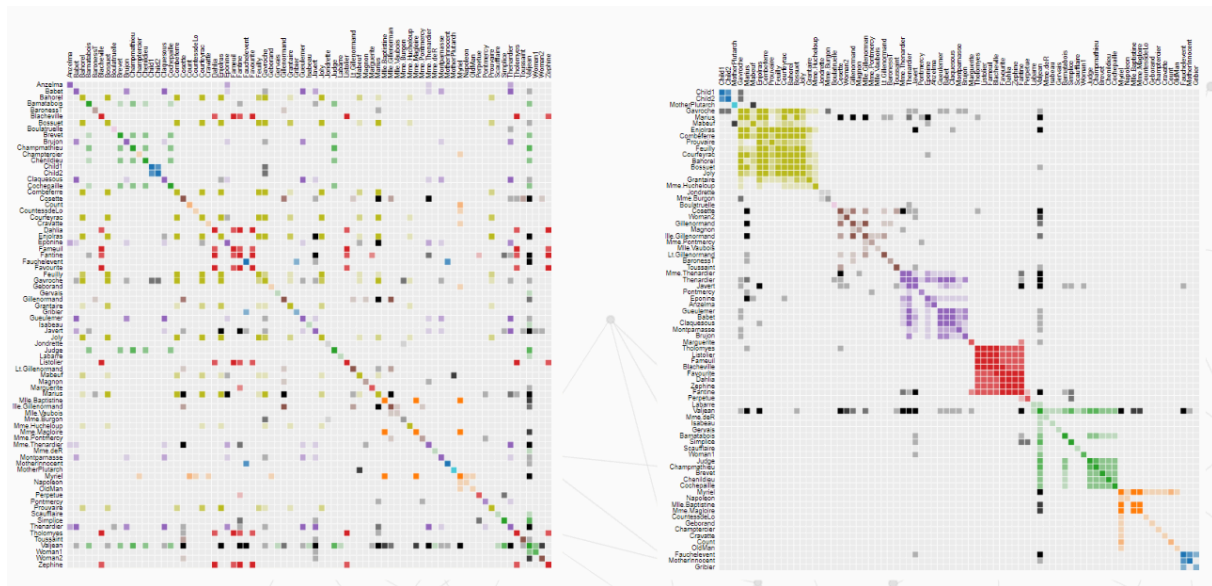


Figure 150: Order matters in matrix representation!

Pros

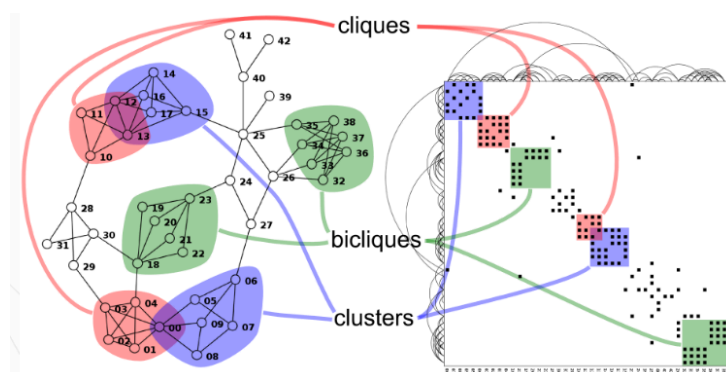
- can represent all graph classes except for hypergraphs
- puts focus on the edge set, not so much on the node set
- simple grid -> no elaborate layout or rendering needed
- well suited for ABT on edges via colouring of the matrix cells
- well suited for neighbourhood-related TBTs via traversing rows/columns

Cons

- quadratic screen space requirement (any possible edge takes up space)
- not suited for path-related TBTs

Connection vs Adjacency - Comparison

- Adjacency matrix strengths: predictability, scalability, supports reordering, some topology tasks, some training needed
- Node-link diagram strengths: topology understanding, path tracing, intuitive, no training needed
- Empirical studies findings: node-link best for small networks, matrix best for large networks (if tasks do not involve topological structure!)



9.1.4. Implicit Representation

In implicit representations we use nested structures to represent the topology of a network, they work well for trees but not graphs.

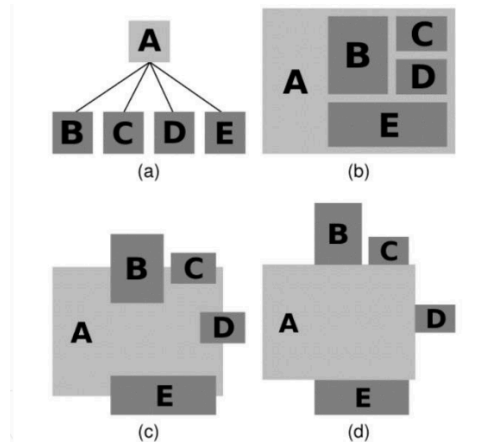
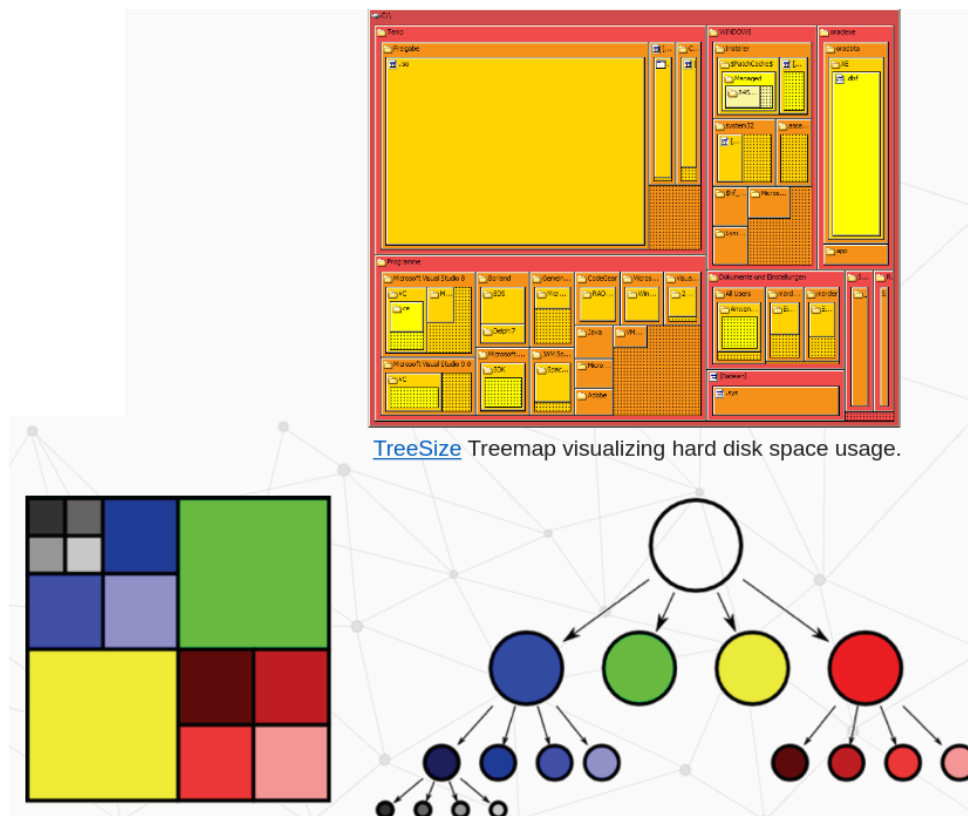


Figure 152: (a) Explicit node-link layout, (b) Implicit layout by inclusion, (c) Implicit layout by overlap, (d) Implicit layout by adjacency

Visualization by Nesting: Treemaps

- Data: Tree e.g. Hierarchical data; 1 quantitative attribute at leaf nodes
- Encoding: Area containment marks for hierarchical structure, Rectilinear orientation, Size encodes quantitative attribute
- Tasks: Query attribute at leaf nodes
- Scalability: 1M leaf nodes



9.1.4 Graph Representations > Implicit Representation

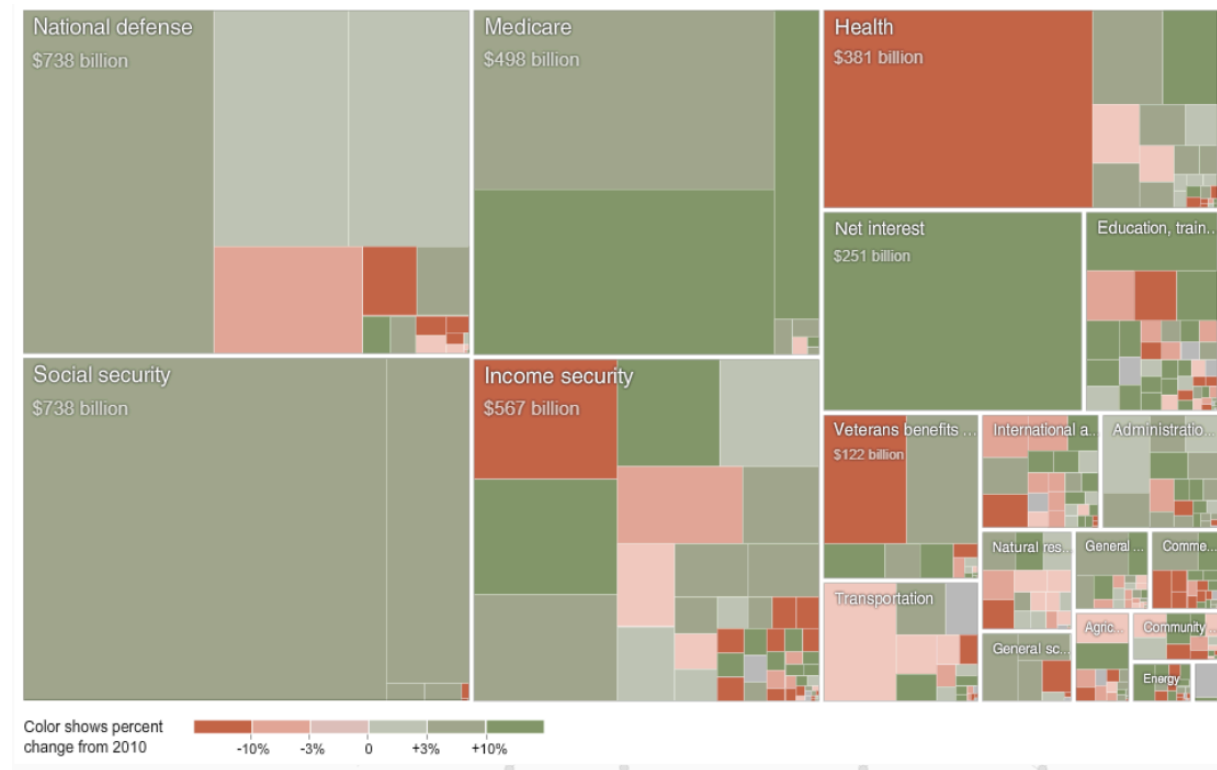


Figure 154: Obama's 2011 Budget Proposal

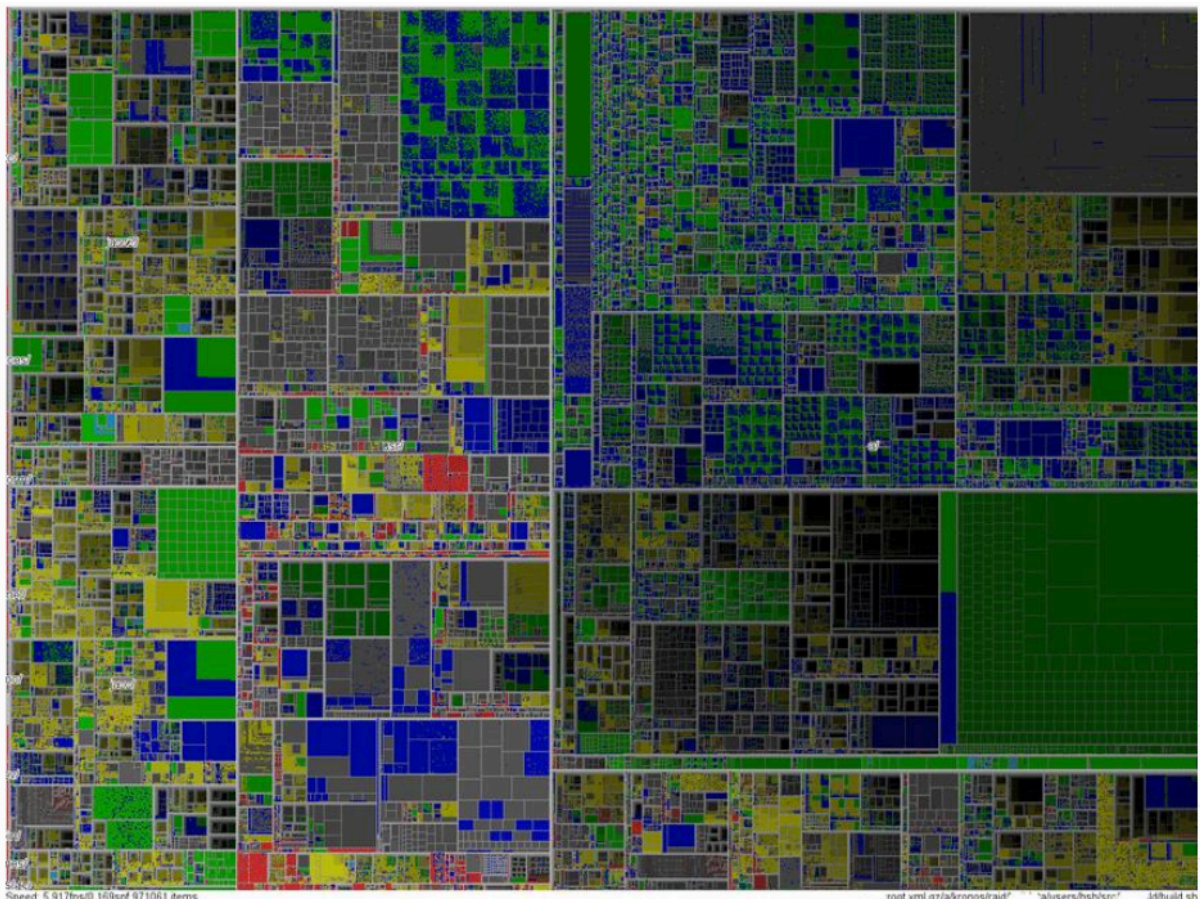
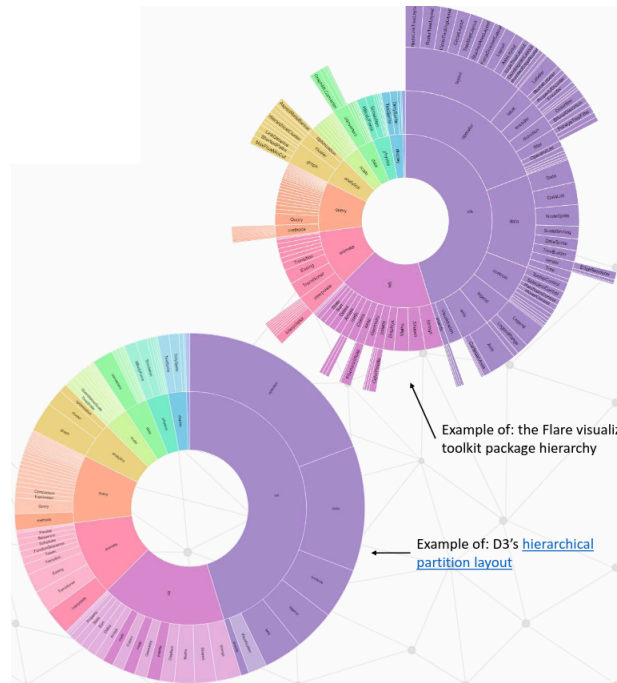


Figure 155: 1M Point Treemap

Visualization by Adjacency: Sunburst Radial Layout

The SunBurst technique is an alternative, space-filling visualization that uses a radial rather than a rectangular layout. Items in a hierarchy are laid out radially, top of the hierarchy at the centre, deeper levels farther away from the centre. The angle swept out by an item and its colour correspond to some attribute of the data.



Implicit Representations Pros and Cons

Pros

- Space-efficient because of the lack of explicitly drawn edges, scale well
- Well suited for ABTs on the node set
- Also useful for some TBTs

Cons

- Can only represent trees
- No free arrangement (maps)
- Useless for edge task

References

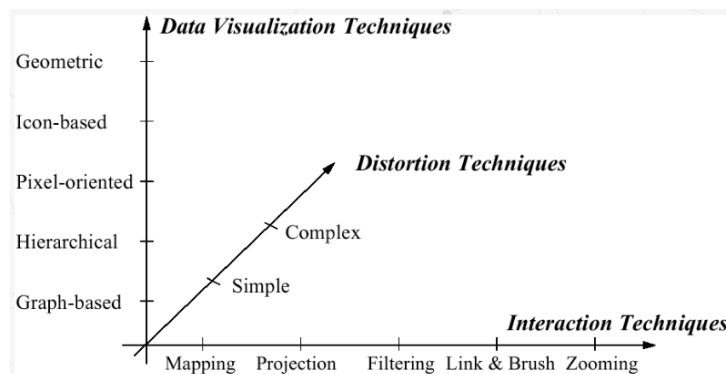
- <https://treviz.net>
- <https://gephi.org>
- <https://cytoscape.org>
- <https://js.cytoscape.org>
- <https://networkx.github.io>

10. Dimensionality Reduction Techniques

In multidimensional visualization, there is no standard mapping into the Cartesian coordinate system: data does not have some inherent two-/three-dimensional semantics.

Techniques for visualizing multidimensional data sets can be classified using three orthogonal criteria:

- visualization technique
- distortion technique
- interaction technique



Major categories:

- Geometric Techniques
- Icon-/Glyph-based Techniques
- Pixel-oriented Techniques
- Hierarchical Techniques
- Graph-Based Techniques
- Hybrid Techniques: arbitrary combinations from above

Interaction and Distortion

- Interaction
 - Single View (e.g. semantic zooming, filtering, focus+context ...)
 - Multiple Views (e.g. linking & brushing, details on demand ...)
- Distortion
 - Simple Distortion (e.g., lenses effect, fisheye views, ...)
 - Complex Distortion (e.g., hyperbolic projections ...)

10.1. Pixel-Oriented Techniques

Basic Idea:

- each attribute value is represented by one colored pixel
 - the value ranges of the attributes are mapped to a fixed colormap
- the attribute values for each attribute are presented in separate subwindows

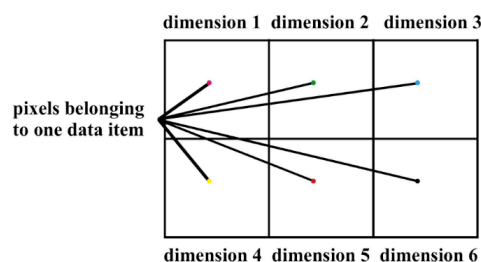


Figure 158: Example of basic arrangement of subwindows for data with 6 dimensions

10.1 Dimensionality Reduction Techniques > Pixel-Oriented Techniques

For a data set of m dimensions, create m windows on the screen, one for each dimension: The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows; The colors of the pixels reflect the corresponding values



Scalability

- 1M items,
- 100s of categorical levels,
- 10 quantitative attribute levels

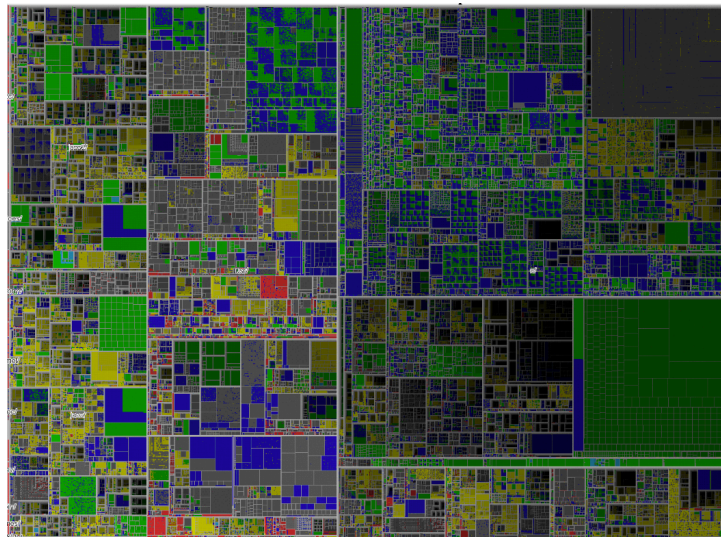
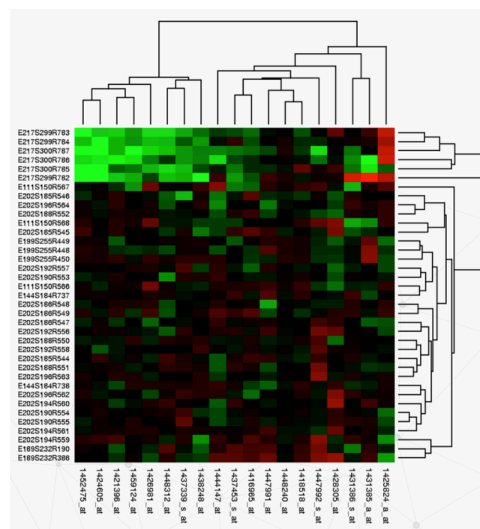


Figure 160: One Million Items Treemap

One example is **heatmaps**, cluster heatmap, re-ordered matrix ordering on 2 attributes, goal is to group similar cells (clusters, patterns); includes trees drawn on edges (dendrograms); trees show hierarchical clustering of data (rows/columns) and ordering of data according to hierarchy

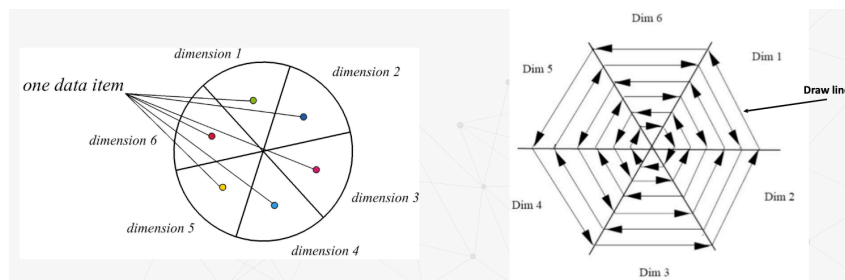


The next important question is whether there exists an alternative to the regular partitioning of the screen into rectangular subwindows.

- The rectangular shape of the subwindows -> good screen usage, but ...
- The rectangular shape of the subwindows -> dispersal of the pixels belonging to one data object over the whole screen.

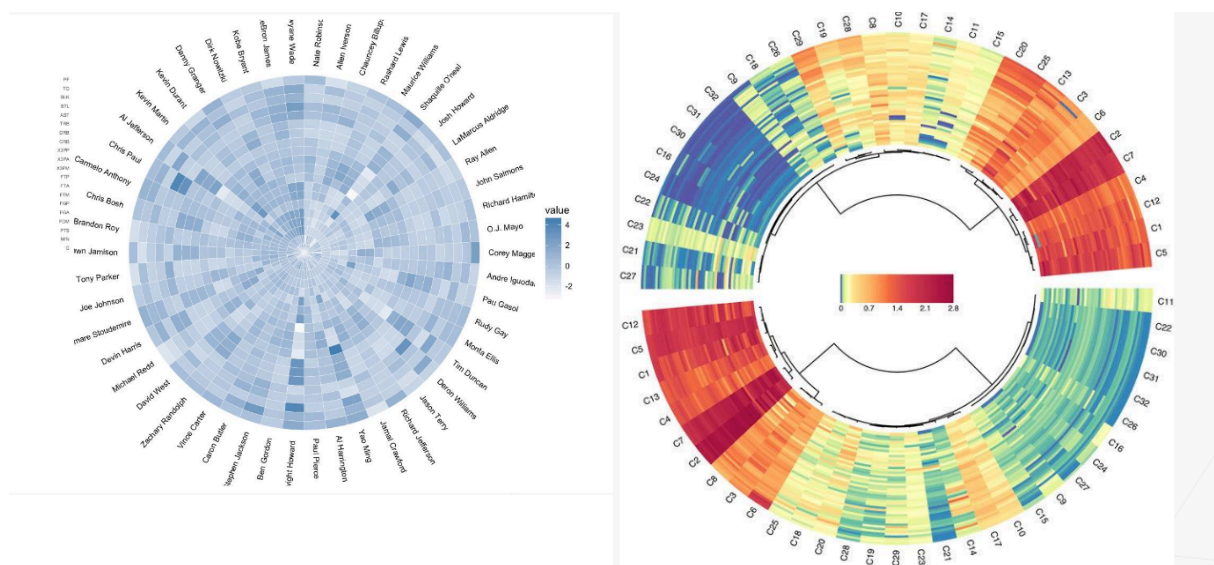
Circle Segment approach: display the data dimensions as segments of a circle

If the data consists of k dimensions, the circle is partitioned into k segments, each representing one data dimension.



Data objects within one segment are arranged in a back and forth manner along the “draw_line”, orthogonal to the line that halves the two border lines of the segment

- The “draw_line” starts in the center of the circle and draws the pixels from one border line of the segment to the other.
- Whenever the “draw_line” hits one of the border lines, the “draw_line” is moved in parallel along the segment-halving
- line to the outside of the circle and the direction of the “draw_line” changes.
- The process is repeated until all data objects of one dimension are visualized and, then, the whole procedure is restarted for the remaining dimensions

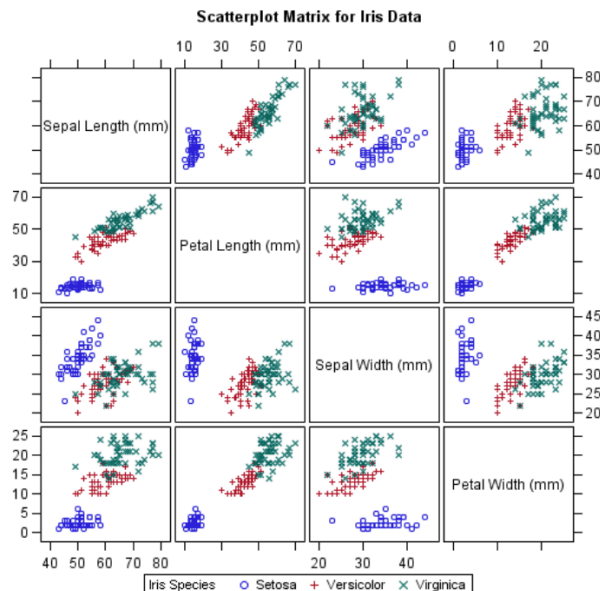


10.2. Geometric Techniques

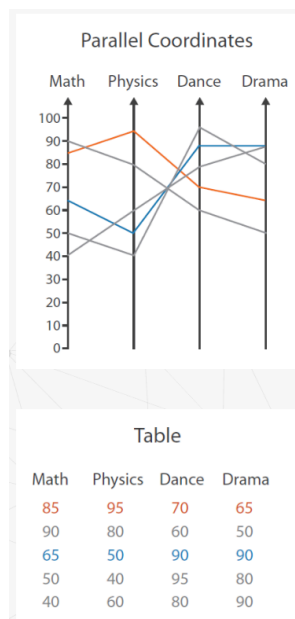
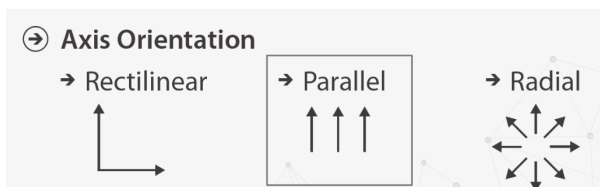
Geometric Techniques: visualisation of geometric transformations and projections of the data, e.g. scatterplot-matrices, parallel coordinates

10.2.1. Scatterplot Matrices

Scatterplot Matrices (SPLOM): each cell contains an entire scatterplot chart; shows all possible pairwise combinations of attributes; task is to find correlation, trends, outliers; scalability to 12 attributes and hundreds of items



10.2.2. Space Re-organization: Changing Axis



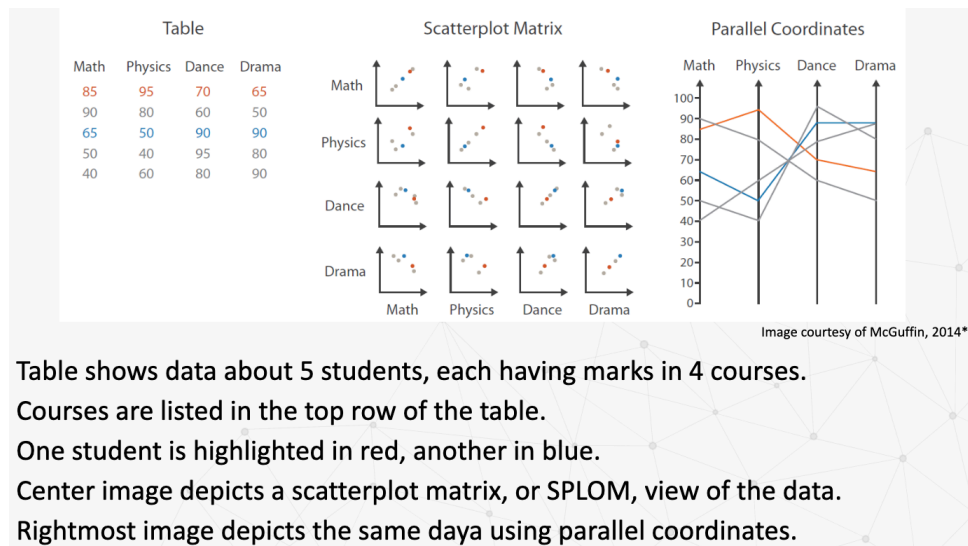
Parallel Coordinates

- To show correlation between neighbouring axis
- Dot in scatterplot corresponds to line in parallel coordinates
- Similarly to scatterplots we look for patterns
- Ideal for comparing many variables together and seeing the relationships between them.
- Example, if you had to compare an array of products with the same attributes (comparing computer or cars specs across different models).
- Each axis can have a different scale, as each variable has a different unit of measurement, or all the axes can be normalised to keep all the scales uniform

Basic Idea:

- Every Cartesian coordinate becomes a parallel axis
- Coordinate values are mapped to the respective axis
- Values on the parallel axis are joint together to form a line
- Points in Cartesian coordinates become line in Parallel coordinates

10.2.2 Geometric Techniques > Space Re-organization: Changing Axis

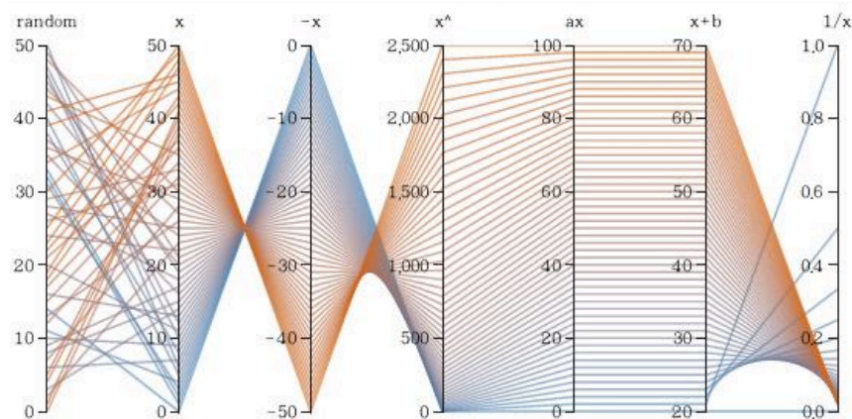


Parallel Coordinates uses parallel axes (rectilinear axis, typically vertical and equally spaced, one axis per dimension); task is to find correlation (SPLOM are better), trends, outliers, extremes, ranges of attributes; scalability is proportional to the no. of axis, hundreds of items as points on axis; limitations are each axis can have at most two neighbour axes, ordering of axes can hinder patterns, sufficient space between axis to distinguish patterns of intersection or parallelism

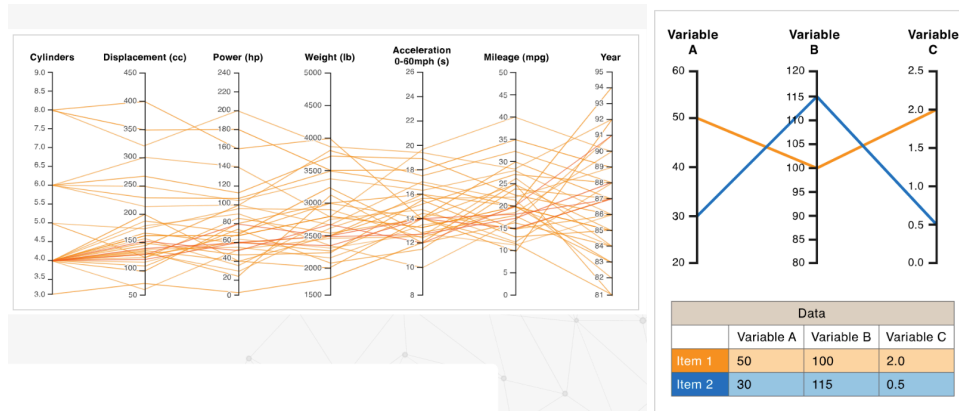
Space between axis is necessary to be able to read relational patterns.

How to visually read out parallel coords' relational patterns:

- positive correlation: parallel line segments
- negative correlation: all segments at halfway point
- uncorrelated: scattered crossings



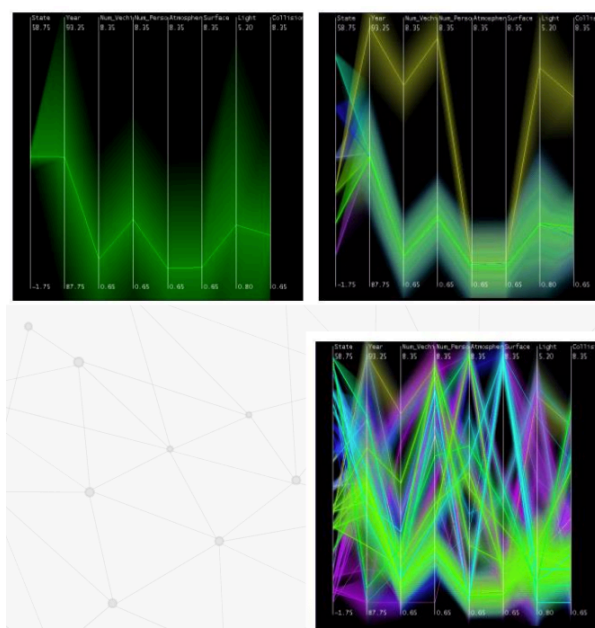
The downside to Parallel Coordinates Plots, is that they can become over-cluttered and therefore, illegible when they're very data-dense. The best way to remedy this problem is through interactivity and a technique known as “Brushing”. Brushing highlights a selected line or collection of lines while fading out all the others. This allows you to isolate sections of the plot you're interested in while filtering out the noise. Interaction is important to manage overcrowding.



Hierarchical Parallel Coordinates

Provides multiple level of details

- Uses interactively controlled aggregation
 - Increase scalability
 - Items 10000 to 100000, clusters 12
- Statistics for each cluster
- Clusters encoded with band of varying width and opacity
 - Mean in the middles
 - Width at each axis = cluster min and max at the axis
- Full plot shows:
 - 230,000 records in a fatal accident database
 - Variable opacity bands show meaningful clusters

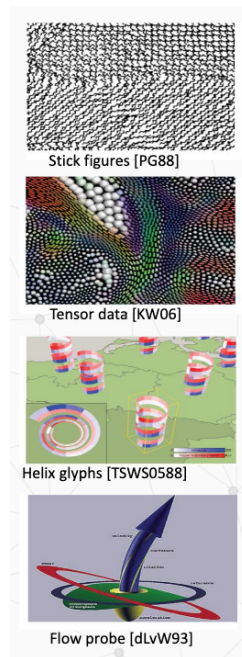


10.3. Icon-/Glyph-based Techniques

Loose definition: glyphs are visual entities which convey one or more data values via attributes such as shape, size, color, and position.

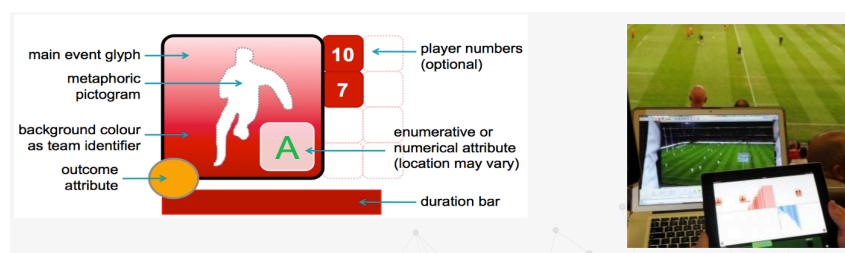
More formal definition [Ward02]: glyphs are parameterized geometric objects with p components, each of which may have r geometric attributes and s appearance attributes:

- geometric attributes: shape, size, orientation, position, direction/magnitude of motion
- appearance attributes: color, texture, and transparency.

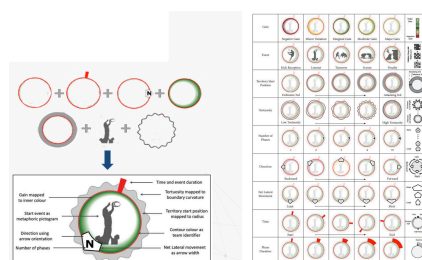


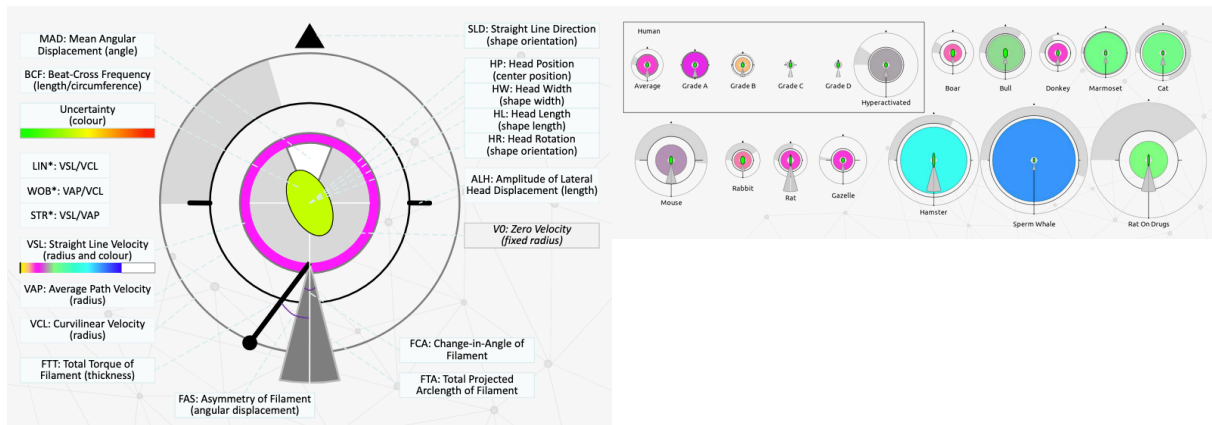
- Each attribute encodes a specific value of the underlying data.
- The encoding allows to integrate multiple variables within a single item and therefore the creation of a unique image, or visual signature, to be constructed for each data point.

Example: SportVis



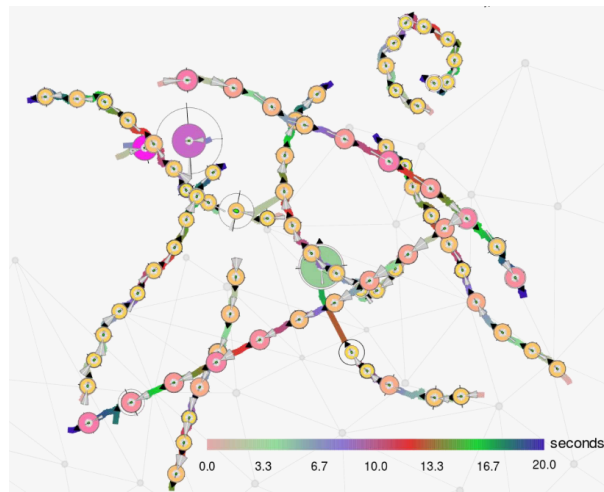
Each attribute encodes a specific value of the underlying data. The encoding allows to integrate multiple variables within a single item and therefore the creation of a unique image, or visual signature, to be constructed for each data point.



Glyph-Based Video Visualization for Semen Analysis B. Duffy, et al. TVCG 2014

Furthermore:

- Glyphs can be placed independently in a display space
- Or can be spatially connected to convey the topological relationships between data records or geometric continuity of the underlying data space



10.4. Dimensionality Reduction

We want to visualize high dimensional data; Some features may be irrelevant; “Intrinsic” dimensionality may be smaller than the number of features

Three Objectives:

1. Reduce high dimensional to lower dimensional space
2. Preserve as much of variation as possible
3. Plot lower dimensional space

How to find the ‘best’ low dimension space that conveys maximum useful information?

- Traditional Methods
 - Principal Component Analysis (PCA): Linear mapping, by order of variance
 - Multidimensional Scaling: Non linear, popular in text analysis
- Recent Methods
 - t-Distributed Stochastic Neighbor Embedding (t-SNE)
 - Uniform Manifold Approximation and Projection (UMAP)

10.4.1. Principal Component Analysis

It is a mathematical tool from applied linear algebra. It provides a roadmap for how to reduce a complex data set to a lower dimension. Minimises the mean squared error (MSE) when reconstructing original data.

Variance

- A measure of the spread of the data in a data set with mean X
- Measure of the deviation from the mean for points in one dimension, e.g., heights

Covariance

- A measure of how much each of the dimensions varies from the mean with respect to each other.
- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions, e.g., number of hours studied and grade obtained.
- Covariance provides insight into how two variables are related to one another.
- Exact value is not as important as its sign.
- A positive value of covariance indicates that both dimensions increase or decrease together, e.g., as the number of hours studied increases, the grades in that subject also increase.
- A negative value indicates while one increases the other decreases, or vice-versa, e.g., active social life vs. performance in school.
- If covariance is zero: the two dimensions are independent of each other, e.g., heights of students vs. grades obtained in a subject.

Covariance Matrix

Represents covariance among dimensions as a matrix, e.g., for 3 dimensions:

$$C = \begin{pmatrix} \text{cov}(X,X) & \text{cov}(X,Y) & \text{cov}(X,Z) \\ \text{cov}(Y,X) & \text{cov}(Y,Y) & \text{cov}(Y,Z) \\ \text{cov}(Z,X) & \text{cov}(Z,Y) & \text{cov}(Z,Z) \end{pmatrix}$$

Properties

- Diagonal: variances of the variables
- $\text{cov}(X,Y) = \text{cov}(Y,X)$, hence matrix is symmetrical about the diagonal (upper triangular)
- m-dimensional data will result in mxm covariance matrix

Important

PCA is already covered in the other two modules, so not even bothering to copy most of this in.

m : number of rows (features/measurement types) in a dataset matrix.

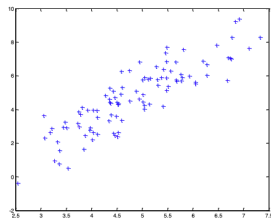
n : number of columns (data samples) in a dataset matrix.

X : given dataset matrix ($m \times n$)

x_i : columns/rows of X as indicated.

PCA algorithm:

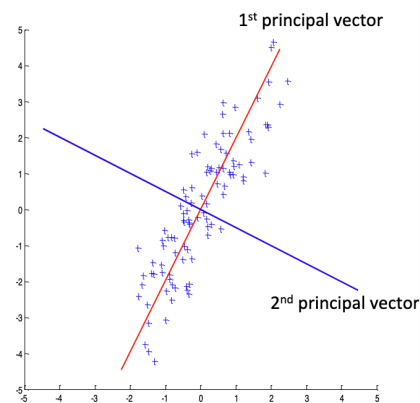
1. $X \leftarrow$ Create $m \times n$ data matrix, with one row vector x_m per data point
2. $X \leftarrow$ subtract mean $\bar{x}$ from each row vector x_m in X
3. $\Sigma \leftarrow$ covariance matrix of X
4. Find eigenvectors and eigenvalues of Σ
5. PCs $\leftarrow$ the M eigenvectors with largest eigenvalues



PCA transforms the data to a new **coordinate system** such that

- the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component),
- the second greatest variance on the second coordinate, and so on.

Gives best axis to project



How many components?

Check the distribution of eigenvalues

Take enough/many eigenvectors to cover 80-90% of the variance

If m dimensions in your data

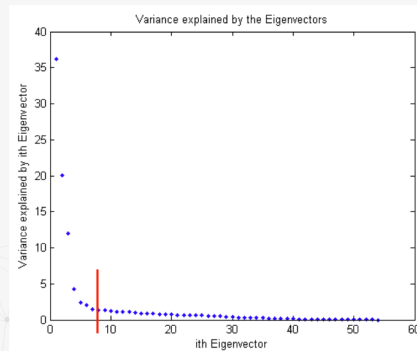
- calculate m eigenvectors and eigenvalues
- choose only the first r eigenvectors
- final data set has only r dimensions.

If the dimensions are **highly correlated**

- there will be a small number of eigenvectors with large eigenvalues and r will be much smaller than m .

If the dimensions are **not correlated**

- r will be as large as m and PCA does not help.



Reconstruction of original Data

- If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard.
- Recall that:
$$\text{FinalData} = \text{RowFeatureVector} \times \text{RowZeroMeanData}$$
- Then:
$$\text{RowZeroMeanData} = \text{RowFeatureVector}^{-1} \times \text{FinalData}$$
- and thus:
$$\text{RowOriginalData} = (\text{RowFeatureVector}^{-1} \times \text{FinalData}) + \text{OriginalMean}$$
- If we use unit eigenvectors, the inverse is the same as the transpose (hence, easier).

Problems and limitations

- What if we have very large dimensional data?
 - e.g., Images ($d \geq 10^4$)
- Problem:
 - Covariance matrix Σ is size (d^2)
 - $d=10^4 \rightarrow |\Sigma| = 10^8$

PCA-like approaches

- Multi-dimensional scaling (MDS):
 - Given a matrix of distances between features
 - We want a lower-dimensional representation that best preserves the distances
- Independent component analysis (ICA):
 - Find directions that are most statistically independent

10.4.2. t-SNE (T Distributed Stochastic Neighbour Embedding)

With dim reduction, you have two choices:

- PCA matrix factorisation; maximise variance
- Use neighbour graphs, e.g. isomap, linear embedding, **t-SNE**, etc



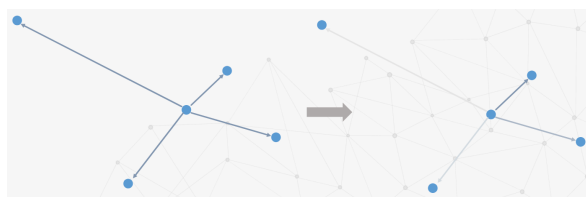
Figure 180: PCA on MNIST



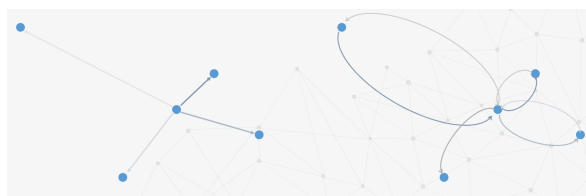
Figure 181: PCA on Fashion MNIST

What about t-SNE?

- Build a directed graph with vertices given by data points; If I weight the edges using a RBF kernel function; Smaller weights get applied to points that are more distant from each other

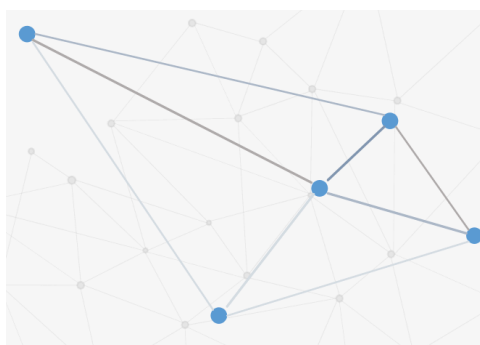


- Normalise weights at each vertex so that the outgoing weight is 1



- Convert to undirected graph by averaging weights between pairs of vertices then we're going to normalize all the

weights on the graph so the total weight of all the edges in the graph sums up to one



Suppose we had a proposed low dimensional representation; We can use a similar process to build a weighted graph in a low dimensional space. To build a weighted graph of those points in the low dimensional space where we want to embed them.

Differences

- t-distribution kernel (not RBF kernel as before)
- Fixed bandwidth => can consider the whole thing as an undirected graph from the start
- Undirected from the start, edges equal in either direction
- We still have the normalisation step.

We want to build a low dimensional representation that gives a graph that matches as closely as possible the graph using the high dimensional case. According to reconstruction error:

$$\sum_{e \in G} w_h(e) \log \left(\frac{w_h(e)}{w_l(e)} \right)$$

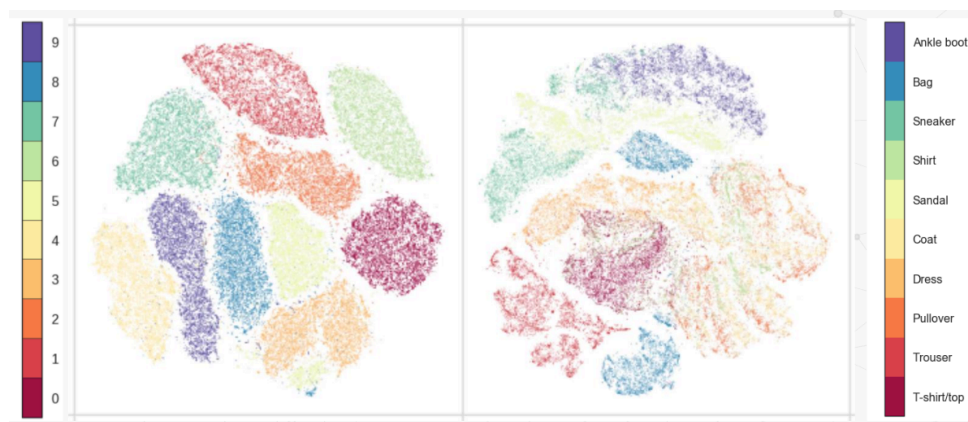


Figure 185: t-SNE on MNIST digits/fashion