Visualization

Bernhard Schmitzer

Uni Göttingen, summer term 2022, May 14, 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

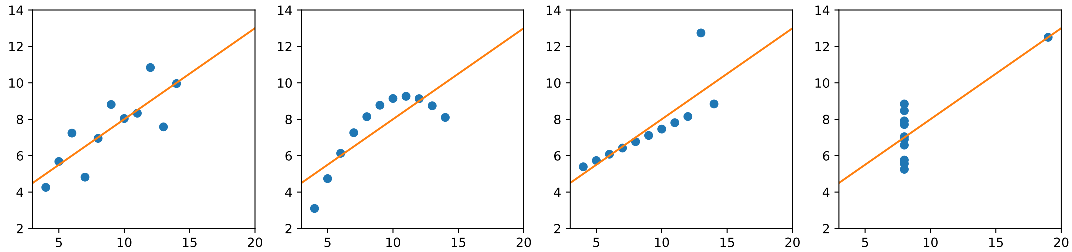
1.3.1 Tables versus plots

- 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

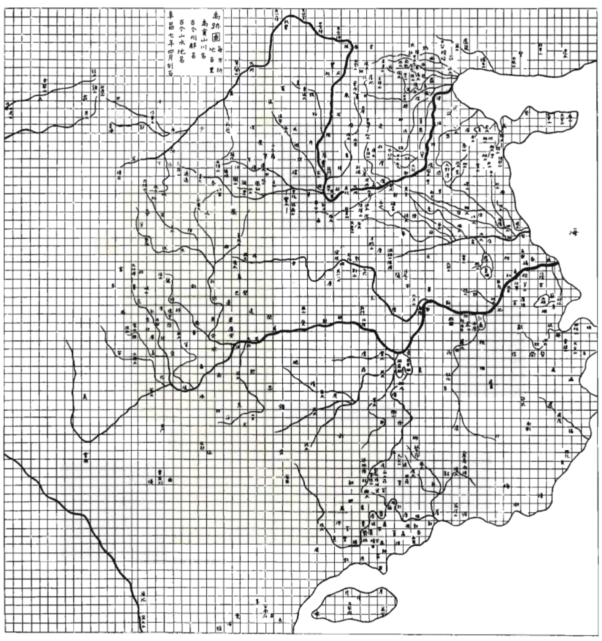
- hard to interpret as numbers in table, try basic statistical analysis
- all x_i and y_i sequences have same mean and variance
- sequences of pairs (x_1, y_1), (x_1, y_2), (x_1, y_3) and (x_2, y_4) all yield essentially the same linear regression:
  - same slope, intercept, correlation coefficient,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



1.3.2 Historical examples

Chinese cartography

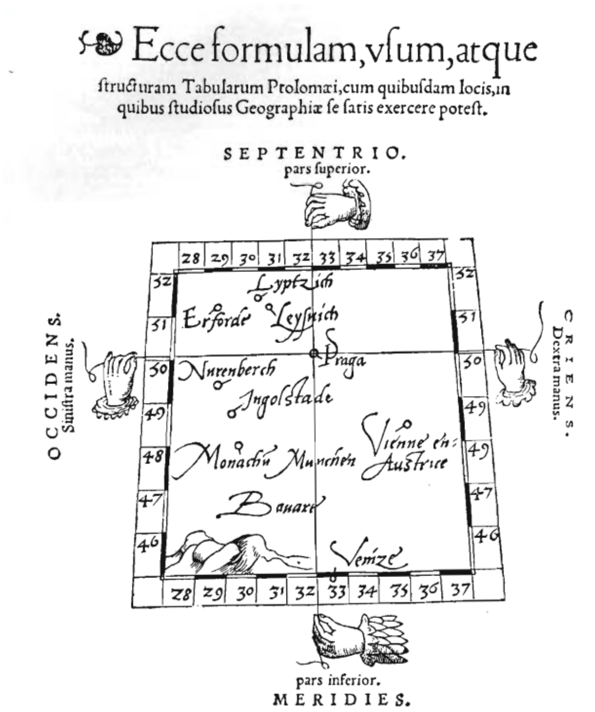
- taken from [Tufte: The visual display of quantitative information]
- approx 1100 AD



European cartography

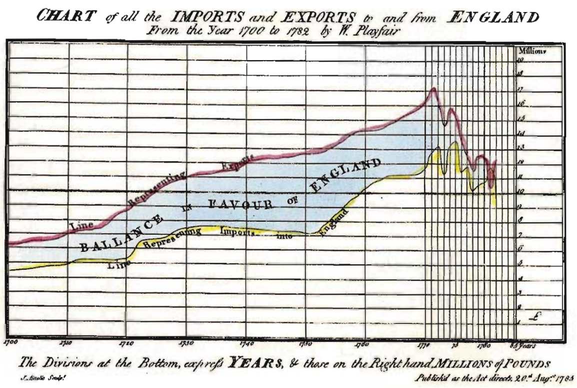
- taken from [Tufte: The visual display of quantitative information]

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



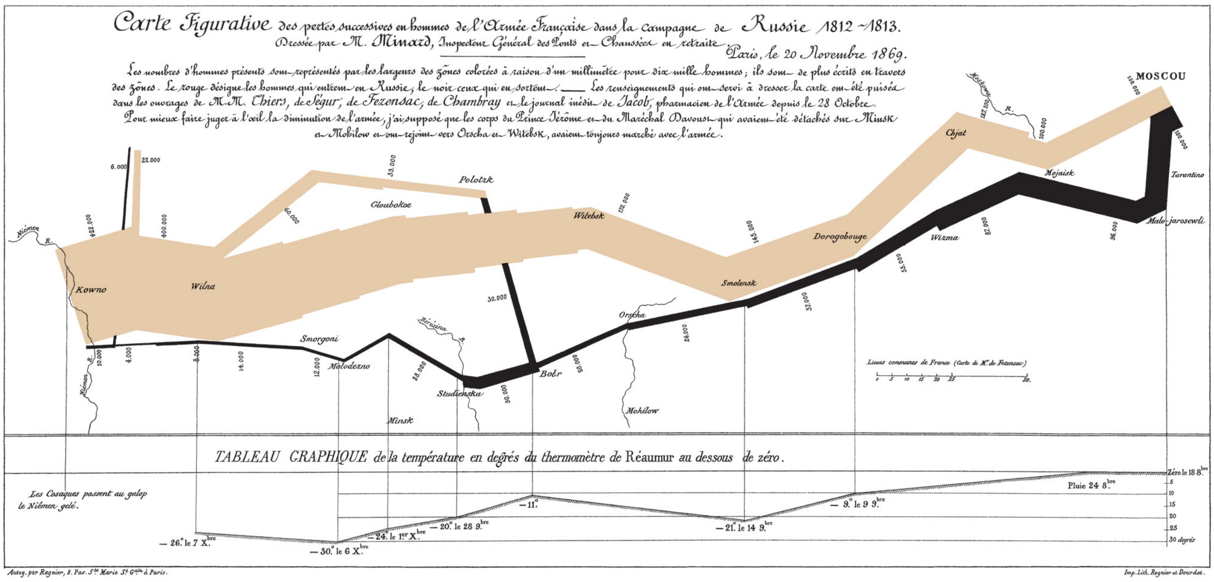
Playfair

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



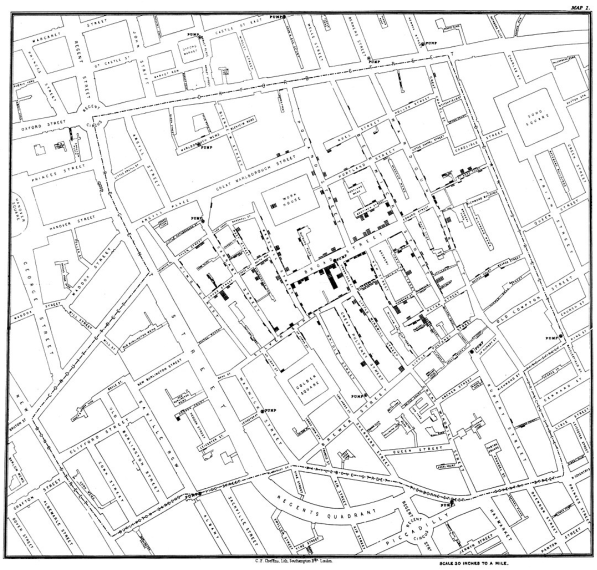
Napoleon's march to Moscow.

- 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.

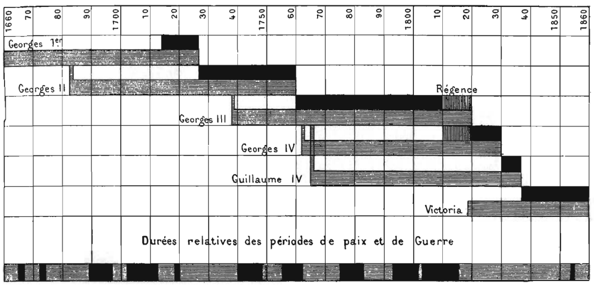
- taken from [Spence: Information Visualization]
- figure in public domain, available at https://en.wikipedia.org/wiki/File:Snow-cholera-map-1.jpg
- John Snow, 1854



Regency chart

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

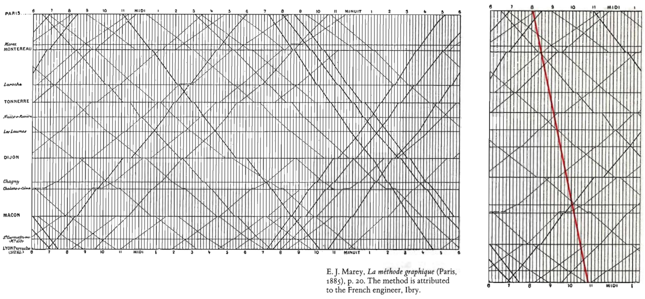
- Note: George II was the founder of Göttingen University, Wilhelmsplatz is named after William IV.



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

Train timetable.

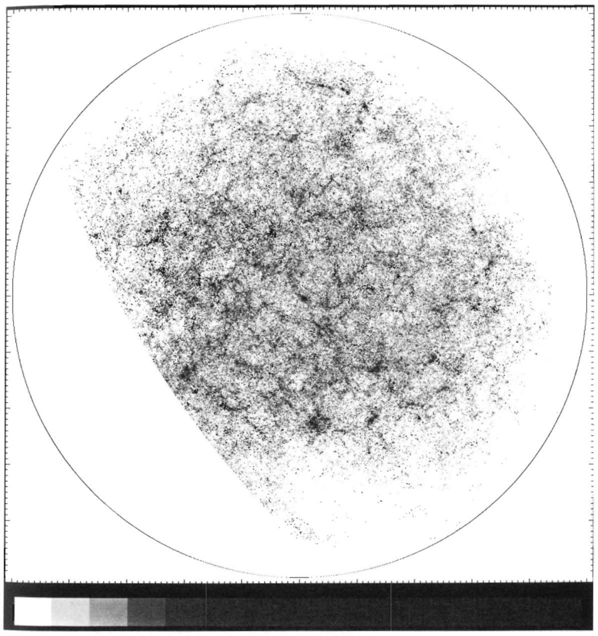
- taken from [Tufte: The visual display of quantitative information]
- E. J. Marey: La méthode graphique, 1885



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.

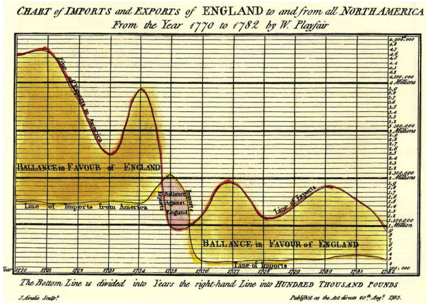
- of course it was already well known that graphics may distort the data, by ignorance or by intention
- Tufte even had the impression that graphics had a general reputation for ‘lying to’ or ‘fooling’ viewers, see [The Visual Display of Quantitative Information, Chapter 2]:
  - ‘For many people the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

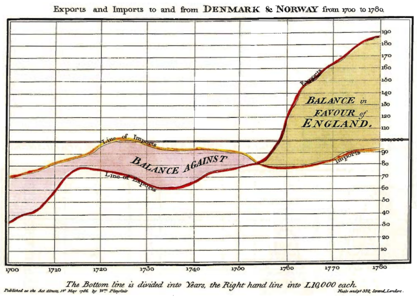
– 'Much of twentieth-century thinking about statistical graphics has been preoccupied with the question of how some amateurish chart might fool a naive viewer.'

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

2.2 Data-ink

Example: Evolution of charts by Playfair.



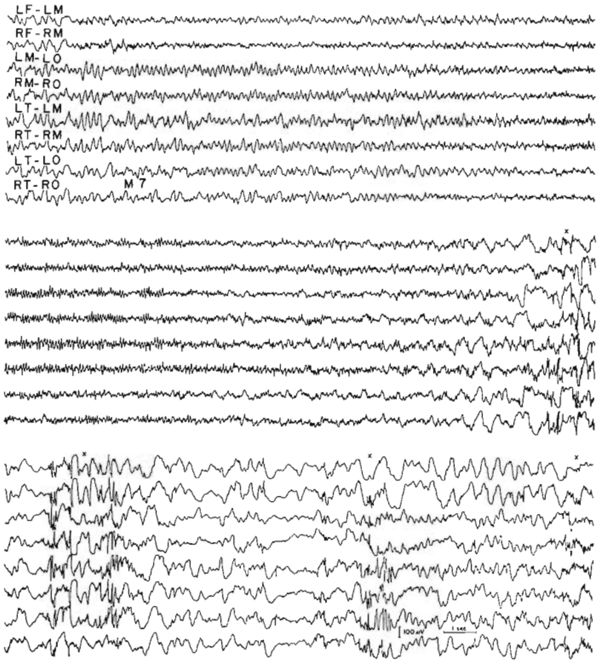


- first example: 1785, early pages of 'The Commercial and Political Atlas'
- second example: created on year later, already much more mature, removed much of the 'background'
- Tufte formulates a fundamental principle: Above all else show the data.

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 $= \dfrac{\text{data-ink}}{\text{total ink}}$

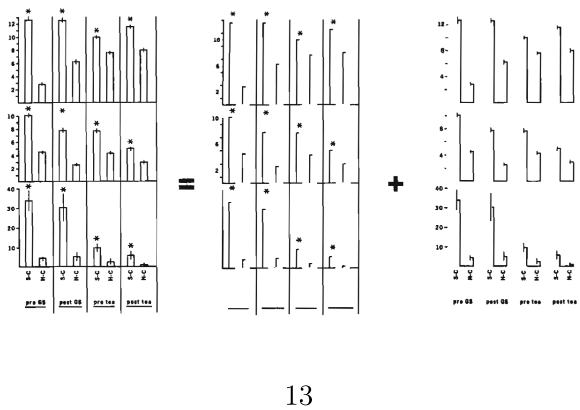
Example: electroencephalogram.



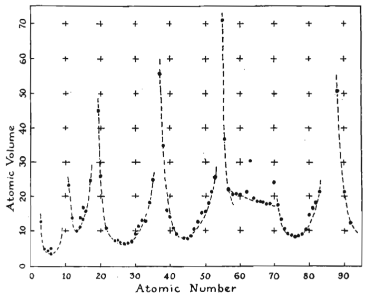
- extreme example: almost exclusively data-ink, but can only be read by specialists

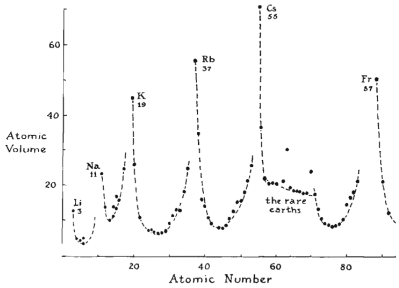
Example: sour taste.

- 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

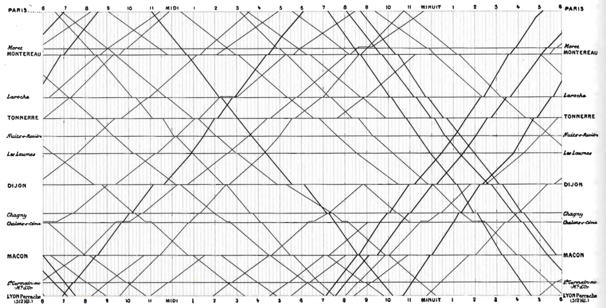
More principles.

- 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.

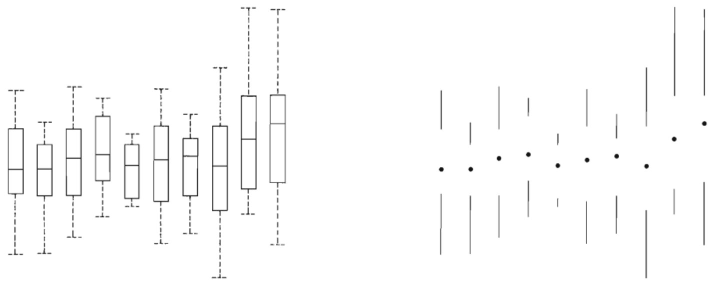


Ducks.

- 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

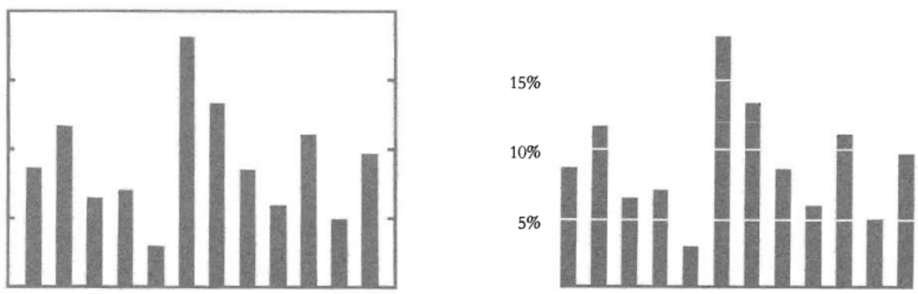
2.4 New graphical design suggestions

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.

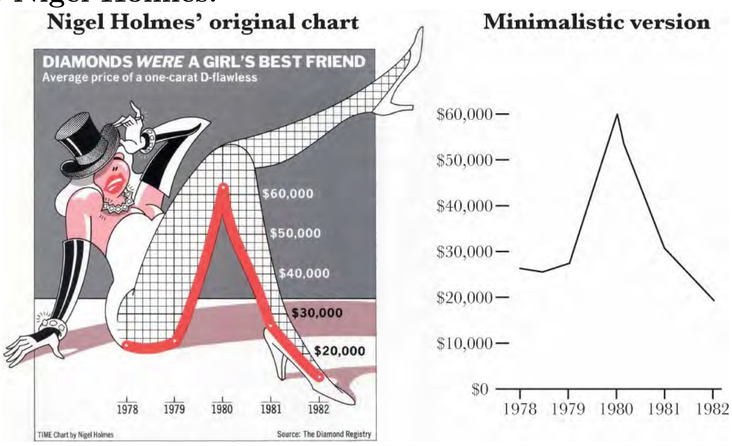
- what is the type of the x-axis?
  - a **time series** or other continuous, one-dimensional variable? then show scatter plot, possibly with lines
  - a **nominal type**? (city, country, ...) then data should be sorted, or at least grouped (e.g. by continent?)
  - does the chart represent a **histogram** 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)

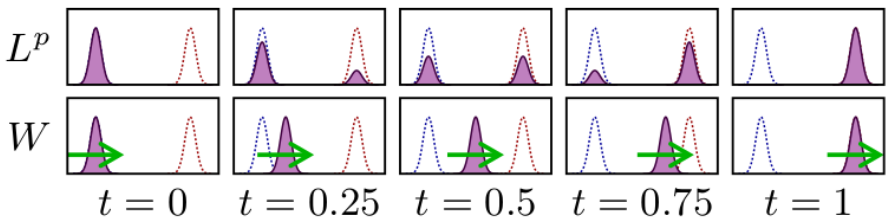
2.6 Data density and small multiples

- 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)

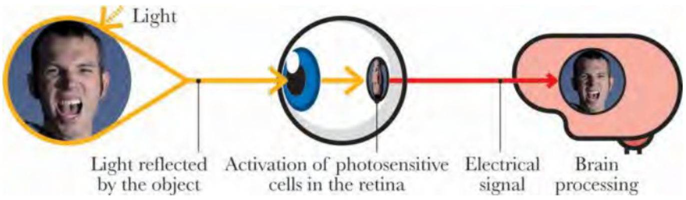


3 Human visual system and cognition

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

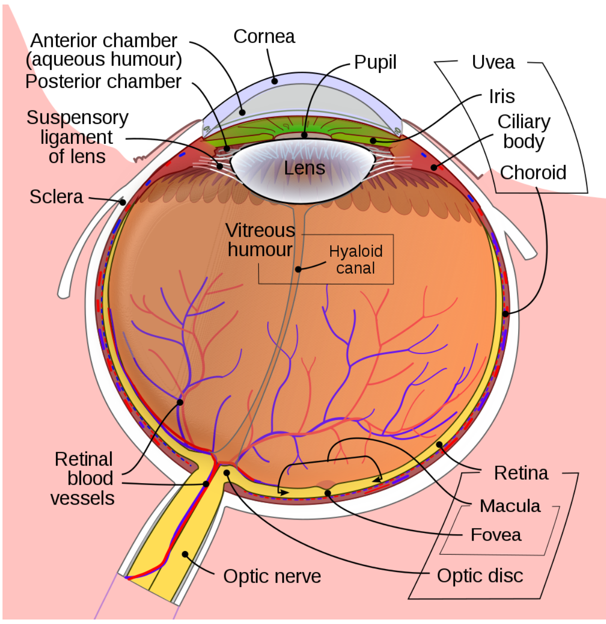
An unhelpful analogy.



- 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.

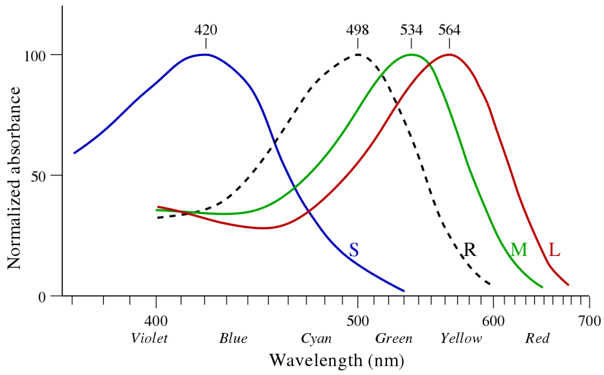


https://en.wikipedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg

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

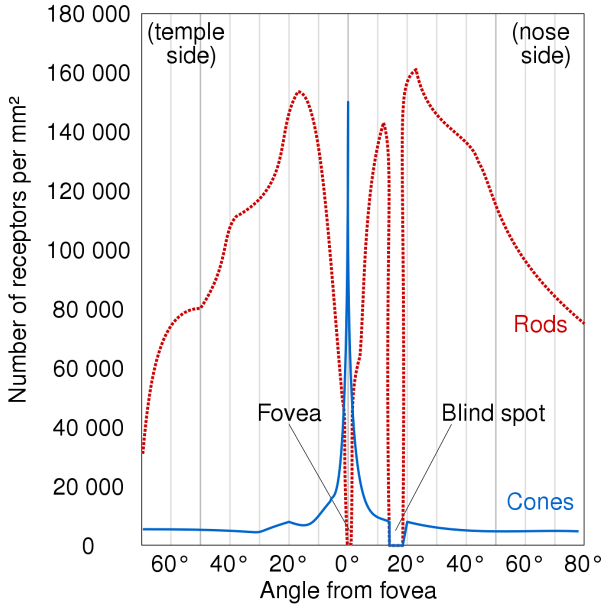
photoreceptors.

- 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



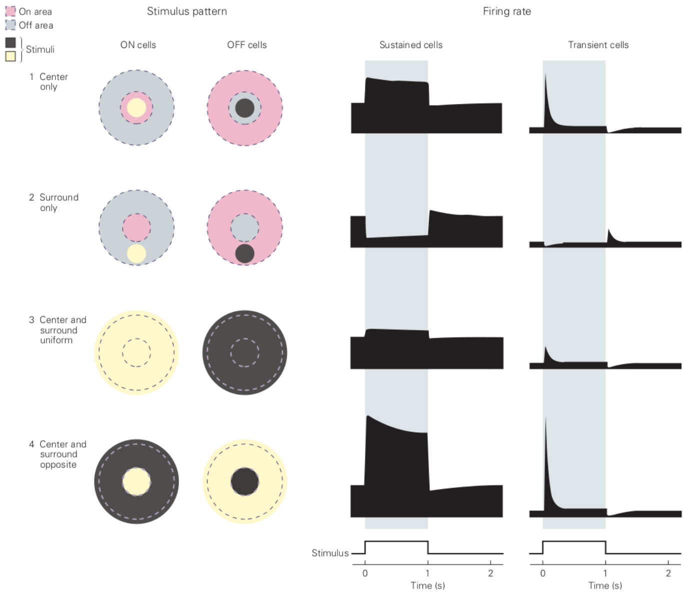
https://en.wikipedia.org/wiki/File:Human_photoreceptor_distribution.svg

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

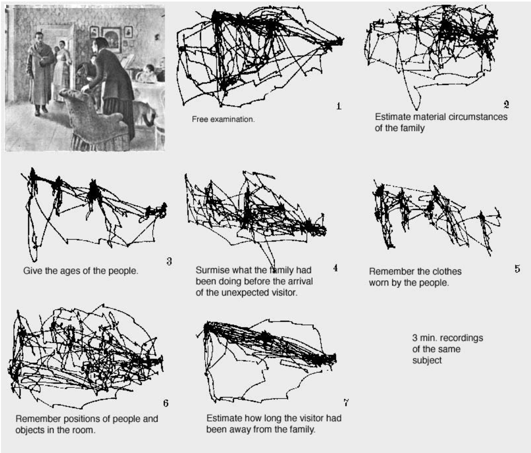


from Kandel, response of ganglion cells

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



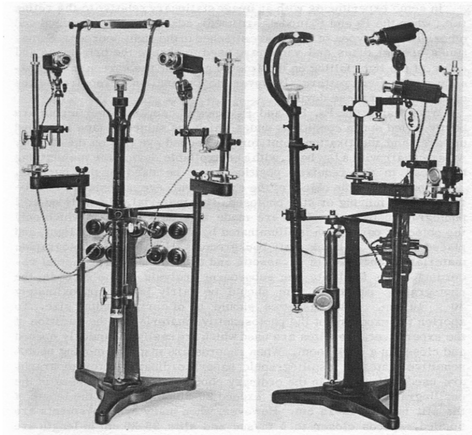
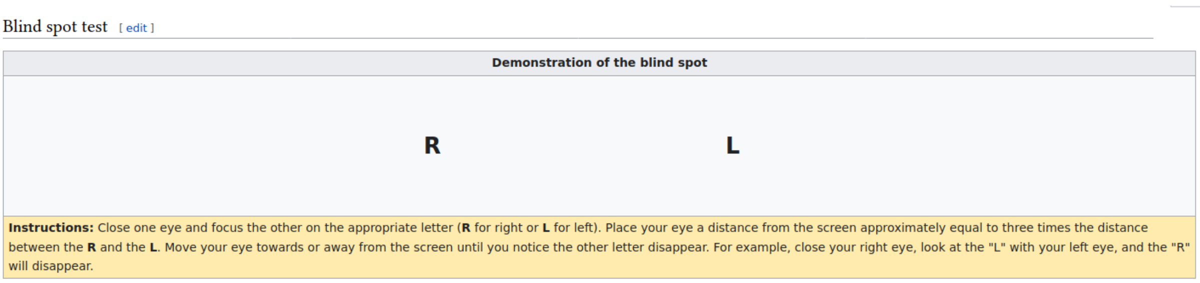


Fig. 21. The apparatus used in recording eye movements.

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


Blind spot test [edit]

Demonstration of the blind spot

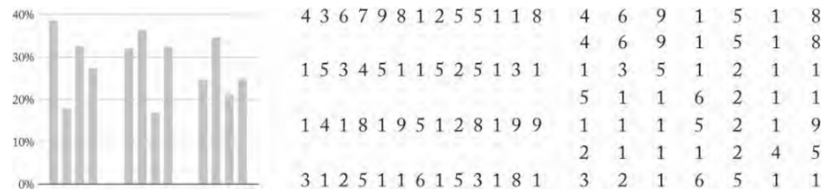
R L

Instructions: Close one eye and focus the other on the appropriate letter (**R** for right or **L** for left). Place your eye a distance from the screen approximately equal to three times the distance between the **R** and the **L**. Move your eye towards or away from the screen until you notice the other letter disappear. For example, close your right eye, look at the "L" with your left eye, and the "R" will disappear.

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

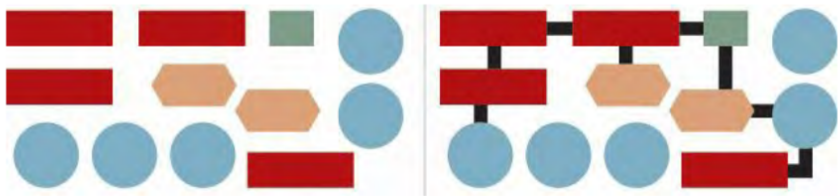
Gestalt grouping principles.

- 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

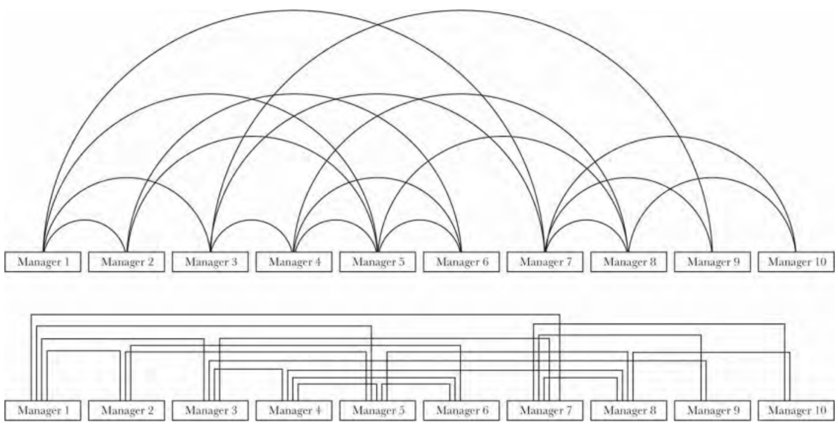


- similarity

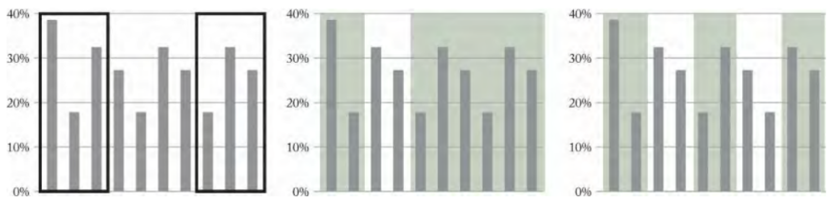
- connectedness



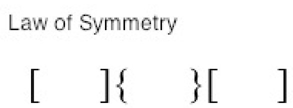
- continuity



- closure



- symmetry



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

3.2 Preattentive processing

famous experiments.

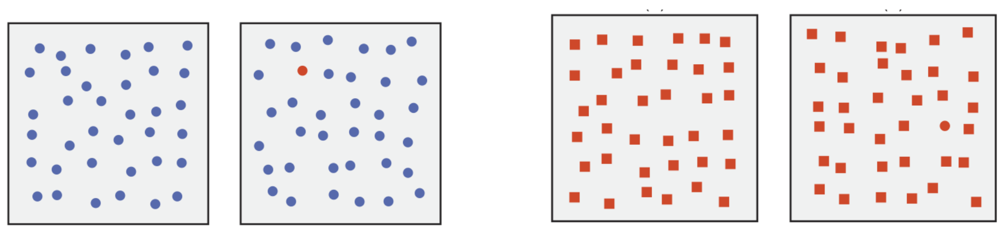
- Treisman and Gormican, 1990: show simple image only for 50ms, viewer can still detect the gist
- Kundel and Nodine, 1975: trained radiologists were shown lung X-rays for 200ms. Could detect anomalies in 70% of cases (97% under unlimited viewing).
- we extract a lot of information from an image in the 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.

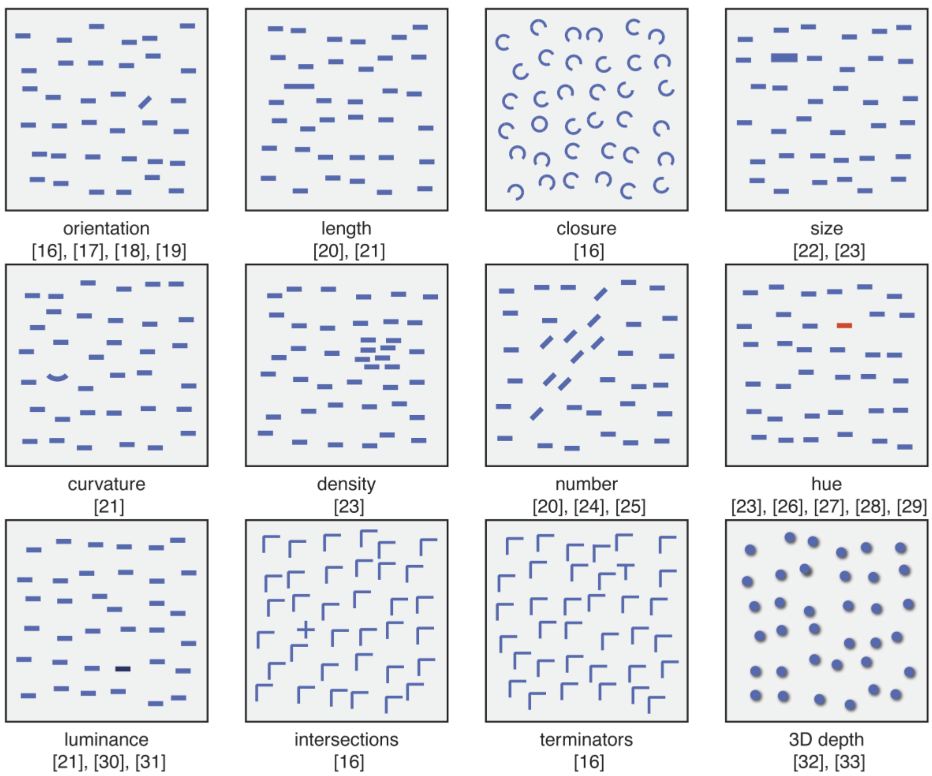
- example: red circle among blue circles (feature: hue)
- example: red circle among red squares (feature: shape)
- identification happens automatically and without exhausting the viewer



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

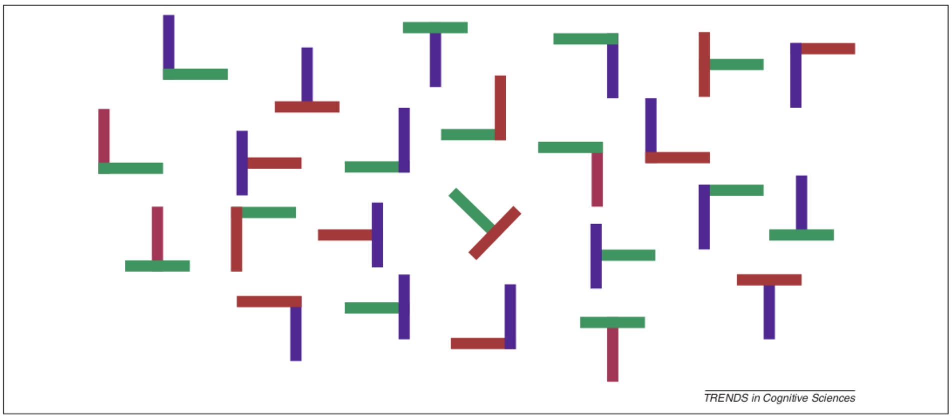
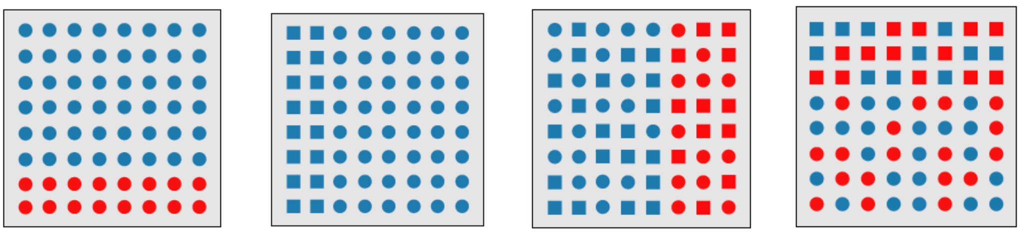


Figure 1. Find the four purple-and-green Ts. Even though it is easy to identify such targets, this task requires search.

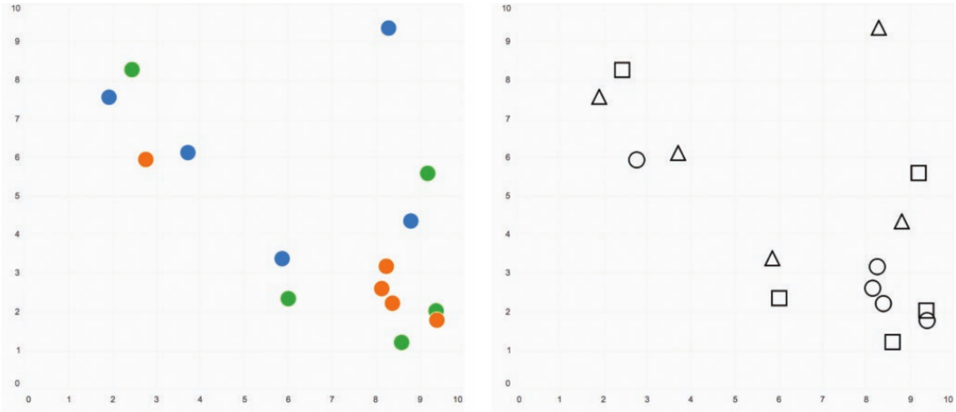
conjunction search is much harder [Wolfe]

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

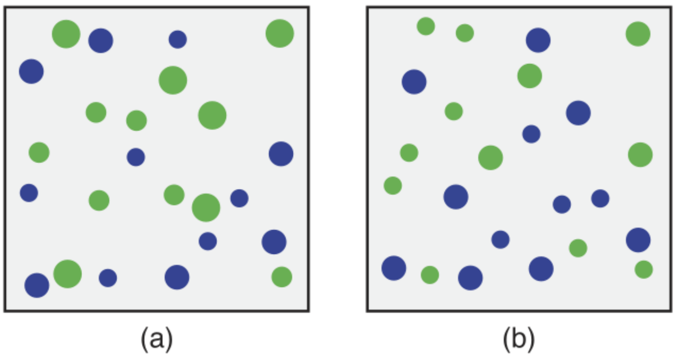


Fig. 9. Estimating average size: (a) average size of green elements is larger; (b) average size of blue elements is larger [66].

  - ‘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

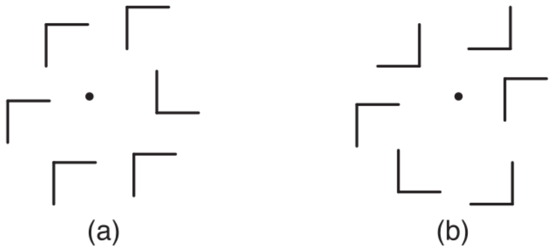


Fig. 5. N-N similarity affecting search efficiency for an L-shaped target: (a) high N-N (nontarget-nontarget) similarity allows easy detection of the target L; (b) low N-N similarity increases the difficulty of detecting the target L [55].

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.

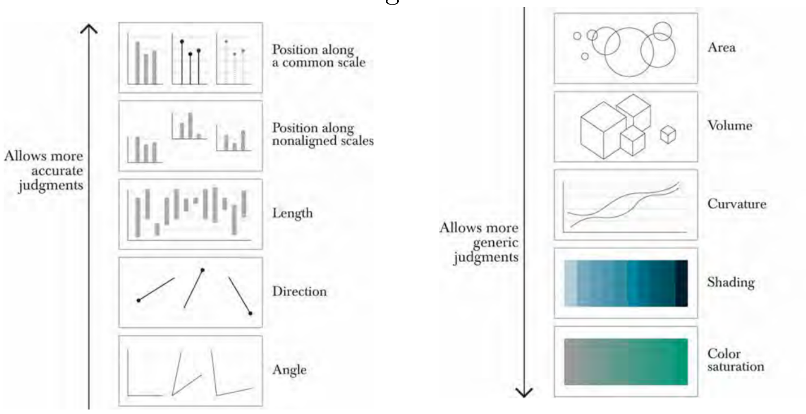
- if a visualization looks familiar, we can very quickly get a good orientation (see ensemble coding)
- so re-use visualization techniques that the reader already knows, e.g. small multiple, smooth animations

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.

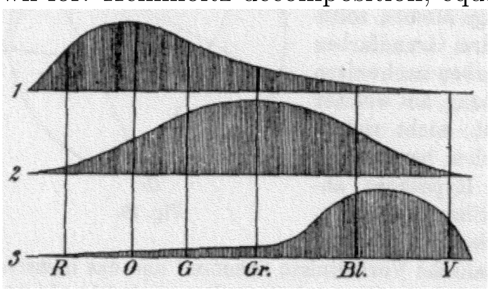
- since internal representation of figure in brain seems to drop many 'unnecessary details' (see change blindness), strong abstraction must happen in visual understanding of scene
- simplify work by using abstracted images.

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, ...

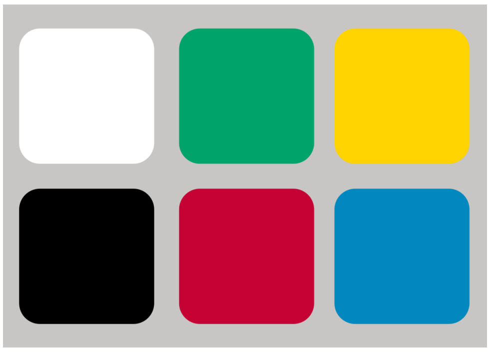


https://en.wikipedia.org/wiki/File:YoungHelm.jpg

- 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

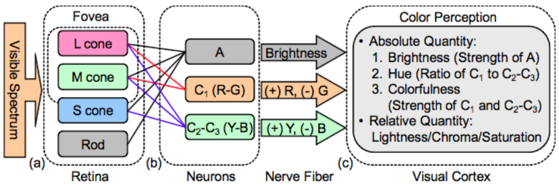
Opponent process theory.

- 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

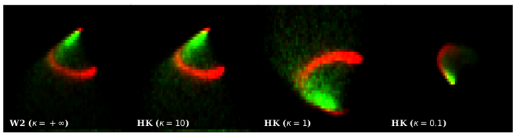


FIG. 5.8. *Distribution of 10k W (red) and QCD (green) jets in the tangent space with respect to the first two dominant PCA modes for* W_2 *and* HK *distances with* $\kappa = 10, 1, 0.1$ *(from left to right).*

  - 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

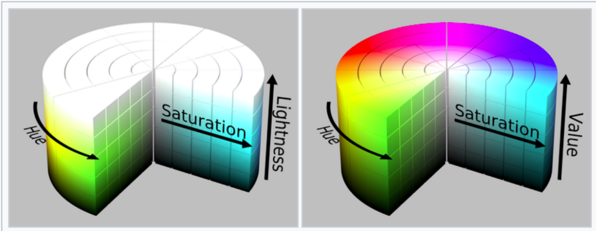


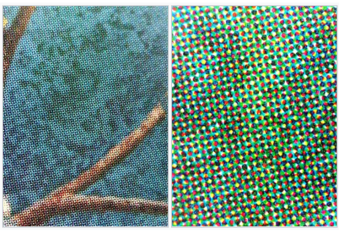
Fig. 2a. HSL cylinder. Fig. 2b. HSV cylinder.

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

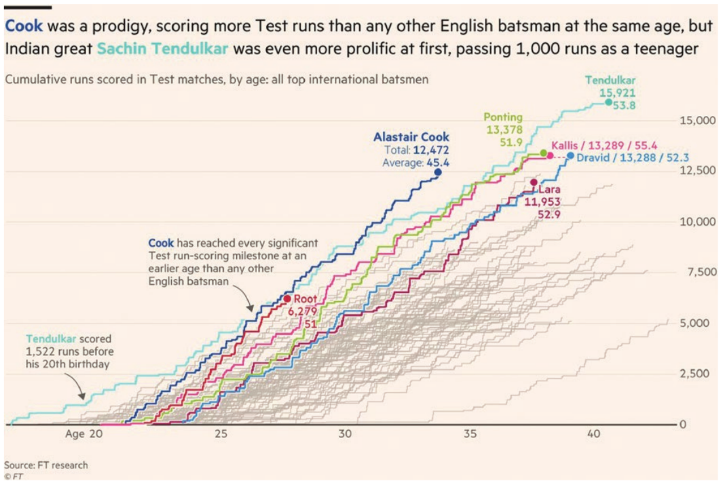
(next paragraphs roughly following [Kirk: Data Visualization, Chapter 9], examples from there unless mentioned otherwise)

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

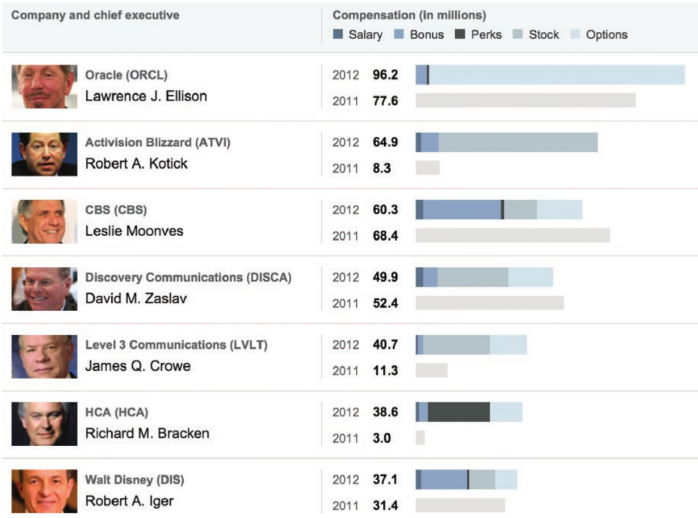
Highlighting.

- use color for pop out effect

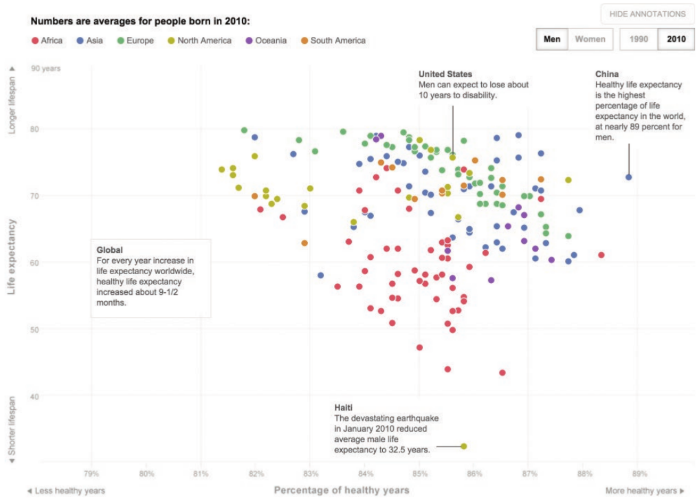


Nominal data.

- 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:

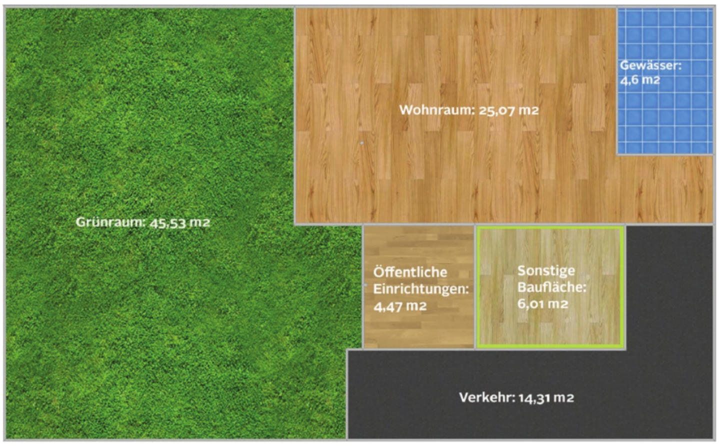


- how many categories can we visually distinguish? Kirk: at most 12 categories via hue (p259). General advice: as few as possible, try to have at most 6.



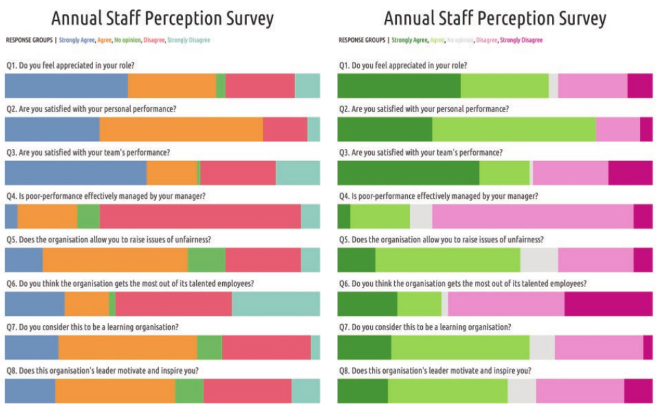
more information can be shown better via annotation, multiple additional plots (e.g. focussing on sub-groups), interactive visualization

- creative example: relative space usage in Vienna

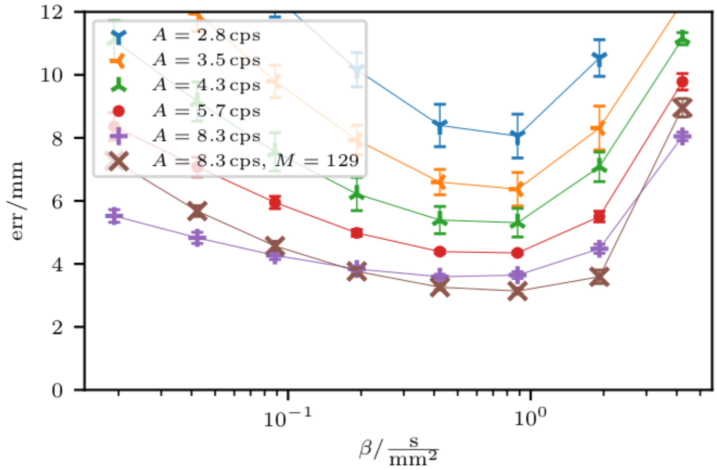


Ordinal data.

- colors should be ordered in an intuitive way

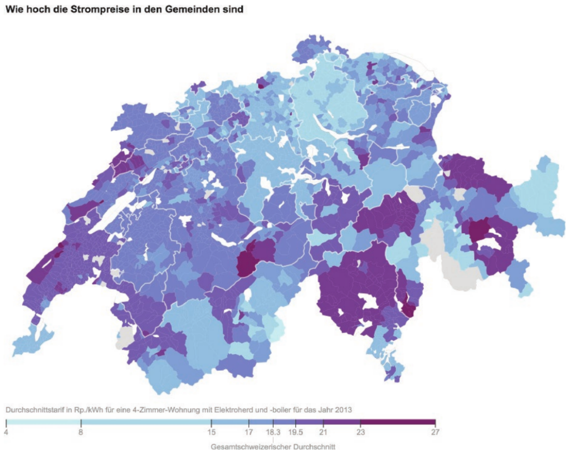


- bad example: dynamic PET.
  categories are ordinal (even rational), but color scheme is not; with direct labeling maybe do not need color at all, since curves are almost 'parallel'
  one exception: last curve, $M = 129$, which could be handled separately

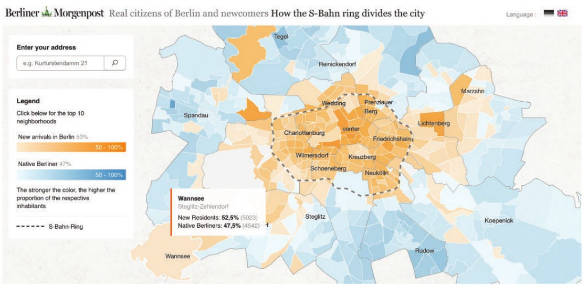


Interval or ratio data.

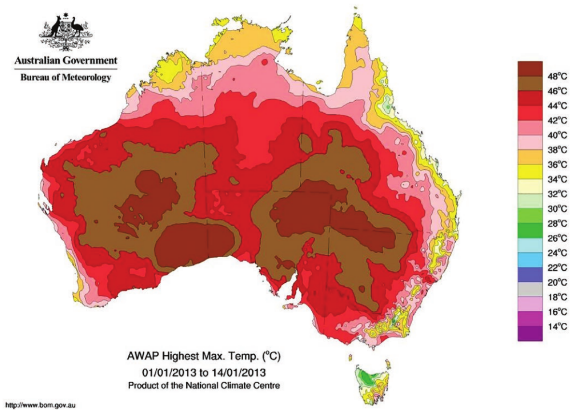
- use color scales / color maps



- 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

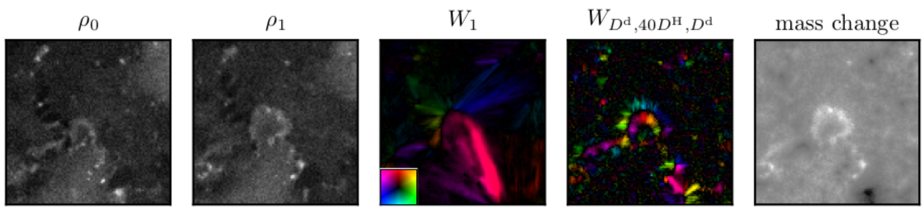
$$S : (x, y) \mapsto (f(x, y), g(x, y))$$

- 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)

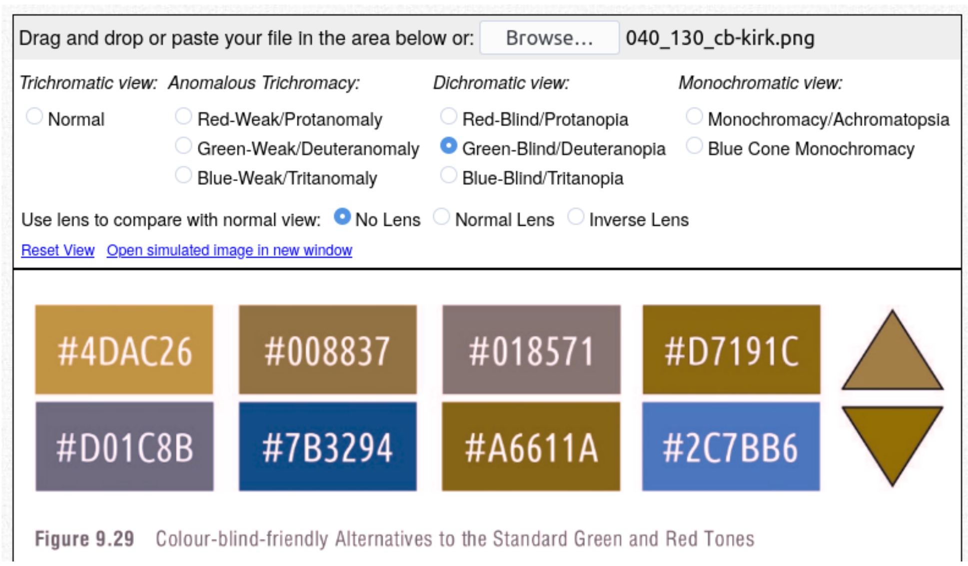
- example: unbalanced W1 field



- comment: can only convey a qualitative picture

4.5 Color deficiency

- example palette from [Kirk]:



- 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.

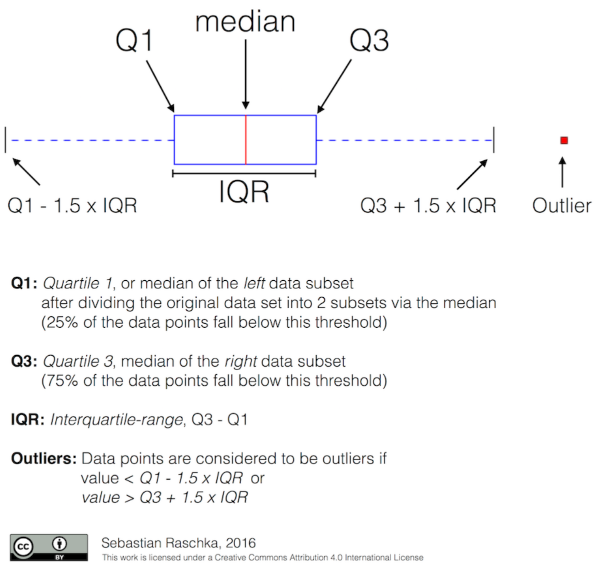
- y=f(x,z) where z is random

Mean and standard deviation.

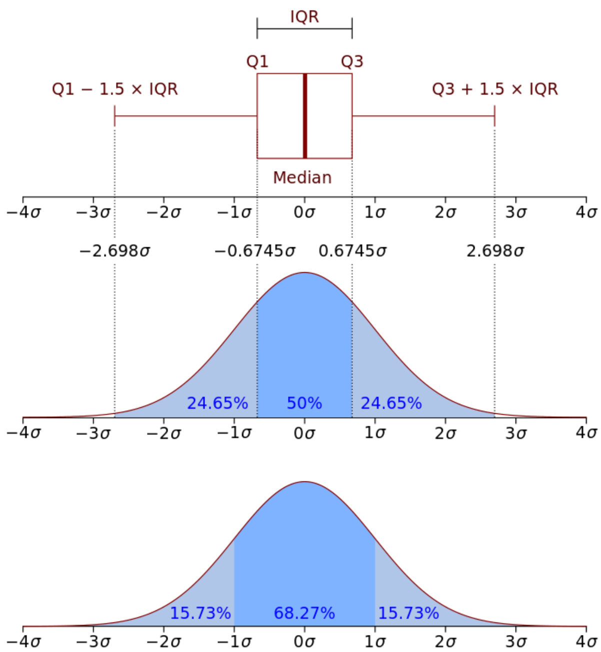
- easy to compute and to visualize
- usually inappropriate for non-Gaussian distributions

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 $\Rightarrow$ 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