Visualization

Bernhard Schmitzer

Uni Göttingen, summer term 2023, July 2, 2023

1 Introduction

1.1 Basic information

Time, BBB, videos.

- Time: Mondays, 10:15-11:45 via the same BBB link. Please share responsibly.
- Lecture will be recorded and made available afterwards in StudIP.
- If you are uncomfortable with your questions being part of the video, please let me know and I will edit them out. But please do not let it stop you from asking and participating.

Exercises and final exam

- One 'problem sheet' every other week, starting next week.
  - Requirement for admission to final exam: 50% of problems solved (or reasonably worked on). General rule: we will be very 'tolerant' here. Not meant as a measure to bar students from taking the exam, more as an 'encouragement' to stay engaged. Submissions will be checked by my doctoral student Olga Minevich.
  - Problem sheets will usually be 'extended examples', not theoretical questionnaires. So it should really be well accessible.
  - Submission online via StudIP. After every sheet there will be a virtual discussion session where the solutions will be discussed. Participation in these is **not mandatory**.
- Final test will be a short practical project, in **groups of two to three**.
  - Should take about 2-3 weeks to prepare.
  - Finally an oral exam in groups, approximately 20mins per student
  - will suggest list of potential topics later during the lecture; suitable suggestions by students are always welcome
- some examples for previous topics:
  - demographics of German parliament (distribution over parties, age, gender, development over time); someone did this for US congress instead
  - trajectories of space probes through the solar system

  - analysis of manga tv shows, how are different genres related, how do viewers decide what to watch and rate shows?
  - extending an open source fitness app for smartphones to provide a better visual presentation of the past workout data
  - analysis of transfers and cash flow between European soccer clubs
  - analysis CO2 emissions or precipitation data, relation to climate change

1.2 What is this lecture about?

What is visualization?

- this lecture is about creating good figures.
  - but not in the sense of: 'how can we render fancy 3d graphics'
  - more in the following sense: good figures can convey large amounts of data or complex structures to the human brain;
  - our brains can process them almost effortlessly, build mental model of data which cannot be achieved by other means (such as reading a description text of the data)
- we want to make the most of this powerful communication tool, need to understand the following things:
  - how do the human eyes and brains process visual information?
  - based on this: what are good design rules, how do we make figures 'compatible' to the human visual system
  - how can we represent data visually (colors, positions, lengths, graphs,...)
  - how can we prepare and transform data for better representation
  - how to keep track of the whole pipeline from data to figure

The role of computers.

- some books on visualization like to emphasize that visualization is not about how to create figures with a computer
  - that a good course on visualization is not a tutorial for a particular piece of software
  - makes sense: software changes, good design principles are independent of software
- but: computers are incredible useful for visualization
  - we can apply much more complicated processing to much more complicated data and transform it into ways that can be visualized
  - we can set up algorithmic pipeline that automatically generates figures for many datasets or after parameter changes
  - we can generate dynamic and interactive visualizations to handle even more complexity
- modern data analysis and visualization impossible without computers, which is why so many universities now offer courses on data science

- similarly: modern data science and visualization require mathematics and statistics, to transform, analyze and simplify data

- in this spirit: this lecture will not be a tutorial session for a particular software environment
  - but: practical examples will be indispensable
  - for this will mainly use python/matplotlib in jupyter notebooks
  - some other programs and libraries will be mentioned here and there

Python: a prototypical computational data analysis environment.

- open source, available on all platforms, long term support, immense availability of / compatibility with libraries and software
- simple installation, management of components via
  - Anaconda/Miniconda: https://docs.conda.io/en/latest/miniconda.html
  - or the Python Package Index and pip
- alternatively, try Google colab (https://colab.research.google.com) or GWDG Jupyter cloud (https://jupyter-cloud.gwdg.de/)
- core packages:
  - numpy/scipy: scientific computing
  - matplotlib: plotting (including high-quality export e.g. for LaTeX manuscripts)
  - imageio: saving/loading raster image formats, animations
  - pandas: managing tables, simple import/export
  - scikit-learn: basic statistical analysis and machine learning tools
  - jupyter (or similar): fast interactive scripting, sharing and presenting results (use jupytext plugin for better compatibility with version control)
- support for data formats (built-in or popular packages):
  - raw binaries, mat, csv, json, xml, ods, xls...

Outline of lecture.

- examples (historical, and 'live' or 'personal')
- brief discussion of 'theory'
  - design principles by Tufte
  - grammar of graphics by Wilkinson
- the human visual system
  - how do the eyes and the brain generate 'the image in our head'?
  - what does this imply for the creation of good figures?

- data types and processing pipeline
- the role of colors
- fundamental visual data representations
  - points, lines, bars, pies, histograms...
  - boxes, violins, errorbars, ...
  - images, heatmaps,
  - contours, vector fields, transformations
  - graphs, meshes, networks
- high-dimensional or complex data
  - aggregations, filters, slices
  - dimensionality reduction and embeddings
- how to lie with charts?
- animations and interactive visualization
- visualization on the web

Literature.

- Alberto Cairo: The Functional Art, New Riders, Berkeley, 2013
- Alberto Cairo: How charts lie, W. W. Norton & Company, 2019
- Andy Kirk: Data Visualisation, SAGE Publications Ltd, 2019
- Robert Spence: Information Visualization, Springer, 2014
- Edward Tufte: The Visual Display of Quantitative Information, Graphics Press Cheshire, 1983
  (and similar, more recent books by same author)
- Leland Wilkinson: The grammar of graphics, Springer, 2005

1.3 A few examples

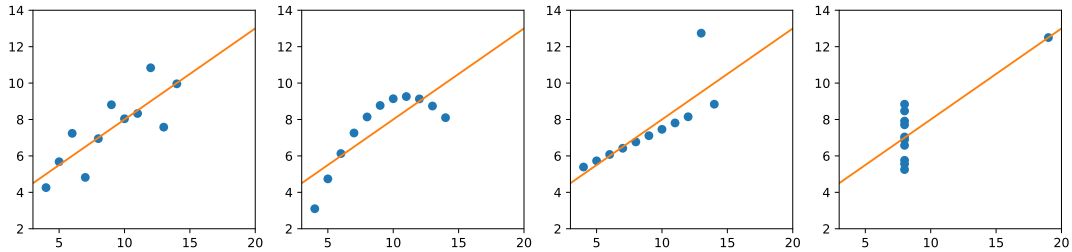
1.3.1 Tables versus plots

- taken from F. J. Anscombe: Graphs in Statistical Analysis, The American Statistician, 1973, 27, 17–21

Data series represented as table:

x_1	10.00	8.00	13.00	9.00	11.00	14.00	6.00	4.00	12.00	7.00	5.00
y_1	8.04	6.95	7.58	8.81	8.33	9.96	7.24	4.26	10.84	4.82	5.68
y_2	9.14	8.14	8.74	8.77	9.26	8.10	6.13	3.10	9.13	7.26	4.74
y_3	7.46	6.77	12.74	7.11	7.81	8.84	6.08	5.39	8.15	6.42	5.73
x_2	8.00	8.00	8.00	8.00	8.00	8.00	8.00	19.00	8.00	8.00	8.00
y_4	6.58	5.76	7.71	8.84	8.47	7.04	5.25	12.50	5.56	7.91	6.89

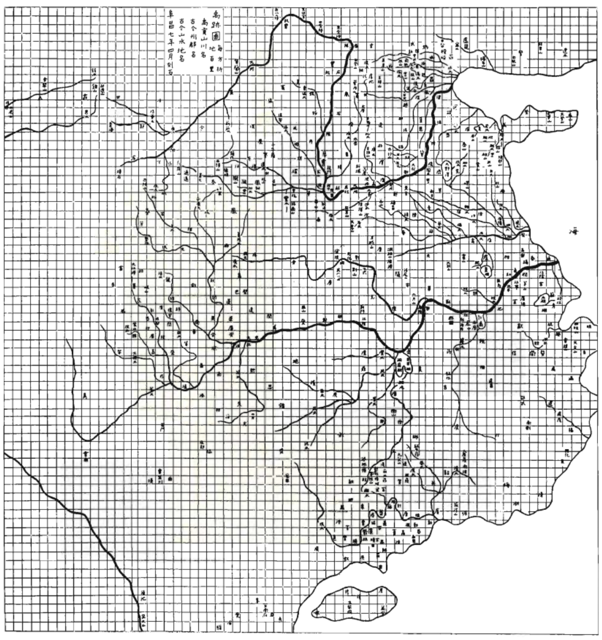
- hard to interpret as numbers in table, try basic statistical analysis
- all x_i and y_i sequences have same mean and variance
- sequences of pairs (x_1, y_1), (x_1, y_2), (x_1, y_3) and (x_2, y_4) all yield essentially the same linear regression:
  - same slope, intercept, correlation coefficient, standard error for slope estimation
- graphic representation immediately tells us four different stories



1.3.2 Historical examples

Chinese cartography

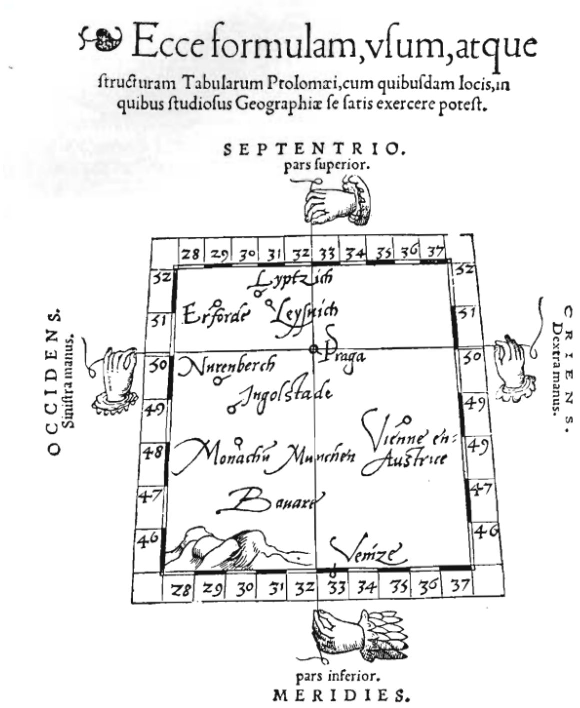
- taken from [Tufte: The visual display of quantitative information]
- approx 1100 AD



European cartography

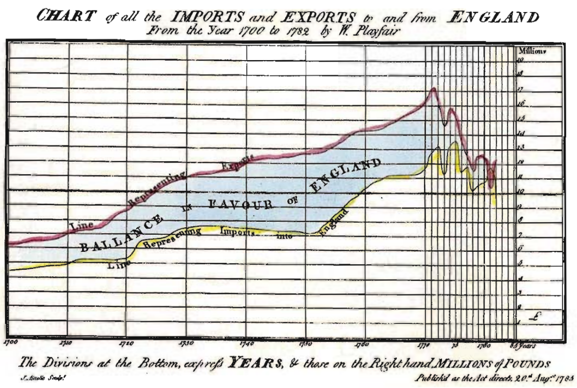
- taken from [Tufte: The visual display of quantitative information]

- 1546 by Petrus Apianus, generalization to two-dimensional plots still took some time



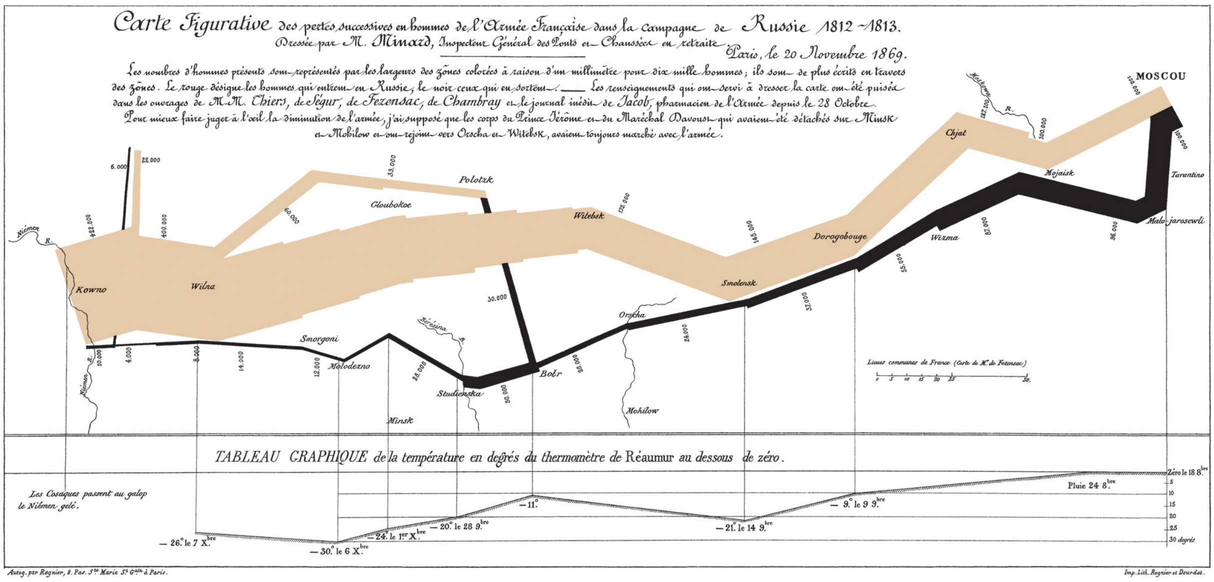
Playfair

- taken from [Tufte: The visual display of quantitative information]
- William Playfair: The commercial and political atlas, 1786



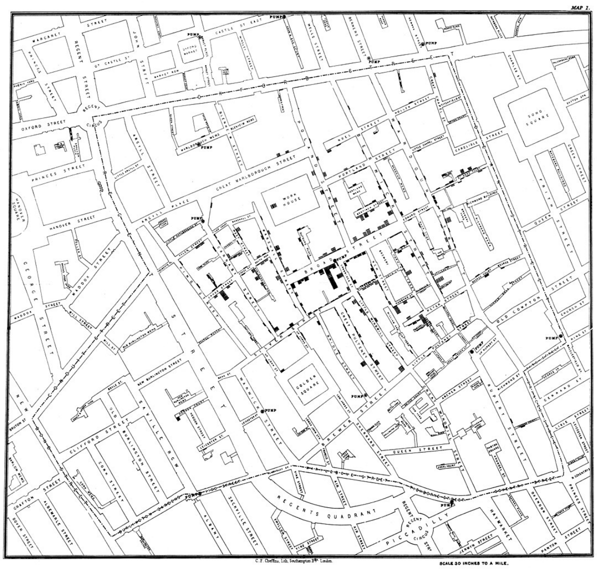
Napoleon's march to Moscow.

- taken from [Spence: Information Visualization]
- figure in public domain, available at https://en.wikipedia.org/wiki/File:Minard.png
- Charles Joseph Minard, 1869



Water pumps in London.

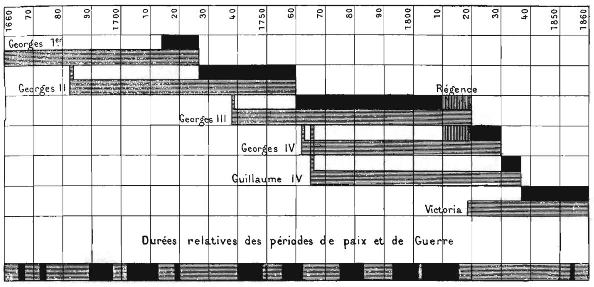
- taken from [Spence: Information Visualization]
- figure in public domain, available at https://en.wikipedia.org/wiki/File:Snow-cholera-map-1.jpg
- John Snow, 1854



Regency chart

- taken from [Tufte: The visual display of quantitative information]
- E. J. Marey: La méthode graphique, 1885

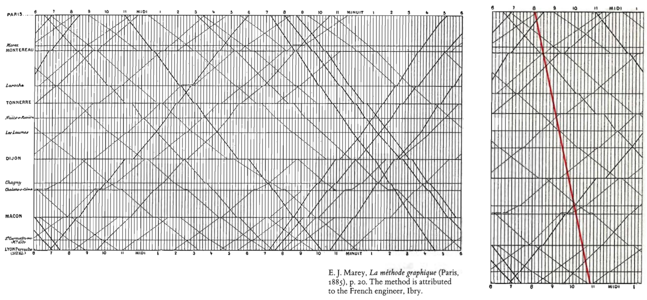
- Note: George II was the founder of Göttingen University, Wilhelmsplatz is named after William IV.



E. J. Marey, *La Méthode Graphique* (Paris, 1885), p. 6.

Train timetable.

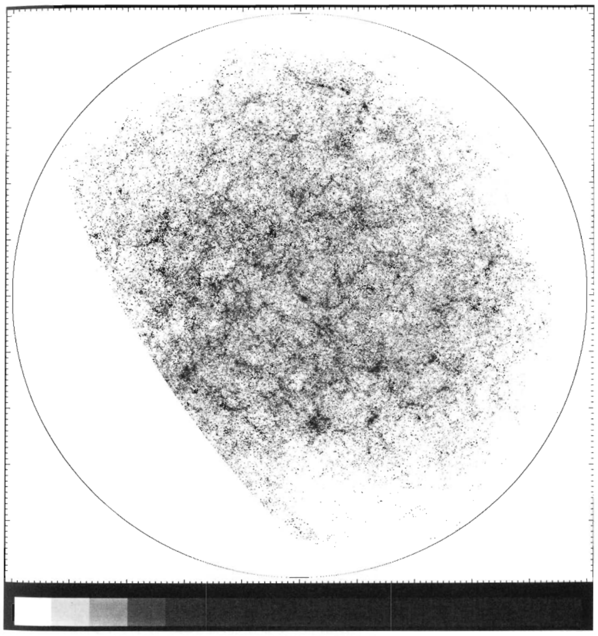
- taken from [Tufte: The visual display of quantitative information]
- E. J. Marey: La méthode graphique, 1885



E. J. Marey, *La méthode graphique* (Paris, 1885), p. 20. The method is attributed to the French engineer, Ibry.

Galaxy distribution.

- taken from [Tufte: The visual display of quantitative information]
- example for computerized cartography, 1977



1.3.3 Example by Alberto Cairo: world population

- taken from [Cairo: The Functional Art, Chapter 1]
- Cairo read book ‘The Rational Optimist: How Prosperity Evolves’ by Matt Ridley
- chapter on world population made the hypothesis that it will soon stabilize
  - rapid decrease in fertility in developing countries
  - slight increase (back to ‘replacement rate’ of 2.1 children per woman) in developed countries
- provided figure did not support the hypothesis, did not display appropriate data, different simultaneous trends cannot be distinguished in summarized data
- showing all individual trajectories not helpful either: all necessary data is shown, but hard to process visually
- final version: highlight representatives from different clusters. supports hypothesis, allows for further more detailed exploration (e.g. China, Brazil, Niger)

1.3.4 EEA: Chart dos and don’ts

https://www.eea.europa.eu/data-and-maps/daviz/learn-more/chart-dos-and-donts

- show full y-axis
- consistent x-axis intervals

- Edward Tufte in a nutshell: remove clutter
- highlight what’s important
- sorting

- do not use 3d or other visual effects
- direct labeling where possible
- avoid pie charts
- avoid stacked charts
- do not use maps for everything with spatial dimension
- avoid animations, use small multiples

- show level of confidence
- tell the ‘why’ and ‘how’
- how to treat missing data
- do not confuse causation and correlation
- do not compare apples with oranges
- adjust for inflation

- do not forget color deficiency
- ask others for opinion

2 Edward Tufte

2.1 Introduction

About the author.

- Born 1942, professor emeritus for political science, statistics and computer science at Yale University. Pioneer in the field of data visualization. 'e' at the end of name is pronounced. (https://en.wikipedia.org/wiki/Edward_Tufte).
- First influential book on topic: The Visual Display of Quantitative Information, 1983.
- Promoted a philosophy of minimalist design in information graphics, apparently driven by a trend that graphics were only perceived as means to dumb down information or to make statistical data less boring, assuming the audience would be stupid or not interested.
- Adopts a polemic language in his books, seems to enjoy deconstruction of bad examples, likes to formulate lists principles.
- The following section is based on [The Visual Display of Quantitative Information] and examples are taken from there.

Graphical excellence according to Tufte. Excellence in statistical graphics consists of complex ideas communicated with clarity, precision, and efficiency. Graphical display should

- show the data
- induce the viewer to think about the substance rather than about methodology, graphic design, the technology of graphic production, or something else
- avoid distorting what the data have to say
- present many numbers in a small space
- make large data sets coherent
- encourage the eye to compare different pieces of data
- reveal the data at several levels of detail, from broad overview to the fine structure
- serve a reasonably clear purpose: description, exploration, tabulation, or decoration
- be closely integrated with the statistical and verbal descriptions of the data set.

Graphical integrity.

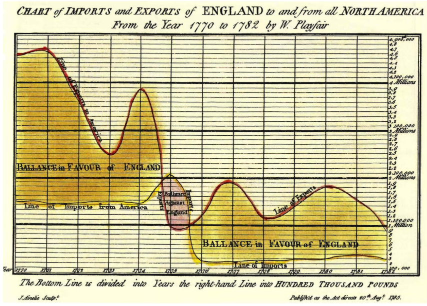
- of course it was already well known that graphics may distort the data, by ignorance or by intention
- Tufte even had the impression that graphics had a general reputation for 'lying to' or 'fooling' viewers, see [The Visual Display of Quantitative Information, Chapter 2]:
  - 'For many people the first word that comes to mind when they think about statistical charts is *lie*.'

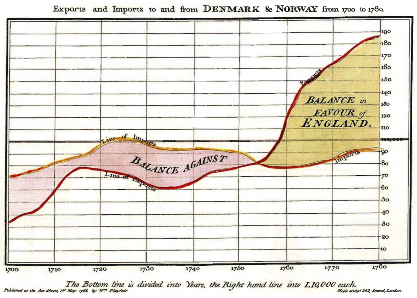
  - ‘Much of twentieth-century thinking about statistical graphics has been preoccupied with the question of how some amateurish chart might fool a naive viewer.’

- postpone detailed discussion until session on ‘how to lie with charts’.

2.2 Data-ink

Example: Evolution of charts by Playfair.



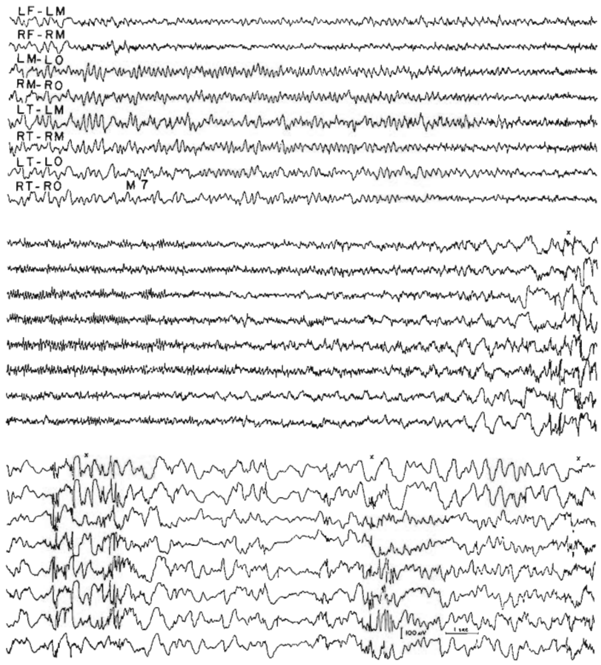


- first example: 1785, early pages of ‘The Commercial and Political Atlas’
- second example: created on year later, already much more mature, removed much of the ‘background’
- Tufte formulates a fundamental principle: Above all else show the data.

Definition.

- data-ink is the ink in a graphic that represents data / information
  - non-data-ink: frames, grids, (unnecessary) ticks, decoration
  - data-ink: data points, (necessary) labels, derived data (e.g. marginal distributions, indication of minimal or maximal values)
- data-ink ratio $= \dfrac{\text{data-ink}}{\text{total ink}}$

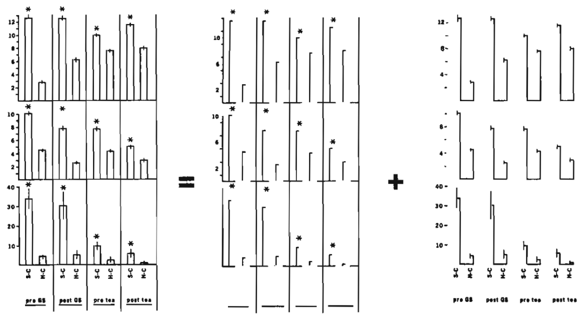
Example: electroencephalogram.



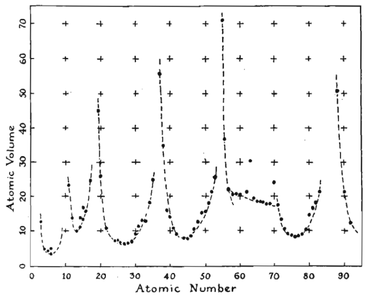
- extreme example: almost exclusively data-ink, but can only be read by specialists

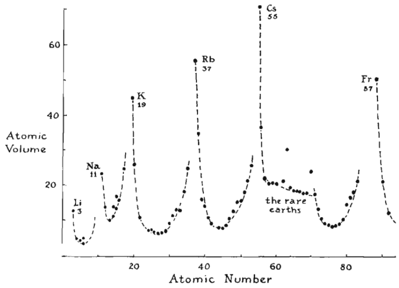
Example: sour taste.

- original image taken from: Kuznicki and McCutcheon: Cross-enhancement of the sour taste on single human taste papillae. Journal of Experimental Psychology: General, 108(1), 68–89, 1979. (study effect of sucrose on the perceived intensity of sour taste)
- Tufte's introduction in book: '[The display] compares each long bar with the adjacent short bar to show the viewer that, under the various experimental conditions, the long bar is longer.'
- Tufte removes: frames, some ticks and labels, one side of each bar, stars (marking the longer bars), text decoration (underline)
- lines connecting adjacent bars are kept (data, since they show which experiments belong together)
- extreme example. My humble opinion: should be seen as illustration of principle rather than as concrete suggestion



Example: periodic system.





- original image created by science illustrator Roger Hayward for chemistry textbook by Linus Pauling, 1947 (intruduced covalent bond in chemistry, two nobel prices, chemistry and peace)
- remove grid, try removing guidelines (but rather not), add individual labels

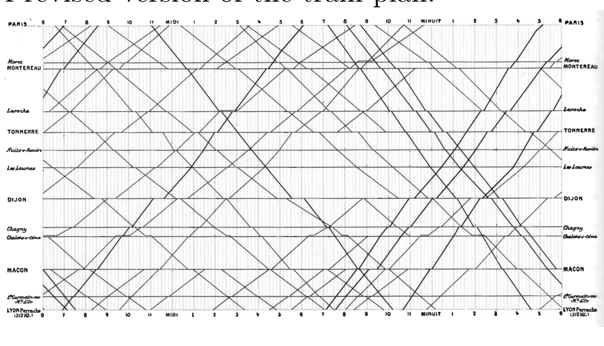
More principles.

- above all else show the data
- maximize the data-ink ratio
- erase non-data-ink
- erase redundant data-ink
- revise and edit
- everything 'within reason':
  - sometimes data-ink ratio is ill-defined or not appropriate
  - sometimes redundancy may be helpful: train schedule example

2.3 Chartjunk: Vibrations, Grids and Ducks

Avoid moiré patterns.

Keep grids subtle. A revised version of the train plan.

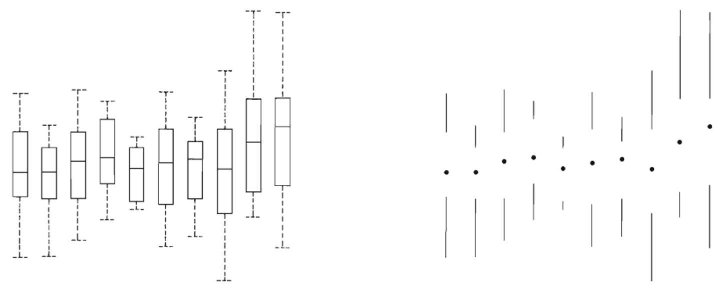


Ducks.

- a duck is a graphic which is just entirely decoration, e.g. self-promotion of graphical techniques instead of information display
- sometimes a table may be better than a pointless graphic

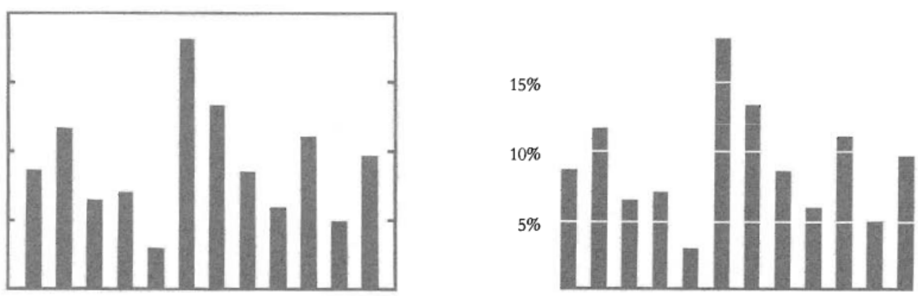
2.4 New graphical design suggestions

Simplification of box plot (quartile plot).



- Tufte: conventional box plot is highly redundant. Suggests minimalistic version, argues via number of placings of straightedge
- my humble opinion: oversimplified
  - (information about) data is there; but weight of ink does not align with weight of data (usually higher density within quartiles, otherwise not proper plotting device anyway)
  - Tufte frequently makes data density calculations: how much numbers are encoded in a figure and equates this with amount of numbers that are transferred into viewers brain
  - but the visual system/brain do not extract a list of numbers from a graphic (at least not at "first glance"), but coarse structures and trends
  - coarse structure more accurately visually reflected by original design

Simplification of bar chart.



- Tufte proposes changes to basic bar chart
- my humble opinion: misses the point. simple bar chart is not the right format for the discussed example in the first place.

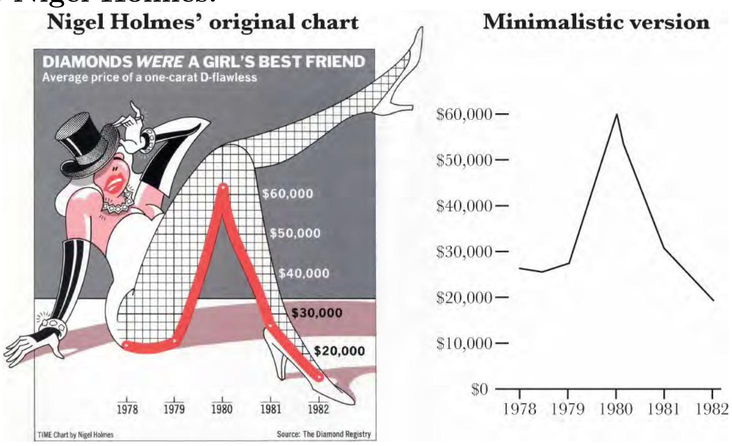
- what is the type of the x-axis?
  - a **time series** or other continuous, one-dimensional variable? then show scatter plot, possibly with lines
  - a **nominal type**? (city, country, ...) then data should be sorted, or at least grouped (e.g. by continent?)
  - does the chart represent a **histogram** with contiguous intervals (of equal width)? then remove gaps between bars
  - if gaps correspond to empty intervals: chart is perfect. it emphasizes a very peculiar property of data.

2.5 Critical discussion by Alberto Cairo

Dumbing down.

- apparently a widespread misconception: graphics are for 'dumbing down' data, flashy presentation of 'boring statistics'
- when struggling with interpretability of a graphic, reflex is to simplify instead of clarifying
- this appears to have been particularly common in 80s (and onward with rise of computers) and seemingly motivated Tufte's work
- Tufte: we should not think that our readers are stupid.
- Of course this is true. But also keep in mind: readers/listeners need time to absorb, process, pause and digest new information. Not all required background-knowledge may be present (or has become a little diffuse). We get tired. Human brains are not computers, eyes are not cameras. Motivation, patience, redundancy and good graphics are key to including the audience.

Edward Tufte vs Nigel Holmes.



- Nigel Holmes was art director for Time magazine
- example: illustration of diamond prices, 1980s

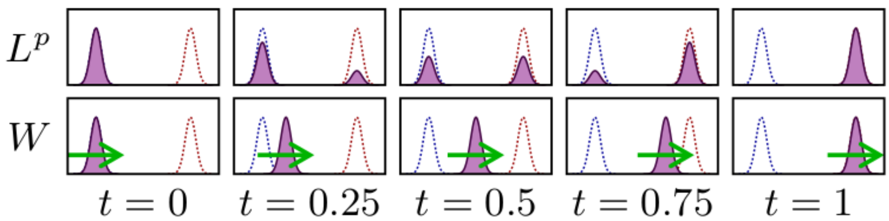
- 'anti-Tufte': very low data-ink ratio, data-density, full of decoration and 'chart junk', (and blatantly sexist)

However:

- Tufte's principles are not rigorously based on scientific research but also on aesthetic preferences
- there is no empirical evidence that data-ink ratio is indeed a good measure for the quality of a graphic (in terms of readability)
- mixed results in studies:
  - Ben-Gurion University, 2007, 87 students: compare bar charts with minimalistic versions.
    No significant difference in interpretation performance;
    students aesthetically preferred 'classical' charts.
  - University of Saskatchewan (Canada), 2010, 20 students: compare four Nigel Holmes illustrations with minimalistic versions.
    Subjects interpreted both versions equally well.
    After a waiting period, subjects could answer questions about Holme's graphics with higher accuracy (were not told that they would be questioned)
  - these are not conclusive, representative studies (e.g. very small sample groups) but tempting naive conclusion: decoration may help the brain remember a graphic (and thus also its data)

2.6 Data density and small multiples

- Tufte: most graphics can be reduced in size without losing readability
- small multiple: use same 'graphical encoding scheme' multiple times in a row, to display sequence/series of data
- one the reader has encoded the first graphic, they immediately can read all the others
- in many contexts preferrable to animations (in print, even in presentations: viewer scan 'scroll' individually, can compare simultaneously)

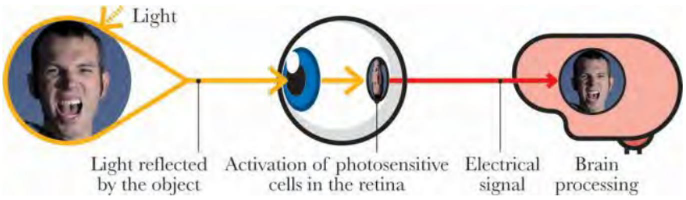


3 Human visual system and cognition

References. Coarse outline based on [Cairo, Functional Art: Part II], [Spence, Chapter 4] Additional sources on low-level visual processing:

- Kandel et al., Principles of Neural Science, 4th ed., McGraw–Hill, New York, 2000, pp. 577ff
- Healey and Enns: Attention and Visual Memory in Visualization and Computer Graphics, IEEE Transactions on Visualization and Computer Graphics, 2012, DOI: 10.1109/TVCG.2011.127
- see also: https://www.csc2.ncsu.edu/faculty/healey/PP/
- Wolfe et al., Visual search in scenes involves selective and nonselective pathways, Trends in Cognitive Sciences, 2011, DOI: 10.1016/j.tics.2010.12.001

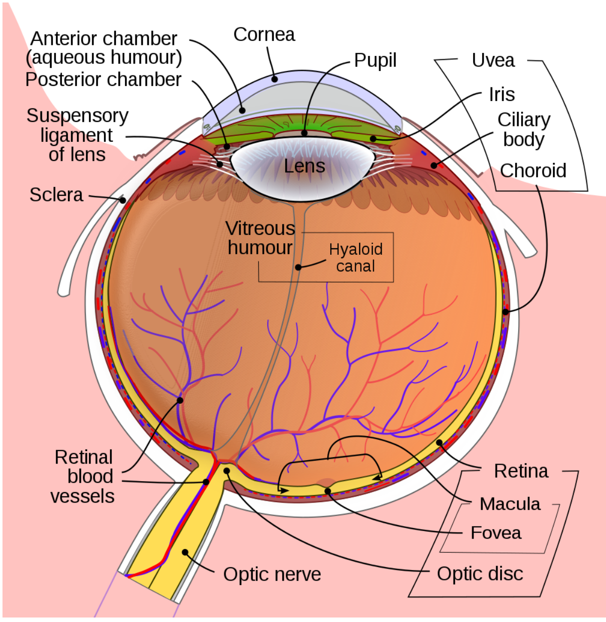
An unhelpful analogy.



- unsatisfying description of the eye/brain: a small literal picture in the brain
- metaphor for visual system: digital camera. eye is lens and chip, nerves are the wires, but the brain is not a hard drive with a serial microprocessor
- seeing, perceiving, knowing are distinct from each other

3.1 The human eye

overview of the eye.

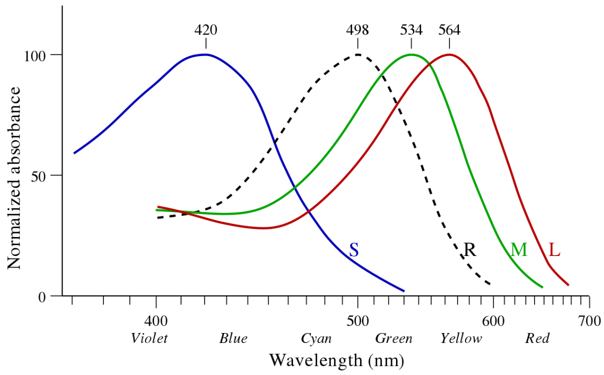


https://en.wikipedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg

- pupil, lens at the front
- retina with photoreceptors at the back

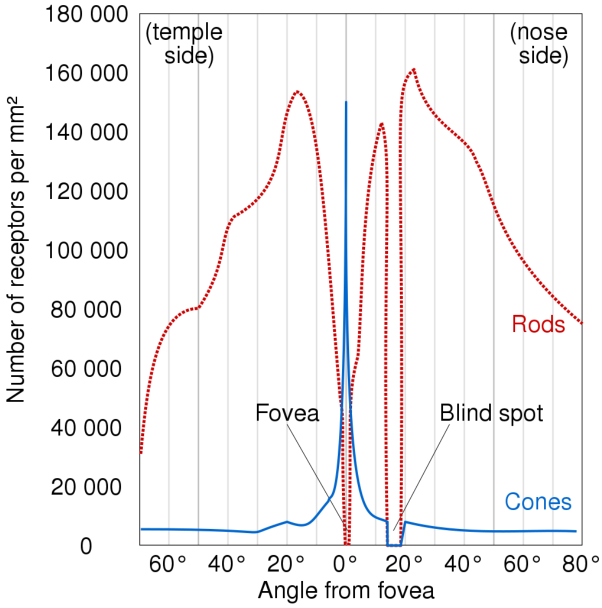
photoreceptors.

- rods
  - approx 100mio, very sensitive (can respond to a single photon),
  - signal is pooled over multiple rod cells, signal is collected over longer time interval $\Rightarrow$ better sensitivity, less spatial and temporal resolution
  - dominant 'in the dark'
- cones
  - 7mio, three types with peak sensitivity in at different wavelengths: long, medium, short $\Rightarrow$ color detection
  - small part of electromagnetic spectrum can be perceived by the eye



https://en.wikipedia.org/wiki/File:Cone-response-en.svg

- distribution of cones and rods
  - rods relatively evenly distributed, density gradually decreasing towards the periphery, two notable gaps.
  - fovea: contains highest density of cones, responsible for sharp color vision
  - but due to higher sensitivity of rods: at night fovea is not so helpful. astronomers know: look slightly past a star to see it
  - blind spot: optimal nerve leaves retina, neigher cones nor rods



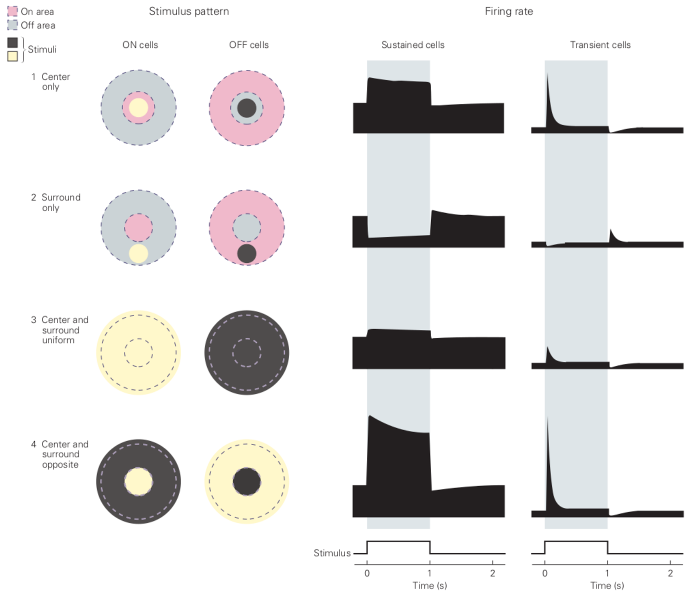
https://en.wikipedia.org/wiki/File:Human_photoreceptor_distribution.svg

structure of retina and first optical processing

- taken from: Kandel et al., Principles of Neural Science, 4th ed., McGraw–Hill, New York, 2000, pp. 577ff
- three layers: photoreceptors and two layers of neurons.
- black layer of pigments at the back to avoid re-scattering of light into eye
- interesting: receptors sit at the back. in fovea front layers are pushed aside. if this is on purpose or by accident is not fully known.
- ⇒ blind spot is necessary consequence of layer ordering: optic nerve must pass through retina to brain
- approx 1 million axons in optical nerve, i.e. approx 1% of number of photoreceptors, ⇒ strong compression and processing must alread happen in retina
- ganglion cells (third layer):
  - ON and OFF type for faster detection of decrease in intensity
  - have center and outer sensitive region, fire most rapidly if stimuli are different ⇒ simple edge detectors

  - transient cells for better detection of temporal variations: higher firing rate on increase of signal, then reduce again (act like a temporal edge detector)

- only 10% of cortical neurons driven by color contrast rather than luminance contrast



from Kandel, response of ganglion cells

saccades and fixations.

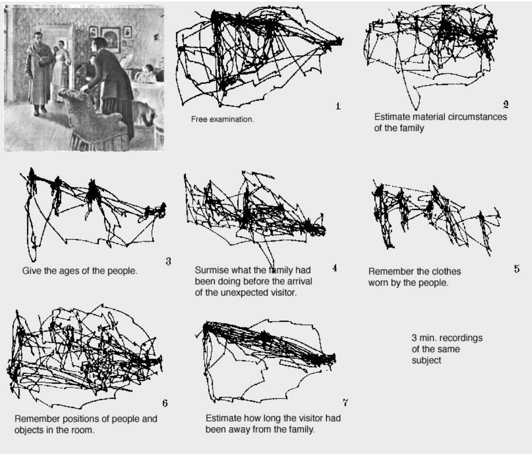
- close eyes for a while, open and keep focused on fixed object:
  from there, really difficult to properly see objects beyond the 'center', and even to see their color

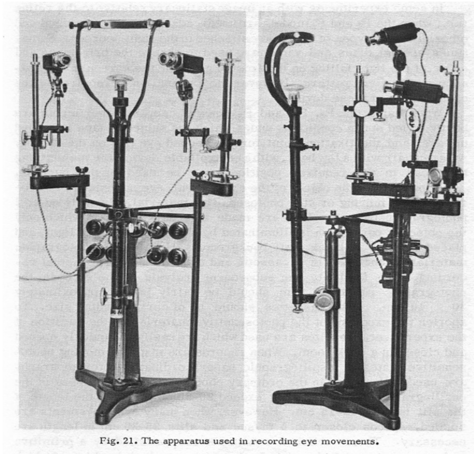
- angular range of vision:
  - peripheral vision $\sim 180°$
  - fovea $\sim 2°$ (approximately thumbnail arm's length)
  - parafovea $\sim 10°$

- how is illusion of 'steady' image in the brain generated?

- focus of eye moves quickly through scene in front of us:
  - a sequence of fixations (lasting typially around 200ms) and quick saccades in between (20ms) where the eye moves
  - location of next saccade mostly chosen unconciously / automatically, based on input from peripheral vision (bottom-up) and task/context (top-down)
  - evolutionary trait, important for survival
  - moving and uncommon objects attract our attention
  - simple example of a deduced design principle for animations: do not start an animation and simultaneously introduce new text box

- eye tracking experiments by Alfred Lukyanovich Yarbus, Russian psychologist, 1960s

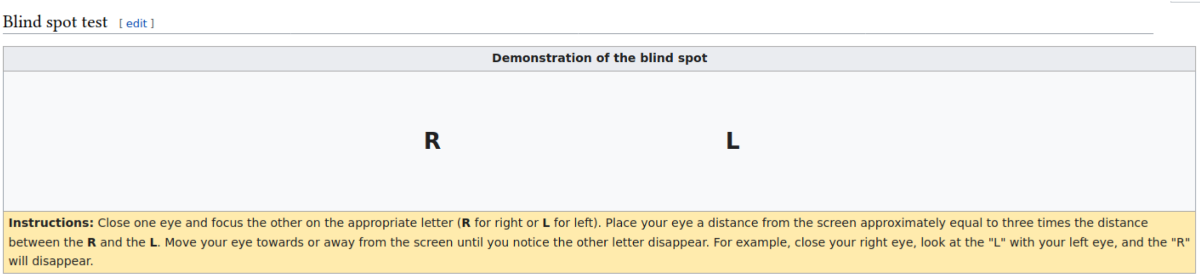




https://en.wikipedia.org/wiki/File:Yarbus_The_Visitor.jpg
https://en.wikipedia.org/wiki/File:Yarbus_eye_tracker.jpg

optical illusions.

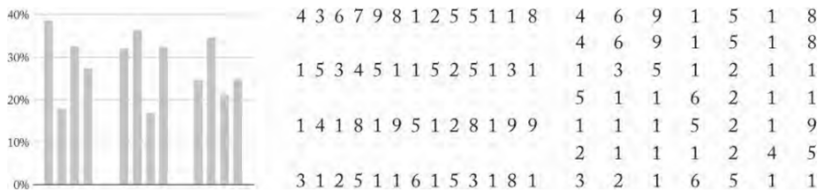
- examples: geometric patterns
- purpose: filling gaps, efficiency
- hiding blind spot is also an illusion
- clear by the above: processing of visual information in brain can be nothing like pixel-level processing of detector input



https://en.wikipedia.org/wiki/Blind_spot_(vision)

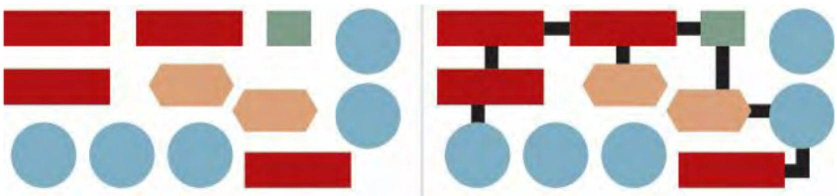
Gestalt grouping principles.

- Gestalt psychology is psychological school of thought, emerging in early 20th century Germany and Austria, among other things concerned with principles of perception
- probably does not hold up to modern standards of rigor in science, unsatisfying in terms of explanations of phenomena, but did collect a list of observed principles that seem to underlie visual perception
- famous example: grouping laws
    - proximity

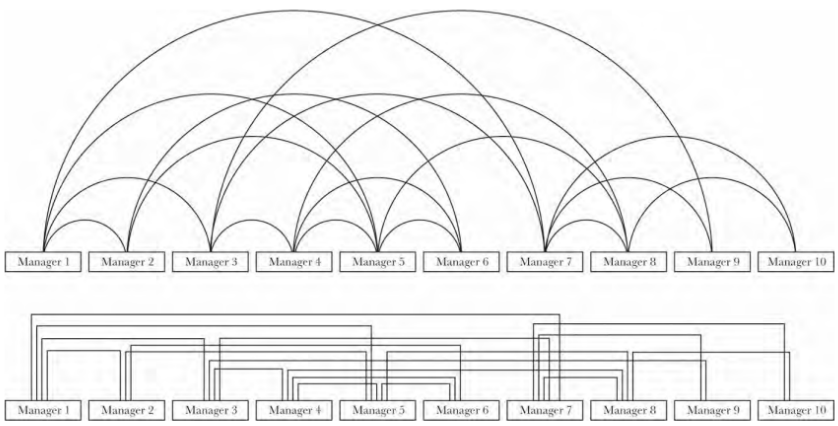


– similarity

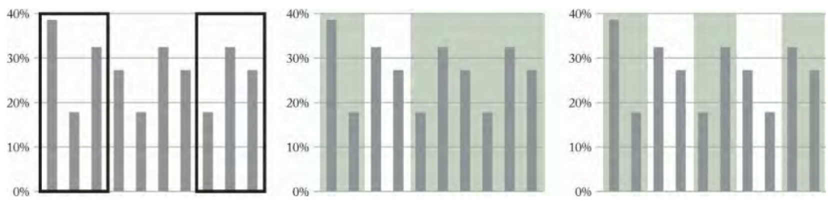
– connectedness



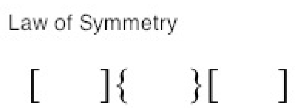
– continuity



– closure



– symmetry



https://en.wikipedia.org/wiki/File:Law_of_Symmetry.jpg

3.2 Preattentive processing

famous experiments.

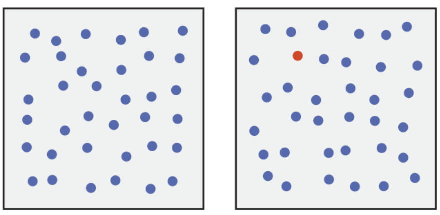
- Treisman and Gormican, 1990: show simple image only for 50ms, viewer can still detect the gist
- Kundel and Nodine, 1975: trained radiologists were shown lung X-rays for 200ms. Could detect anomalies in 70% of cases (97% under unlimited viewing).
- we extract a lot of information from an image in the first 200ms, before we even can consciously think about the image
- this is called preattentive processing. refers to processing that happens approximately within first 200ms after stimulation, on large multi-element displays. eye movement takes about 200ms, so this stuff happens before eyes can focus on something. so processing must happen in parallel in low-level visual system.
- now know: not fully true that independent of attention (bottom-up), since the given task of the individual may influence the exact evaluation of the images (top-down)

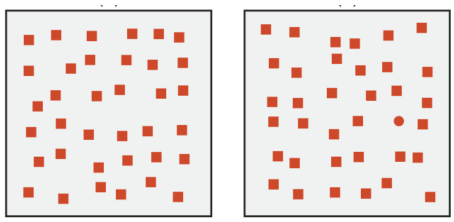
Anne Treisman (1935-2018).

- English cognitive psychologist, taught in Oxford, Berkeley, Princeton. Important contributions to study of visual perception. Devised various experiments and developed 'feature integration theory' for their interpretation.
- motivated by 'edge detection' neurons in the brains of cats (famous experiment by David H. Hubel and Torsten Wiesel in 1959, 1981 nobel prize in medicine)
- designed experiments with two different performance measures:
  - in time experiment: complete task as fast as possible, while being accurate. increase number of distractors. if solving time is approximately constant, then task is preattentive. otherwise, searial search is required and it takes longer with increasing number of distractors.
  - in accuracy experiment: a fixed short time (250ms) to solve the task is given. the eye cannot move in this time. measure accuracy vs complexity. if accuracy remains high with increasing complexity, task is preattentive.

pop-out effect: quickly identifying unique visual properties.

- example: red circle among blue circles (feature: hue)
- example: red circle among red squares (feature: shape)
- identification happens automatically and without exhausting the viewer

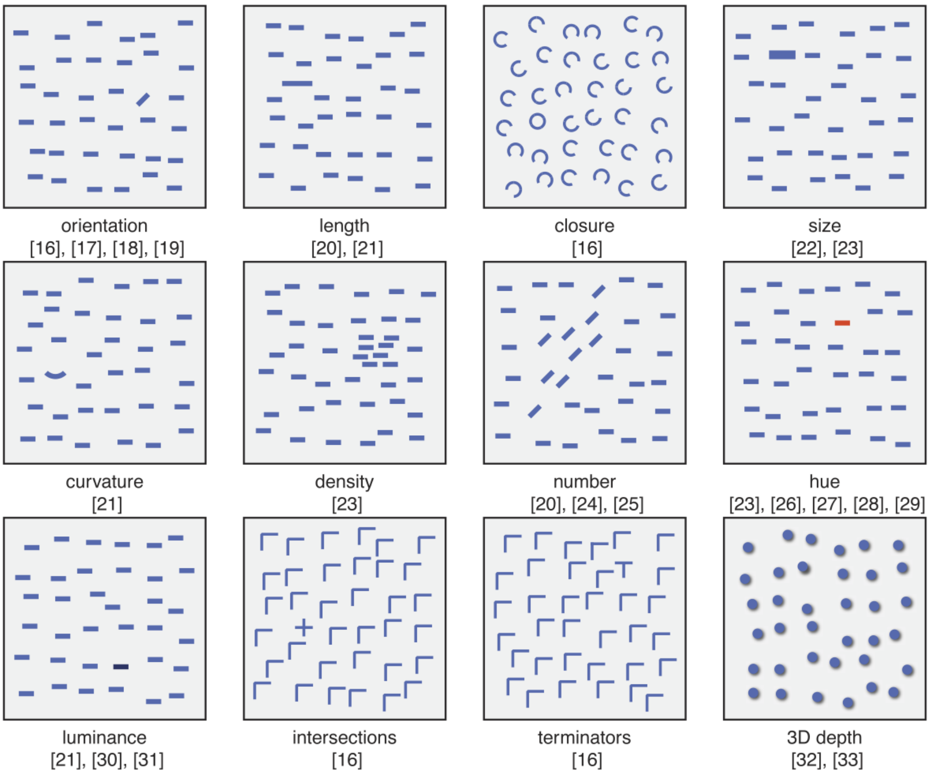




pop-out with hue and shape [Healey]

features for which preattentive processing works (not exhaustive).

- orientation
- length
- closure
- size
- curvature
- density
- number
- hue
- luminance
- intersections
- terminators
- 3d depth



features in preattentive processing [Healey]

tasks solved by preattentive processing.

- target detection: detect the presence or absence of a 'target' element with a unique visual feature within a field of distractor elements
- boundary detection: detect a boundary between two groups of elements, where all of the elements in each group have a common visual property
- region tracking: track one or more elements with a unique visual feature as they move in time and space
- counting and estimating number of elements with a unique visual feature

conjunction search.

- find target with several specific features (e.g. hue and shape) that are not unique. typically much harder, does usually not happen preattentively.
- exceptions possible in 'extreme' cases

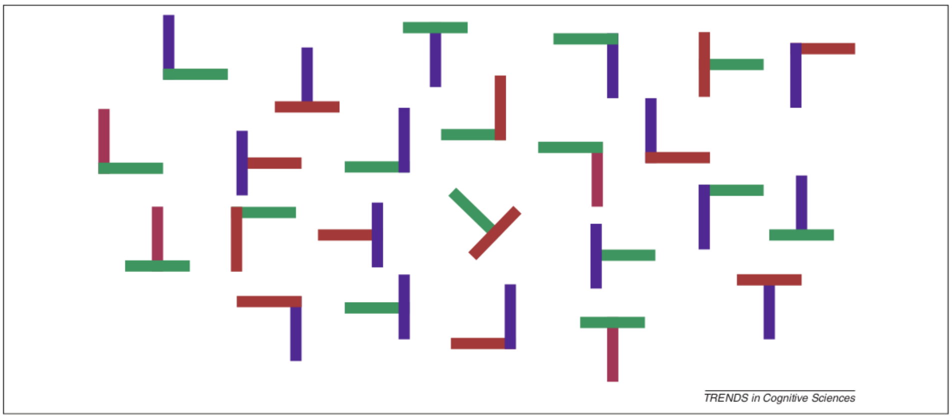
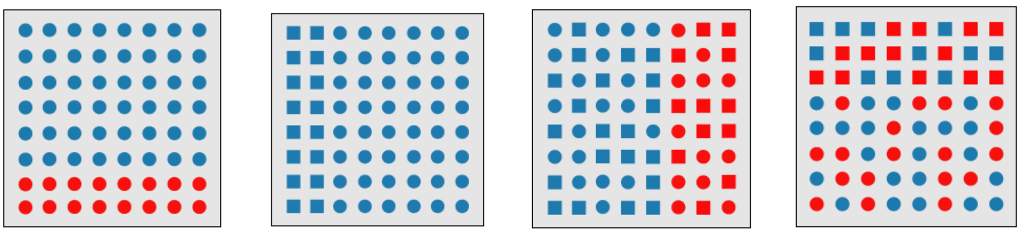


Figure 1. Find the four purple-and-green Ts. Even though it is easy to identify such targets, this task requires search.

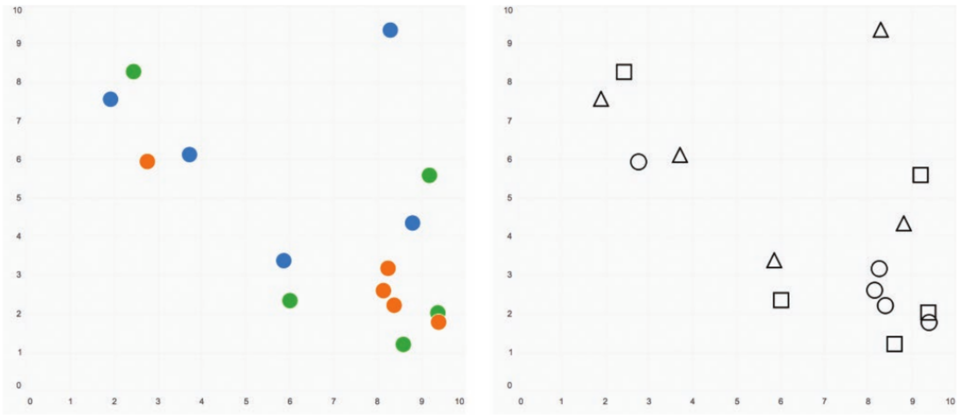
conjunction search is much harder [Wolfe]

feature hierarchy.

- 'hue trumps shape' (caveat: color deficiency): pop-out effect of some feature may overshadow 'weaker features'



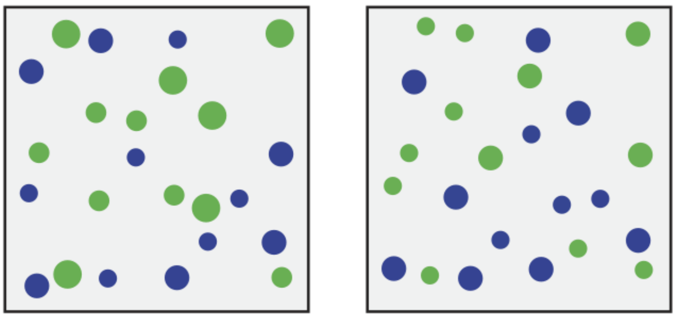
feature hierarchy between hue and shape [Healey]



'color beats shape' from [Kirk: Data Visualisation]

ensemble coding.

- humans are good at getting overall impressions at first glance. examples:
  - average size of points



(a) (b)

Fig. 9. Estimating average size: (a) average size of green elements is larger; (b) average size of blue elements is larger [66].

  - 'average emotion' in group of faces
  - 'setting' of a scene (e.g. kitchen)
- the overall impression might help to perform target search faster, e.g. bread in kitchen. this suggests that there are several parallel processing pathways in the brain.

3.3 Theoretical models for preattentive processing

feature integration theory (Treisman).

- simple model: feature maps
- each feature has its own map: where in the image does this feature occur
- combination search needs attentive look-up of master map
- of course: reality not as simple; perception is not as binary; sometimes even conjunction search can be preattentive

other theoretical models.

- texton theory: things that look distinct in isolation can look almost identical in large crowds

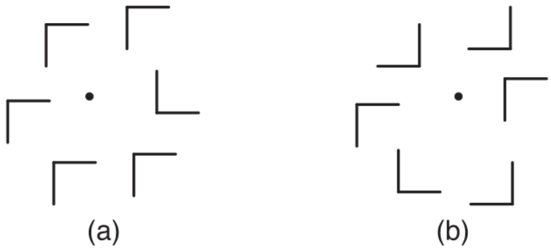


Fig. 5. N-N similarity affecting search efficiency for an L-shaped target: (a) high N-N (nontarget-nontarget) similarity allows easy detection of the target L; (b) low N-N similarity increases the difficulty of detecting the target L [55].

low vs high NN similarity

- similarity theory: TN vs NN similarity; not purely based on presence of preattentive features, conjunction is possible to some extent
- boolean map theory: to some extent we are able to intersect features

3.4 Visual perception and memory

guided search.

- perception is not just botom-up; there must be some top-down component
- feature maps: bottom-up; search query: top-down; queries are in format/features provided by the visual cortex;
- activation map: combination of bottom-up, top-down activity; look at regions of most activation; this would explain TN-NN results
- also explains why pure bottom-up attempt to predict location of next saccade failed (saliency theory: Itti and Koch, 'Computational Modeling of Visual Attention', Nature Rev.: Neuroscience, 2001)
- ⇒ preattentive is not entirely independent of attention (recall bread and kitchen example)

postattentive amnesia.

- does looking at scene for longer time generally improve our understanding?
- this is true when we can link objects in image to familiar representations from long-term memory (LTM). LTM can be queried almost instantaneously.
- but quering short-term memory is slow ($\approx$ 50ms), so if objects in image cannot be detected preattentively or recognized semantically, then allowing some extra time to look at image in advance, does not speed up subsequent target search

- confirmed by simple experiment: target search for arbitrary conjunction of features (e.g. 'green vertical') among several objects. in this case, showing the image for 300ms prior to showing the target description did not speed up the search
- but again: reality more complicated. sometimes can measure positive effect (keyword 'contextual cuing')
- our takeaway: if visualization looks unfamiliar and is not accessible via preattentive processing, then it will take a long time and mental effort to parse and understand

change blindness.

- the photography metaphor from above is inaccurate: it is not true that an increasingly accurate and faithful model of the scene is built in brain over time as the image is gradually processed in greater detail. only short-lived models specifically guided by the current vision task are created.
- good experiment: change blindness. if two similar images are shown next to each other (or separated by a short blink) then we may sometimes not spot the difference, even when it is not particularly small or subtle
- obviously our visual system detects the change in the pictures, but our attention is not drawn to it
- some theories:
  - overwriting. new image overwrites old image, everything that is not saved via abstraction is lost. but: we would immediately see change if no blink in between (have time difference filters in visual cortex, these are tricked by the blink)
  - first impression. abstraction: we do not encode scene as accurate pixel rasterized image, but more like a naive, abtract vector graphic ('elderly couple in front of sphinx statue, trees in background'). if change in image does not require change in this encoding, it is not noticed. example: experiment with movie actor changed mid-film.
  - everything stored, but not compared: illustrated by experiment. subject is asked for directions by person with basketball. students show up, cause some distraction, basketball disappears. only few individuals notice that it is suddenly missing, but more half of the others remember the ball afterwards.

inattentional blindness.

- if our attention is drawn onto a specific task, other things may be missed, even if they are obvious
- simple example: individuals had to examine a cross on slide and see which arm was longer. after two or three trials, a small 'critical object' was added to the image. 25% of subjects missed this; but essentially 100% noticed the object when asked to look out for it
- extreme example: gorilla in ball passing video (available on the internet)

attentional blink.

- subject looks at sequence of images, approx 100ms per image, need to judge if target was present or not, by pressing some button
- when pre-attentive processing has identified relevant candidate (and thus 'activates' conscious processing), the pre-attentive processing is disabled for a short interval
- thus a target present shortly after another one is usually missed

3.5 Derived principles and mechanisms for visualization

becoming aware of our limitations.

- no general purpose vision exists, i.e. one which fully extracts and holds all information in an image, at least after some time of observation
- our vision seems to provide some specific fast low-level info extraction mechanisms, that can then be used to solve a specific set of high-level tasks, the latter probably selected by evolution

reduce number of saccades required for understanding image.

- recall: look-up in short term memory is slow; saccades take time
- use direct labelling
- use systematic coloring (more on this later)

preattentive processing.

- obvious application in data-feature mapping, i.e. when mapping data attributes to visual features;
- if used properly, we can automatically detect distinct trends in dataset, identify 'outliers'
- can also be used for highlighting and drawing attention
- be careful to not produce interference between features! recall feature hierarchies.
- also keep in mind similarity theory: aim for good TN-NN-ratios

familiarity improves understanding.

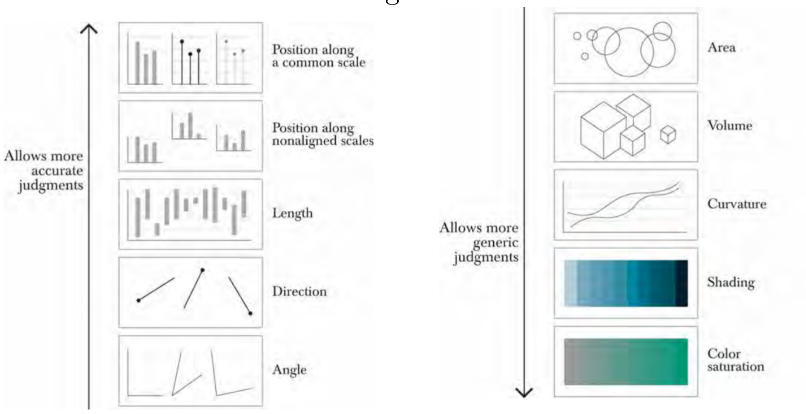
- if a visualization looks familiar, we can very quickly get a good orientation (see ensemble coding)
- so re-use visualization techniques that the reader already knows, e.g. small multiple, smooth animations

attention management.

- make sure that viewers know what to focus on, they may miss important things when focussing on wrong parts of image
- attention is limited in space and time, do not ask for too much at once

visual encoding of numbers.

- William S. Cleveland and Robert McGill: 'Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods', Journal of the American Statistical Association, Vol. 79, No. 387 (1984).
- How accurate are various visual encodings of numbers?



- Example: scale vs disk vs color
- Example: obesity vs BA degrees

abstraction.

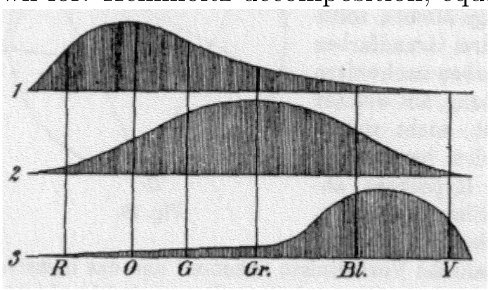
- since internal representation of figure in brain seems to drop many 'unnecessary details' (see change blindness), strong abstraction must happen in visual understanding of scene
- simplify work by using abstracted images.

4 Color

4.1 Color perception

Young–Helmholtz theory / trichromacy.

- https://en.wikipedia.org/wiki/Young%E2%80%93Helmholtz_theory
- developed in 19th century
- Young postulated in 1802 the existence of three types of photoreceptors,
  - https://en.wikipedia.org/wiki/Thomas_Young_(scientist)
  - British polymath, obtained medical doctor degree from Uni Göttingen in 1796
  - also known for advocating the wave theory of light, introducing the early form of the double slit experiment
- Helmholtz in 1850: violet, green, red. Their relative signal strength is perceived as color. Can increase intensity without changing hue.
  - https://en.wikipedia.org/wiki/Hermann_von_Helmholtz
  - also known for: Helmholtz decomposition, equation, ...

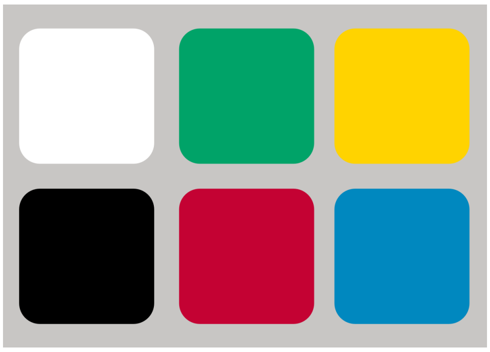


https://en.wikipedia.org/wiki/File:YoungHelm.jpg

- some historical context:
  - Maxwell's equations were published in 1850s to 1870s
  - experimental validation of different wavelength sensitivity in retinal cells of fish in 1956, in humans in 1983
- Tetrachromacy: some animals have four types of cone cells, some can see ultraviolet. example: goldfish

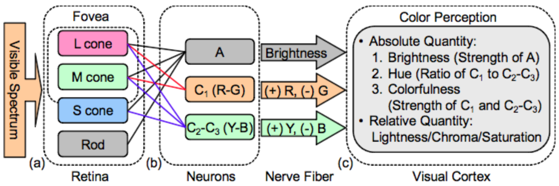
Opponent process theory.

- white vs black, green vs red, yellow vs blue
  color perception based by position on these three axes, 'there is no reddish green, or bluish yellow'



https://en.wikipedia.org/wiki/File:Opponent_colors.svg

- proposed by Ewald Hering in 1892, later formalized by Leo Hurvich and Dorothea Jameson
- present in Goethe's discussion of color, 1810 (Goethe made accurate observations with prisms and the like, but his interpretation was more based on aesthetics than scientific rigor)
- there is a rough equivalence of the (w-bk), (r-g), (y-b) channels in the earliest visual processing in the retina, e.g. [Kandel ER, Schwartz JH and Jessell TM, 2000. Principles of Neural Science, 4th ed., McGraw–Hill, New York. pp. 577–80]



https://en.wikipedia.org/wiki/File:Diagram_of_the_opponent_process.png

- so not in contradiction of trichromacy, latter is essentially the first processing step of the former

4.2 Color models

RGB.

- Additive model, three channels: red, green, blue. Each has intensity between 0 and 1. Can denote a color as point in $[0, 1]^3$.
  - $(0, 0, 0)$ is black, $(1, 1, 1)$ white, (s, s, s) is grey for s in $[0, 1]$.
  - $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ are 'pure' red, green, blue.
  - $(1, 1, 0)$: yellow, $(0, 1, 1)$: cyan, $(1, 0, 1)$: magenta; secondary colors
- used mostly for describing colors on screens, basic image storage, websites,. . .
  - $[0, 1]$ is usually discretized into 256 discrete values (=8bit), given as natural numbers between 0 and 255, or as hex code: 00 to FF
  - color specification in HTML: #RRGGBB
- python example:

  - create simple images of color gradients
  - load image, extract and visualize color distribution (2d/3d pointcloud)

- application examples:

  - use two red and green channel to visualize two densities

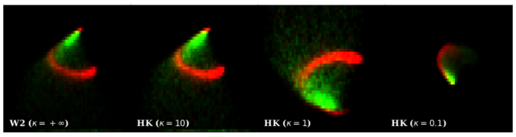


FIG. 5.8. *Distribution of 10k W (red) and QCD (green) jets in the tangent space with respect to the first two dominant PCA modes for* W_2 *and* HK *distances with* $\kappa = 10, 1, 0.1$ *(from left to right).*

  - color histogram matching

- RGB is a relative color space:

  - depends on what red,green and blue are used for actual mixing. so in principle is device dependent.
  - distances in the 'RGB-cube' are not necessarily accurate reflections of perceived similarity by humans

Absolute color spaces, example: CIE color spaces.

- specified by the International Commission on Illumination, (Commission internationale de l'éclairage)

- attempt to find objective descriptions of color, based on 'standard observers' and 'standard illuminants'

- specification is copyrighted, precise standards and documentation has to be purchased from the CIE

- CIELAB: three dimensional, axes are L (lightness), a (red-green), b (yellow-blue), distances in these coordinates are supposed to be similar to perceived color differences

- can be transformed to RGB if an identification profile is fixed

- similars concepts: Adobe RGB, sRGB

HSL and HSV.

- RGB is intuitive at level of additive mixing of signal to three cone types

- at level of perception other parametrization seems more appropriate:

  - hue and saturation describe color in '(blue-yellow)-(red-green)-plane'
  - lightness / value: interpolation between black,color, and white

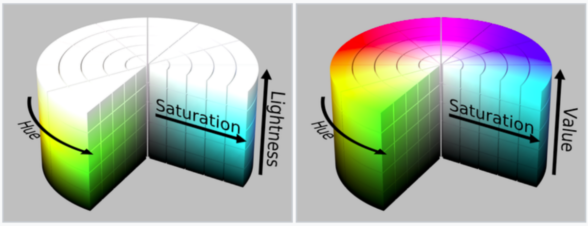


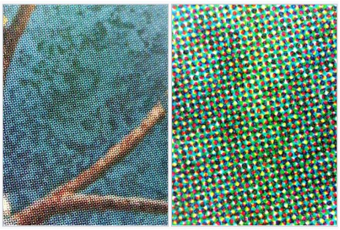
Fig. 2a. HSL cylinder. Fig. 2b. HSV cylinder.

https://en.wikipedia.org/wiki/HSL_and_HSV

- transformation between RGB and HSL/HSV values
  - note: so HSL/HSV are merely alternative parametrizations for RGB color space, so also still a 'relative' color space
  - is a parametrization in which certain 'natural' operations are easier to express
- python example:
  - construct a hue circle
  - simple saturation filter

CMYK.

- subtractive color model. base colors: cyan, magenta, yellow, black (k stands for key, usually the printing plate with most structural detail)
- color mixing laws from kindergarden
- usually used in print



https://en.wikipedia.org/wiki/CMYK_color_model

- CMYK is 'overparametrization': CMY channels would be enough. K is used to save expensive color ink, and to get deeper black
- again device dependent, conversion to and from RGB requires some registration profile
- we do not need to be particularly concerned about it for now, but if you ever go into print, remember this

Informative blog post.

- https://medium.com/the-philipendium/the-contradictions-in-how-we-explain-colors-99f11c894fe7

4.3 Color in visualization

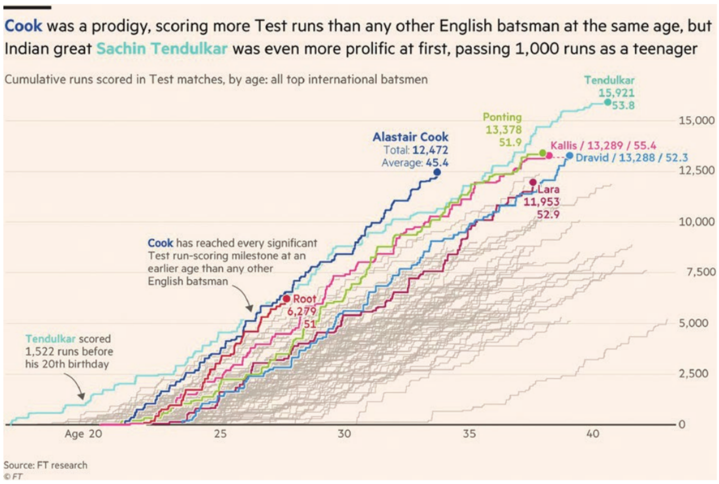
(next paragraphs roughly following [Kirk: Data Visualization, Chapter 9], examples from there unless mentioned otherwise)

Preliminaries.

- in principle: space of perceivable colors is three-dimensional, could encode three dimensions via color
- but: color shading and saturation are at the bottom of the Cleveland-McGill scale, so only qualitative presentation of data, not precise and quantitative
- also: 'unpacking' three dimensions from color within a 2d plot can be challenging for viewers, due to our difficulties of conceptualizing more than three dimensions

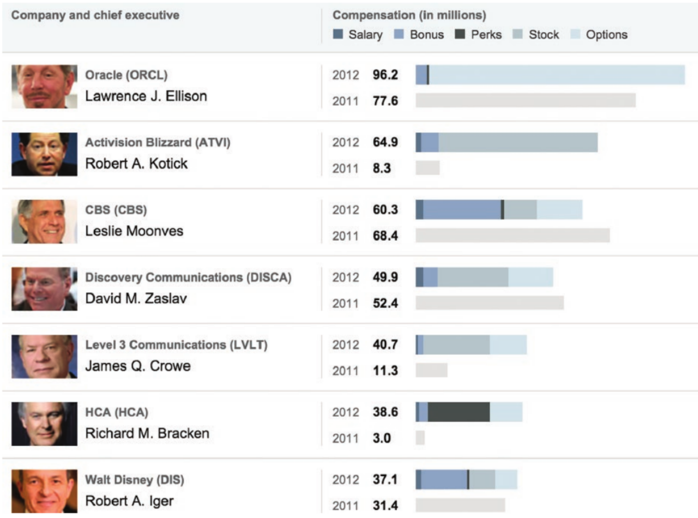
Highlighting.

- use color for pop out effect

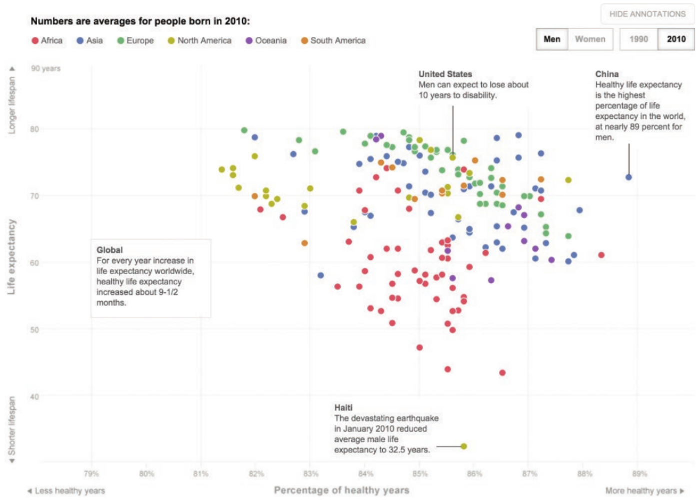


Nominal data.

- need clearly distinct colors, ideally without obvious order
- simple example: different hues for same saturation and value in HSV
- usually the default colors in plotting software bad example:

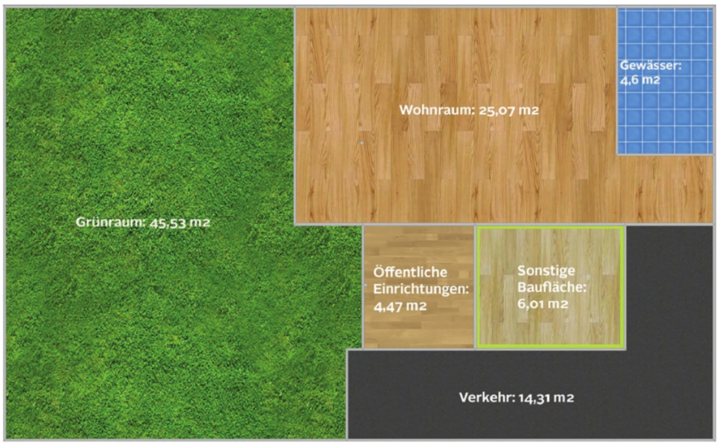


- how many categories can we visually distinguish? Kirk: at most 12 categories via hue (p259). General advice: as few as possible, try to have at most 6.



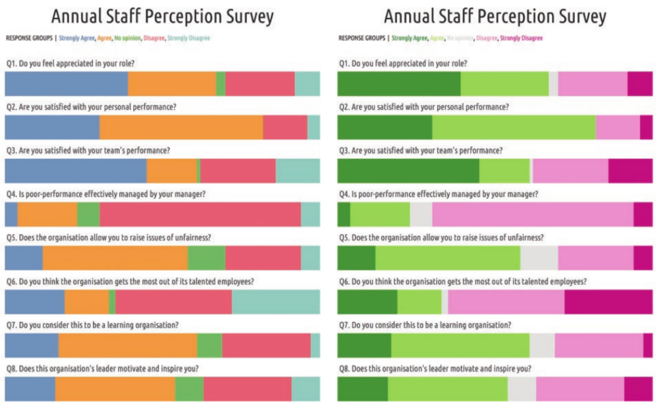
more information can be shown better via annotation, multiple additional plots (e.g. focussing on sub-groups), interactive visualization

- creative example: relative space usage in Vienna

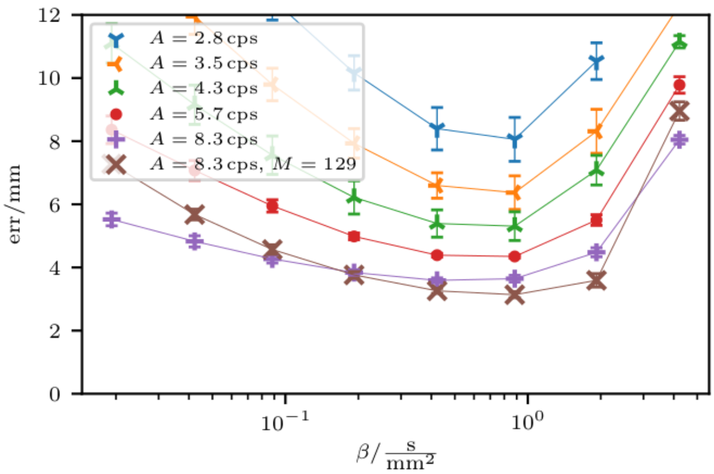


Ordinal data.

- colors should be ordered in an intuitive way

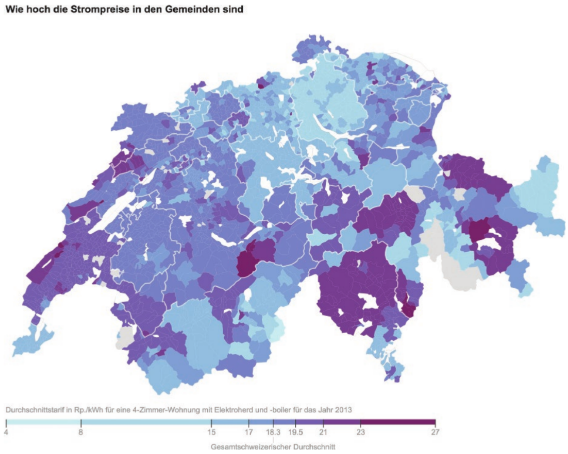


- bad example: dynamic PET.
  categories are ordinal (even rational), but color scheme is not; with direct labeling maybe do not need color at all, since curves are almost 'parallel'
  one exception: last curve, $M = 129$, which could be handled separately

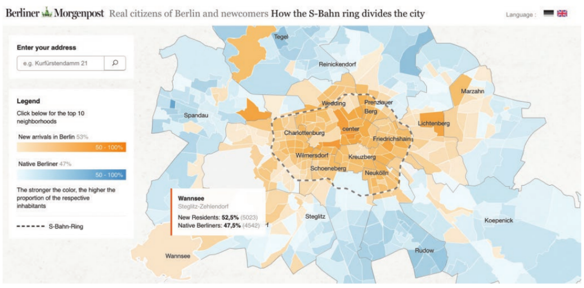


Interval or ratio data.

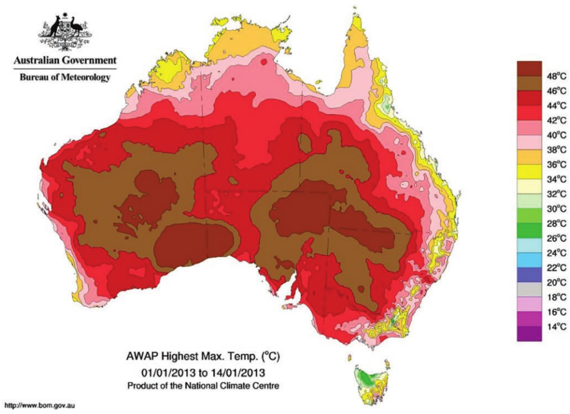
- use color scales / color maps



- signed vs unsigned data ([-1,1] vs [0,1])



- do not use rainbow scale (except maybe for cyclic data! see below)



4.4 Color maps

Python example.

- value variation for fixed hue in HSV
- two opposing hues for signed data

- saturation variations
- summary: in principle can easily create color maps, but they are not perceptually uniform, do not make the most of our color perception

Example: more general color maps from matplotlib.

- https://matplotlib.org/stable/gallery/color/colormap_reference.html
- categories: sequential, diverging, cyclic, qualitative
- how to choose a good color map? https://matplotlib.org/stable/tutorials/colors/colormaps.html
- observation: brightness variations within colormaps can be very different

Perceptual uniformity.

- many standard colormaps in various software is not perceptually uniform, can cause substantial perceptual 'gaps'
- matplotlib contains a few perceptually uniform ones, e.g. viridis or plasma
- good resource for more perceptually uniform color maps: https://colorcet.holoviz.org/
  maintained by Peter Kovesi, research fellow at School of Earth Sciences - Centre for Exploration Targeting, The University of Western Australia.
  installation via `conda install colorcet` or `pip install colorcet`
- python examples

Other ways to improve perceptual resolution.

- discretize values into bins, so boundaries between values become more visible
- use a high-frequency cyclic scale,
- draw contour lines

Normalization / truncation / transformations.

Multidimensional data.

- maps and vector fields: 2d to 2d

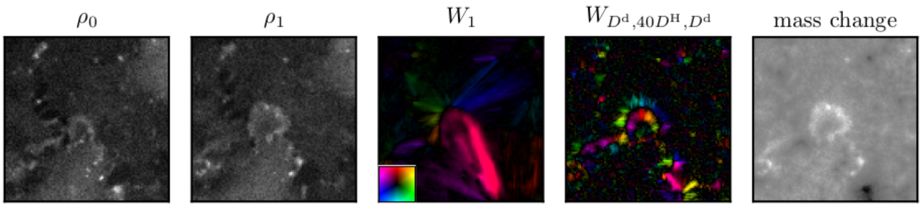
$$S : (x, y) \mapsto (f(x, y), g(x, y))$$

- in polar coordinates:

$$\begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix} = r(x,y) \cdot \begin{pmatrix} \cos(\varphi(x,y)) \\ \sin(\varphi(x,y)) \end{pmatrix}$$

- can in principle be represented by color function. For example: use φ as hue (scale $[0, 2\pi]$ to $[0, 1]$), use scaled r as value (need to truncate at some maximum value)

- example: unbalanced W1 field

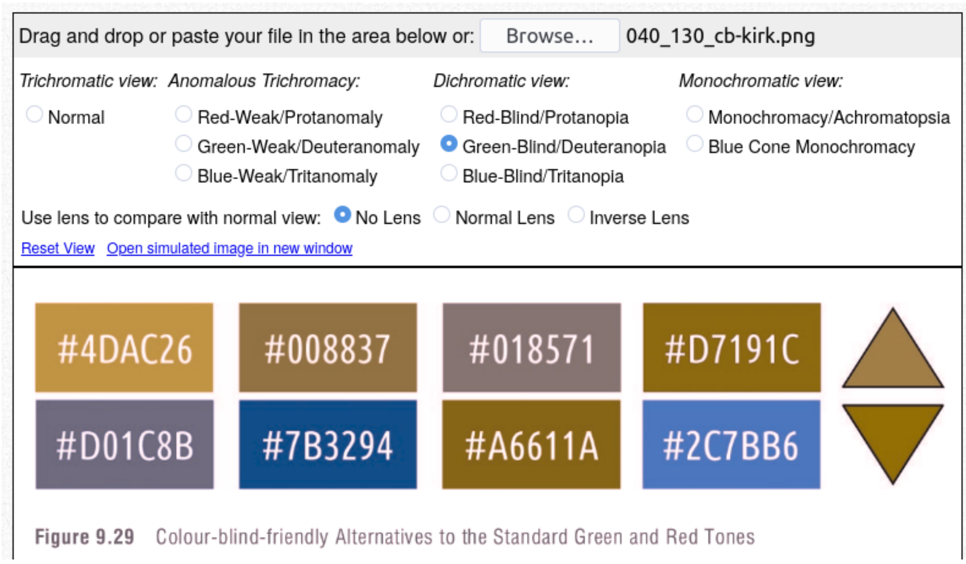


- comment: can only convey a qualitative picture

4.5 Color deficiency

- example palette from [Kirk]:

Figure 9.29 Colour-blind-friendly Alternatives to the Standard Green and Red Tones



- matplotlib viridis seems to be ok

- simulation of effects at https://www.color-blindness.com/coblis-color-blindness-simulator/

5 Data

Sources for this section were:

- [Kirk: Data Visualization, Chapter 4]
- Online course "R for data science", https://r4ds.had.co.nz/introduction.html, Sections 5,7,13.
- Pandas documentation, 'Getting started', https://pandas.pydata.org/docs/getting_started/index.html#getting-started

5.1 Data types and structures

Primitive data types Classification of data types due to psychologist Stanley Stevens, 1946. No rigorous, universal system; some debate and more refined proposals exist. But the general idea seems to be consensus. No need to be dogmatic about this. Just be aware of some conceptual properties of data.

- textual: unstructured passages of text
  - responses to 'Any other comments?' in a questionaire
  - abstract for academic research article
  - product description in an online shop most difficult to handle in automated / quantitative analysis → natural language processing
  - in programming: string types (not primitive in that sense, but highly complex substructure)
- nominal: from a fixed set of categories
  - gender of survey participant
  - meals available on restaurant menu
  - city, country of birth
  - edges in a graph (described by indies of two vertices)
  - in programming: enum types, or int types but we never use ordered comparison or addition, only use equality test
  - can have a hierarchical structure: cities can be grouped into countries
- ordinal: categories can be ordered, but no notion of 'distance' or 'difference' between categories.
  - options on a survey to which extent you agree with a statement
  - 'non-quantitative' size of clothing: XXS to XXL
  - rank of police officer
  - rank in a competition (but not the finishing time or score)
  - in programming: int or char types, but do not use differences, only equality and order comparison

- interval: there is a notion of difference, but not of ratio
  - temperature in degrees Celsius ($20°$ is not twice as hot as $10°$)
  - dates (but ratios between differences can be meaningful)
  - in programming: floating point types
- ratio:
  - most physical measurements: length, duration, mass, temperature in Kelvin
  - in programming: floating point types

Composite data structures. Primitive types can be combined into more complex data structures.

- lists / tuples: ordered list of entries. Entries may be different data types, primitive or nested lists. In principle can build arbitrarily deep and rich structures.

Simplify things by imposing more structure:

- array: one- or multi-dimensional Cartesian grid of values, usually of a common primitive type
- struct or composite types: defined by set of fields (or variables). Each field has a specified data type (can be primitive or another struct). An instance of a struct must provide a value for each field (maybe have a convention for missing values).
- table / 'data frame': a list of instances of the same struct; fields/variables are usually referred to as columns; different entries as rows

Composite data often represents some of the following structures:

- images: usually represented as rectangular two-/three-dimensional array of pixel intensities
- time series: measurement of some quantity over time, e.g. temperature evolution throughout the day. If time interval is regular, it can be represented as one-dimensional array of values. Otherwise could be list of pairs of time stamp and value (e.g. times when a car passed on the road + type of car).
- functions: abstract definition of a function is list of associated pairs of input and output values. Computationally and in experiment the concrete representation can vary extremely.
  - store values at fixed locations such as grid points (e.g. image, fixed-interval time series)
  - store pairs of location and values where measurements are available (e.g. weather measurement stations)
  - $\Rightarrow$ visualizing functions poses several challenges, high dimensionality, irregular availability of values, representation of uncertainty
- graphs / networks: usually consists of two parts:
  1. list of vertices or nodes, with properties such as label, type, weights, ...
  2. list of edges between nodes, with properties such as orientation, type, length, ...

5.2 Data processing / pipeline

- acquisition
  - source depends strongly on context (internet, company records, scientific measurements obtained by ourselves or collaborators)
- examination
  - is meant here at very fundamental level: what even is my data? which files? what types, format, relation? amount and range? Can be very relevant 'in the wild'
  - quality and representativeness? missing values, errors, formatting inconsistencies, corruption, duplicates, out of date
- cleaning / pre-processing
  - fix formatting errors, typos, add missing values
  - apply filters
  - compute derived data (averages, reductions, aggregations, clustering, binning,...), but often it is too early to know what will be relevant

  personal recommendations:
  - keep all iterations/versions of the data (if possible/practical), ideally in a format where changes can be traced transparently (via diff or with dedicated analysis script)
  - add meta data:
    * document source of external data
    * for generated / processed data: document used parameters, conventions; examples: encoded directly in filename; added as comment at beginning
    * use meaningful directory structures
  - automate and standardize cleaning and pre-processing as much as possible via scripts, save these scripts with the data for documentation
- data exploration
  - more detailed and quantitative than examination: what stories does the data tell?
  - is it compatible with hypotheses? what are even good hypotheses? are there statistically significant dependencies?
  - usually main challenge: find clear low-dimensional structure encoded in vast high-dimensional dataset
  - 'Exploratory data analysis is an attitude, a flexibility, and a reliance on display, not a bundle of techniques, and should be so taught.' (John Tukey: We Need Both Exploratory and Confirmatory, The American Statistician, 1980)
  - statistics, visualization, quick flexible movement play a key role, will be demonstrated throughout lecture
- main step: actual work on data, extracting information required for visualization
- backup, archiving, publishing

- use version management, repositories, redundant storage
- when project is done, make sure everything is stored and documented such that you can later retrace your steps and reproduce all findings
- in spirit of reproducible research data is also often published along with the paper

5.3 Basic data transformations

This section is not meant to be an exhaustive tutorial, or even a pandas-specific tutorial. It intends to illustrate a few basic common operations on tabular data. See python example for actual section.

6 A tour of various chart types and data representations

- this section is primarily presented by python examples, this script merely collects a few keywords

6.1 Bar charts

- data format: interval vs nominal (or ordinal)
- if multiple series: grouping, stacking (my recommendation: at most 4 series)
- horizontal might be useful for better usage of space
- annotations to avoid look-up of values
- can often be replaced by a scatter or line plot, when x-axis has more structure
- careful with visual vibrations due to strong contrasts between bars and gaps

6.2 Scatter plot

basic scatter plot.

- most basic visualization for a point cloud in 2d
- error bars for encoding uncertainty

additional degrees of freedom.

- marker style: nominal degree of freedom, separate different dataseries; if possible, support with color
- bubble size: encode weight or count. careful: use area, not radius for encoding quantity
- marker color: separate different dataseries; or encode height or temperature; recall lesson on colors and color maps

3d.

- visualize additional dimension
- works best with animation or interactive rotation
- perspective is often problematic in 2d static renderings, use with caution; use auxiliary lines such as a mesh grid or color
- data has even more dimensions? will be addressed later

Lines

- when is adding lines appropriate/helpful?
  - if x-axis is interval data, actually or practically continuous
  - to visually connect a data series for better distinction from other series
  - to emphasize trends (line gives explicit encoding of slope) $\rightarrow$ regression lines
  - if line is decent approximation of intermediate values that might also be measured
- do not use lines if data is highly oscillating, or subject to strong stochastic fluctuations
  - on data with clear trend + overlap by noise, maybe a smoothed interpolation is appropriate (always be careful to mention how it is generated)
  - simple example: linear regression
  - simple example: kernel based interpolation

interesting variant: 'trajectory plot'

- show temporal trajectories in scatter plot as connected markers
- clearly indicate initial and final point (=direction of time)
- show intermediate markers for impression of velocity

Logarithmic axis scaling

- applicable to positive data of ratio type
- may be appropriate when values range over several orders of magnitude, when ratios (quotients) are meaningful
- to visualize certain dependencies: exponential or power laws
- careful with regression in log plots!

6.3 Distribution of points

Scatter plot.

- in principle can visualize distribution of points, but difficult to visually evaluate high-density regions

Histograms.

- better representation of distribution of large amount of points, 1d and 2d
- no general rule for proper choice of bin number
- extension: weighted samples
- optional but handle with care: variable bin widths

Density estimation.

- simple histograms are sometimes unstable under small shifts in the bin sizes or locations
- for some applications a more sophisticated density estimation may be more appropriate
- kernel density estimation is a simple method that is easy to apply, e.g. https://scikit-learn.org/stable/modules/density.html
- choosing the width of the kernel is analogous to the width of the bins
- encode density as color, up to three
- density estimation and contour lines

Higher dimensions.

- representation as voxel image, https://matplotlib.org/stable/gallery/mplot3d/voxels.html, https://terbium.io/2017/12/matplotlib-3d/

6.4 Point data with stochastic functional relation

Motivation.

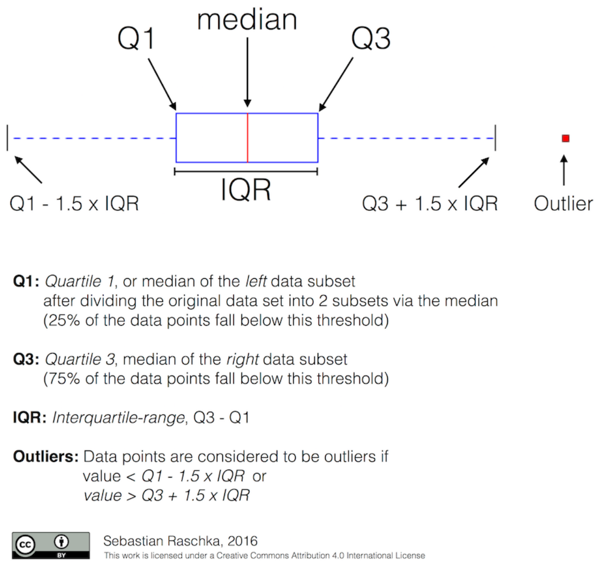
- y=f(x,z) where z is random

Mean and standard deviation.

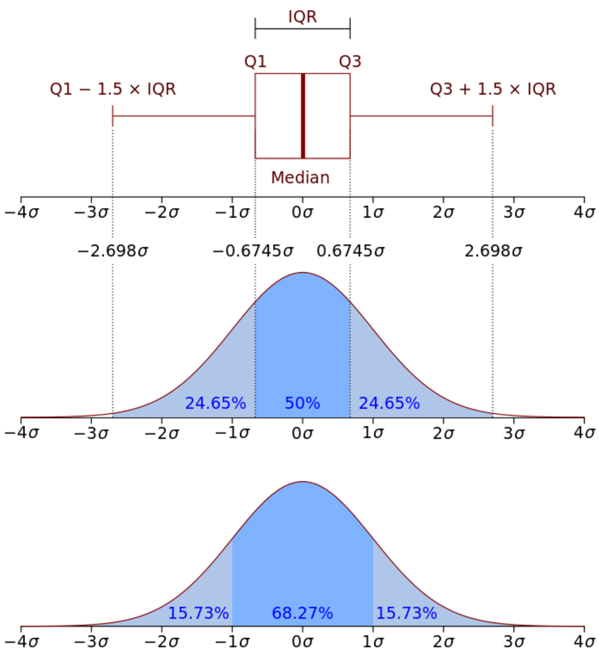
- easy to compute and to visualize
- usually inappropriate for non-Gaussian distributions

Box plot.

- mark median, first and third quartiles (Q1 and Q3), heuristic for ouliers: more than 1.5 interquartile ranges (Q3-Q1) below or above Q1,Q3
- explanation and illustration:
  - https://en.wikipedia.org/wiki/Box_plot
  - matplotlib documentation: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html



– comparison with Gaussian density: https://en.wikipedia.org/wiki/File:Boxplot_vs_PDF.svg



- extended version with variable width, confidence intervals for median, ...
- better representation of non-Gaussian, but 'sufficiently simple' distributions (unimodal)

Violin plot.

- run a small local Gaussian kernel density estimation on each Y-dataset
- visualize the density as a small vertical 'mini-graph'
- can also add median, quartiles, and outliers

Plot percentile levels.

- very intuitive, contains essentially all the information, but hard to read

Summary and some comments.

- for large batches (points in y per x bin) or many x bins we need a compact graphical summary of the y-distribution for each x
- for 'simple distributions', 'simple representations' are possible, e.g. mean + standard deviation
- increasing level of detail: quartiles, outliers, density estimation. individual combinations are possible (violin plot with outliers, etc.), easy to implement with matplotlib

6.5 Vector fields

Quiver plot.

- represent vector field by drawing many small arrows
- anchor/pivot of arrows at start can lead to biased perception, sometimes anchor/pivot in middle is more reasonable
- careful about aspect ratio and orientation of arrows
- number / scale of arrows: trade-off between resolution and over-cluttering
- scale of arrows: 'to-scale' vs 'fitted'
- additional degree of freedom: color
  examples: hue for orientation, length as saturation/value

Deformation plot.

- quiver plots work well for showing tangential or infinitesimal directions, not so well for large displacements
- maps $\mathbb{R}^2 \to \mathbb{R}^2$ that act as transformations can in principle be visualized via HSV color coding, but this only givey a very rough qualitative impression
- visualize such maps as deformations acting on a grid, gives good impression of contractions and compressions, rotations
- make sure, points in grid and deformed grid can be identified with each other
- be careful: when creating such a map, numerically one sometimes needs the inverse of the actual map to get the desired figure

Stream plot.

- stream lines of vector field: solve differential equation / 'integrate' vector field
- intuititvely: visualize trajectories of individual particles in flow field
- appropriate for flow, velocity, force,...
- additional degree of freedom: line width, color (for field strength, orientation)

6.6 High-dimensional data

Historical curiosity: Chernoff faces.

- introduced by physicist Herman Chernoff in 1973, informally: "how many degrees of freedom can be encoded in the marker?", leverage our familiarity with human faces
- encode approximately 18 values in the 'marker' by drawing a little face
- while we cannot absorb all 18 values, we are still able to detect outliers from a relatively smooth distribution of values
- but to be honest: other visualization techniques work just as well / even better, more of a curiosity in my opinion

Principal component analysis (PCA).

- prototypical example for dimensionality reduction method
  - often reasonable assumption that high-dimensional data does not 'occupy' full high-dimensional space, has low 'intrinsic' dimensionality, just embedded into larger space
  - fundamental data analysis problem: identify low-dimensional sub-manifold, ongoing research
  - PCA is very simple case: assume sub-manifold is linear sub-space
- brief overview
  - cannot give mathematical details here. basic idea: iteratively find the directions along which data has most variance
  - then compute projections of samples onto these axes ⇒ get lower-dimensional approximation
  - useful for visualization or further processing (reduce risk of overfitting)
  - can visualize coefficients of projections as point cloud, or individual projections themselves
- usually also look at spectrum of eigenvalues (=how much variation along subsequent axes)
- example: (almost) Gaussian distributions
- limitations of linear sub-space assumption:
  - missing overlap between Gaussians
  - non-linear low-dimensional manifold
  - non-linear extensions and manifold learing are active fields of research, see also: graph embeddings later in lecture

Comparing point clouds.

- if a sample is not given by an individual point, but by a whole point cloud, the situation becomes even trickier
- plotting the point clouds may not be helpful, in particular in higher dimensions
- good strategy: find simpler representation for each point cloud
- simplest case: covariance matrix
- can then visualize / compare the representations instead
- this is just an extremely simplified example, just for illustration:
  - in practice data samples can be more complicated (far beyond Gaussian distribution, high dimensions, images, videos, graphs, ...)
  - finding good representation and mode of comparison is extremely non-trivial, theoretically and computationally

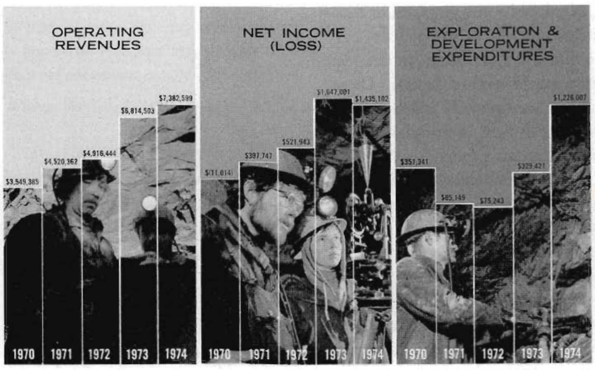
7 Interlude: How to lie with charts

- This section follows roughly the book 'How to lie with charts' by Alberto Cairo (see Literature). Most examples are taken from there, marked with [Cairo]. Some examples are taken from 'The Visual Display of Quantitative Information' by Tufte, marked by [Tufte].
- Cairo identifies several ways that charts may lie to us:
  - by being poorly designed
  - by displaying dubious data
  - by displaying insufficient or inappropriate data
  - in the way uncertainty is treated
  - by suggesting misleading patterns
  - by pandering to our expectations and prejudices
- **DISCLAIMER:** The data and charts shown in this section are often fictional and have not been verified by me. Due to the sheer mass of examples not each example can be discussed in detail. The examples should therefore not be misinterpreted as 'sources' on the respective topics but merely as illustrations for certain misrepresentations of data and relations. The list given in this section is by no means exhaustive.

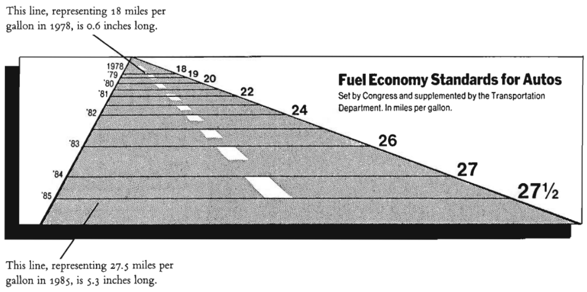
7.1 Poor design

Skewed encoding.

- graphical representation of numbers is not consistent with numbers
- seen previously: area and radius of bubbles
- example [Tufte]: missing base line

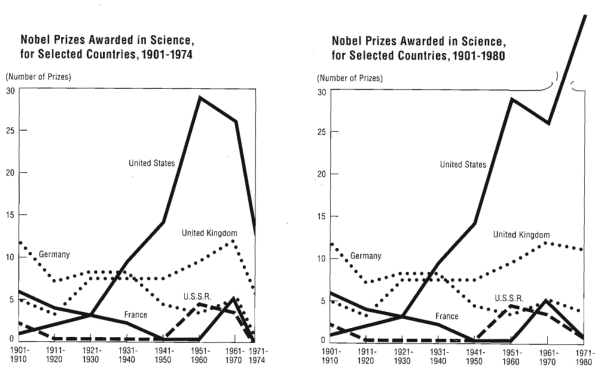


Tufte's lie factor: size of effect in graphic / size of effect in data.



- example [Tufte]: fuel economy, miles per gallon requirements
- 1978: 18mpg represented by 0.6 inches
- 1985: 27.5mpg represented by 5.3 inches
- lie factor=(5.3/0.6)/(27.5/18)=5.78 (computed slightly differently in Tufte's book)

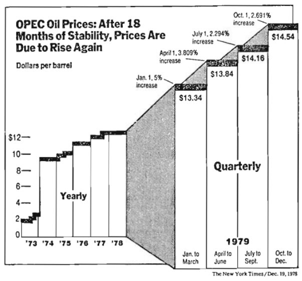
Inconsistent design.



- example [Tufte]: nobel prices (left: original version, right: fixed version)
- number of nobel prizes per decade is shown, but last data point is only over four years

Distorting 3d effects.

- example [Tufte]: oil prices



New York Times, December 19, 1978, p. D-7.

Five different vertical scales show the price:

During this time	one vertical inch equals
1973–1978	$8.00
January–March 1979	$4.73
April–June 1979	$4.37
July–September 1979	$4.16
October–December 1979	$3.92

And two different horizontal scales show the passage of time:

During this time	one horizontal inch equals
1973–1978	3.8 years
1979	0.57 years

- example [Cairo]: 3d charts for fictional company

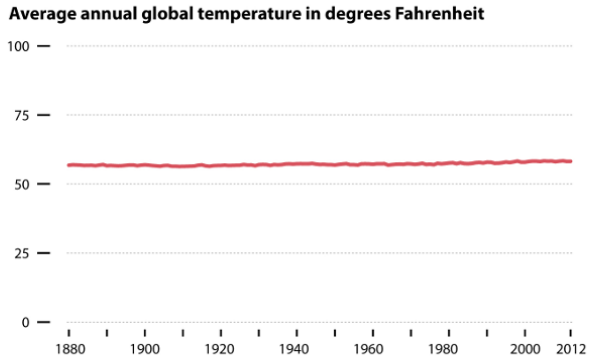
Why it matters?

- taken from [Cairo]
- study on the deceptiveness of deceptive graphics: Anshul Vikram Pandey et al.: 'How Deceptive Are Deceptive Visualizations? An Empirical Analysis of Common Distortion Techniques', New York University Public Law and Legal Theory Working Papers 504 (2015)
- charts comparing the drinking water supply situation in two fictional cities
- subjects were shown representative or skewed versions, all charts in addition contained the correct numbers
- subjects were subsequently asked to rate the severity of the differences between the two cities
- subjects with wrong chart versions were substantially misled, effect weaker in more educated subjects, but they also misled

7.1.1 Axes and scaling.

Role of the baseline.

- previously: emphasis that y-axis should start at zero
- but it is not always so easy: zero is sometimes arbitrary
- example [Cairo]: degrees Celsius or Fahrenheit in global warming plot



- similar example [Cairo]: what is appropriate vertical scale for national debt?
- relevant vertical scale depends on context
- starting at zero may not be necessary if numbers are not encoded by length (bar chart) but by position (scatter / line plot)
- but need to make sure that readers are not misled (e.g. do not draw axis at non-zero value, additional hint in caption)

Aspect ratio.

- changing the aspect ratio of a graphic can change its perception
- example [Cairo]: life expectancy

Logarithmic scale.

- sometimes non-linear scales of the axes are more appropriate
  - allows separation of data points at various orders of magnitude
  - allows accurate representation of exponential growth
  - careful: can potentially be misleading
- example [Cairo]: GDP vs life expectancy
- example [Cairo]: exponential growth

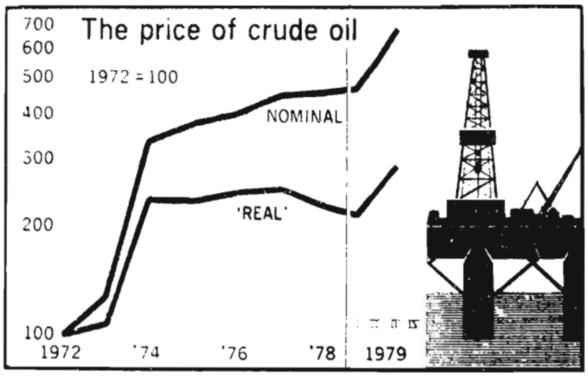
Multiple plots with different scales.

- example [Cairo]: population in the Americas. In a single plot scaling imposed by large countries makes it hard to see differences in smaller countries. Breaking the axis is a common technique but it can be misleading. Alternative: show additional second chart with a 'zoom' on the smaller countries.

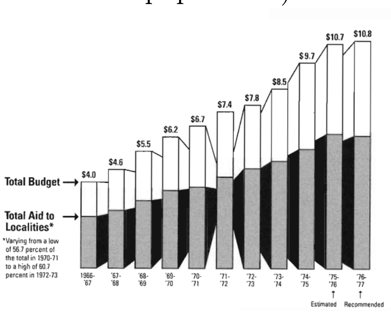
7.2 Dubious data

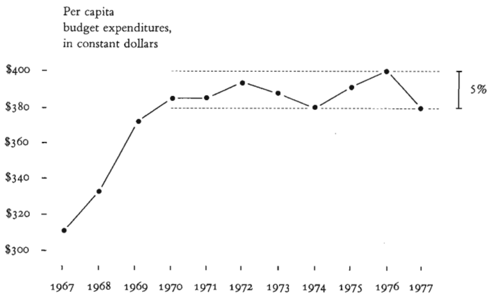
Inflation.

- adjust monetary values for inflation, possibly for purchasing power parity
- example [Tufte]: oil prices and inflation



- example [Tufte]: budget increase (contains also several design flaws, and does not correct for increase of population)





- example [Tufte]: Playfair (has also published a chart on the increase of the national debt, in nominal currency and with dramatic portrait format, but has included also a 'de-sensationalized' landscape format version with inflated-adjusted numbers)

Faulty data.

- the displayed data can simply be wrong / contains glitches that were not noticed
- example [Cairo]: pornhub (pornhub published statistics about access numbers in various states of the US based on IP addresses. Kansas was a strong outlier. Actual explanation is not immense porn consumption in Kansas, but that IP addresses that were assigned to US, but not to a federal state, were incorrectly assigned to Kansas)

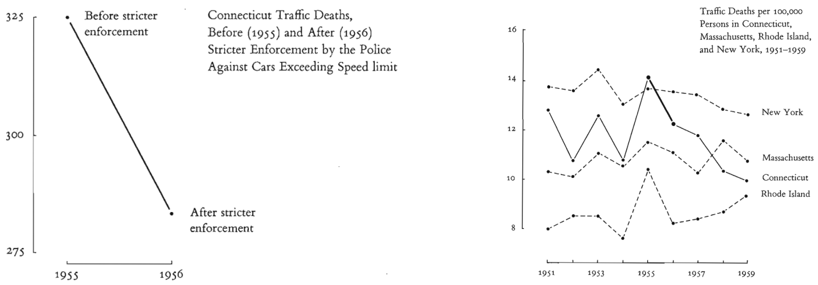
Non-representative data / apples and oranges.

- example [Cairo]: prices for health care (appeared on a news website, confirms bias one may have as European. no details on data given with chart, only source. After looking up source: US data is averaged over large number of medical claims. Data for rest of countries are given by insurance companies that are members of the agency which published the data, not representative for each country, entirely different source. So hard to compare the numbers.)

7.3 Insufficient data

Context: preceding and subsequent data points, related data series.

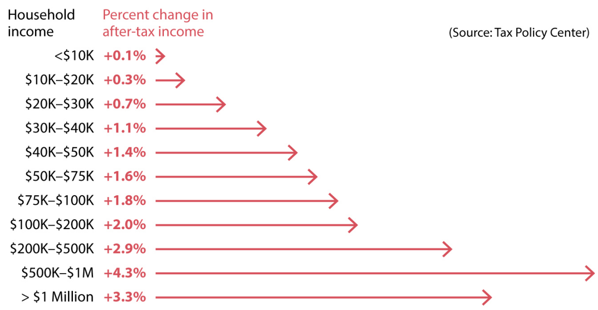
- example [Tufte]: traffic deaths and stricter speeding enforcement (shown chart only shows to subsequent data points in one state. is this part of a larger trend? how did the number evolve in other states without the change in enforcement?)



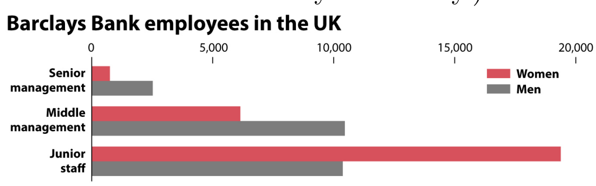
- example [Cairo]: Dow Jones since Trump inauguration

Too much summarizing.

- example [Cairo]: tax cuts and jobs act (Paul Ryan: 'average family saves 1182$'. but how are savings distributed over different income groups? larger incomes get a larger relative tax cut.)



- example [Cairo]: Barclays gender pay gap (BBC: 'women earn up to 43% less than men at Barclays'. Impression: difference due to unequal pay for similar positions. more detailed data: difference due to much fewer women in senior and management positions. So still a stark inequality, but must be addressed very differently.)

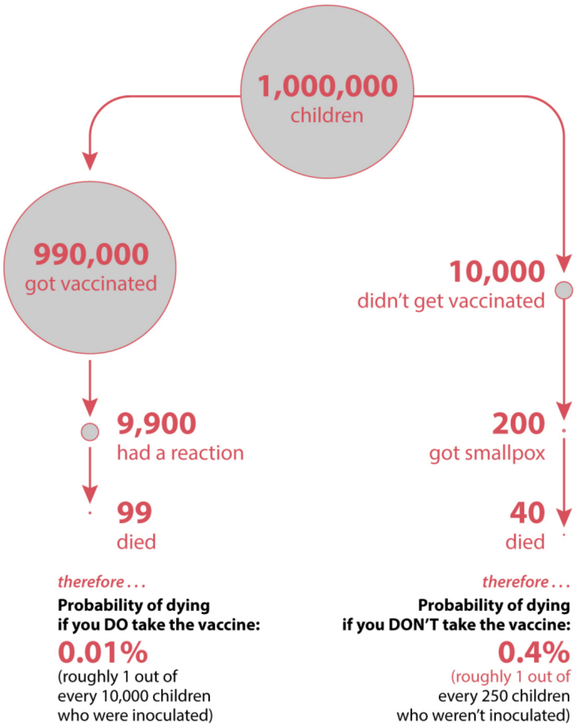


Too much data.

- try to hide the message by burying it
- example: all 100+ fertility curves without any highlighting in the example from section 1

Absolute numbers vs ratios and conditional probabilities.

- example [Cairo]: vaccination (fictitious example (inspired by reality): more children dying due to the vaccine than due to smallpox. but: ignores total number of vaccinated or non-vaccinated children. conditional probability for death among vaccinated children is much lower than among un-vaccinated children.)



- example [Cairo]: Charleston church shooting 2015 (Shooter was partially motivated by statistics of 'black-on-white crime'. Chart made it look as if black violent offenders were specifically targeting white people. This ignores the composition of the general population. See python example.)

What is the right metric for success?.

- example: movies and box office (inflation, number of cinemas, marketing and filming budget,. . .)

7.4 Dealing with uncertainty

- example [Cairo]: climate change (Bret Stephen in New York Times: 'modest warming since 1880 by 0.85 degrees Celsius and its human cause are indisputable, scientific models for future trend are subject to uncertainty and should be treated as such' (paraphrasing by me). But: 0.85 degrees Celsius is by no means modest. While future models carry uncertainty, they have shown to be conservative in the past, and the general trend in the overwhelming majority of models points upwards. This should not be discarded by a mere reference to associated uncertainty.)

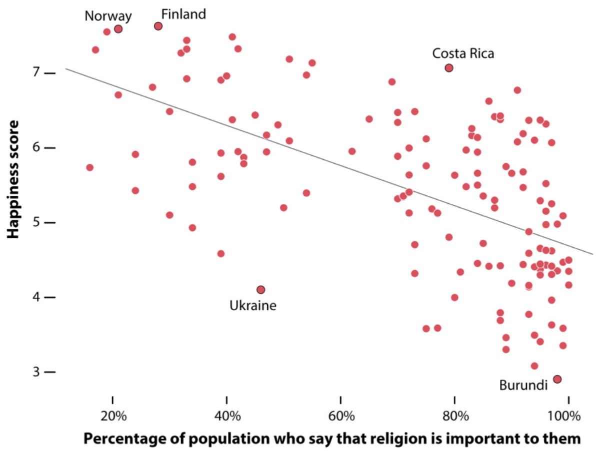
- example: election polls

- example: unemployment rate (4.3% to 4.4% in one month means almost nothing. it is mostly noise. look at longer time-frame to see signal. similar: daily corona virus infection numbers)

7.5 Suggesting misleading patterns

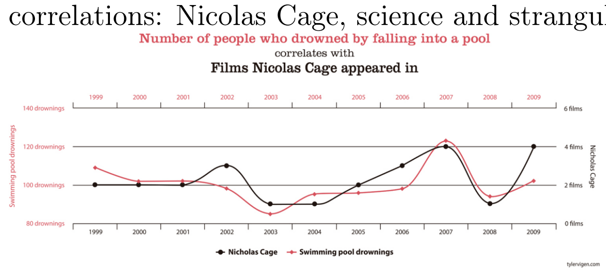
correlation vs causation.

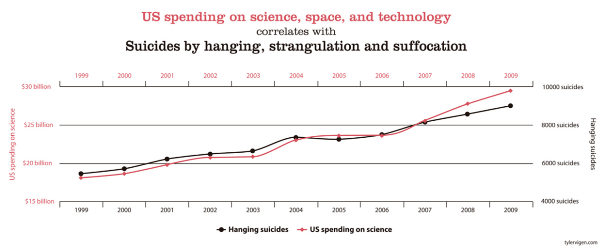
- example [Cairo]: Obamacare and unemployment (signing of Affordable Care Act coincides with a beginning of economic recovery. but no causal relation can be inferred at this point)
- example [Cairo]: religion and happiness



  - on the country level the two seem to be negatively correlated. but: cannot infer causal relation.
  - maybe confounding factor: inequality
  - at the individual level the correlation may even be reversed in countries with high inequality

- example: spurious correlations: Nicolas Cage, science and strangulation





population fallacy.

- example [Cairo]: cigarettes vs life expectancy (trend at the country level seems like a positive correlation, because wealth may be a confounding factor. at the individual level we recover well-established negative correlation)
- example [Cairo]: Kansas voting against their interest

Simpson's paradox.

- example: kidney stones (https://en.wikipedia.org/wiki/Simpson%27s_paradox)

Stone size \ Treatment	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

misreading exponential growth.

- example [Cairo]: Cambrian explosion (creationists argue that sudden increase in diversity of genera is proof of intelligent creator, scientific data suggests it is not 'sudden' and part of a longer growth trend)
- in linear scale, a period of exponential growth can look like an 'off-on'-situation

7.6 Epilogue

For chart creators: Tufte's principles of graphical integrity.

- The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented.
- Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data.
- Show data variation, not design variation.
- In time-series displays of money, deflated and standardized units of monetary measurement are nearly always better than nominal units.
- The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data.
- Graphics must not quote data out of context.

For chart consumers: Cairo's principles.
With the internet we consume a lot of charts from a wide range of sources, often not professional or acting in good faith, and often without sufficient context. It is impossible to scrutinize every chart that we see. But we should be wary. Cairo gives some advice [Cairo, end of Chapter 3, subset of full list]:

- Don't trust any chart built or shared by a source you're not familiar with until you can vet either the chart or the source, or both.
- Don't trust chart authors and publishers who don't mention the sources of their data or who don't link directly to them. Transparency is another sign of appropriate standards.
- Expose yourself to sources you disagree with, and assume good faith on their part.
- Don't assume ill intentions when haste, sloppiness, or ignorance is the more likely explanation for a bad chart.
- But trust has its limits. If you begin spotting a pattern of misdeed in a source on your list, erase it.
- Follow only sources that issue corrections when they ought to and that do it visibly.
- All journalists have political views. Who doesn't? But most try to curb them, and they do their best to convey, as famed Watergate reporter Carl Bernstein likes to say, 'the best obtainable version of the truth.'
- Telling the difference between a merely partisan source, there are reliable ones all over the ideological spectrum, and a hyperpartisan one can be tricky. It will require some time and effort on your part, but there is a very good clue you can begin with: the tone of the source's messages, including whether that source employs ideologically loaded, bombastic, or aggressive language. If it does, stop paying attention to it, even if it's just for entertainment.
- Expertise matters, but it's also specific. When it comes to arguing over a chart about immigration, your judgment as a layperson is as valid as that of a mechanical engineer or someone with a PhD in physics or philosophy. And your opinion is less likely to be accurate than are the ones expressed by statisticians, social scientists, or attorneys who specialize in immigration. Embrace intellectual modesty.
- It's easy to be overly critical of charts that depict realities we'd rather not learn about. It's much harder to read those charts, to assume that their creators likely acted in good faith, and then to coolly assess whether what the charts show has merit. Don't immediately snap to judgment against a chart just because you dislike its designers or their ideology.
- Conversely: The more ideologically aligned you are with a publication, the more you should force yourself to read whatever it publishes with a critical eye. We humans find comfort in charts and stories that corroborate what we already believe and react negatively against those that refute it.
- We are prone to lying to ourselves.

Reasoning and rationalization .
Some points from the epilogue in Cairo's book.

- We are not rational thinking machines as we may sometimes think.
- We are prone to lying to ourselves, to all sorts of logical fallacies and mental biases.
- 'Forming beliefs is easy; changing them is hard work.' (from 'The Believing Brain' by psychologist Michael Shermer)
- example of cheating students (from 'Mistakes Were Made (but Not by Me)' by Carol Tavris and Elliot Aronson):
- start with two students, initially with mild opinions about cheating
- one day, during exam, both are tempted to cheat. one goes through with it, the other one not.
- the one who resisted will likely become more self-righteous about not cheating
- the other may start to say cheating is not that bad, or it was justified in this case, because the exam was unfair, everyone does it, etc. (I'm a good person, so it's impossible that cheating is really that bad!)
- rationalization is the habit of 'defending' views/beliefs we already hold, even against new evidence. we are very prone to it.

8 Graphs and meshes

8.1 Graphs (with given embedding)

Definition.

- vertex list (with positions)
- edge list

Basic degrees of freedom / signals for visualization.

- vertices: color, size, marker/shape
- edges: color, width, linestyle, arrow tips, orientation, shrinking
- annotations

Example: TikZ.

- language for generating vector graphics within LaTeX
- Takes some getting used to, but results usually look very slick. A lot of built-in functions for common use cases, you just have to know about them.

Example: flows.

- flow is signed signal on edges, sign is related to edge orientation $\Rightarrow$ sign can be visualized as arrow
- alternatively: represent sign as hue on large graphs
- flow induces divergence on vertices, adjacency/incidence matrix

Example: stress and tension.

- stress is signed signal on (undirected) edges, visualize by signed color map

8.2 Meshes

Definition.

- vertices, triangles
- extract edges, orientation and adjacency

Plotting functions (with values on vertices).

- on pure pointcloud: can only represent function on the vertices
- with additional mesh structure: can assume that function is piecewise affine on each triangle (make sure this assumption is appropriate!)
- encode function via 3d and/or color shading
- visualization becomes much smoother

3d Models.

- with this, in principle, can visualize general 3d objects
- rendering options are immense: textures, shading, lighting,...
- sophisticated rendering can make a big difference in perception of quality
- relevant questions:
  - is the 3d structure perceivable in the chosen medium? (print / video / interactive)
  - are applied rendering effects necessary / do they improve the spatial perception / add information?

Boundaries, a bit of differential geometry / topology.

- mesh structure allows assigning 1d boundaries to 2d regions; and 0d boundaries to 1d paths
- cycle paths have no boundary. on a mesh that represents a square: each cycle is the boundary of some region
- this is not true on general meshes; usually when they have 'holes'
- this leads to the notion of homotopy classes: a homology class of cycles are cycles that 'only differ by a boundary'
- the 'trivial' class are those cycles that are boundaries; on a torus there are two more classes

8.3 Graph embedding

Description and motivation of problem.

- in previous graph examples: vertex positions were already known / given
- often we obtain large graphs from data, without explicit vertex locations (e.g. social networks, biochemical data,...)
  - for simplicity: undirected graphs
  - edge weights can represent similarity (low weight: little similarity, no edge: 0 similarity)
  - or distance (low distance: vertices are close, no edge: $+\infty$); mathematically: distance is symmetric and satisfies triangle inequality
  - if no "real" distance is available, just use similarity as proxy; common convention: similarity $\sim \exp(-\text{dist}/r)$ where r is a length scale parameter which must be tuned
- these can be represented computationally or mathematically without any problem as matrices (distance, adjacency, weighted adjacency),
  but difficult for humans to assess structure visually
- we want: locations for vertices, such that we can draw the graph; good locations characterized by:

    - distance of vertex locations approximates graph distance (or represents graph similarity)
    - vertices fill drawing region
    - few edge crossings
    - symmetries / 'topology' of graph are well represented
- for a given matrix, how do we find such vertex positions?

Layout methods.

- finding a satisfying graph layout is in general a hard problem. so perfect algorithms can be expected. some solution strategies are:
- fixed layout: nodes are placed on a fixed position set (e.g. circles, multiple straight lines,...)
- structured fixed layout. example: tree graphs can be arranged on concentric circles of increasing radius
- explicit minimization: minimize energy functional that is based on attractive and repulsive potentials between nodes; or tries to achieve prescribed edge lengths
- spectral methods / implicit minimization: node coordinates are given by coordinates of eigenvectors of some matrix defined over the graph, such as graph Laplacian
- PDF: overview of options in Mathematica's GraphPlot function

Observation: not all finite metric spaces are embeddable into $\mathbb{R}^d$, even for arbitrary dimension d.

- see:
  https://mathoverflow.net/questions/12394/representability-of-finite-metric-spaces
- simple example: "square graph": shortest paths in graph must lie on straight lines in Euclidean space, thus cannot embedd two different shortest paths between the same two vertices properly

Tuttle embeddings and Schlegel diagrams.

- Polyhedral graphs (https://en.wikipedia.org/wiki/Polyhedral_graph) can be embedded into $\mathbb{R}^d$ in a systematic way, by fixing a 'frame' and minimizing a spring energy these layouts are called Tuttle embeddings
- similar, related technique: Schlegel diagram (https://en.wikipedia.org/wiki/Schlegel_diagram) where one projects a polytope into the plane
- only applicable for very specific graphs, not so useful in general

Force based embeddings.

- Coulomb potentials and similar: keep connected vertices close; otherwise far
  - example: Fruchterman, Reingold: Graph Drawing by Force-Directed Placement, Software – Practice & Experience 21(11), 1991, doi:10.1002/spe.4380211102
  - if two vertices are connected by an edge, attractive force that scales as: $f_a(d) = d^2$
  - any two vertices repel each other with force that scales as: $f_r(d) = -1/d$
  - computing pairwise forces between any two vertices is expensive; use coarse-to-fine approximation schemes
  - the energy is highly non-convex; can only expect local minimizers; need techniques such as coarse-to-fine methods and simulated annealing to improve results
- spring energy to approximate desired edge lengths:
  - ideal lengths of (some) edges is prescribed: $d_{i,j}$; minimize deviations from these lengths:
  $$E(x) = \sum_{(i,j) \in E} \left(\|x_i - x_j\| - d_{i,j} \right)^2$$
  - Taylor expansion of this energy around the minimum was used for bridge model in previous session

Multidimensional scaling (MDS): Spectral embedding via Gram matrix.

- Gram matrix: if metric were generated from Euclidean embedding, pairwise inner products can be recovered by clever combination of squared distances
- eigendecomposition of Gram matrix is closely related to eigendecomposition of covariance matrix of samples
- if Euclidean embedding exists, (scaled) eigenvectors of Gram matrix provide such an embedding
- can control accuracy by picking only largest eigenvalues
- otherwise, if no Euclidean embedding exists, Gram matrix will be indefinite. can still use largest positive eigenvalues to obtain approximate embedding

Spectral embedding via graph Laplacian.

- input: weighted undirected graph
- mathematical background: graph Laplacian, eigenfunctions
- intution: vibrating string or plate
- eigenvectors provide embedding coordinates
- for distance matrices: tuning of weight / length scale parameter

Example gallery: Gephi.

- Gephi is a program for finding graph layouts and visualizing graphs
- twitter network for digital humanities conference
- http://www.martingrandjean.ch/dataviz-digital-humanities-twitter-dh2014/
- PDF: general introduction to Gephi: http://www.martingrandjean.ch/gephi-introduction/
- PDF: refugee map (these are not embedding problem! but still instructive lesson on visualization)
  - http://www.martingrandjean.ch/data-visualization-map-refugees/

Example gallery: graph viz.

- https://graphviz.org/gallery/
- software module dependencies
- network structure
- PDF: data relation model
- PDF: neural networks
- interactive web demo: http://www.ryandesign.com/canviz/
- http://yifanhu.net/GALLERY/GRAPHS/index.html

Example gallery / convenient implementation: sklearn.

- https://scikit-learn.org/stable/modules/manifold.html
- various classical manifold learning methods are implemented in sklearn and easy to use from there, such as isomap, spectral embedding, multi-dimensional scaling, t-SNE

Example gallery: umap.

- https://umap-learn.readthedocs.io/en/latest/basic_usage.html
- relatively recently proposed method, seems to work quite well on large variety of data
- intuitive mechanism: preserve local connectivity of some form of symmetrized kNN-graph, initialize with spectral embedding, then run a stochastic gradient descent on the embedding energy which takes the form of a force-based algorithm

8.4 Relational graphs

Qualitatively different types of graphs.

- so far: vertices were 'simple objects' such as points in space with little additional attributes or internal structures
  - likewise: edges were 'simple', few attributes such as length, weight, stress
  - but graphs were potentially very large
- now: graphs with more substructure. Vertices have more attributes, or even 'identity'.
  - edges describe more detailed interaction / relation.
- examples for vertices: persons; agents; classes; theorems; elementary particles
  - for edges: (blood) relation; interaction (calls / blocks / reserves / buys); derived from / calls method / holds instance of; proof relies on; decays into / collides with / has bond with
- to encode this information we need a bit more 'decoration':
  - vertices: text, box styles
  - edges: text, arrows, line styles

Example: UML class diagram.

- https://en.wikipedia.org/wiki/Class_diagram
- internal structure of class:
  - attributes, methods
  - visibility (public, private, protected), static or instance member
- relations between classes:
  - association
  - aggregation
  - inheritance
- related: component diagram

Example: UML activity diagram.

- https://en.wikipedia.org/wiki/Activity_diagram
- conditionals
- distinguish flow of control and objects
- input and output

Example: flow charts, UML sequence and state diagrams.

- https://en.wikipedia.org/wiki/Flowchart
  - simple syntax for conditionals, events, ...
- https://en.wikipedia.org/wiki/Sequence_diagram
  - represent interactions between various agents with temporal ordering
- https://en.wikipedia.org/wiki/UML_state_machine
  - describe system with discrete states as vertices
  - transitions and actions
  - finite-state machine
  - visual abstraction: not all states will be represented as individual box, use 'state variables'

Example: graphical model.

- data analysis / machine learning: parametrize high-dimensional probability distributions
- for tractability: exploit / impose sparse dependency structure, based on system knowledge (can also attempt to infer this from the data)
- this structure can be encoded / represented by a graph
- typical examples: undirected graph, directed acyclic graph

Example: Theorem relation diagram.

- for each formulation: short handle, full name, preview formula
- position matters
- [X]: literature references, (X): contributions of the paper
- uni- and bi-directional arrows
- dashes for partial relations

Example: Feynman diagram.

- in pop culture: pictorial representation of elementary particle interactions
- for experts: compact representation of a long mathematical expression
- in quantum field theory one does perturbative expansion of full theory (like a Taylor expansion)
  - need to keep track of all relevant terms and their order (=power in the Taylor expansion)
  - this is much easier to do in the visual representation

9 Animations and interactive visualization

9.1 Motivation

- seen in previous chapters: compressing high-dimensional data into graphics is considerable challenge
- time offers an additional dimension
- one additional dimension can often be accessed by a pre-defined animation
- even more dimensions accessible via interaction

9.2 Relevant effects in human visual system

Recall: Change blindness.

- example: altimeter
- example: housing filter

Recall: Attentional blink.

- task: identify targets in fast sequence of images
- when pre-attentive processing has identified target (and thus 'activates' conscious processing), the pre-attentive processing is disabled for a short interval

Saccadic blindness.

- recall: eye movement composed of fixations ($\approx$ 200ms) and fast saccades between them ($\approx$ 20ms)
- during saccade we are blind, if we quickly move gaze from one part of room to another, we do not see scene continuously moving by
- intuitively clear: our visual designs should reduce number of saccades

Rapid serial visual representation [Spence].

- experimental results
  - mode 1: subjects first look at given reference image, then see rapid sequence of images (100ms per image), need to determine whether reference image was part of sequence. success rate: 80%-90%
  - mode 2: no prior instruction; subjects are shown reference image right after sequence; still success rate of $\approx$ 90%.
  - but this drops to 10%-20% if one pauses for $\approx$ 5s after the sequence, before showing the reference image
  - mode 3: images are still only shown for 100ms, but blank pauses of 1.5s are introduced between images. success rate $\approx$ 90%. (unfortunately, unclear from Spence's text if this experiment was performed with or without pause after sequence, and if sequence contained fewer pictures)

- interpretation:
  - 100ms seems to be enough to capture gist of image
  - this processing must be pre-attentive
  - but not enough time for consolidation in short-term memory
- applications proposed by Spence
  - folder preview (show contents of file directory as quick succession of thumbnails)
  - film preview: 'gist' of a film can be conveyed with 20 carefully selected keyframes over 2s (Tse at al., Dynamic key-frame presentation techniques for augmented video browsing. ACM Proceedings of conference on advanced visual interfaces, 1998)
  - news website on phone display: rapid sequence of main stories
  - web browser back button: rapid sequence of screenshots
  - in my humble opinion: most of these sound terrible and stress-inducing
  - video streaming websites implement different types of 'video preview' schemes. typical choices: a special trailer, a 'typical scene', or a 'fast-forward'

Moving RSVP.

- different presentation modes:
  - collage
  - shot
  - float (with brief 'capture' of pictures near center)
- experimental comparison (Corsato at al., An eye tracking approach to image search activities using RSVP display techniques. ACM Proceedings of conference on advanced visual interfaces, 2008)
  - identify 40 out of 2000 images in the three presentation modes
  - compare eye movement ('gaze travel'), success rate
  - float mode is most efficient, why?
  - for humans task consists of pre-attentive identification of relevant images; then conscious confirmation
  - float mode allows easy identification at center, without gaze movement; during subsequent movement pictures are still visible long enough for confirmation; we may even have enough time to outlast an attentional blink

Visual momentum.

- Spence, Section 5.3.8; Woods: 'Visual Momentum: A Concept to Improve the Cognitive Coupling of Person and Computer', International Journal of Man-Machine Studies 1984
- goal of visualization is to help user build mental model of data
- discontinuous changes in display easily disrupt this model

- examples:
  - rotating a scatter plot
  - interpolating values in bar charts and similar
  - ‘appearing houses’ also fall in this category
- comment: implementing these things in practice can be extremely complex, may consume most of the coding time, need proper environment

Some implications for animations.

- When viewers cannot control playback (e.g. in talk), show animation in loop or at least several runs. Before understanding content of video, viwers need to decode it and learn its visual language.
- When animation is running, let viewers watch, do not compete for attention. In short clips it is often possible to explain what they are supposed to see before, or maybe after a first run (→ recall attentional blindness).
- Choose suitable playback speed. Keep in mind: viewers usually see video for the first time. Leave extra time when you explain things during the video.
- When an animation shows temporal evolution over time, do not immediately cut away after last time-step. Give viewers time to study the final state.

9.3 Control elements

Norman’s action cycle.

- Donald A. Norman: The Design of Everyday Things, 1988
- how to design common devices? door handles, coffee makers, bicycle brakes
- general ideas also apply to interactive graphics

Design principles.

- affordance, perceived and actual
  - what does a control element / device ‘offer’?
  - should be intuitive, extends to design in general
  - there are social conventions and expectations
  - typical examples: doors, shower knobs, lamps
- metaphors in graphical user interface: exploit familiarity of users with other concepts for fast adaptation
  - windows mimic overlapping sheets of paper that can be rearranged
  - manipulation of objects on screen: drag and drop, scrolling by ‘pushing’
  - input elements on windows are design similar to ‘analog’ forms (by now: probably the converse is true)

  - filesystem with folders mimics large filing cabinet with various drawers, separators, and dossiers

- feedback, examples:
  - ‘click’ animation (or sound?)
  - reliable progress bars
  - traffic light without feedback

- constraints
  - example: ‘swipe animation’ at end of picture gallery on phone
  - grey out buttons; not make them entirely invisible; make sure ‘inactive’ look is not confusing

- consistency
  - minimize user time for learning how to use the program
  - objects that look the same should behave the same
  - there are many conventions already on what design elements look like

- what are typical user goals?
  - make sure these can be obtained efficiently
  - example: Deutsche Bahn app, looking for an alternative connection

9.4 Navigating, browsing and filtering complex data

Standard navigation operations.

- open, close, and move windows
- drag and drop
- sort and rearrange
- search and filter
- scroll, pan, zoom
  - on ‘map’ move in all four directions
  - for text pdf: vertical scrolling more useful than horizontal
- for all of these there are by now conventions on design and control interfaces
- these can usually be combined in natural ways

Context.

- thumbnails of neighouring pages or images
- document outline / section structure in PDF reader, hyperlinks within document

Level of detail.

- travel navigation: before digital navigation assistance, printed maps faced a difficult trade-off:
  - high resolution: impossible to print for large area, would constantly need to turn pages, difficult to get large scale impression of travel
  - low resolution: not enough detail within cities, and at critical junctions
  - digital navigation can store all high resolution information, automatically track location on map, and choose zoom level based on appropriate level of detail
- engineering drawing: distinguish between full details for all parts, and reduced version that explains how to replace a certain part

Hierarchies.

- example: hyperbolic graph browser. Better than 'regular zoom' since it still provides context.
- file system explorer
  - Directories and filenames themselves are a means of visualization of data. With proper use we can effortlessly remember (or at least retrieve) the locations of thousands of files.
  - list vs items vs thumbnails

Moving within a program / an app.

- When using software via a graphical interface, users need to move between different menus and query interfaces.
- Multiple instances of same interface might be required at same time. They can be organized in tabs or floating windows.
- relatively easy on computer: large screen, complex input devices
- extremely challenging on smart phones:
  - much smaller screen, touch screen much coarser than mouse, no keyboard, only 'back button'
  - navigation often only in 'tree graph', cumbersome for complex applications
  - recall example: Deutsche Bahn navigator app
- on laptops: mouse less precise

9.5 Sensitivity and dynamic exploration

Extend mental model of users.

- user should quickly be able to extend their mental model to include expectations to reactions on changing parameters
- make experience as 'smooth' as possible
- exploration should be simplified, in particular in high dimensions
- so ideally: from current point can glimpse a little bit into all directions

Sensitivity data.

- how did parameter change affect change of outcome?
- anticipate if a small parameter change can lead to a desired outcome? show 'near misses'

Dynamic exploration.

- short-cut in action cycle, get immediate feedback

9.6 Examples

Kirk (see book references).

- simple selection and filter operations
- temporal evolution of data
- interactive quiz with non-trivial input (intra-German border)

Matplotlib.

- animation
- simple dynamic interaction
- feels old-fashioned, essentially works like old-school GUI implementation

PlotLy.

- another plotting library with support for many languages, including python
- python tutorial available at: `https://plotly.com/python/`
- very interactive
- core library implemented in JavaScript, hence simple export of dynamic figures as HTML

Manim.

- script-based creation of animations in python
- ideal for algorithmic creation of data-based animations
- easy to keep 'visual momentum'
- 'community' fork available at https://www.manim.community/
- example video from our group: https://ot.cs.uni-goettingen.de/video_histology2022.html (partially done with manim)

Tensorflow Playground.

- https://playground.tensorflow.org/
- interactive arrangement and training of a neural network

9.7 Visualization for websites

Context.

- for creating a website one has to solve visualization problems on (at least) three layers
- layer 1: global structure
  - how is the whole content broken down into individual documents? what is the hierarchical structure of documents?
  - how is this structure presented to the user? how can users navigate it? (only moving in the tree, use search function, many internal hyperlinks)
- layer 2: design of a single page/document
  - how are navigation, text and embedded media combined?
- layer 3: media within a single page
  - embedded media can be (interactive) figures, videos, external content

Some comments on navigation.

- menu structure: tall & thin vs flat & fat
  - how complex is each choice? how many choices necessary? when looking for a leaf node, are the names of the parent nodes clear?
  - categories can be clarified by keywords: 'look ahead information' ('residue', Furnas: Effective view navigation, ACM proceedings of CHI'97, 1997)
- breadcrumbs: visualization of past trajectory of user
  - often supplied by 'back' button in browser, but in interactive sessions often in conflict with website functionality (re-sending of requests?)

- ‘see also’ suggestions and cross-references
- how often do you use page-internal navigation vs google?
- increasingly popular: virtually no ‘static predefined hierarchical navigation tools’; content selection solely based on search query interface and machine learning (‘users also liked…’)

9.8 Examples: Plotting libraries for the web

- there is a plethora of java script charting libraries: https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_libraries

echarts

- https://echarts.apache.org/en/index.html
- license: BSD-3
- very easy to get started

d3.js

- https://d3js.org/
- license: BSD-3
- nice example gallery: ‘open data show room’ of Uni Bern, Institut für Wirtschaftsinformatik (German only, unfortunately), https://opendata.iwi.unibe.ch/

PlotLy

- see previous section

9.9 Examples: Websites

Dark patterns.

- In a perfect world the design of websites would be optimizes for user experience. In practice they are often designed to encourage the best outcome for the company that runs the website. When website (or generally application) design intends to bias the user against acting in their best interest this is referred to as a ‘dark pattern’.
- example: accepting the default privacy settings (which are very open) is much easier (and encouraged by suggestive link coloring) than choosing a more restrictive setting
- example: pressuring users into booking overpriced accommodation or transportation tickets buy creating the impression that availability is very limited; this is often amplified by dynamic methods, such as pop-up warnings
- example: it is sometimes very easy to start a subscription for a service with very few steps that are easy to follow, whereas cancellation may be only possible via much more subtle paths, including artificial obstacles (possibly not being possible online at all)

Wikipedia.

- Wikipedia allows different ways for navigating its vast network of articles.
- follow links on individual expressions within an article (e.g. to look up missing relevant background knowledge)
- use boxes with context-specific meta data, e.g. predecessor in a sequence (presidents), other members of same category (movie franchise), other layers in a hierarchical structure (administrative units of a country), ...
- preview for links on hover, allows mini-look-up without interrupting reading flow
- entry point to wikipedia is often a search engine (possibly wikipedia itself) or a theme portal

Documentation.

- providing documentation is vital for complex software projects
- usually divided into:
  - overview ('what is this about? what can I do with it? do I need this?')
  - installation instructions ('how can I get this?')
  - tutorials ('how can I quickly solve this one problem that I came for?')
  - API reference ('what is the full functionality of a given component')

Github.

- what information is presented when browsing a repository?
  - file structure
  - contributors
  - activity (list of commits, commit statistics) and project status
  - introduction text
  - next steps: download files, browse documentation, check project history
- role of text formatting
  - documentation is parsed markdown
  - inline code examples with syntax highlighting
- in total very complex, many underlying visual conventions $\Rightarrow$ gets easier over time