Data Analytics and Machine Learning

Visualisation

Henrik Karstoft
Carl Schultz
Alexandros Iosifidis

TODAY'S OUTLINE

quick recap

1. why, and what, are we visualising?

2. tools for visualisation

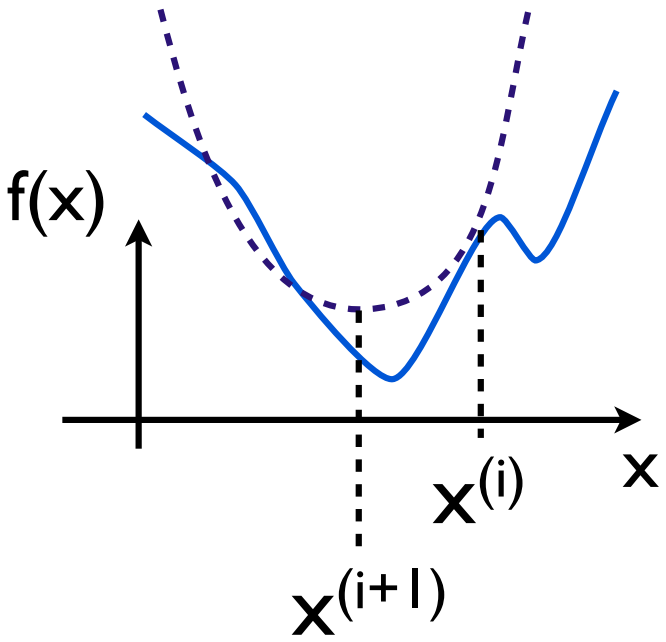
3. dimensionality reduction

say we want to solve this system:

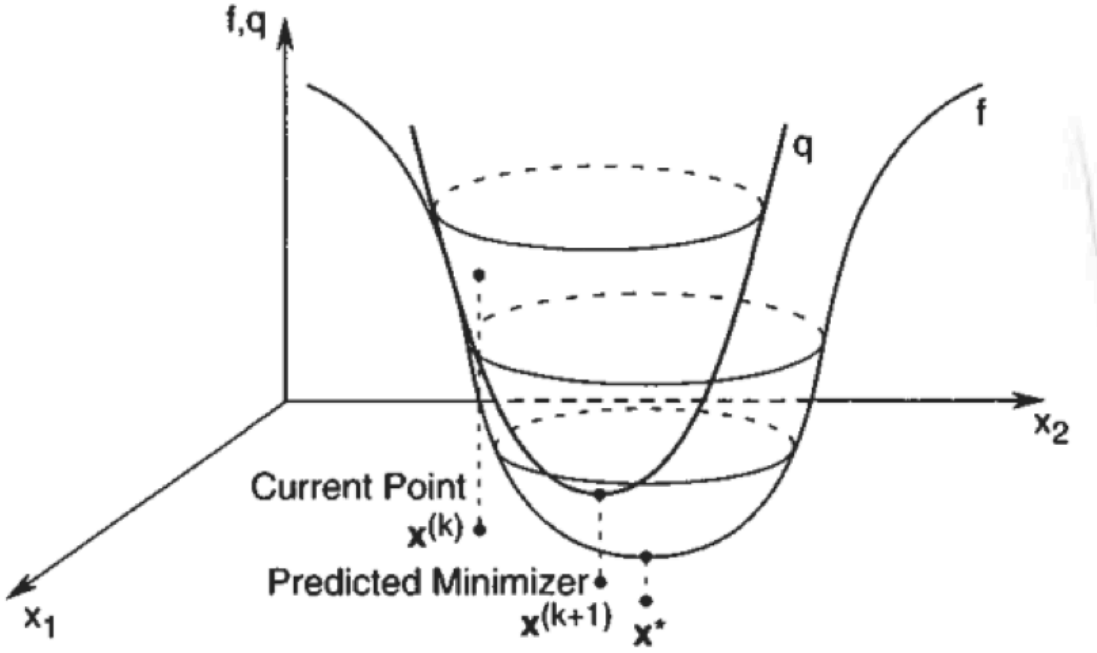
$$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22 = 0\,,$$
$$f_2(z_1, z_2) = 2z_1^2 + z_2^2 - 17 = 0\,.$$

Newton procedure for opimisation

1. guess initial value $X^{(0)}$
2. make quadratic approximation at $X^{(i)}$
3. find min of quadratic system to get better guess $X^{(i+1)}$
4. repeat from step 2 with $X^{(i+1)}$



1D case



nD case

simulated annealing

- Let $s = s_0$
- For $k = 0$ through $k_{\max}$ (exclusive):
  - $T \leftarrow \text{temperature}(k/k_{\max})$
  - Pick a random neighbour, $s_{\text{new}} \leftarrow \text{neighbour}(s)$
  - If $P(E(s), E(s_{\text{new}}), T) \geq \text{random}(0, 1)$:
    - $s \leftarrow s_{\text{new}}$
- Output: the final state s

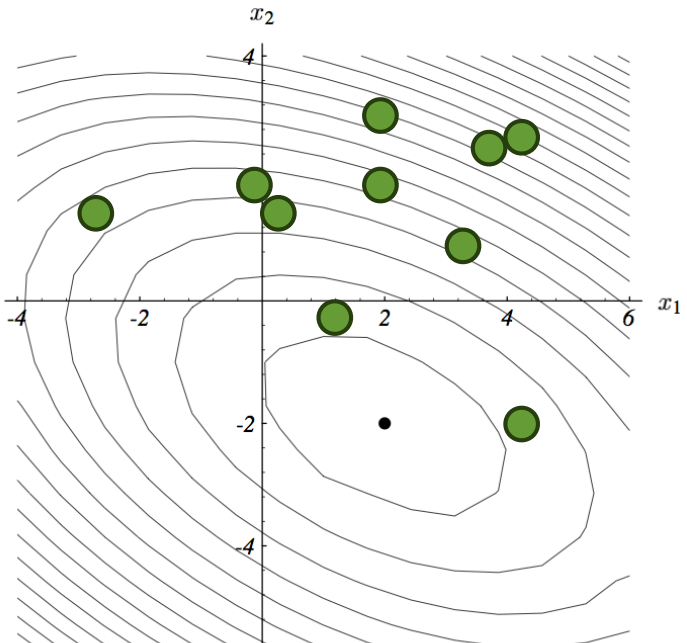
https://en.wikipedia.org/wiki/Simulated_annealing

1. evaluate fitness of each particle

2. update individual and global fitnesses

3. update individual velocity and position

if stop condition not met, then repeat



initial population can be random or seeded

given initial population generation t_0

1. select intermediate population based on fitness
   selection
2. use intermediate population to create offspring
   crossover ***mutation***
3. population (t+1) is offspring

repeat until stop conditions met

genetic algorithms:

why do they work?

hypothesis on why genetic algorithms work

why does this process perform optimisation rather than nonsense?

hypothesis: this process makes it very likely to **find** "building blocks" within the chromosomes that are:

(a) small
(b) above average fitness

hypothesis on why genetic algorithms work

why does this process perform optimisation rather than nonsense?

hypothesis: this process makes it very likely to **find** "building blocks" within the chromosomes that are;

note. the genetic algorithm will **automatically** find these building blocks just by following "selection-crossover-mutation" process, that is the argument - i.e. we do **not** need to say **what** these building blocks (called "schema") are, in fact we don't ever get to see them, we just get the candidate solutions

schemata

chromosome `00101001000010111010 1`

schema `0**********0**********`

this chromosome matches the schema
(* can match with either 0 or 1)

schemata

chromosome `0010100100001011110101`

schema `0**********0**********`

"building block"

again, we do not provide these schema, the argument is that "automatically" the genetic algorithm is finding good "schema" just by following "selection-crossover-mutation", and this is why it performs well overall as an optimisation algorithm

schemata

chromosome `0010100100001011110101`

schema `0**********0**********`

imagine that many many chromosomes that match this schema perform well (high fitness)

what is the chance that **children** of such chromosomes will **not** match the schema

called “disrupting” the schema

schemata

chromosome `0010100`**`10`**`000010111010 1`

schema `*******10**************`

selection favours high performing schema

out of those high performing schema, the ones that will survive are the ones that don't get disrupted

i.e. have few bits (low order), small distance between outer-most bits (short length)

schemata

chromosome `00101001000010111 0101`

schema `*******10**************`

therefore, genetic algorithms find small, high-performing "building-blocks" that can be combined in useful ways to solve problems

that's the hypothesis

particle swarm optimisation demo

Petr Krýže

Part 1

why, and what, are we visualising?

why visualise?

- helping you get insight into your:
  data, process, candidate solution, etc.
- helping communicate insights to others

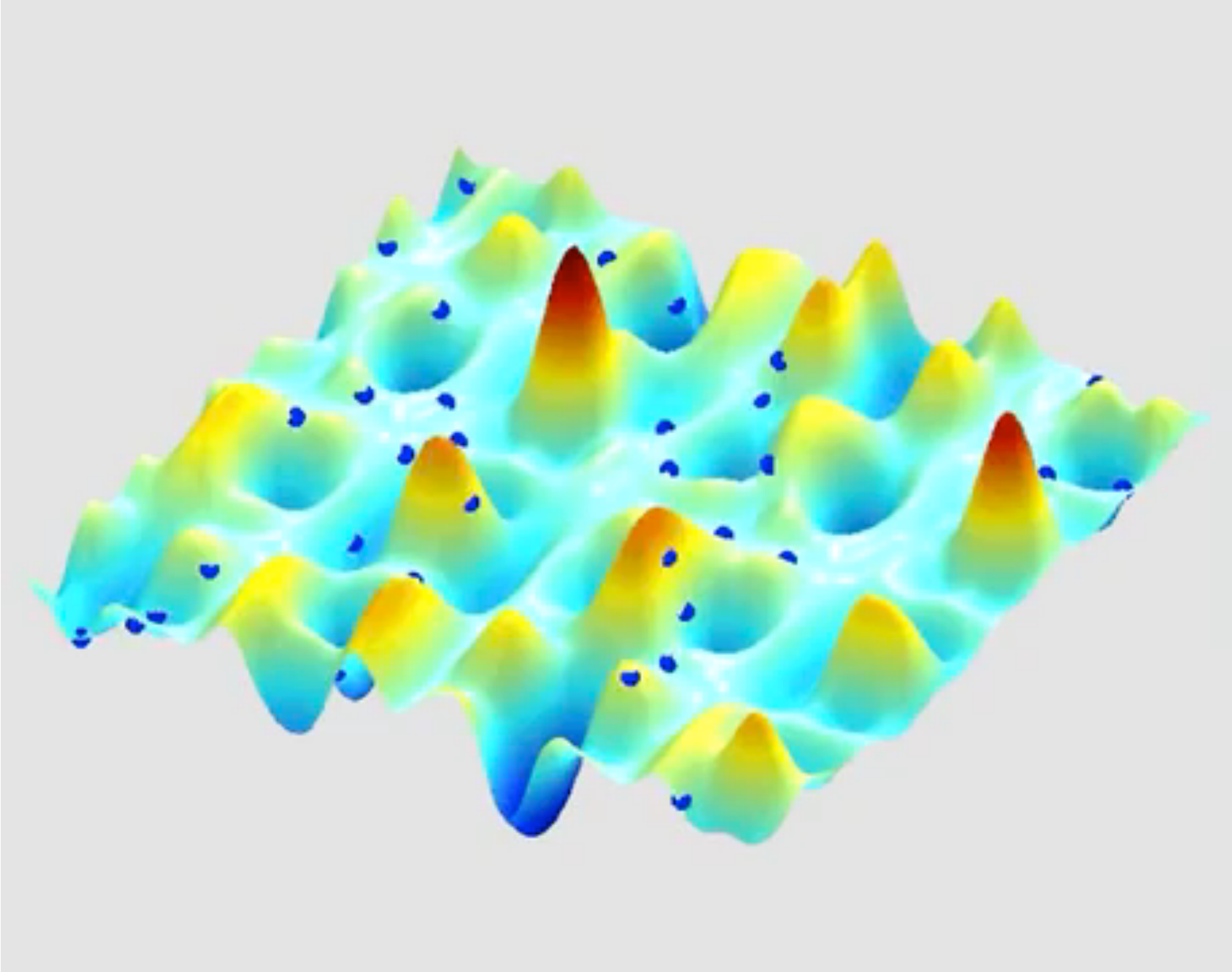
what exactly are we visualising?

?

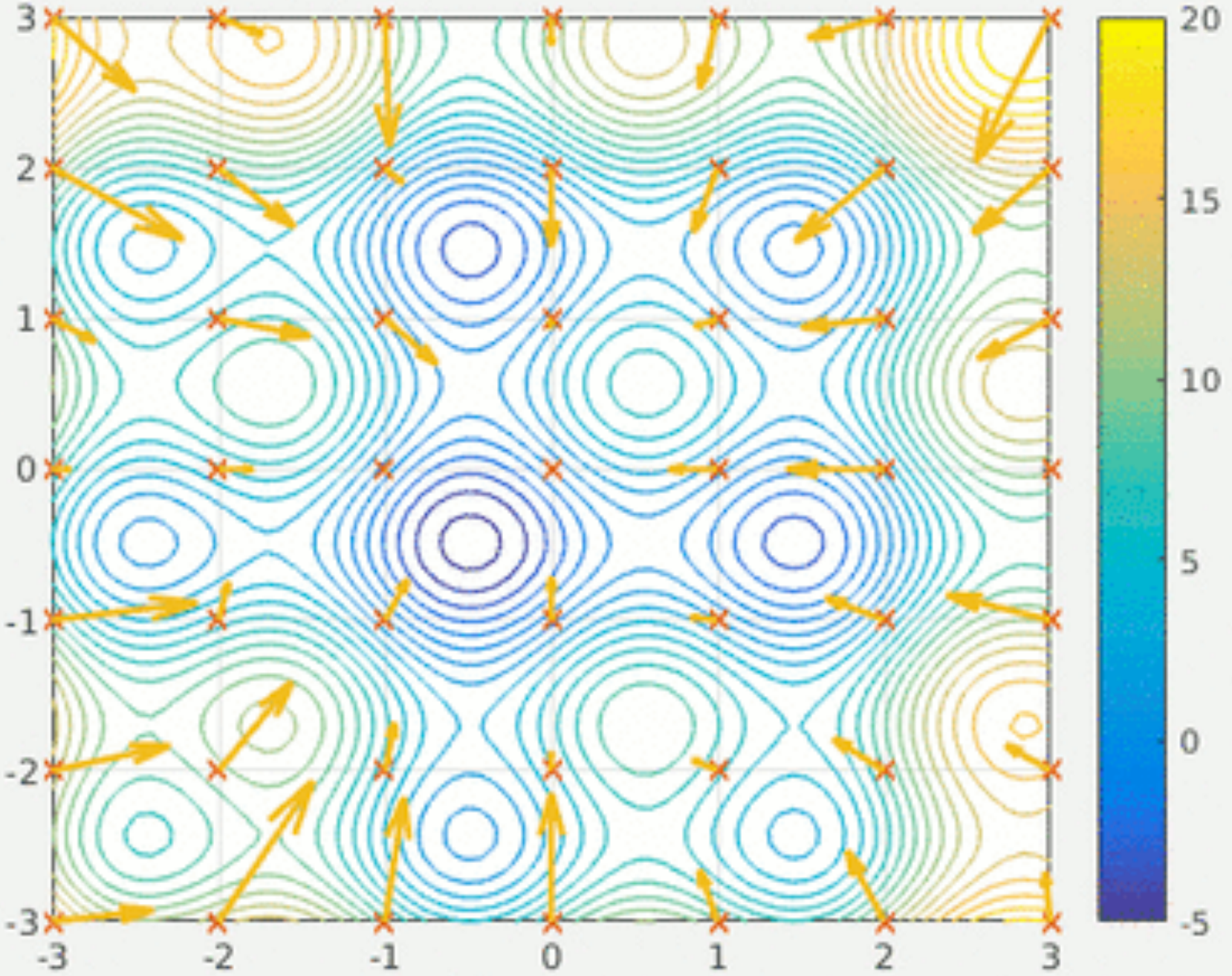
what exactly are we visualising?

- process of optimisation
- candidate solution
- data set
- a “model” (think about curve fitting)

in the following examples, what are we visualising?

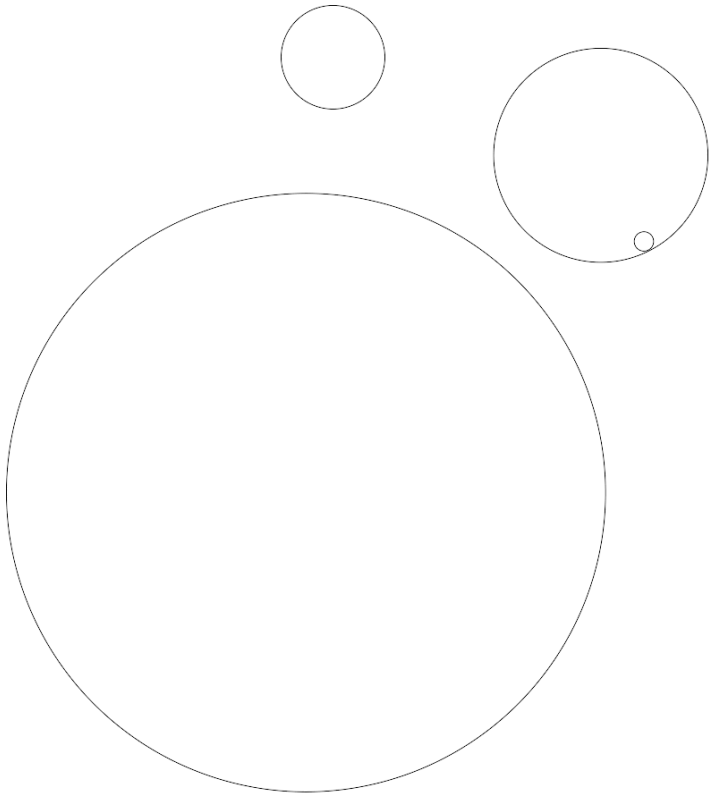


https://www.youtube.com/watch?v=VAASmSSsFaY

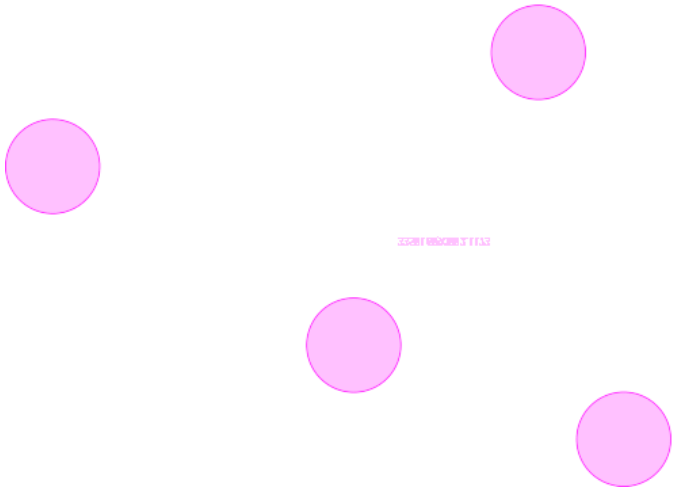


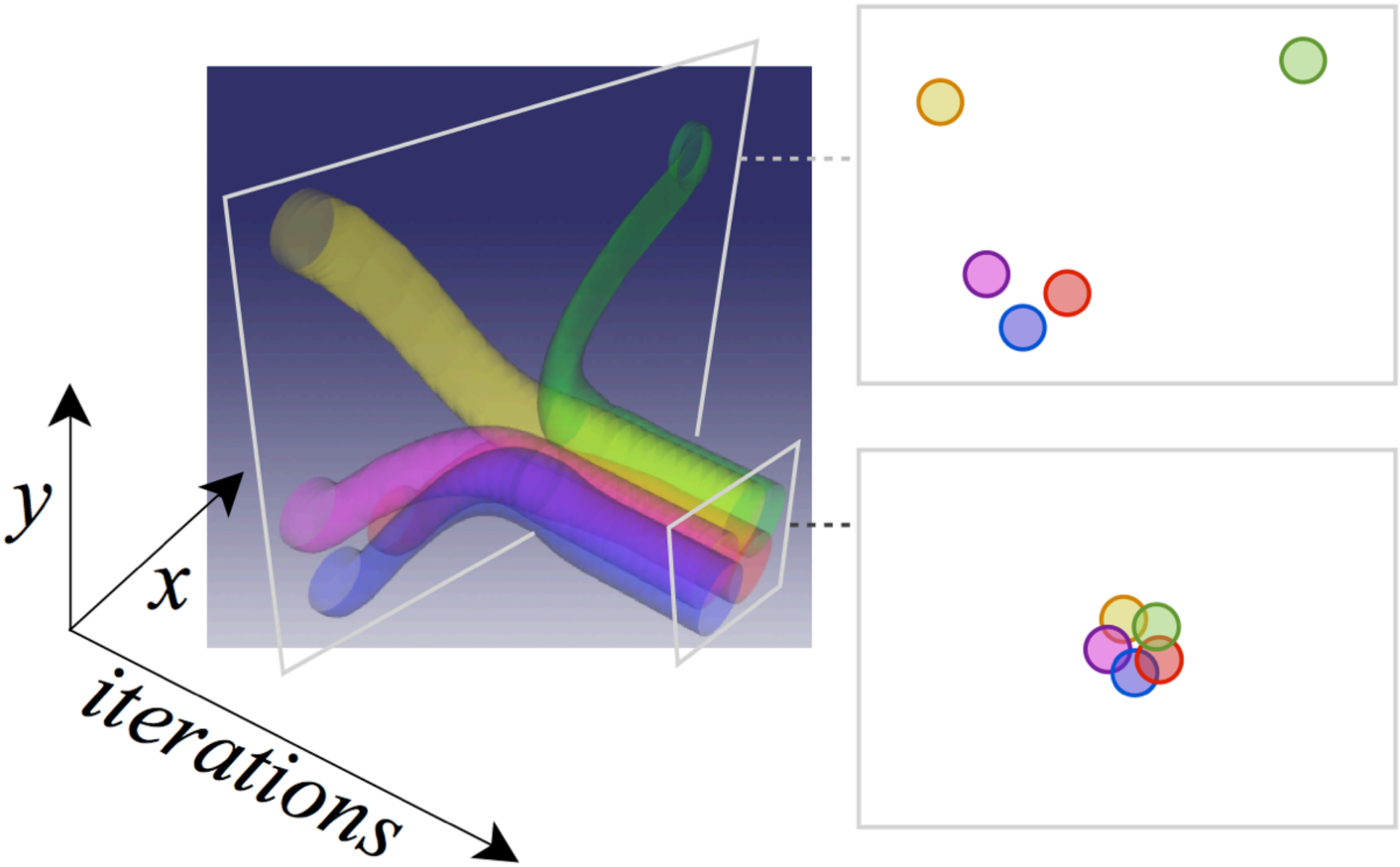
https://en.wikipedia.org/wiki/Particle_swarm_optimization

4 circles
touching



4 circles
touching, same size




y
x
iterations

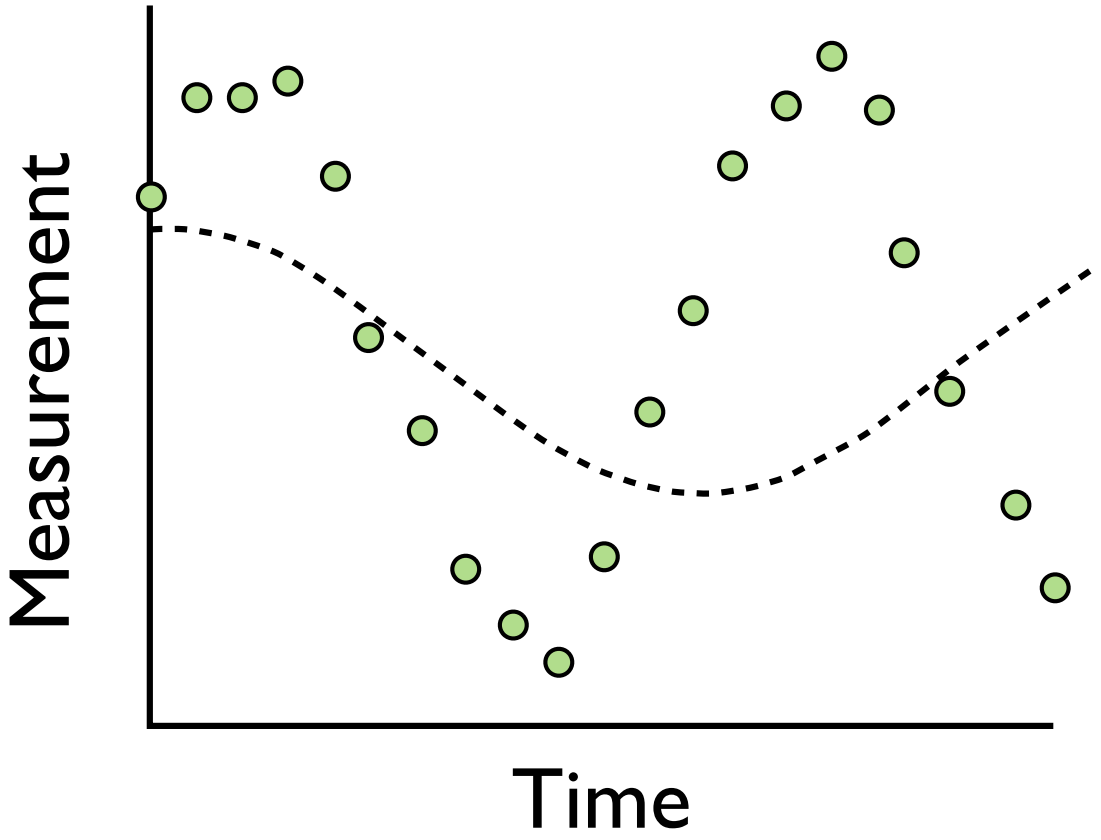
https://www.youtube.com/watch?v=uwz8JzrEwWY

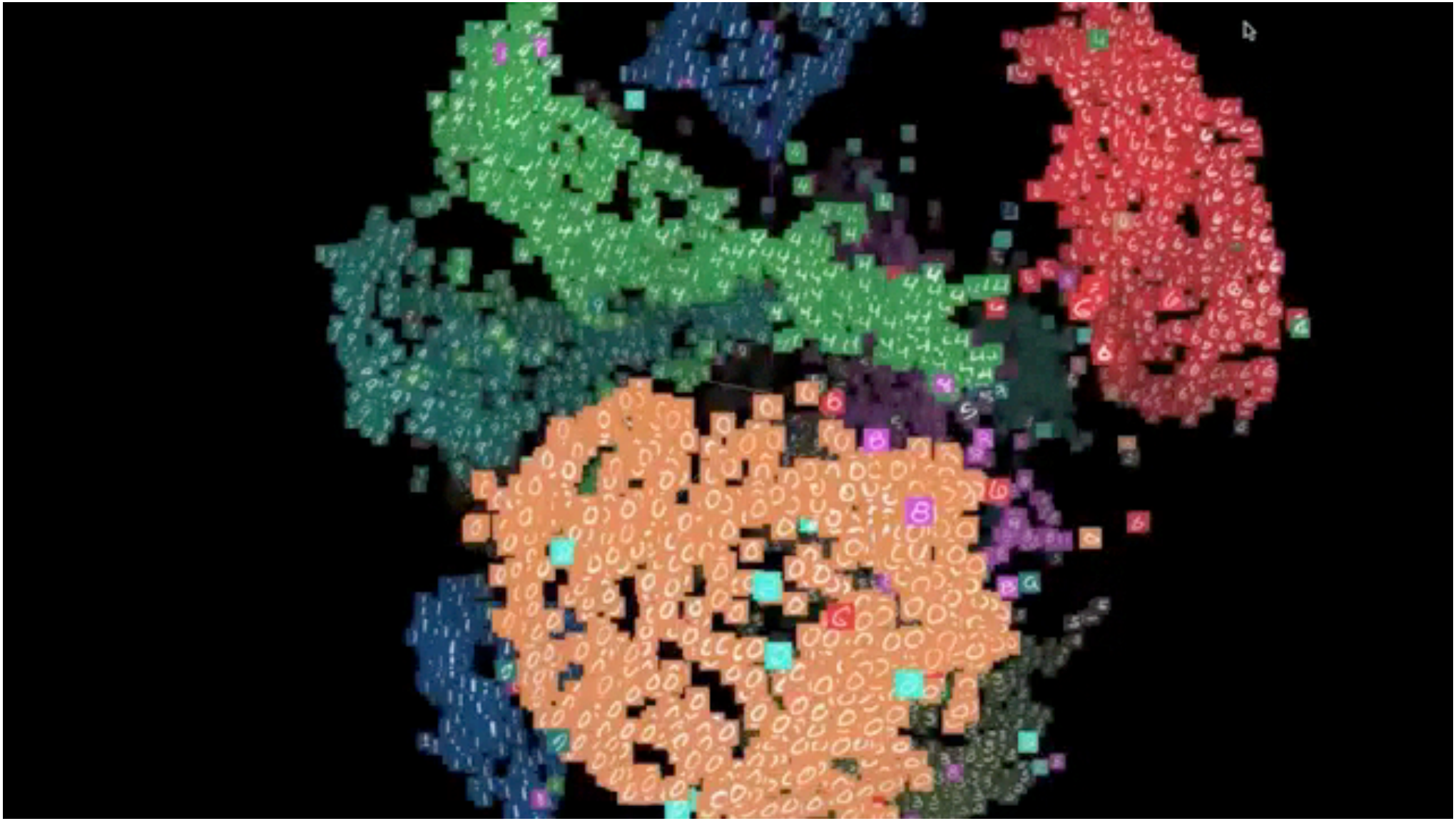
curve fitting

$$f(t) = A \sin(\omega t + \Phi)$$

(initial guess)

$A=1.01 \quad \omega=0.592 \quad \Phi=1.541$





https://www.youtube.com/watch?v=wvsE8jm1GzE

SUMMARY Part 1. visualisation - why and what?

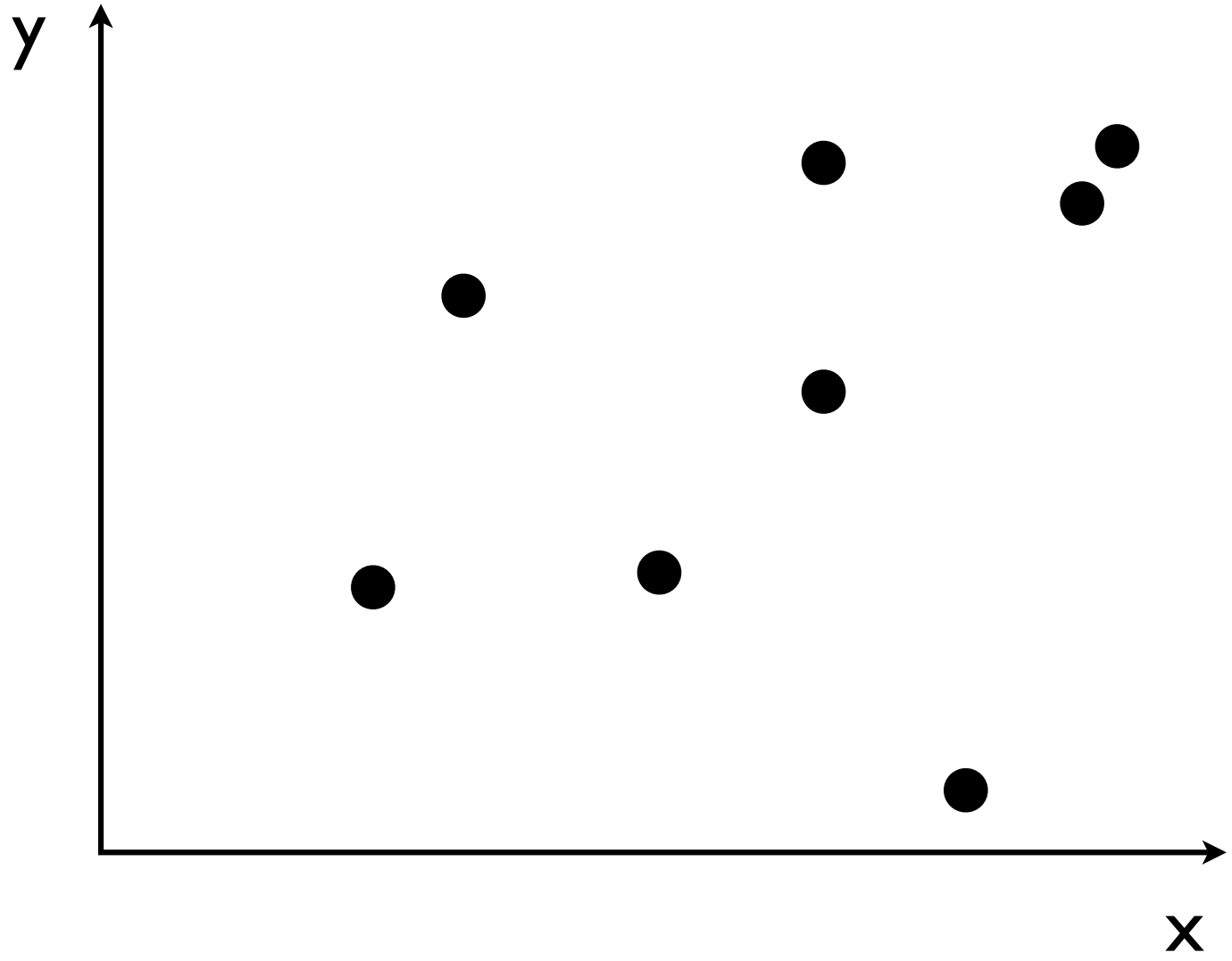
- insight into data, models, processes
  (us and reader)
- what?
  process of optimisation
  candidate solution
  data set
  a "model" (think about curve fitting)

Part 2

tools for visualisation

- high dimensionality:
  lots of inputs and an objective function value
- each data point can be described by many
  “features”

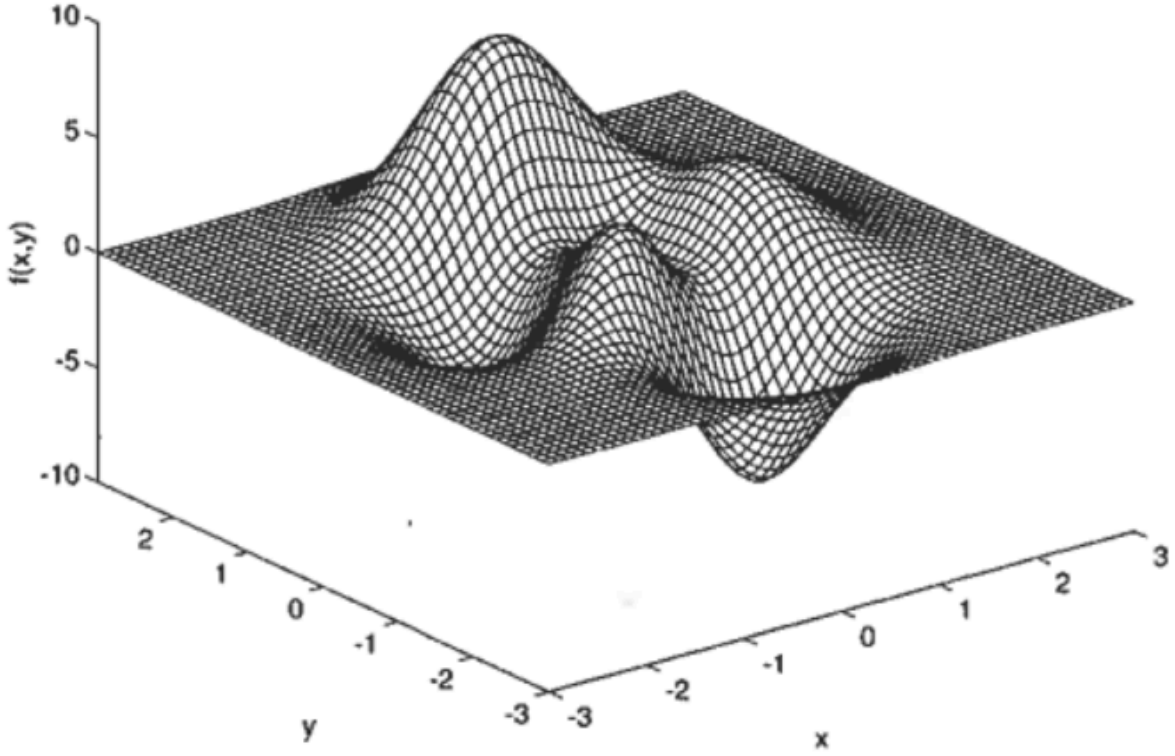
scatterplot



what if each data point has three components (x,y,z)?
what "tools" do we have to visualise more dimensions?

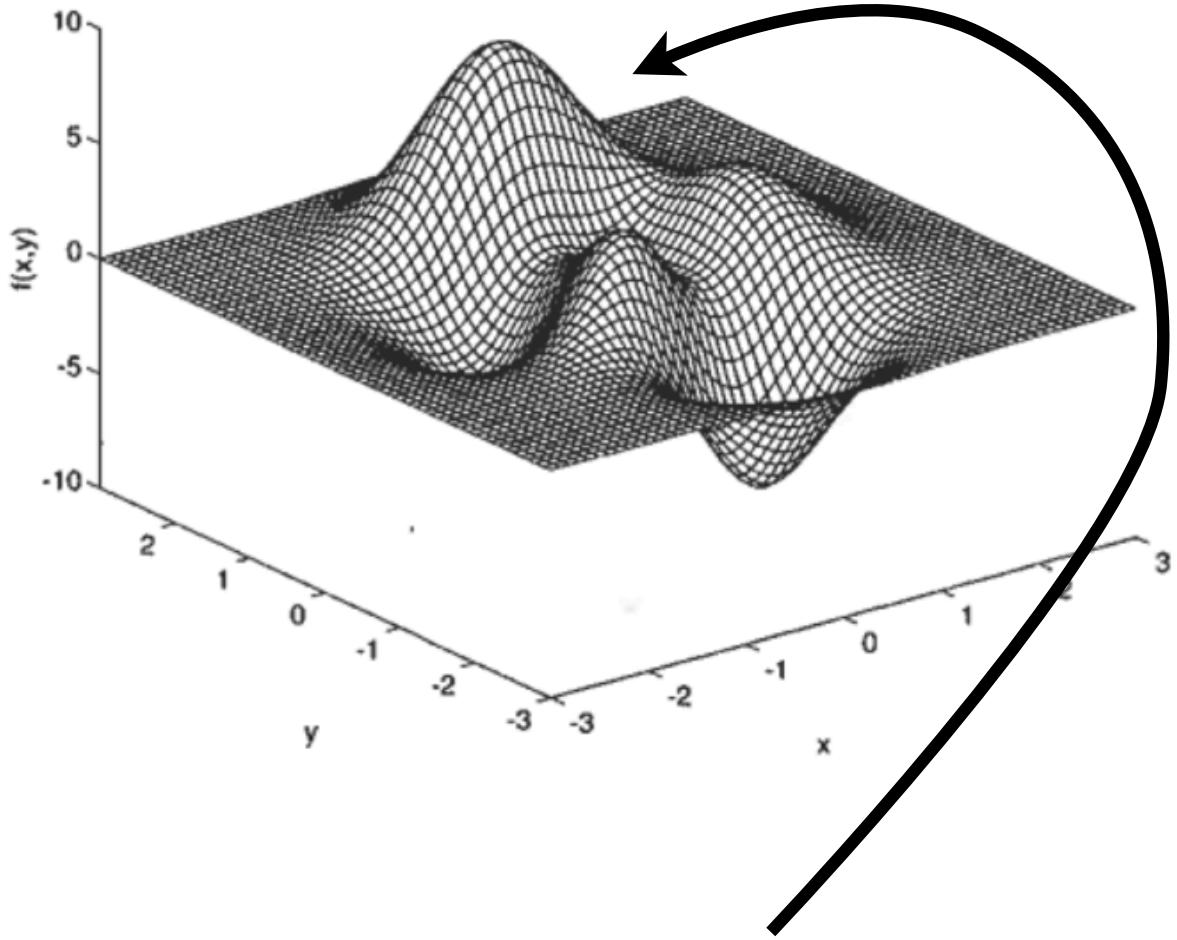
- projection
- size
- colour
- shapes
- labels
- animation
- interactivity

- **projection**
- size
- colour
- shapes
- labels
- animation
- interactivity



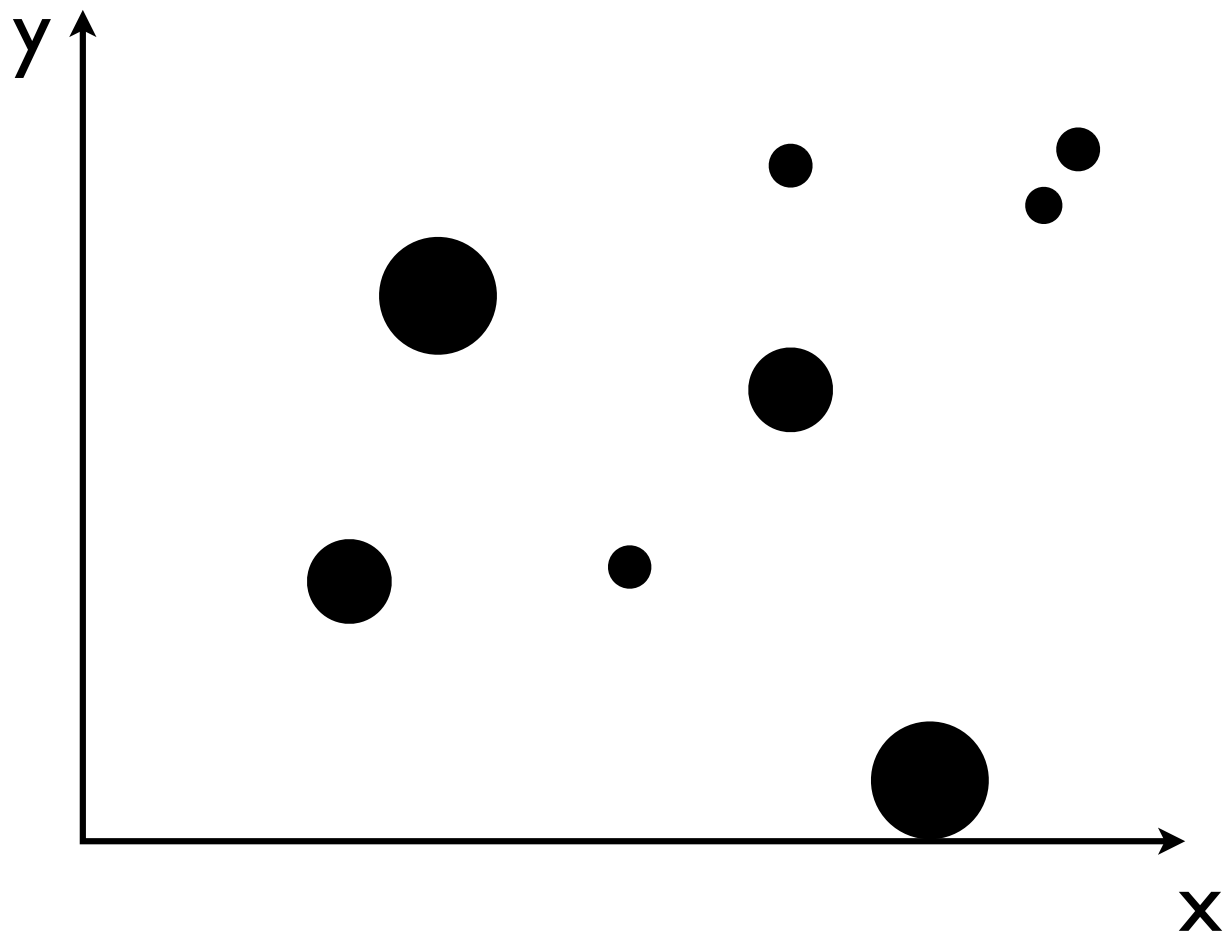
we are projecting 3D information onto a 2D surface (e.g. flat piece of paper)

- **projection**
- size
- colour
- shapes
- labels
- animation
- interactivity

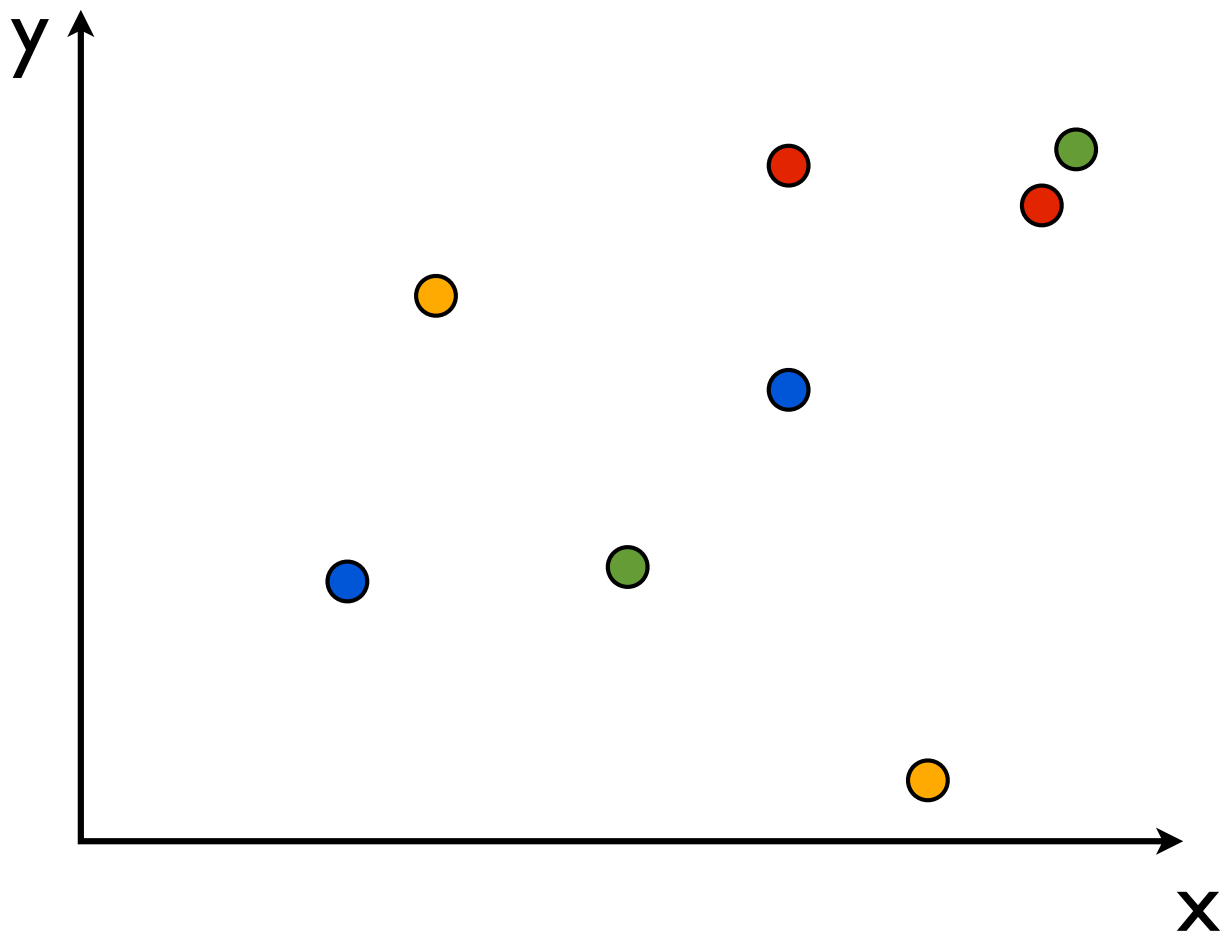


e.g. you can't see behind the peak

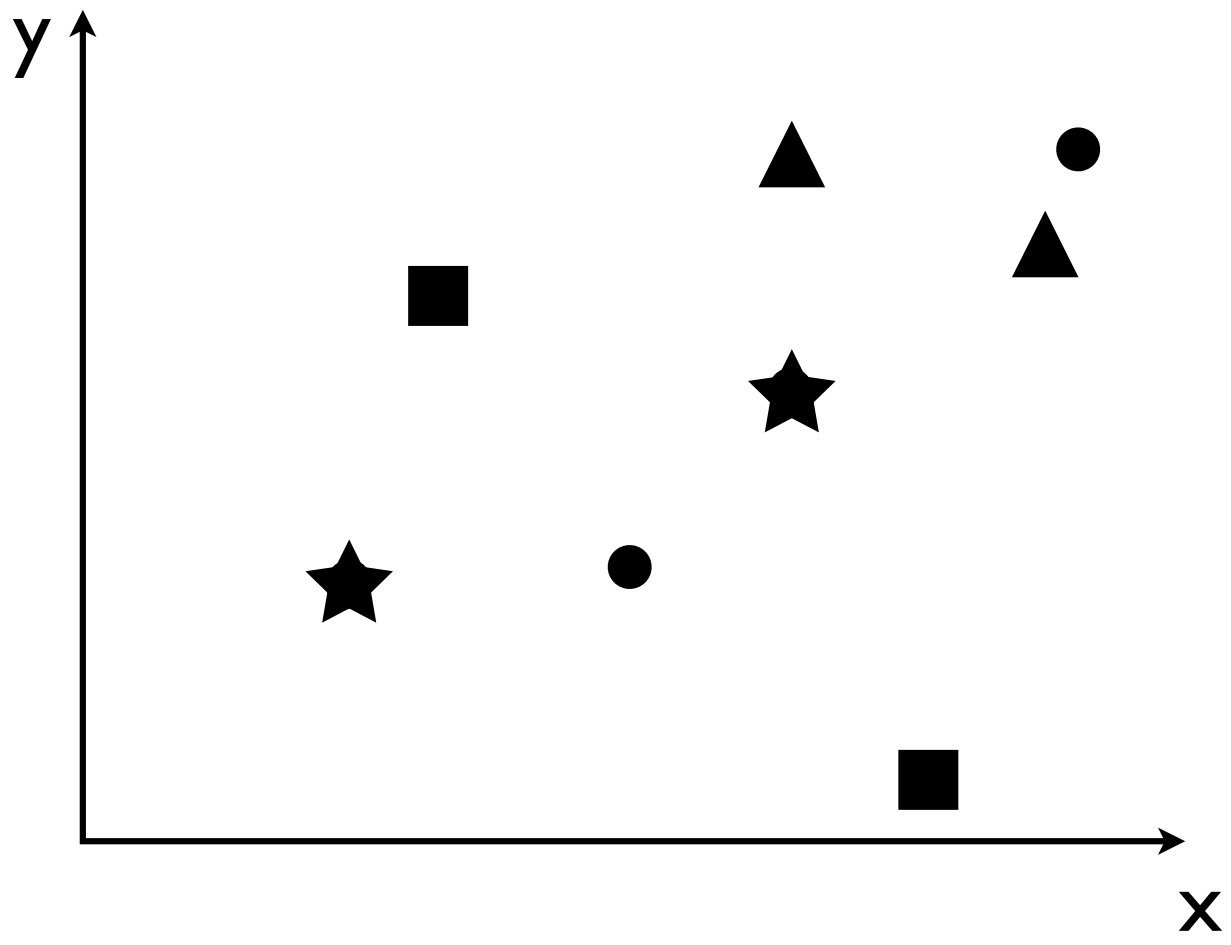
- projection
- **size**
- colour
- shapes
- labels
- animation
- interactivity



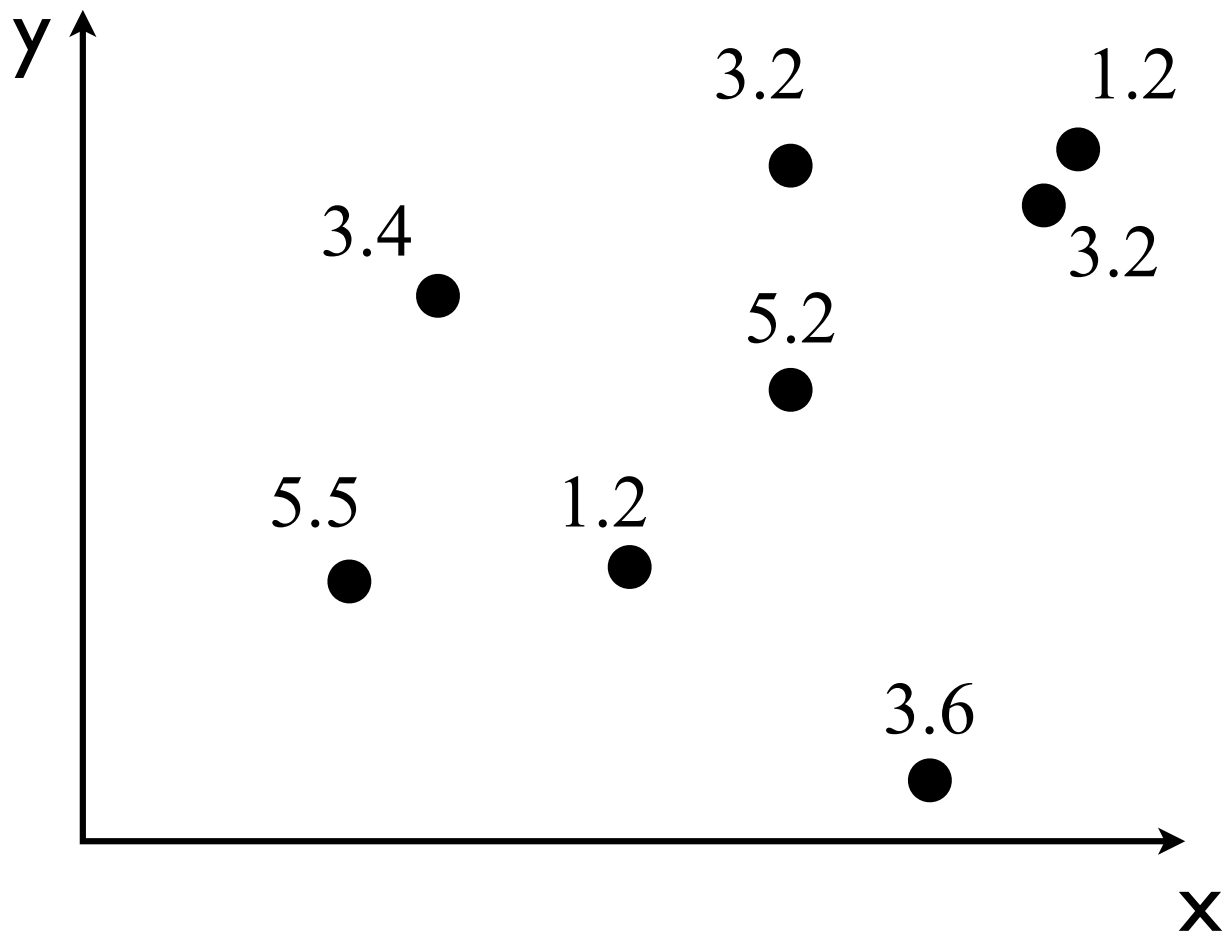
- projection
- size
- **colour**
- shapes
- labels
- animation
- interactivity



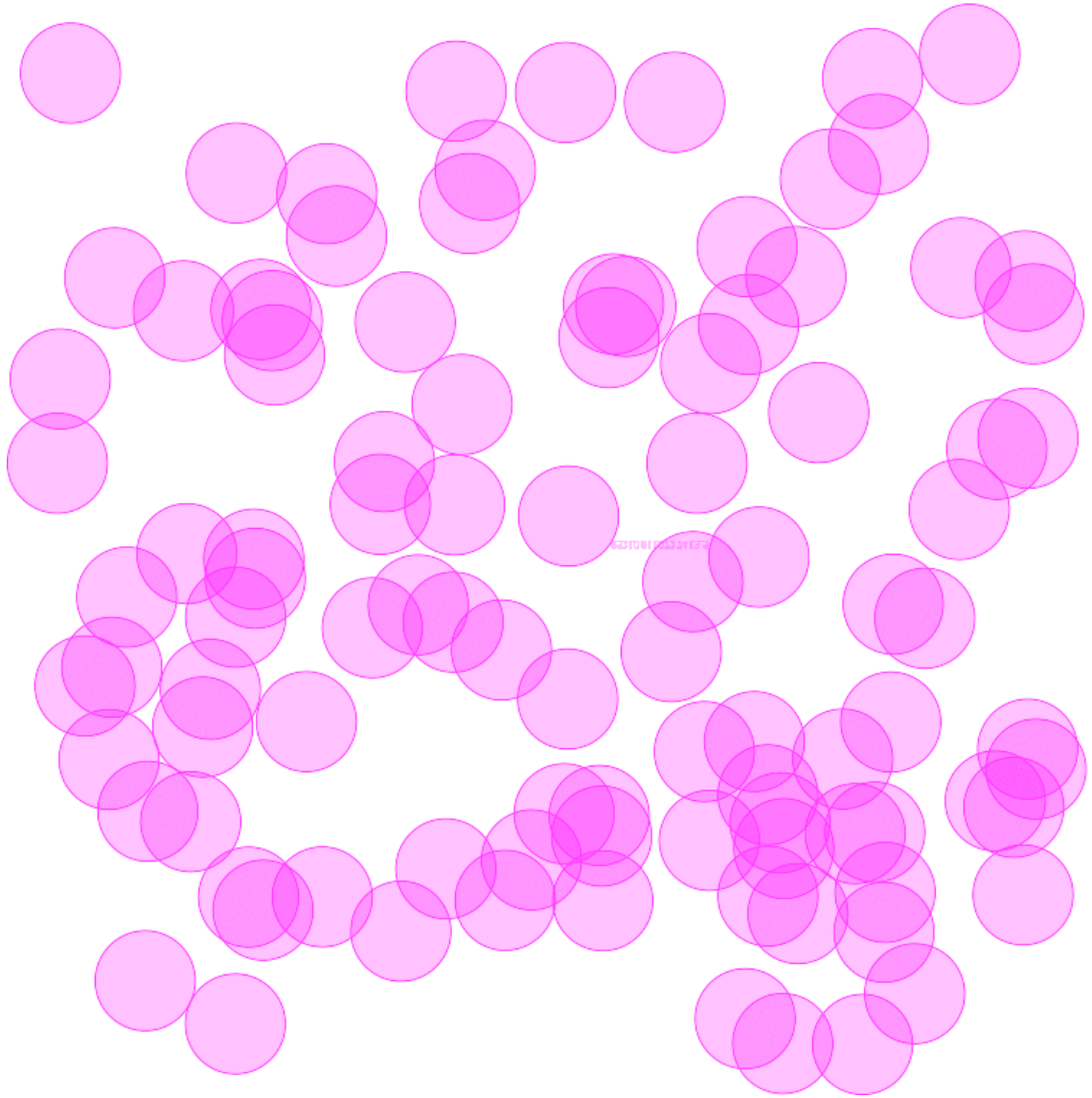
- projection
- size
- colour
- **shapes**
- labels
- animation
- interactivity



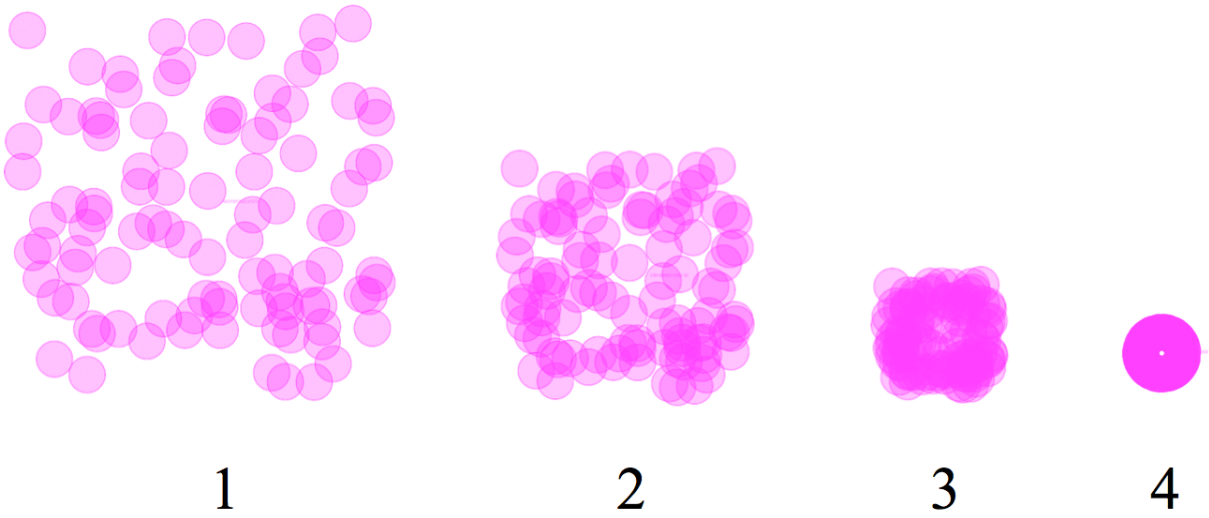
- projection
- size
- colour
- shapes
- **labels**
- animation
- interactivity



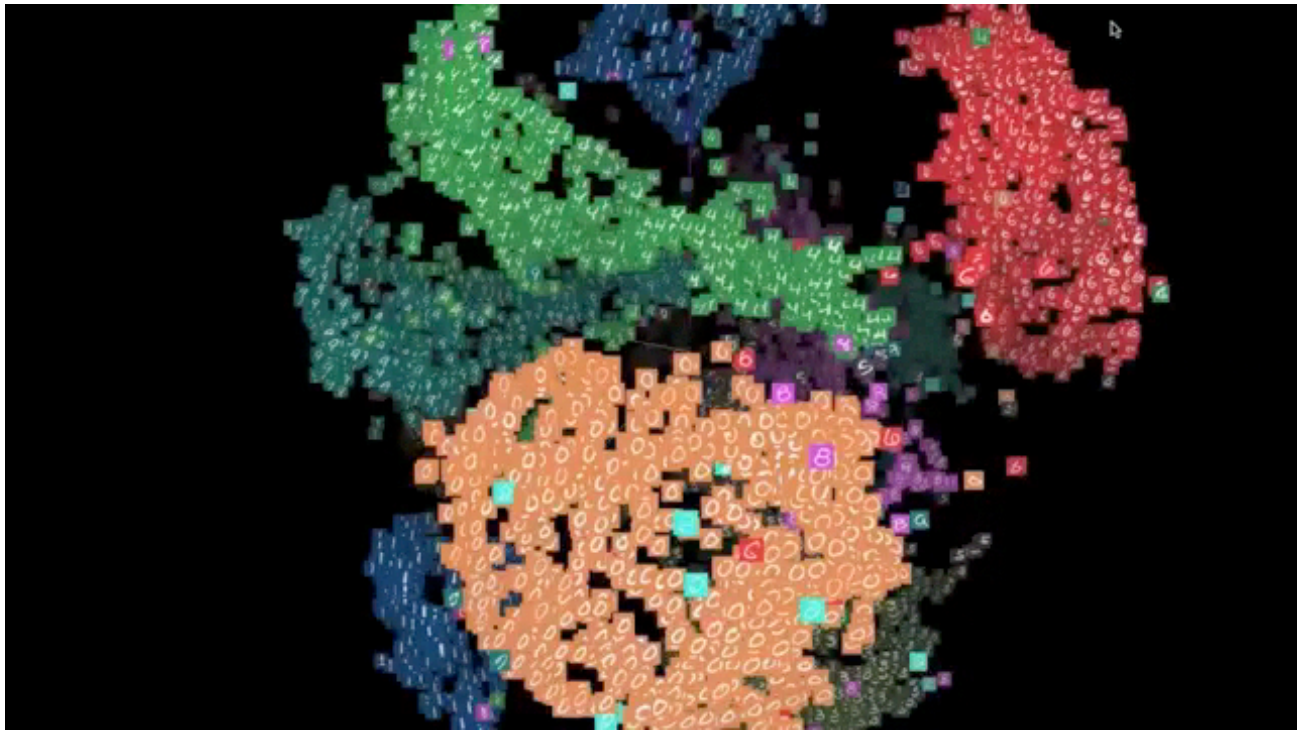
- projection
- size
- colour
- shapes
- labels
- **animation**
- interactivity



- projection
- size
- colour
- shapes
- labels
- **animation**
- interactivity



- projection
- size
- colour
- shapes
- labels
- animation
- **interactivity**



- projection
- size
- colour
- shapes
- labels
- animation
- interactivity

also keep in mind:
are we visualising discrete or continuous values?

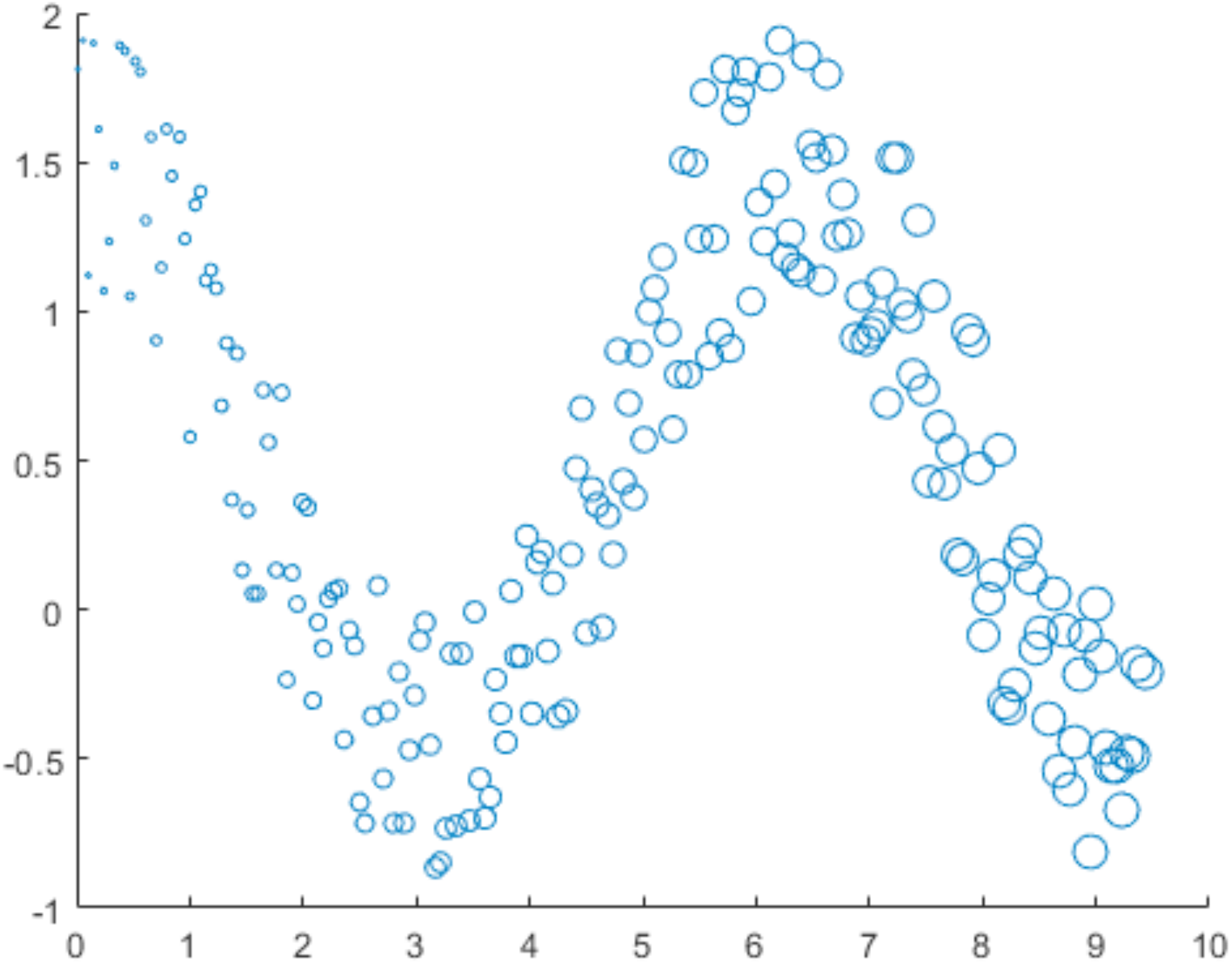
- projection
- size
- colour
- shapes
- labels
- animation
- interactivity

for more info about data visualisation, this is a great person to read up on:

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

- scatterplot
- scatterplot matrix
- parallel coordinates

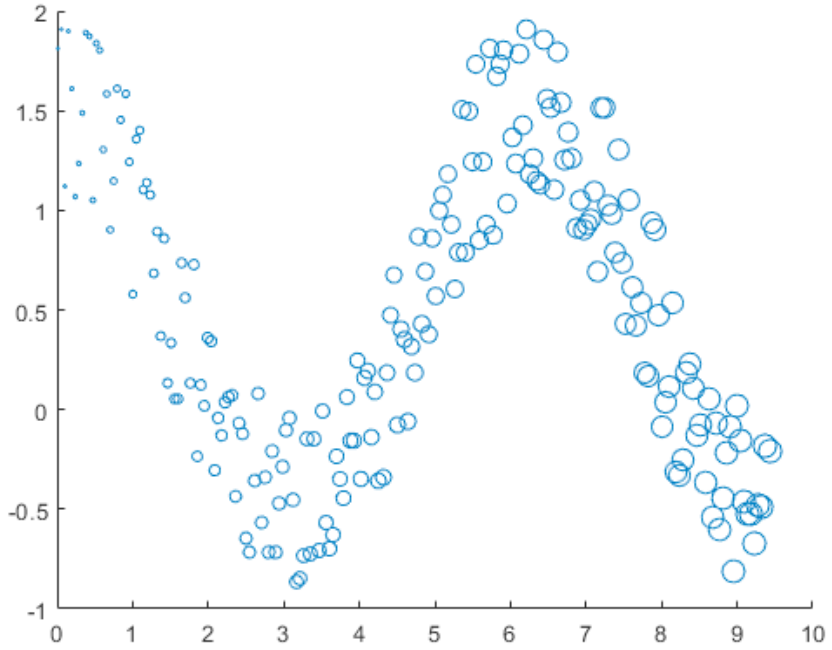
scatterplot



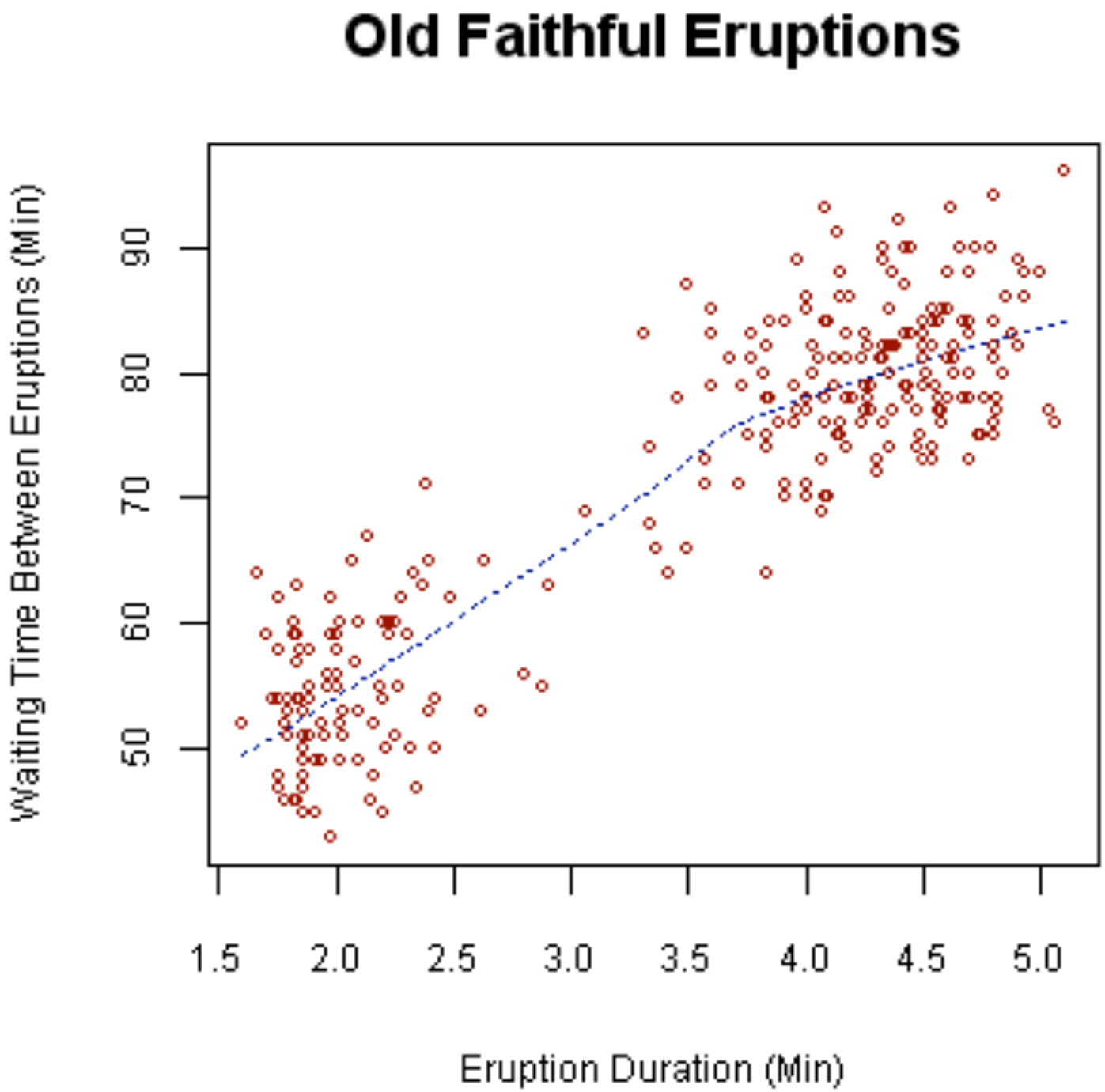
https://nl.mathworks.com/help/matlab/ref/scatter.html

scatterplot

- using Cartesian coordinates to display (normally 2D) data
- one variable (normally on vertical axis) can be controlled by other variable (normally horizontal axis)
- ...or neither variable is controller/dependent
- great for highlighting correlations and clusters between variables



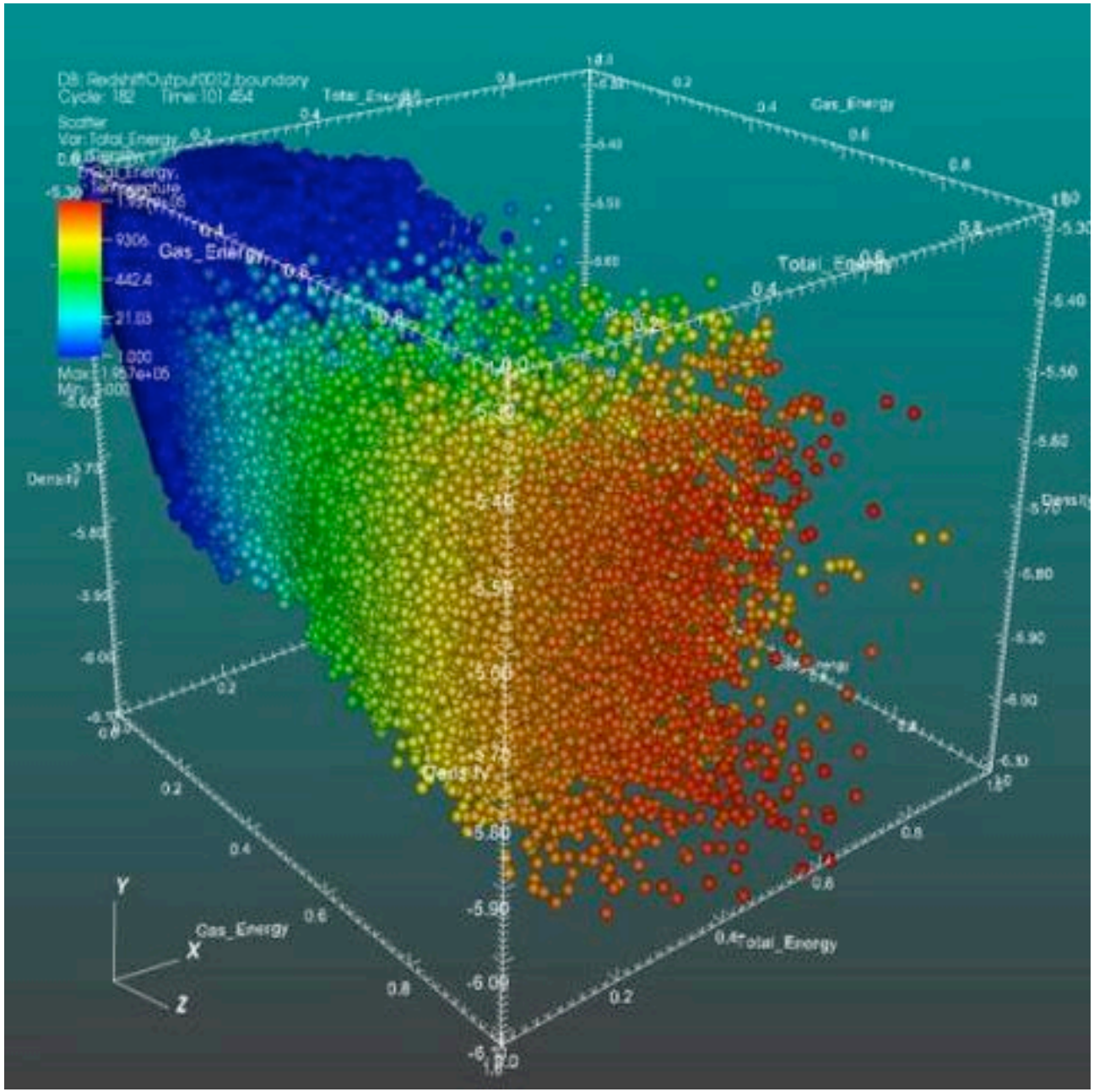
https://nl.mathworks.com/help/matlab/ref/scatter.html



correlations? clusters?

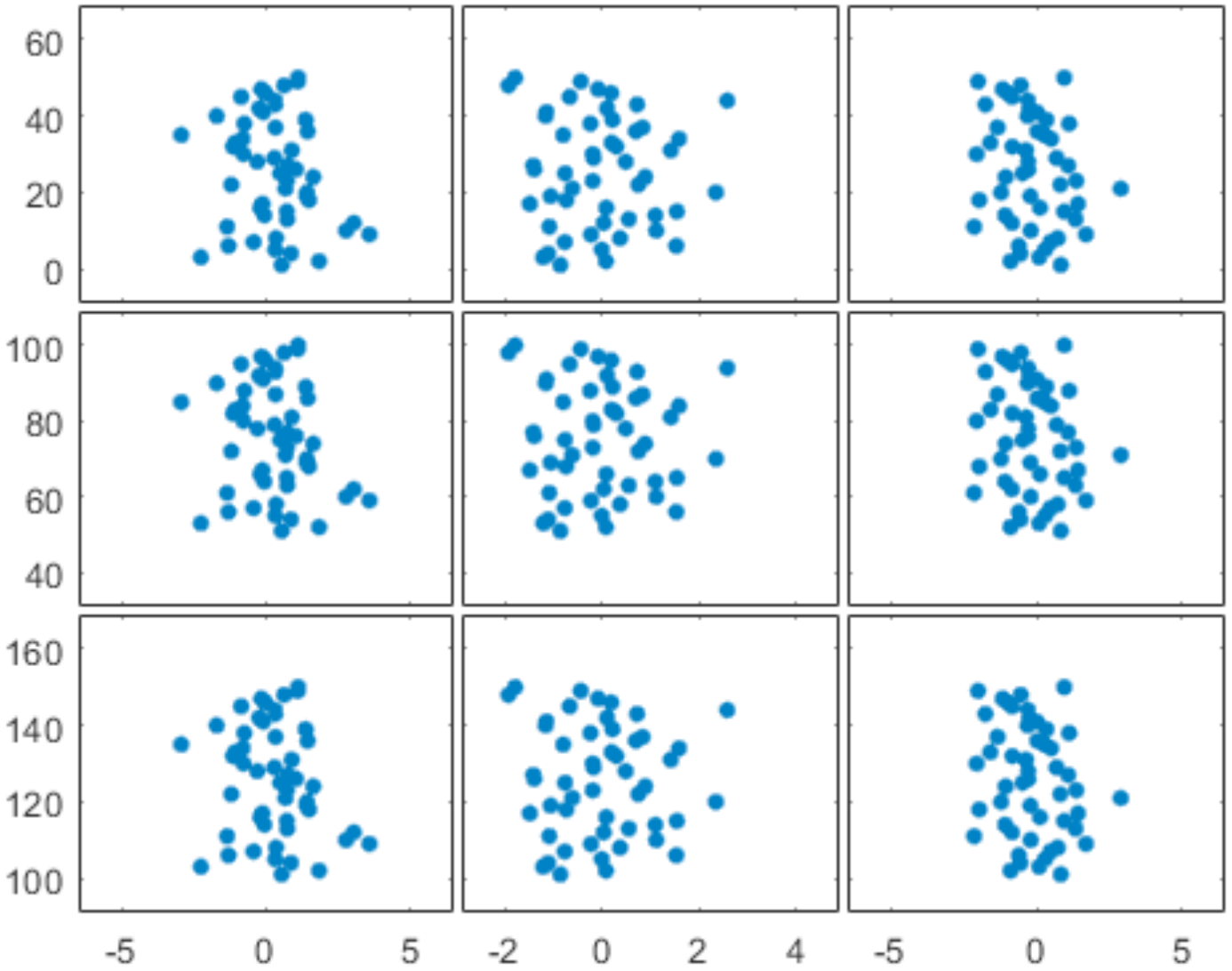
https://en.wikipedia.org/wiki/Scatter_plot

3D scatterplot



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

scatterplot matrix

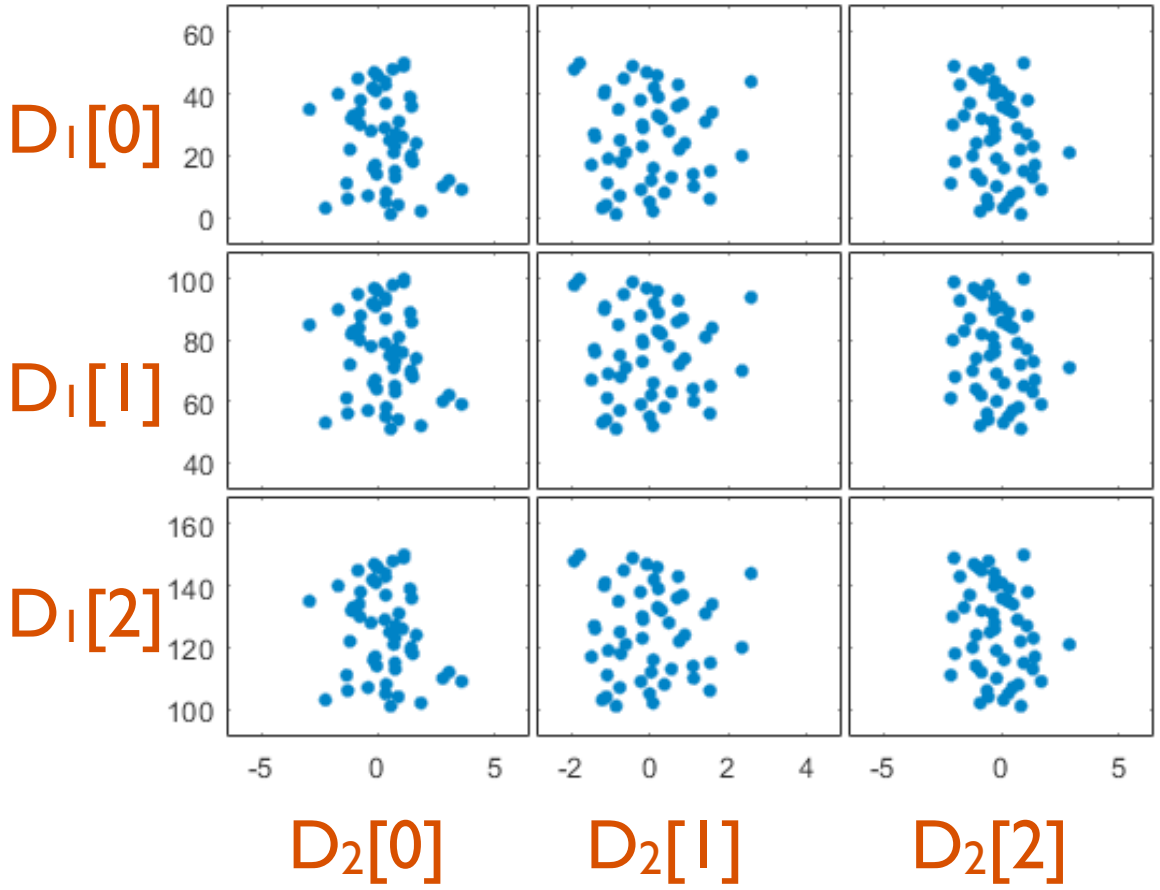


https://nl.mathworks.com/help/matlab/ref/plotmatrix.html

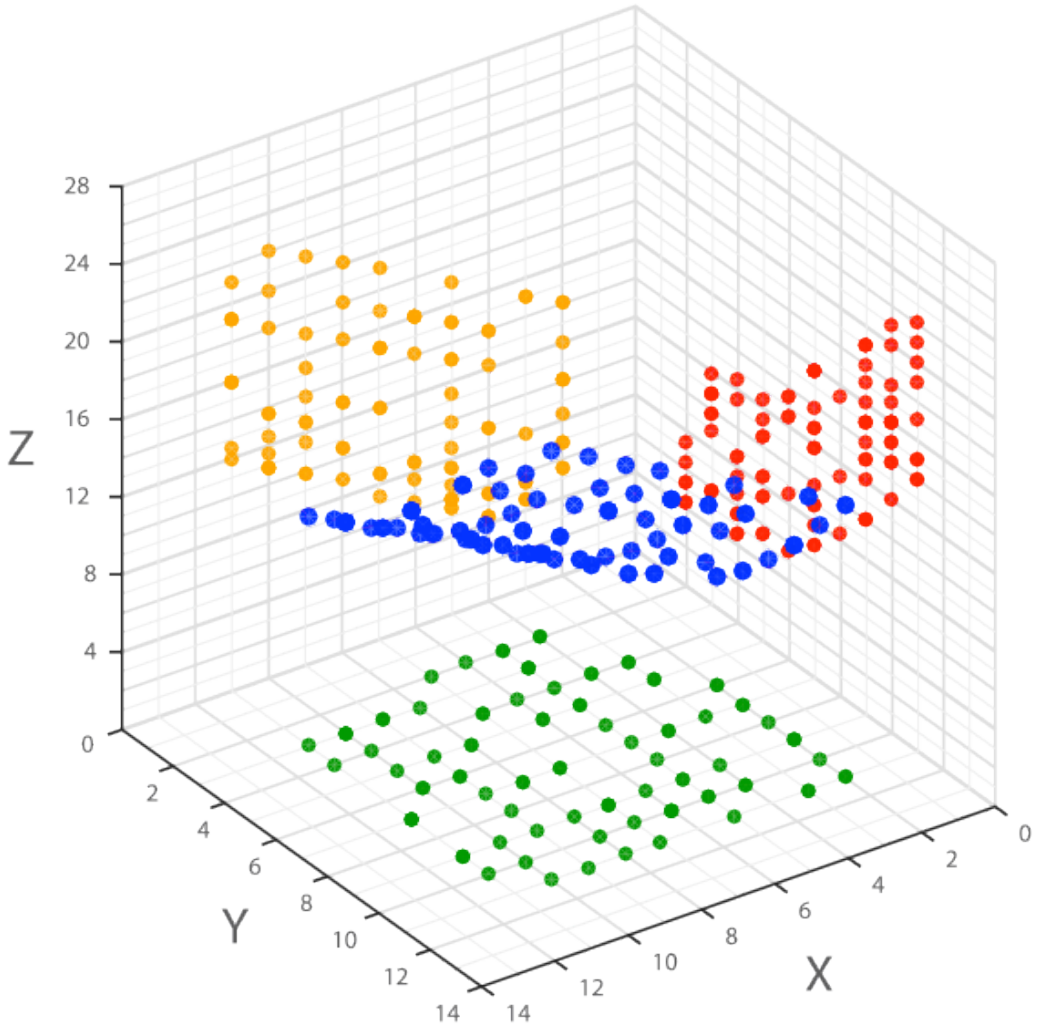
scatterplot matrix

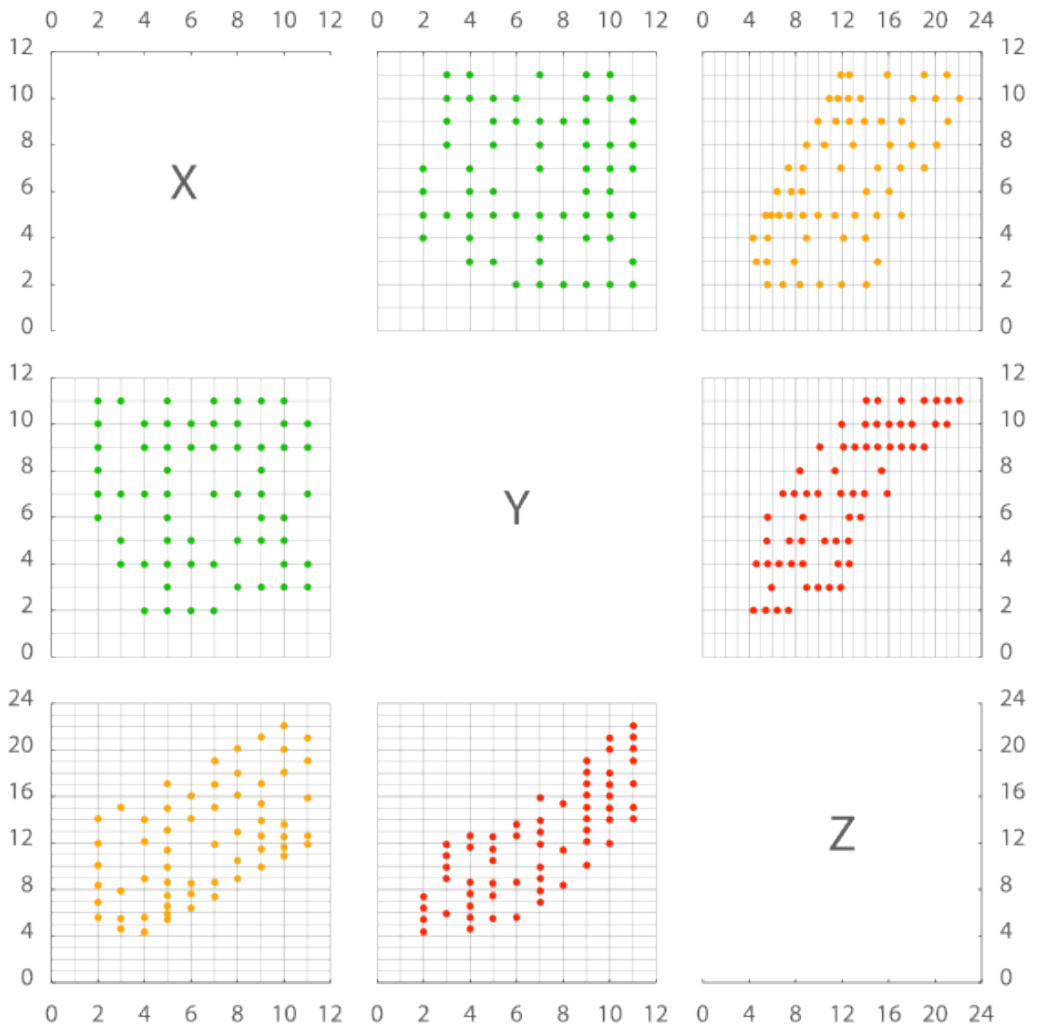
this example compares to different datasets D_1, D_2 that each have k=3 dimensions

- given data with *k* dimensions, each cell has scatterplot comparing two of these dimensions (pairwise)



https://nl.mathworks.com/help/matlab/ref/plotmatrix.html



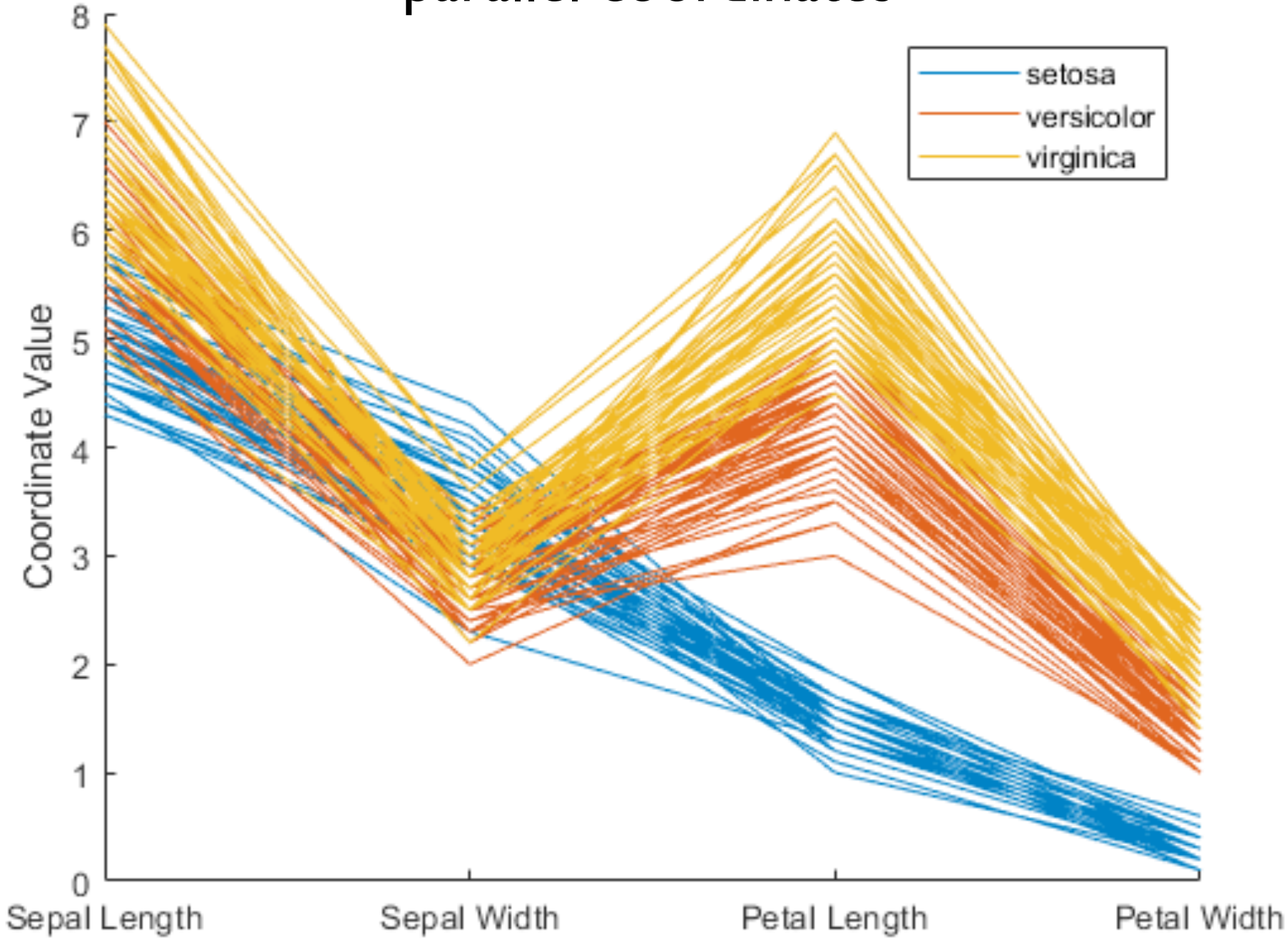


data in **blue**
(other colours are projections)

why not plot diagonal?

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

parallel coordinates

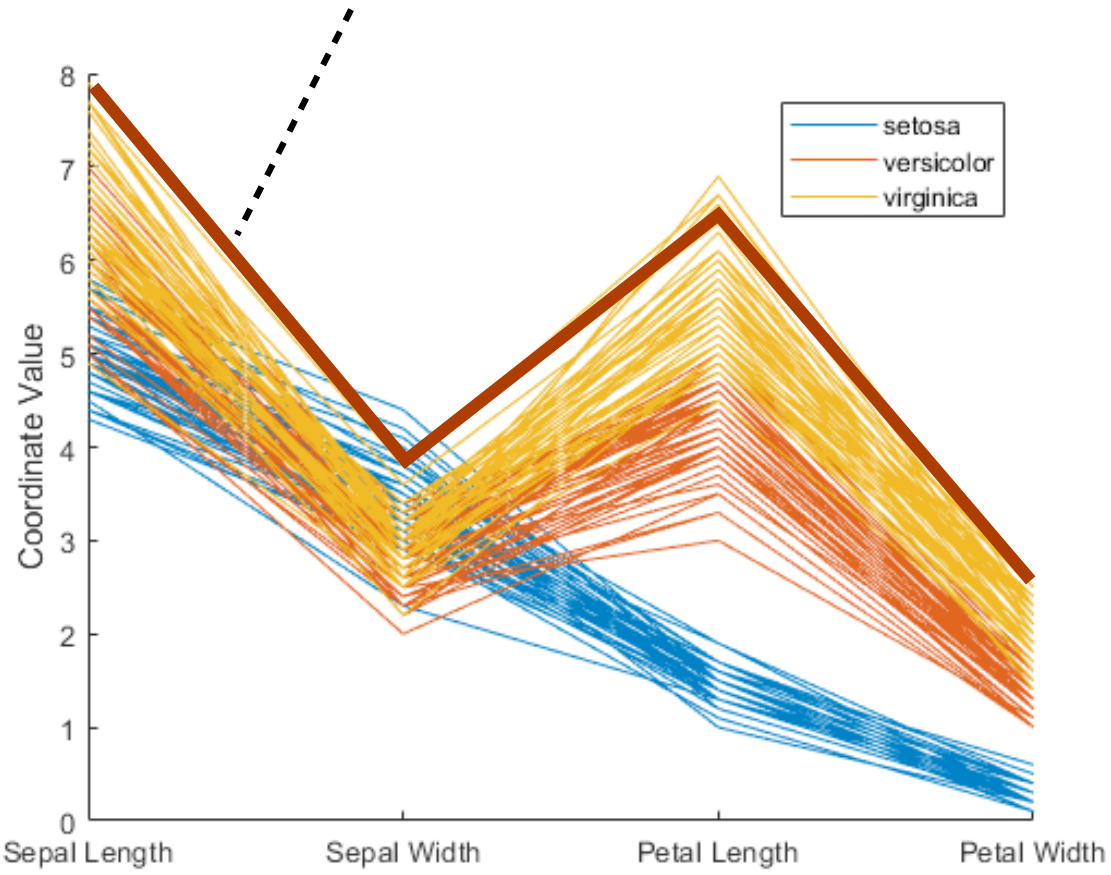


https://nl.mathworks.com/help/stats/parallelcoords.html

parallel coordinates

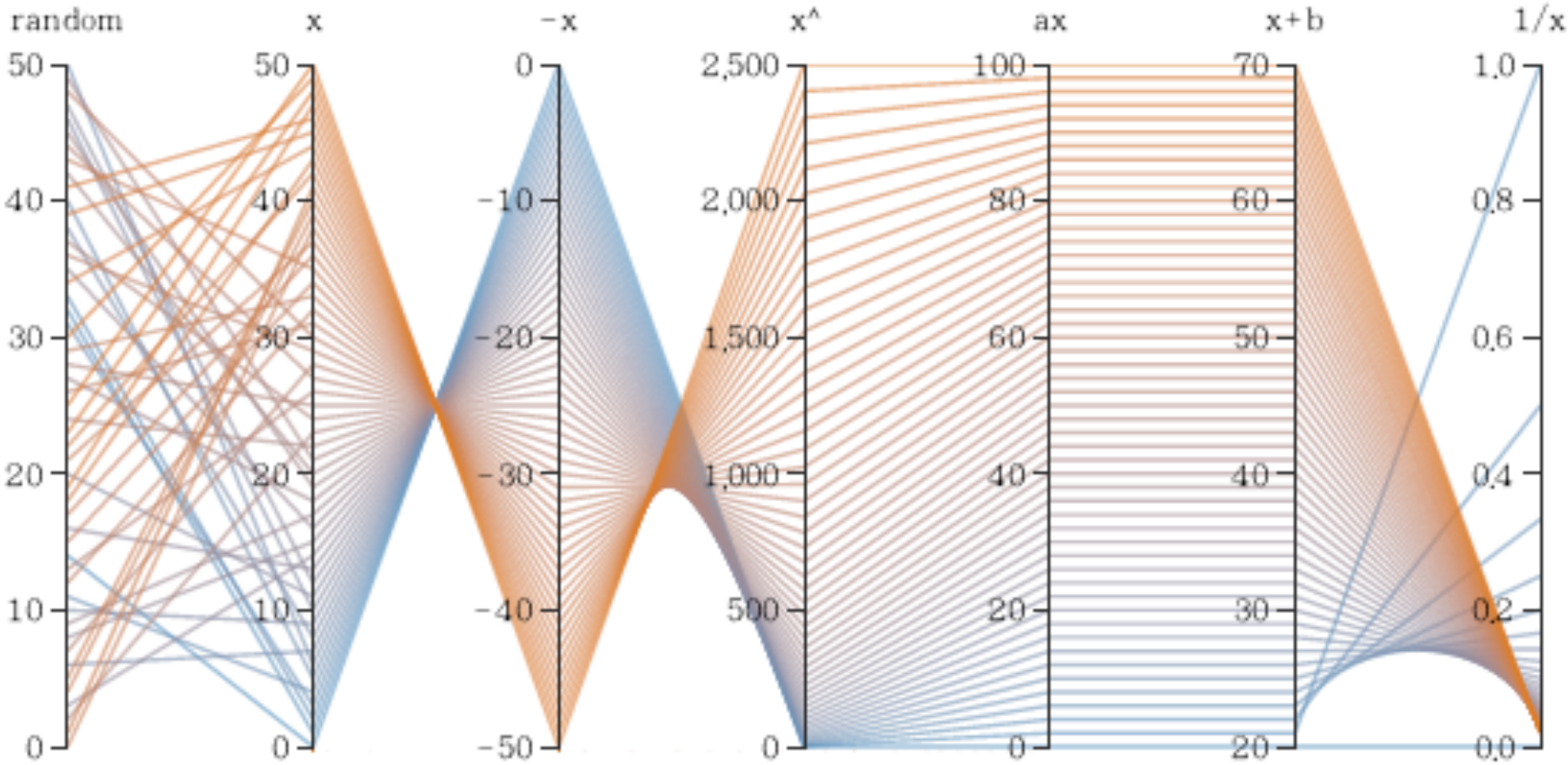
- show data points in nD space
- each "polyline" represents one data point
- can highlight clusters (e.g. compare setosa and versicolor)
- often choice of hoizontal ordering (e.g. sepal length could be swapped with petal width etc.)
- can get cluttered if too many dimensions and categories

each polyline is one data point



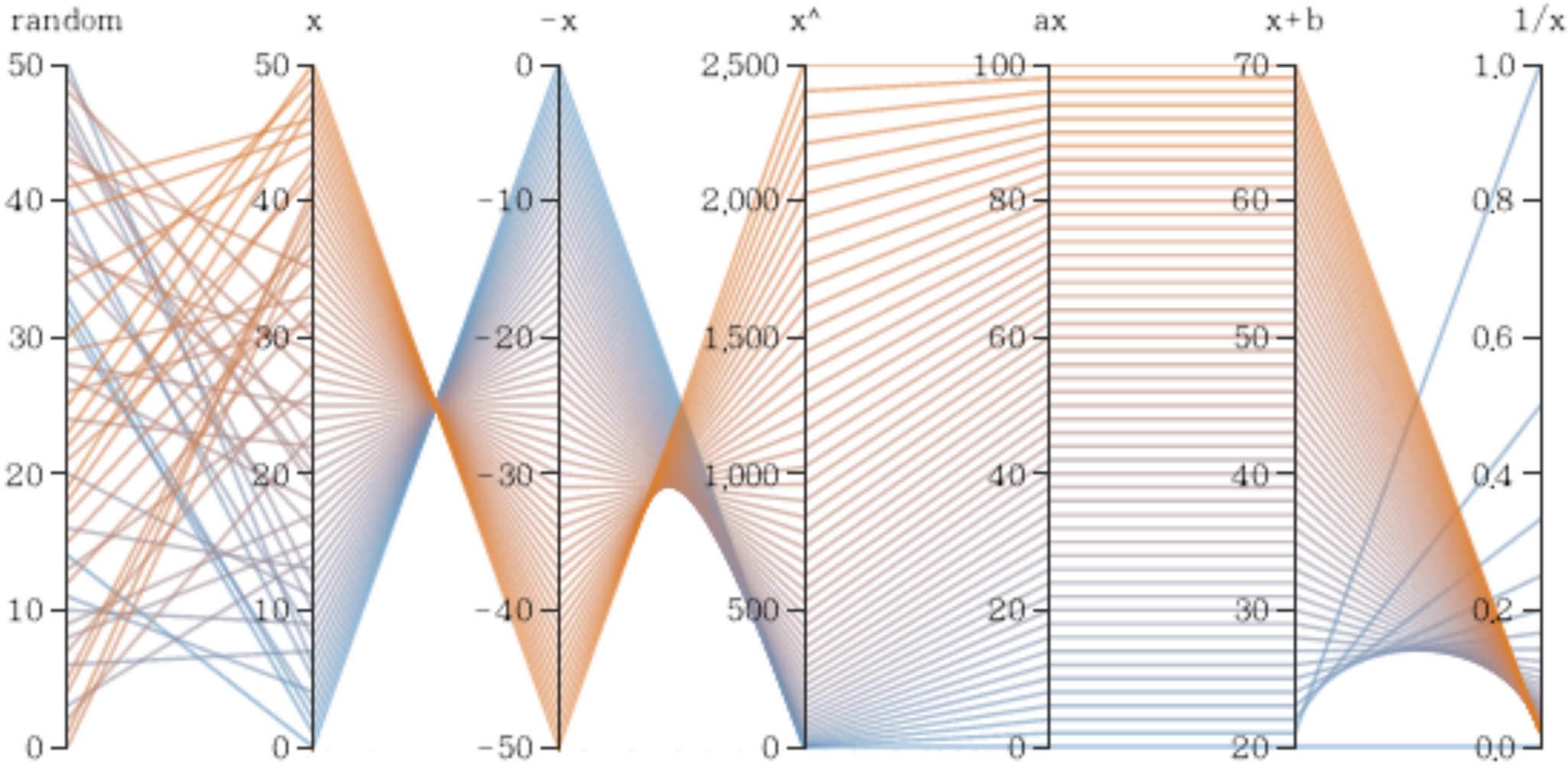
nD
(here 4 dimensions)

“signature” visual patterns corresponding to patterns in data



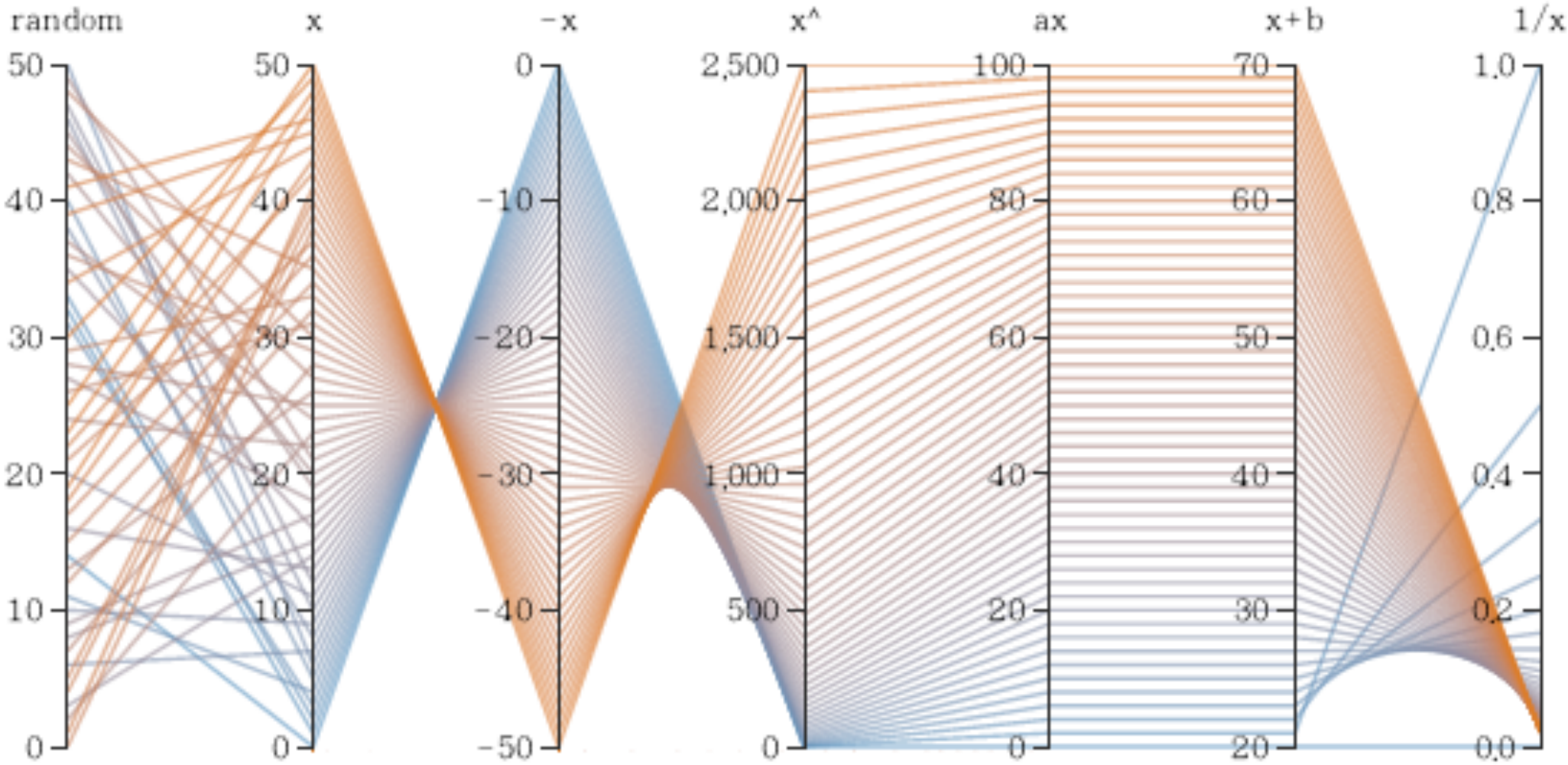
https://en.wikipedia.org/wiki/Parallel_coordinates

"signature" visual patterns corresponding to patterns in data



if you see this,
variables could be independent

“signature” visual patterns corresponding to patterns in data



if you see this,
variables could be inversely linearly proportional

SUMMARY Part 2. visualisation tools

- axes, colour, shape, size, labels, animation
- scatterplot
  scatterplot matrix
  parallel coordinates

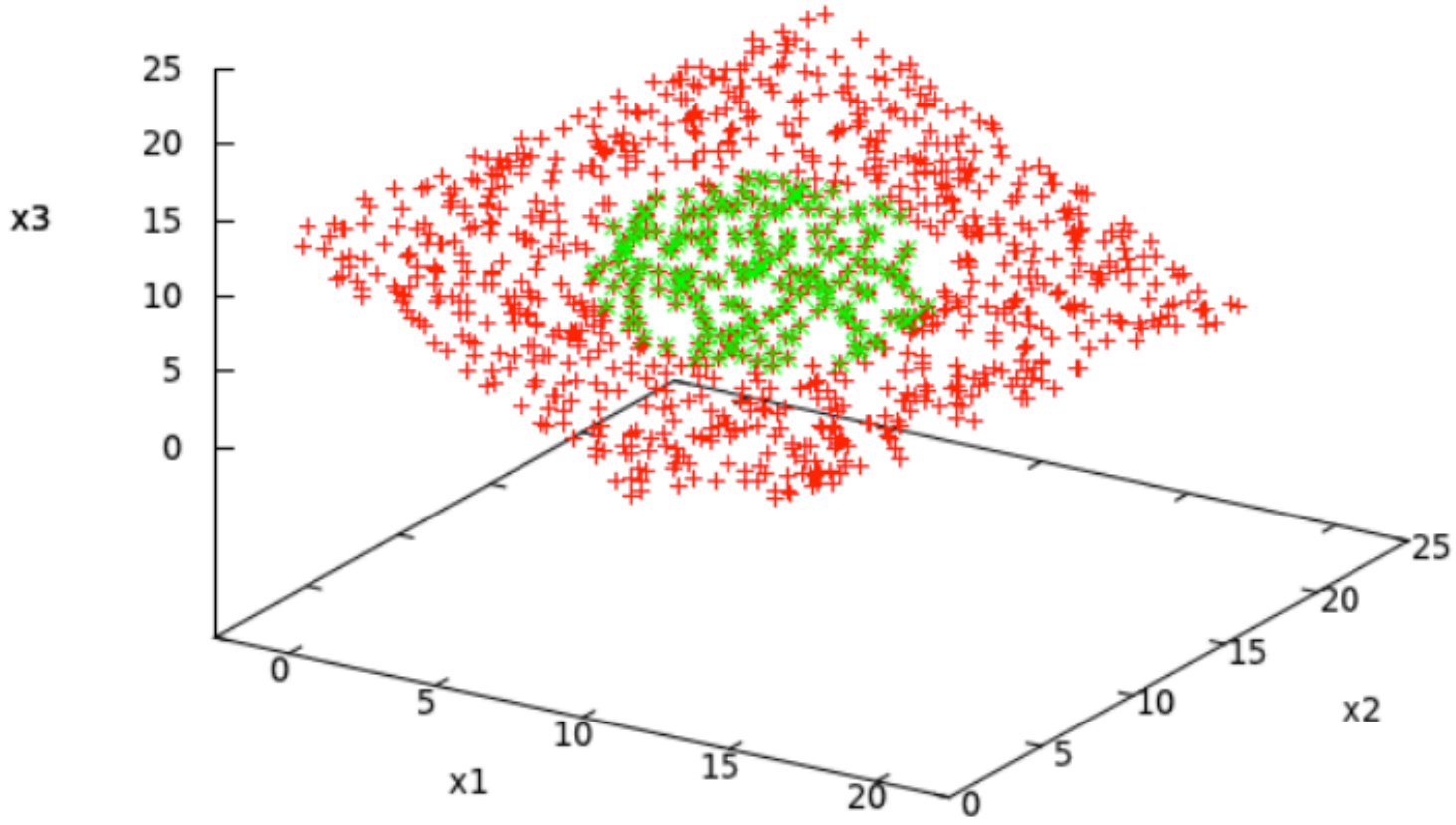
Part 3

dimensionality reduction

dimensionality reduction

- the challenge - our data has too many dimensions to visualise
- often what we need to see is "distribution" of data across dimensions - which data points are "near" and which are "far" with respect to certain dimensions
- perhaps some dimensions can be "combined" because they don't really help to distinguish data

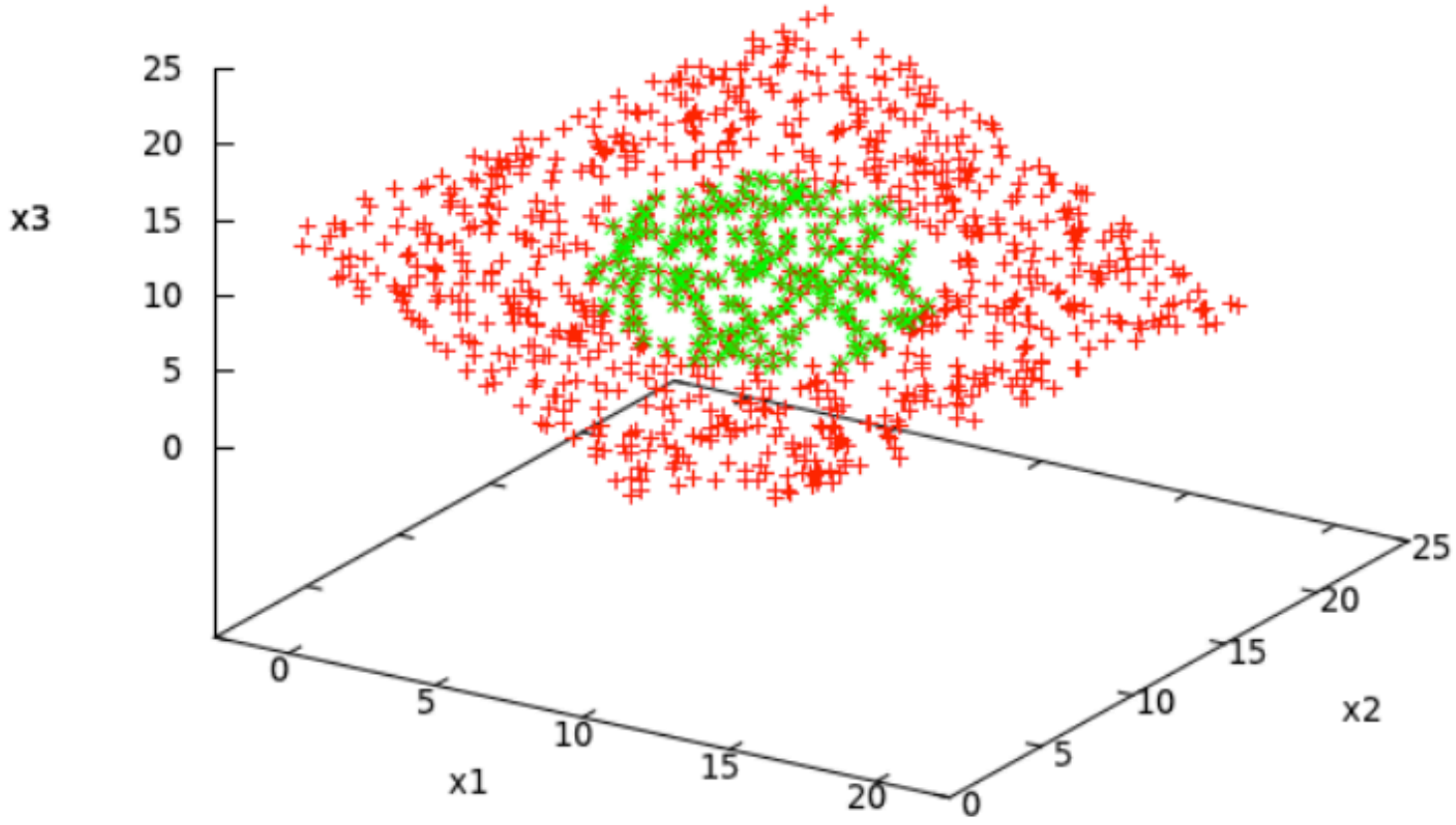
principle component analysis



http://nghiaho.com/?page_id=1030

principle component analysis

in this example, data sits on a 3D plane, tilted 45°

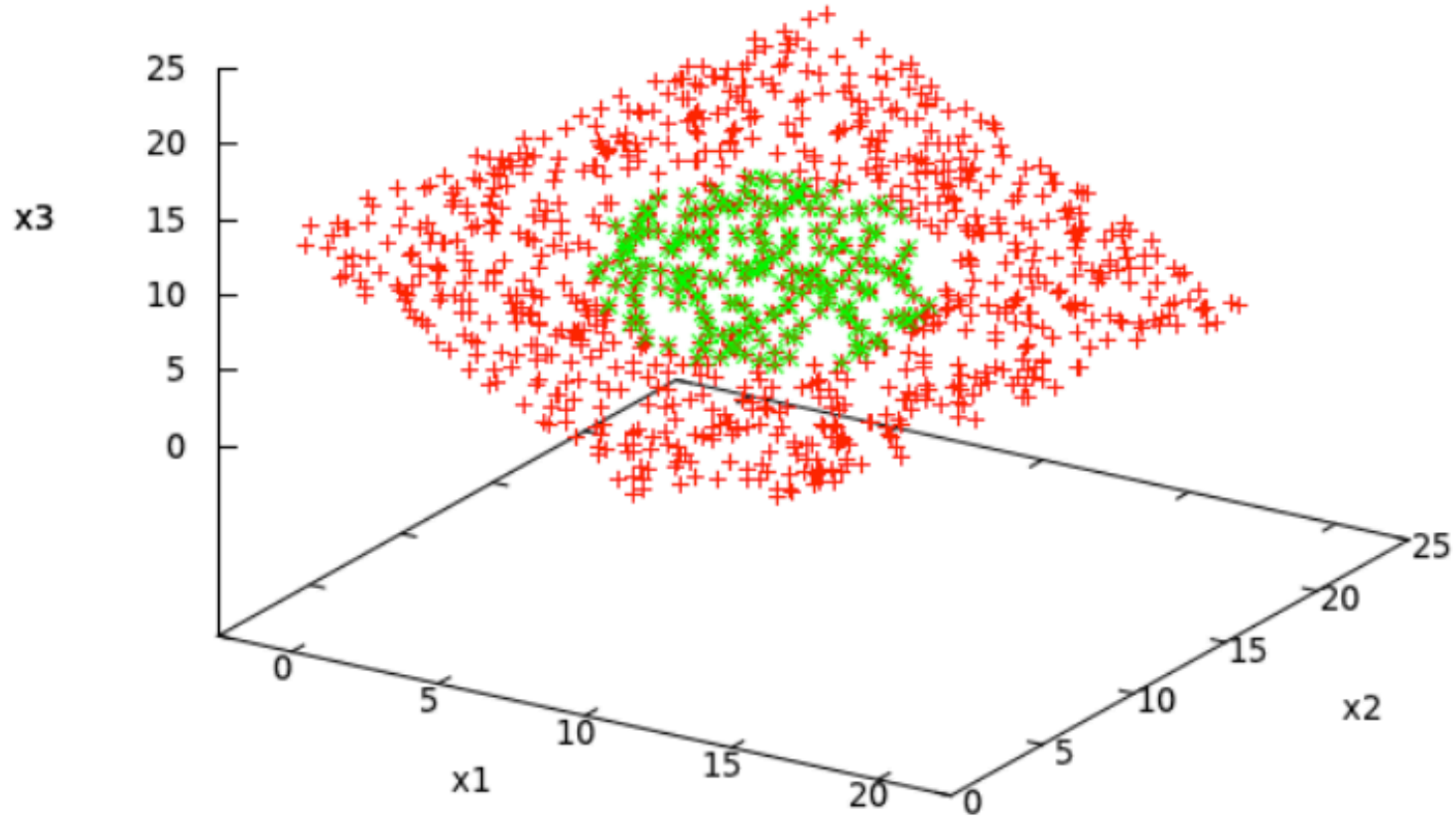


http://nghiaho.com/?page_id=1030

principle component analysis

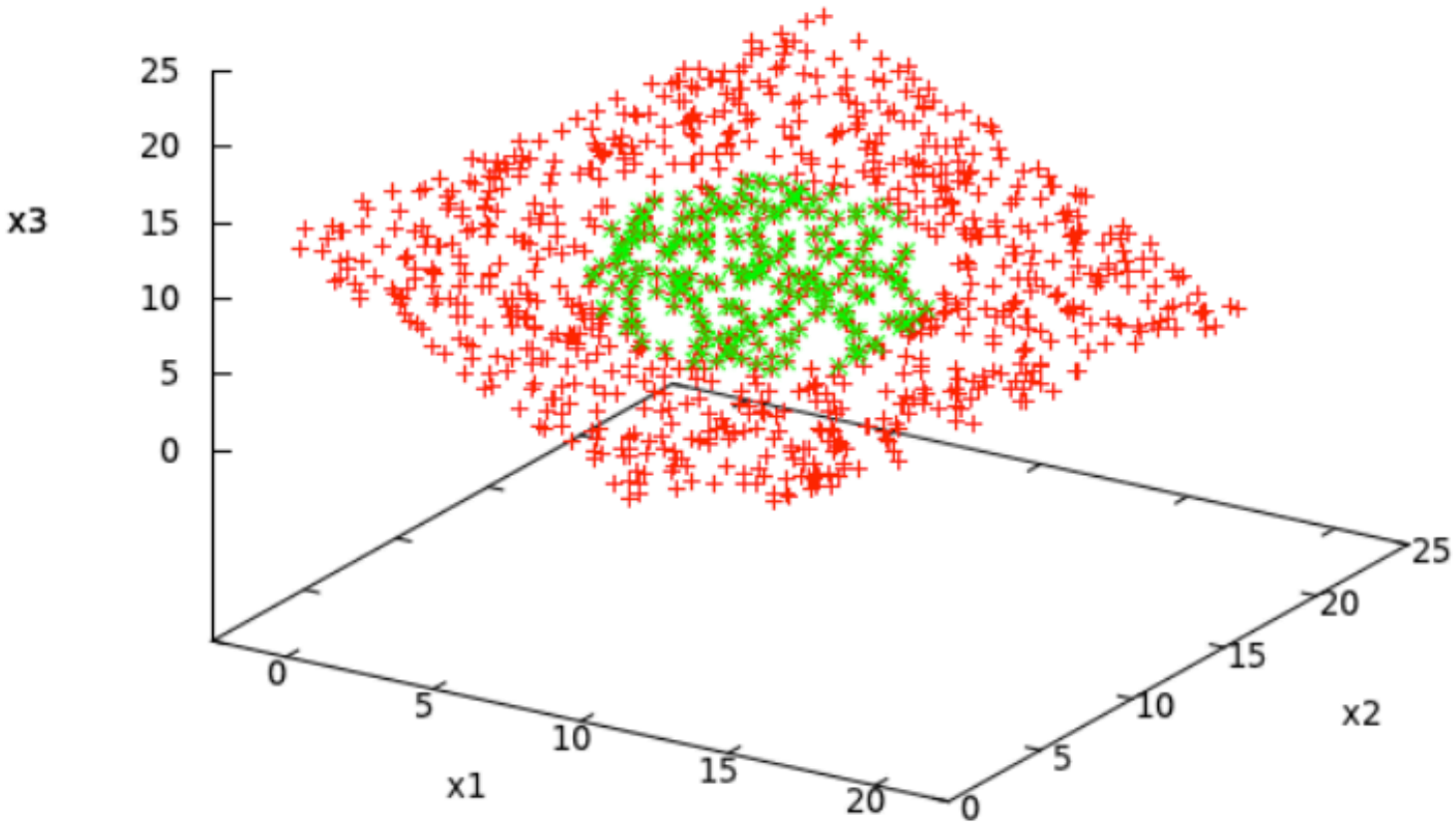
want to view in 2D

want to see as much **spread/variance** as possible



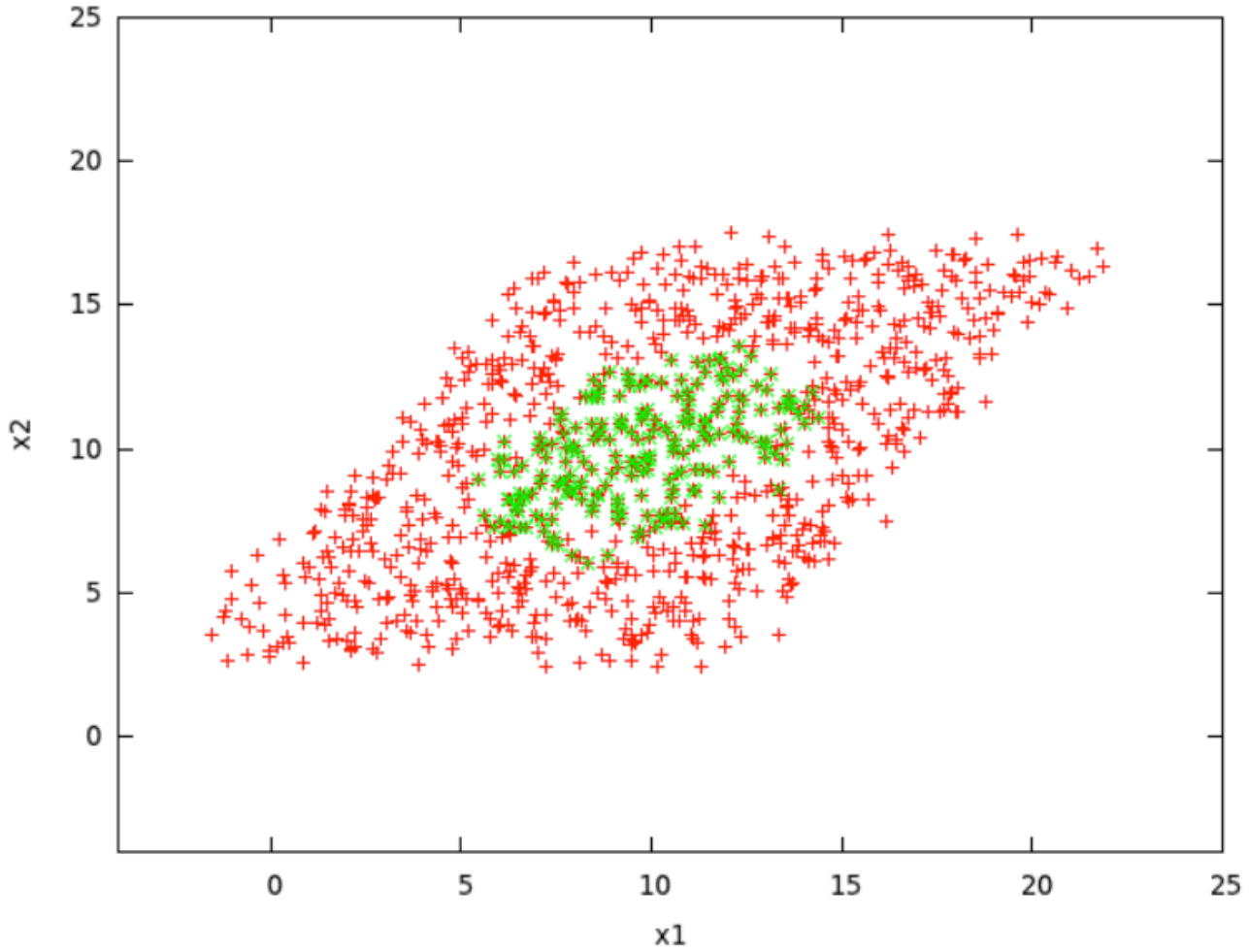
http://nghiaho.com/?page_id=1030

let's look at this top-down



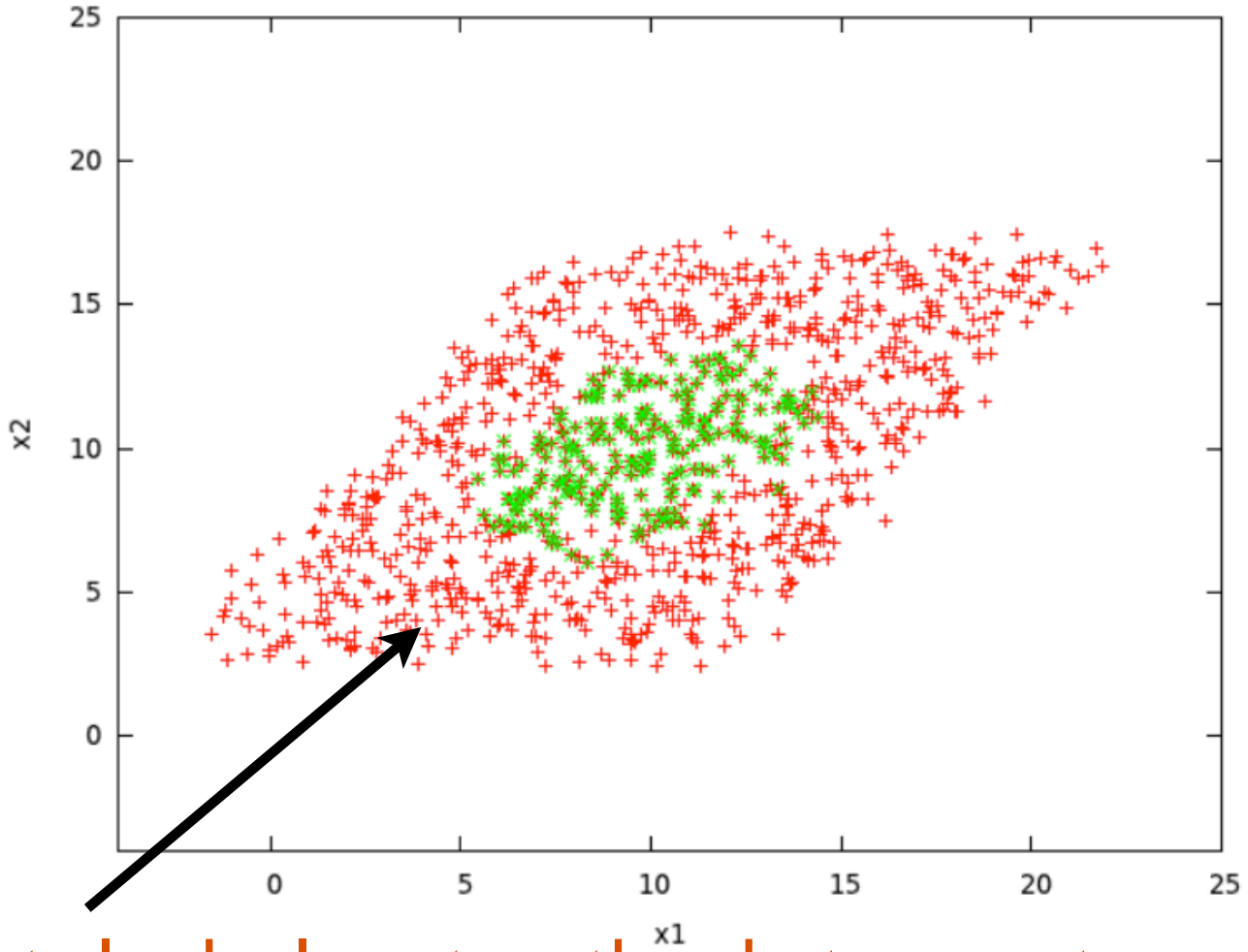
http://nghiaho.com/?page_id=1030

not bad, can see data clearly, but it's all skewed
(other axis projections also look like this)



http://nghiaho.com/?page_id=1030

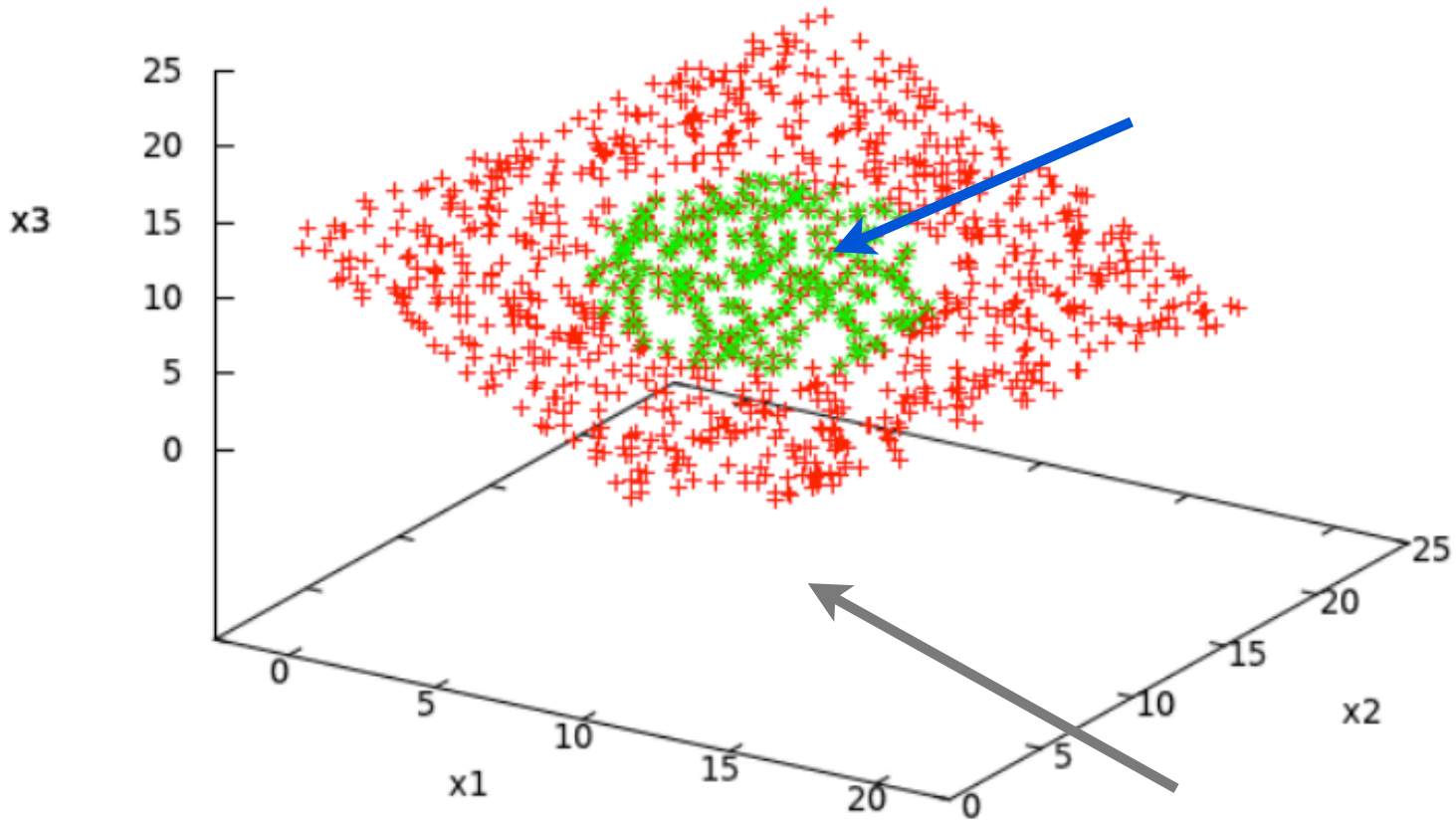
not bad, can see data clearly, but it's all skewed
(other axis projections also look like this)



some points look close together, but are not
really close in the 3D dataset

http://nghiaho.com/?page_id=1030

best view: perpendicular to this plane the data sits on

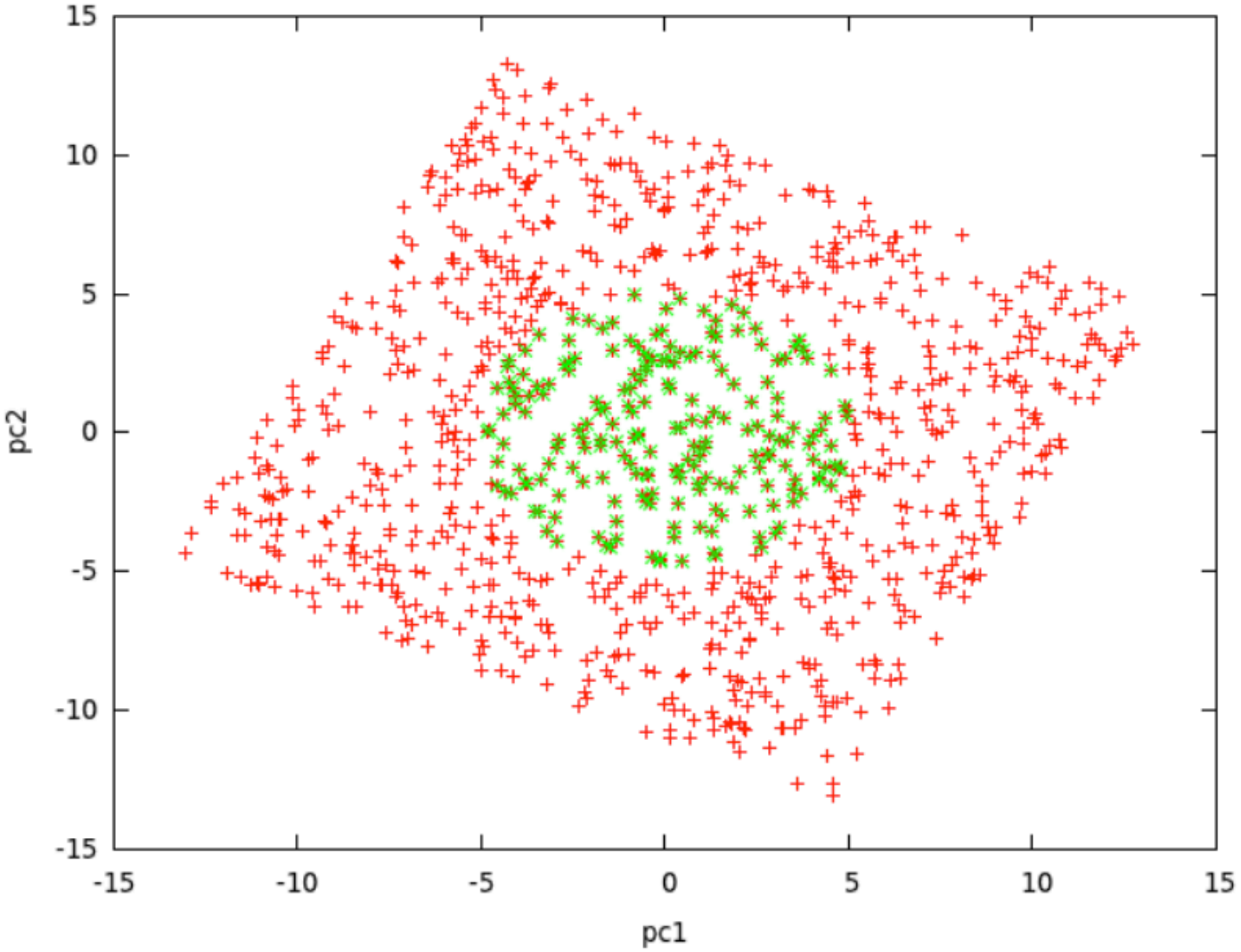


(this is just the shadow of arrow,
to get a 3D feeling)

http://nghiaho.com/?page_id=1030

principle component analysis (PCA)

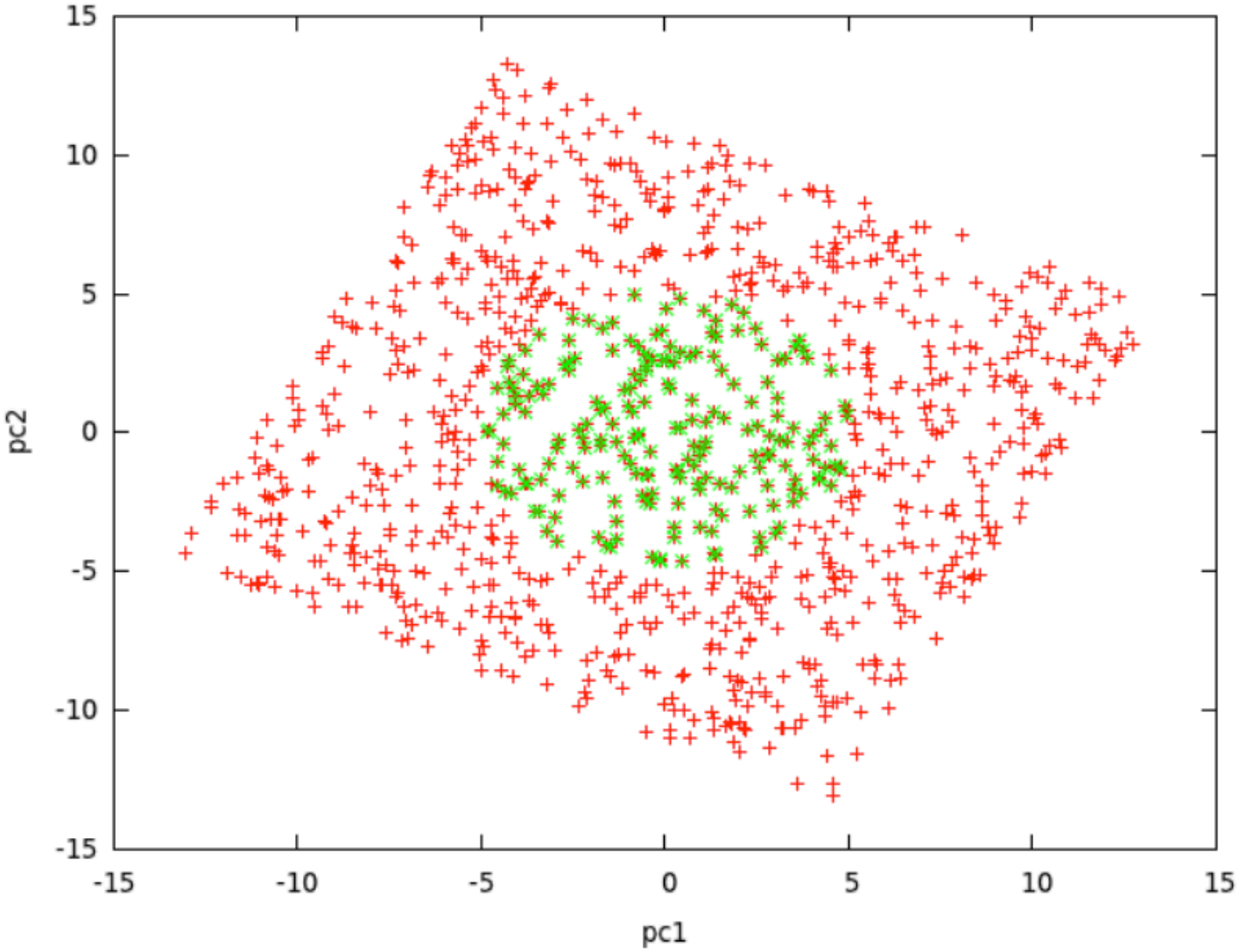
rotates data so we "see" the most spread/variance



http://nghiaho.com/?page_id=1030

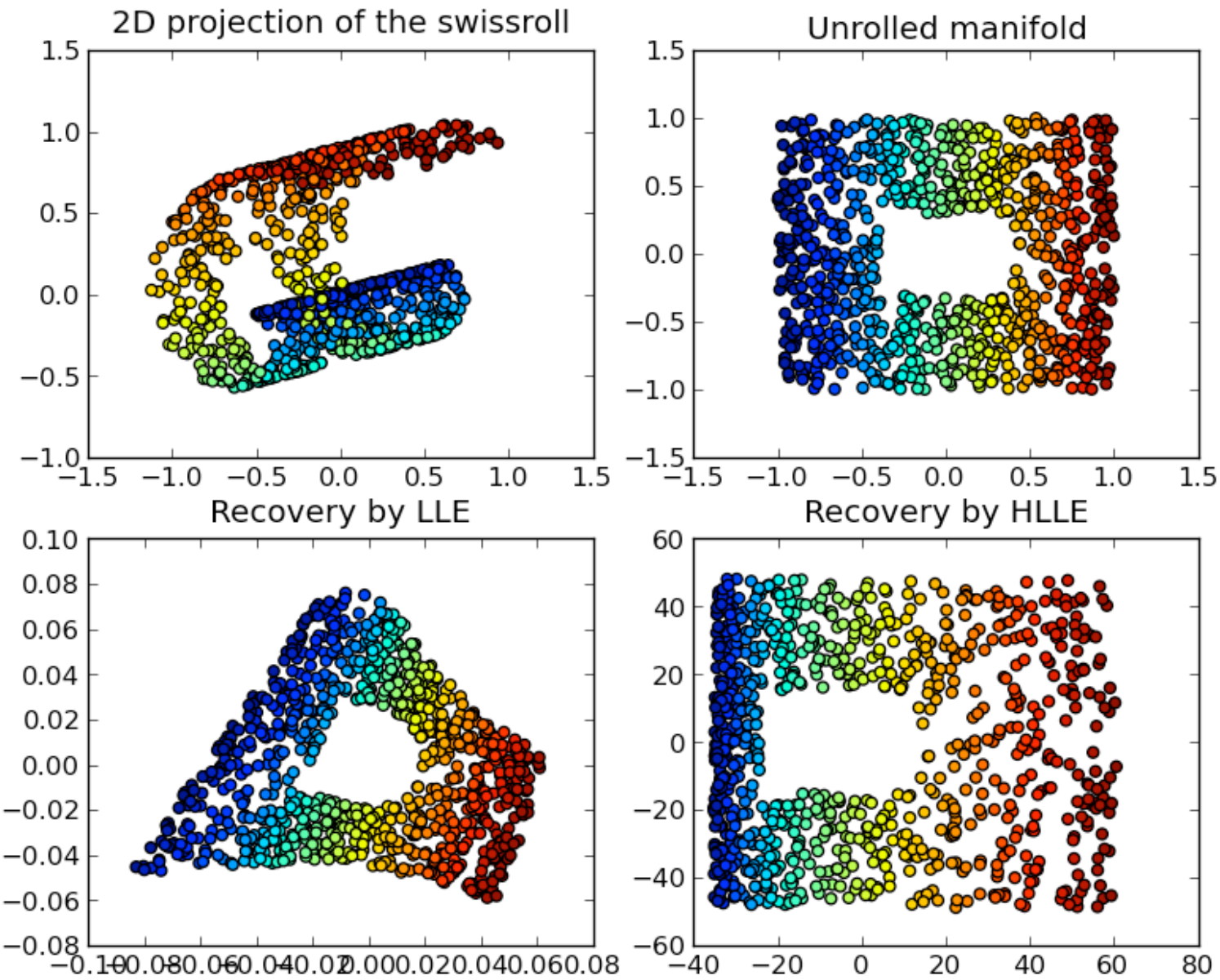
principle component analysis (PCA)

same idea can be applied to nD - keep rotating and projecting until we get down to 2D



http://nghiaho.com/?page_id=1030

non-linear dimensionality reduction



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

some examples from the wild

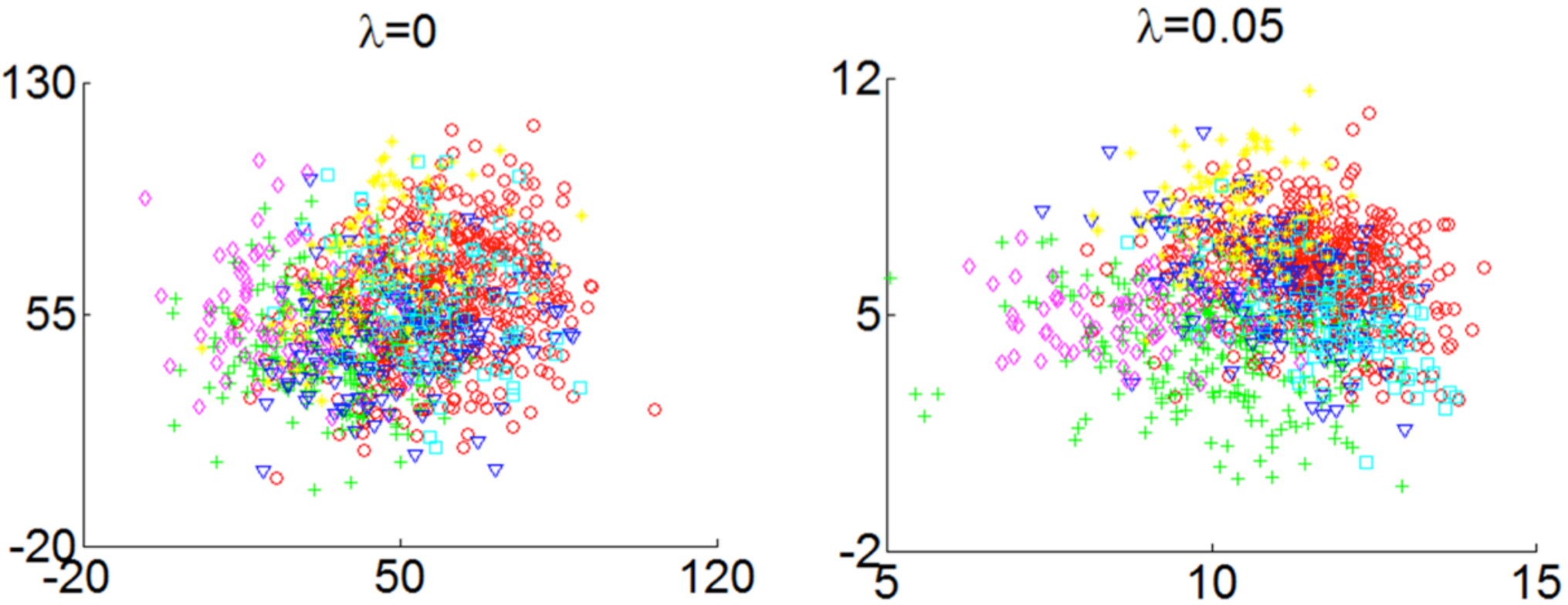


Figure 6: The first two PCA dimensions of DeepID2 features extracted from six identities in LFW.

Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. In Advances in neural information processing systems (pp. 1988-1996).

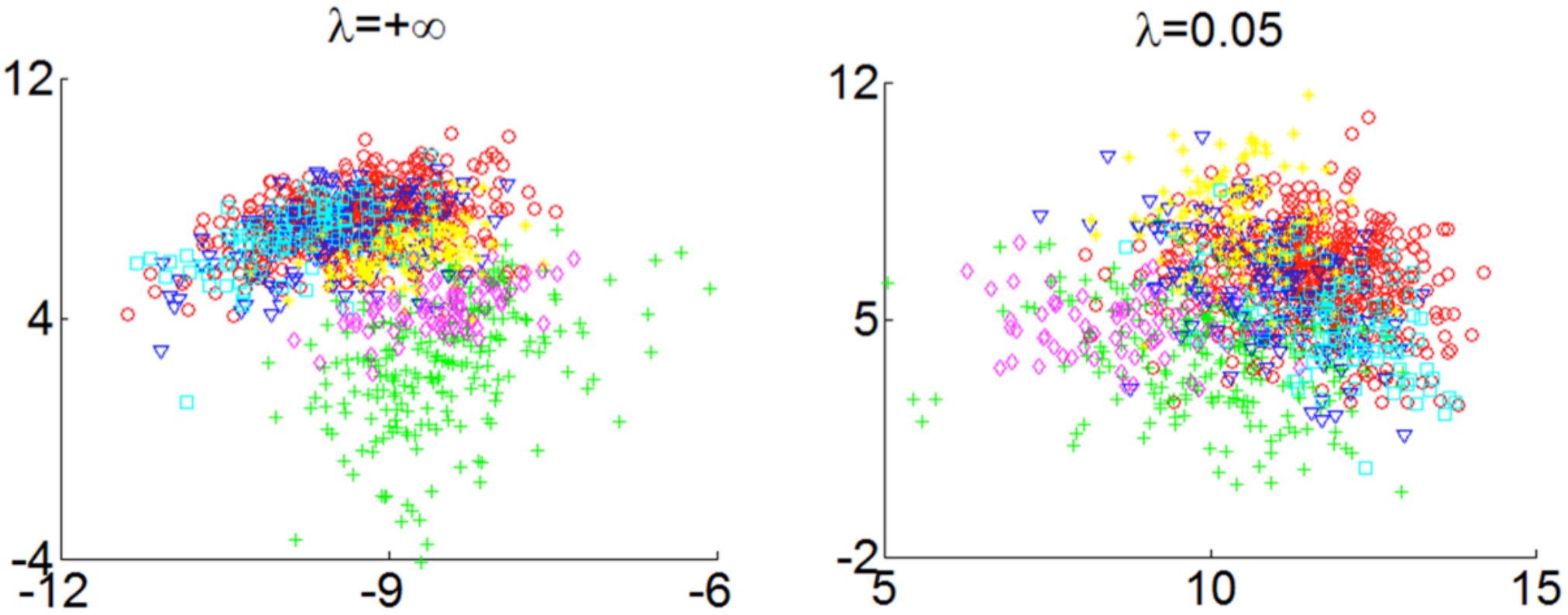


Figure 6: The first two PCA dimensions of DeepID2 features extracted from six identities in LFW.

Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. In Advances in neural information processing systems (pp. 1988-1996).

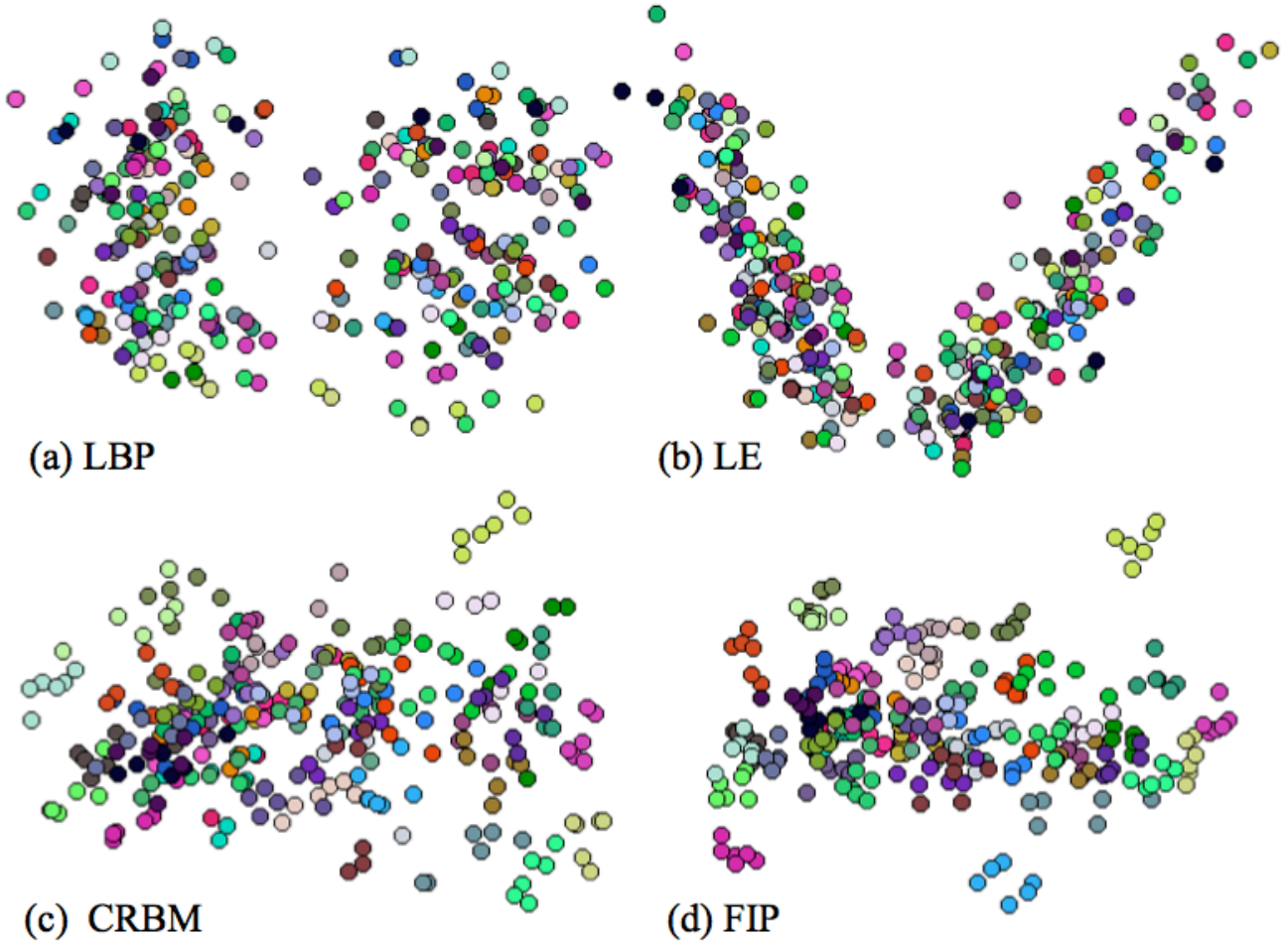


Figure 2. The LBP (a), LE (b), CRBM (c), and FIP (d) features of 50 identities, each of which has 6 images in different poses and illuminations are projected into two dimensions using Multidimensional scaling (MDS). Images of the same identity are visualized in the same color. It shows that FIP has the best representative power. **Best viewed in color.**

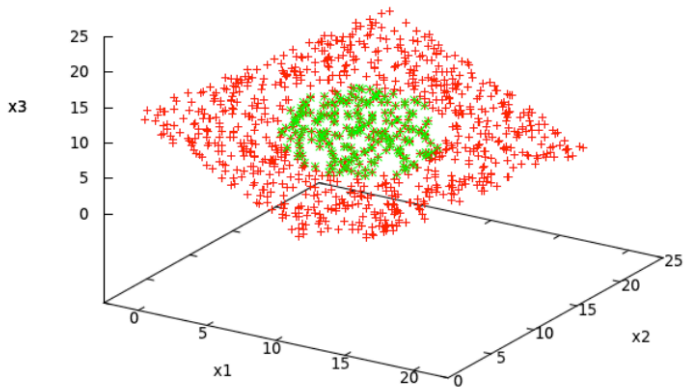
Zhu, Z., Luo, P., Wang, X., & Tang, X. (2013). Deep learning identity-preserving face space. In Proceedings of the IEEE International Conference on Computer Vision (pp. 113-120).

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

SUMMARY Part 3. dimensionality reduction

- too many dimensions!
- linear reduction (PCA),
  non-linear reduction
- examples from the wild

TODAY'S SUMMARY



quick recap

1. why, and what, are we visualising?

2. tools for visualisation

3. dimensionality reduction