CS-E4840
Information Visualization
Lecture 8b

Tassu Takala <tapio.takala@aalto.fi>

25 March 2019

PART III
Dimensionality Reduction

Literature on dimensionality reduction for visualisation

- MDS: Borg, Kroenen, Modern multidimensional scaling: theory and applications. Springer, 1997.
- PCA: any book on matrix algebra.
- Jarkko Venna 2007, Academic Dissertation, http://lib.tkk.fi/Diss/2007/isbn9789512287529/
- Lee & Verleysen, 2007. Nonlinear dimensionality reduction. Springer.
- For a reasonably recent brief review see Verleysen & Lee, 2013. https://doi.org/10.1007/978-3-642-42054-2_77
- (Not to be confused with dimensionality reduction for machine learning where target dimensionality is often higher!)
- See http://www.iki.fi/kaip/p/dimensionality_reduction_1.nb.html

Dimensionality reduction

- Assume you have n m-dimensional **data points**
- Further assume that we can define a *meaningful* **distance** p_{ij} between data points i and j
- Assume dimensionality m is so large that a data point cannot be visualised by "traditional" methods

- *Problem statement:* Given a dimensionality k (typically k=2 or k=3), find an embedding X of data points into k-dimensional space (=locations of data points) such that the Euclidean distance between data points i and j $d_{ij}(X)$ in the embedding matches (a function of) the original distance p_{ij} as well as possible.
- See http://www.iki.fi/kaip/p/dimensionality_reduction_1.nb.html

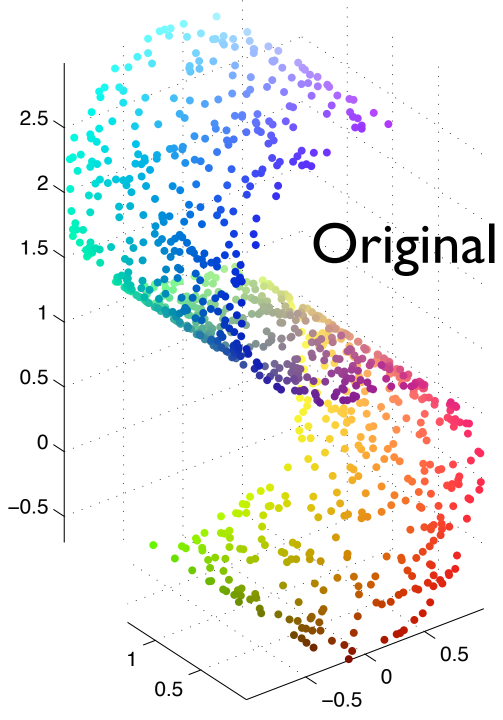
Dimensionality reduction

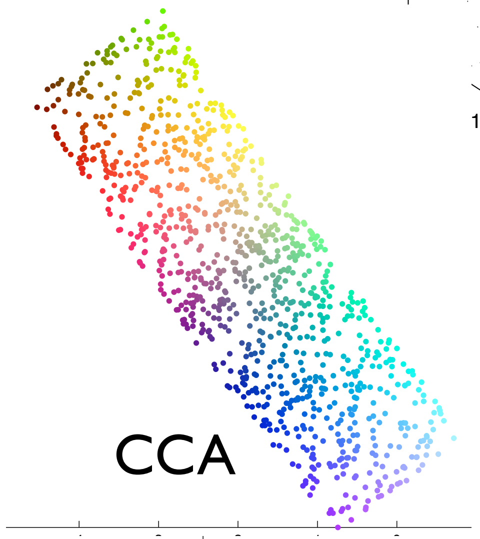
- *Problem statement:* Find an embedding X of (n-D) data points into k-dimensional (k=2..3) space such that the distances between data points in the embedding match those between corresponding original points as well as possible.

- What does "as well as possible" mean?
  - Long distances?
  - Short distances?
  - Neighbourhood relations?
- All embeddings have to make compromises. We will first study embeddings that preserve long distances.
- See http://www.iki.fi/kaip/p/dimensionality_reduction_1.nb.html

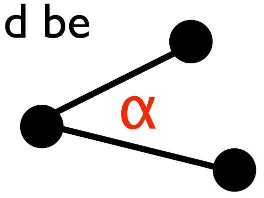
Dimensionality reduction

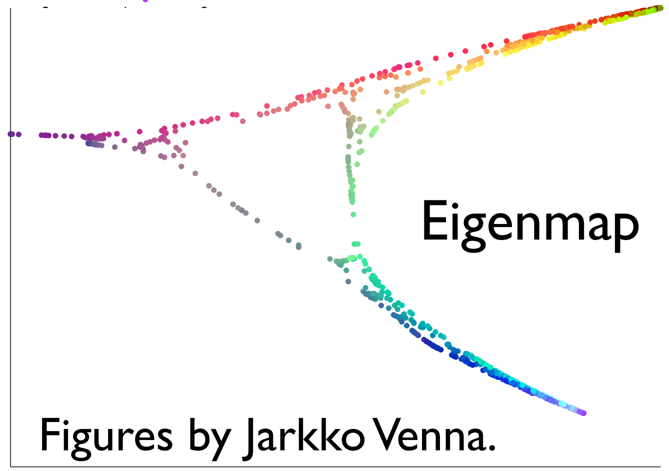
- Goal: project the data into a low-dimensional (1-3D) space, while maintaining the correct (visual perception of) relations between the nodes
- Obviously, in a general case, some information will be inevitably lost in the projection (e.g., there is a trade-off between precision and recall)
- There are several methods, with varying optimisation goals and complexities
- Some optimisation goals:
  - If the nodes are nearby in the original representation, they should also be nearby in the projection (***recall***, sometimes called continuity, or *preservation of the original neighborhoods*).
  - If the nodes are nearby in the projection, they should also be nearby in the original representation (***precision***, sometimes called *trustworthiness*).
  - The (global) distances between nodes should be preserved as well as possible.
  - Angles between nearby nodes should be preserved as well as possible (*conformality*).





RGB color coding is used to help to match the items between representations.





Figures by Jarkko Venna.

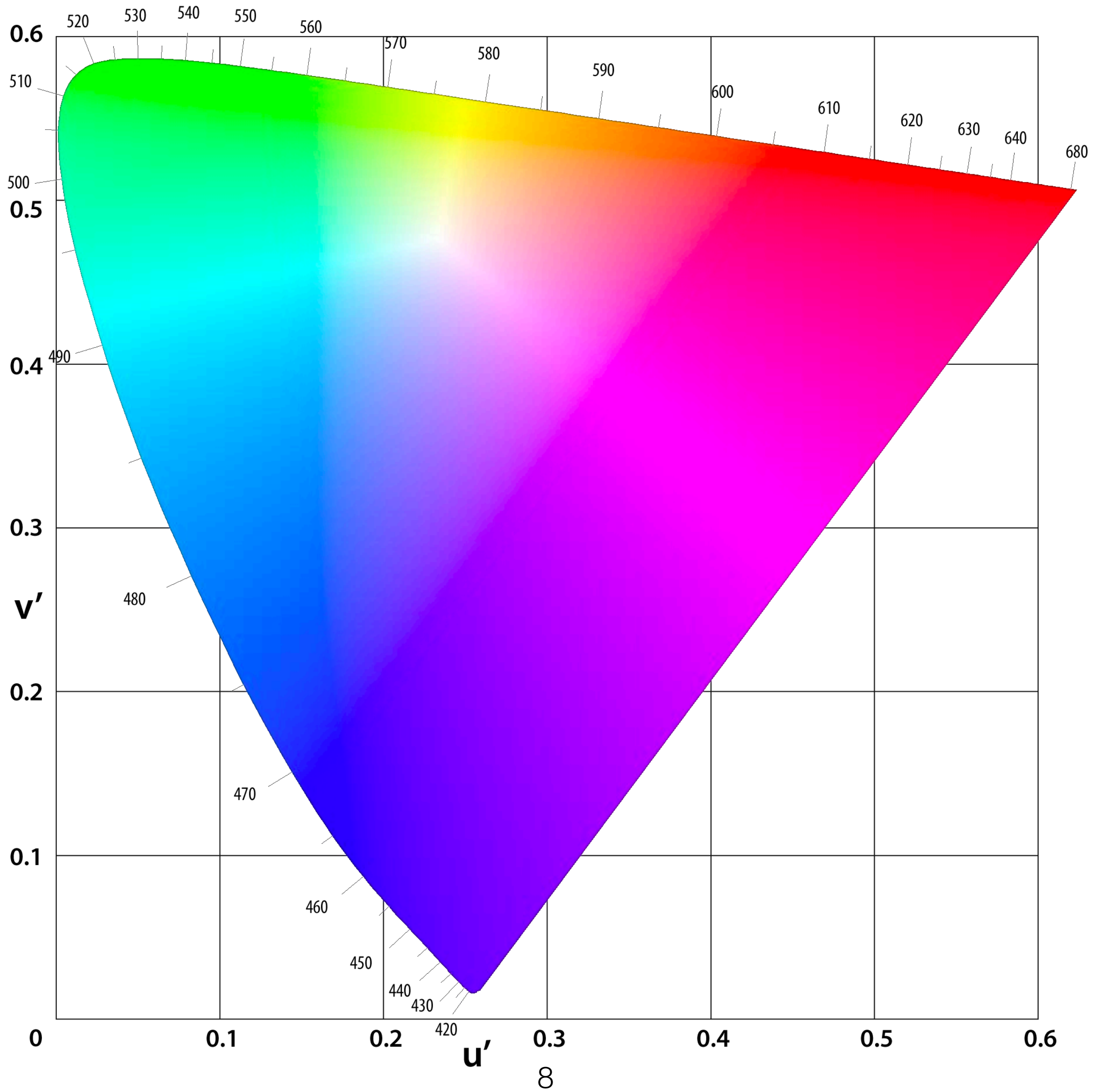
Dimensionality reduction

- *Problem statement:* Given a dimensionality k (typically k=2 or k=3), find an embedding X of data points into k-dimensional space such that the Euclidean distance between data points i and j $d_{ij}(X)$ in the embedding matches (a function of) the original distance p_{ij} as well as possible.

- Today, we will review methods that try to preserve **long distances** as well as possible:
  - Metric multidimensional scaling (MDS)
  - Nonmetric MDS
  - Sammon mapping
  - Principal component analysis (PCA)
  - Independent component analysis (ICA)

- [The alternative is to preserve short distances (neighborhoods), leading to manifold embeddings]

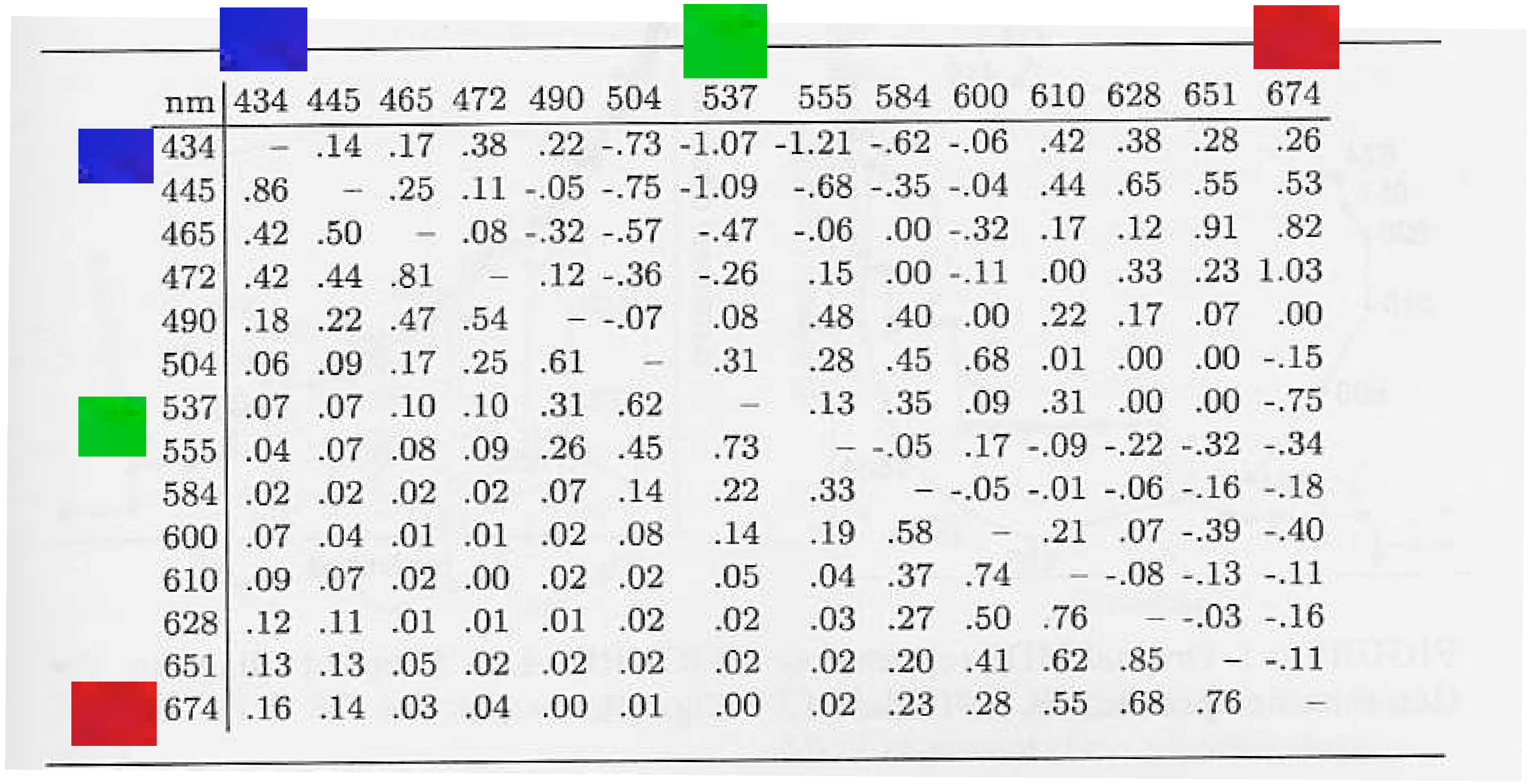
Example: colours



CIELUV

Example: colours

- How is the similarity of colors perceived?
- Pairs of 14 colors were rated by 31 people. Ratings were averaged (Ekman 1954, https://doi.org/10.1080/00223980.1954.9712953).



nm	434	445	465	472	490	504	537	555	584	600	610	628	651	674
434	–	.14	.17	.38	.22	-.73	-1.07	-1.21	-.62	-.06	.42	.38	.28	.26
445	.86	–	.25	.11	-.05	-.75	-1.09	-.68	-.35	-.04	.44	.65	.55	.53
465	.42	.50	–	.08	-.32	-.57	-.47	-.06	.00	-.32	.17	.12	.91	.82
472	.42	.44	.81	–	.12	-.36	-.26	.15	.00	-.11	.00	.33	.23	1.03
490	.18	.22	.47	.54	–	-.07	.08	.48	.40	.00	.22	.17	.07	.00
504	.06	.09	.17	.25	.61	–	.31	.28	.45	.68	.01	.00	.00	-.15
537	.07	.07	.10	.10	.31	.62	–	.13	.35	.09	.31	.00	.00	-.75
555	.04	.07	.08	.09	.26	.45	.73	–	-.05	.17	-.09	-.22	-.32	-.34
584	.02	.02	.02	.02	.07	.14	.22	.33	–	-.05	-.01	-.06	-.16	-.18
600	.07	.04	.01	.01	.02	.08	.14	.19	.58	–	.21	.07	-.39	-.40
610	.09	.07	.02	.00	.02	.02	.05	.04	.37	.74	–	-.08	-.13	-.11
628	.12	.11	.01	.01	.01	.02	.02	.03	.27	.50	.76	–	-.03	-.16
651	.13	.13	.05	.02	.02	.02	.02	.02	.20	.41	.62	.85	–	-.11
674	.16	.14	.03	.04	.00	.01	.00	.02	.23	.28	.55	.68	.76	–

Similarities of colors with different wavelengths (lower half, Ekman 1954) and residuals of 1D MDS representation (upper half) [B 4.1].

Multidimensional scaling (MDS)

- Formally, an MDS algorithm is given as input the original distances p_{ij} (called ***proximities***) between data points i and j
- MDS algorithm then tries to find a k-dimensional (usually k=2 or k=3) representation X for the points
- MDS tries to find representation X that minimises the error function (called ***stress***, by convention)

$$\sigma_r = \sum_{i<j} \left(f(p_{ij}) - d_{ij}(X)\right)^2$$

- where $d_{ij}(X)$ is the Euclidean distance between the data points *i* and *j* in representation X and *f* is a function that defines the MDS model (next slide).

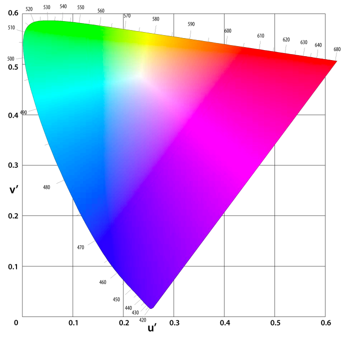
Multidimensional scaling (MDS)

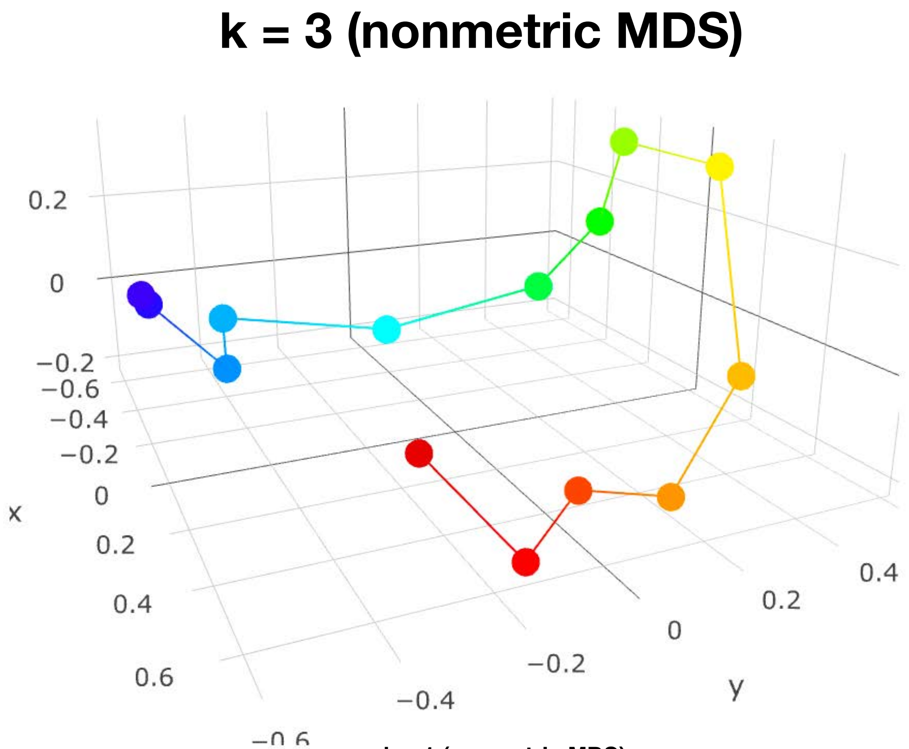
$$\sigma_r = \sum_{i<j} \left(f(p_{ij}) - d_{ij}(X)\right)^2$$

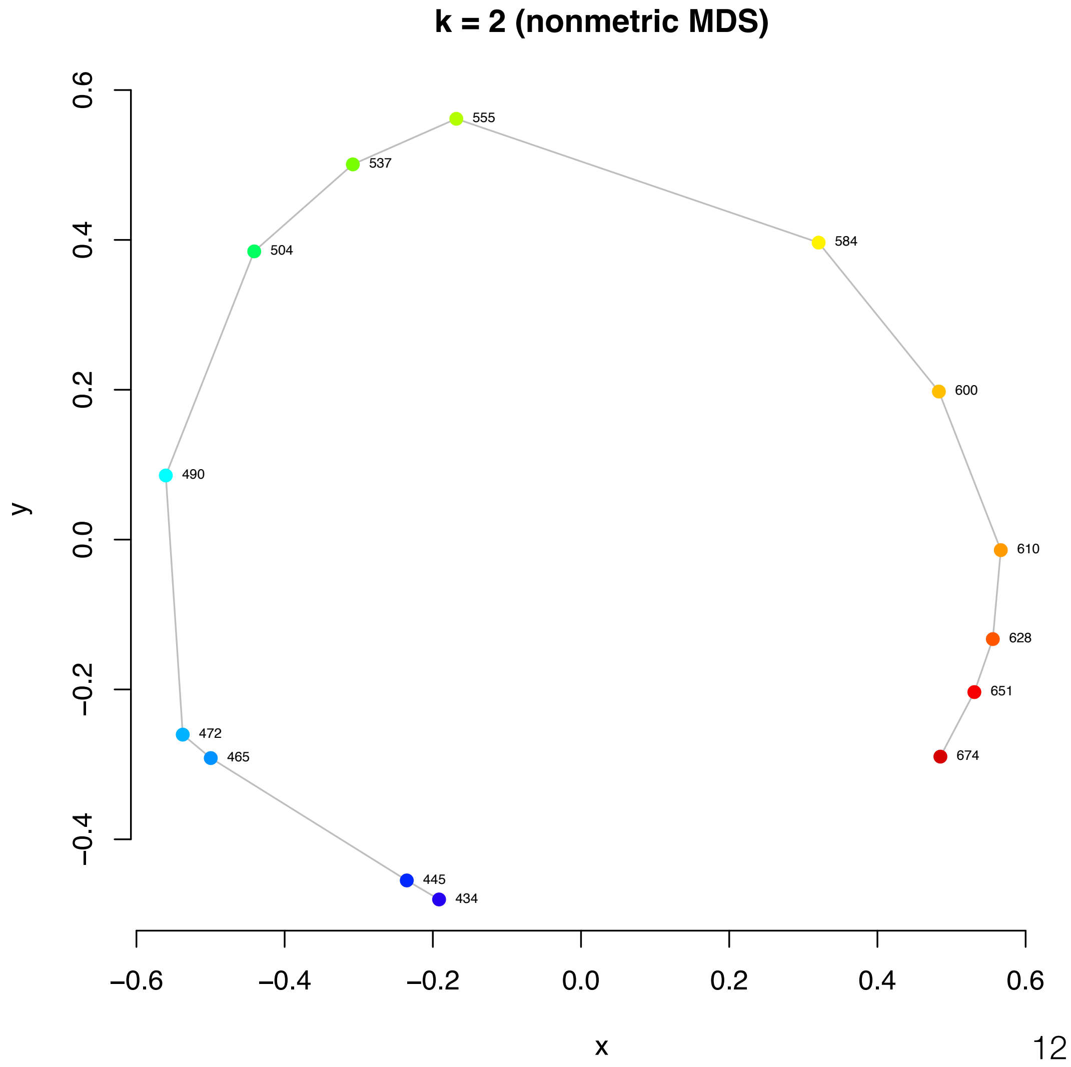
- The choice of *f* defines the MDS model. For example:
  - $f(p_{ij})=p_{ij}$ - absolute MDS (linear model)
  - $f(p_{ij})=b\ p_{ij}$ - ratio MDS (linear model)
  - $f(p_{ij})=a+b\ p_{ij}$ - interval MDS (linear model)
  - $f(p_{ij})=a+b \log p_{ij}$ - useful in psychology (logarithmic)
  - $f(p_{ij})$ can be any monotonically increasing function (ordinal or **nonmetric MDS**)
    - this would be the most important special case of MDS

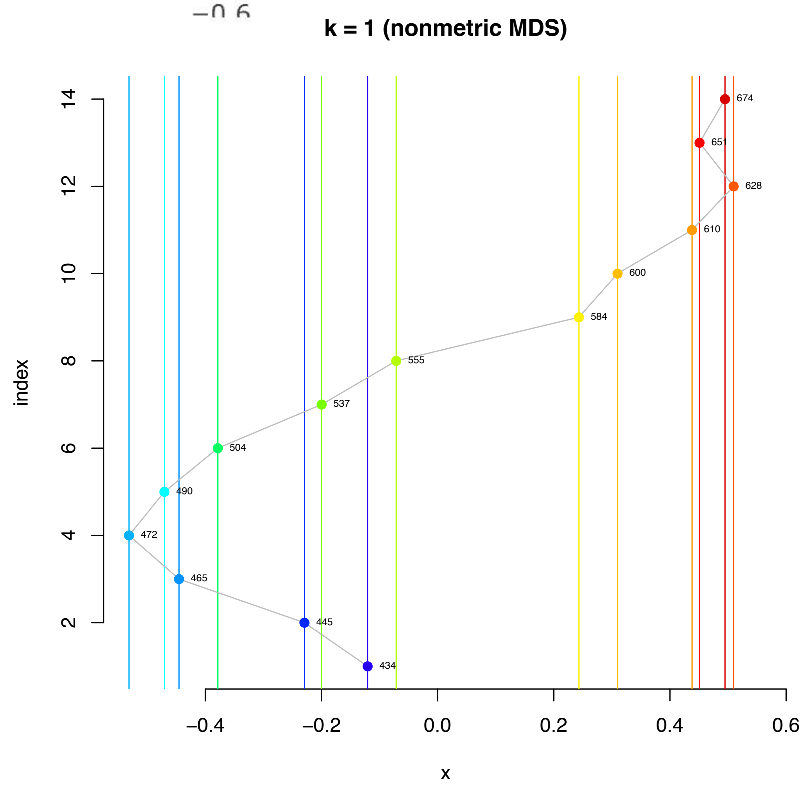
- The parameters of f (such as a and b above) are optimised at the same time as the representation X (i.e., the locations of the projected points)
    - details of the optimisation algorithms is outside the scope of this course

Example: colour

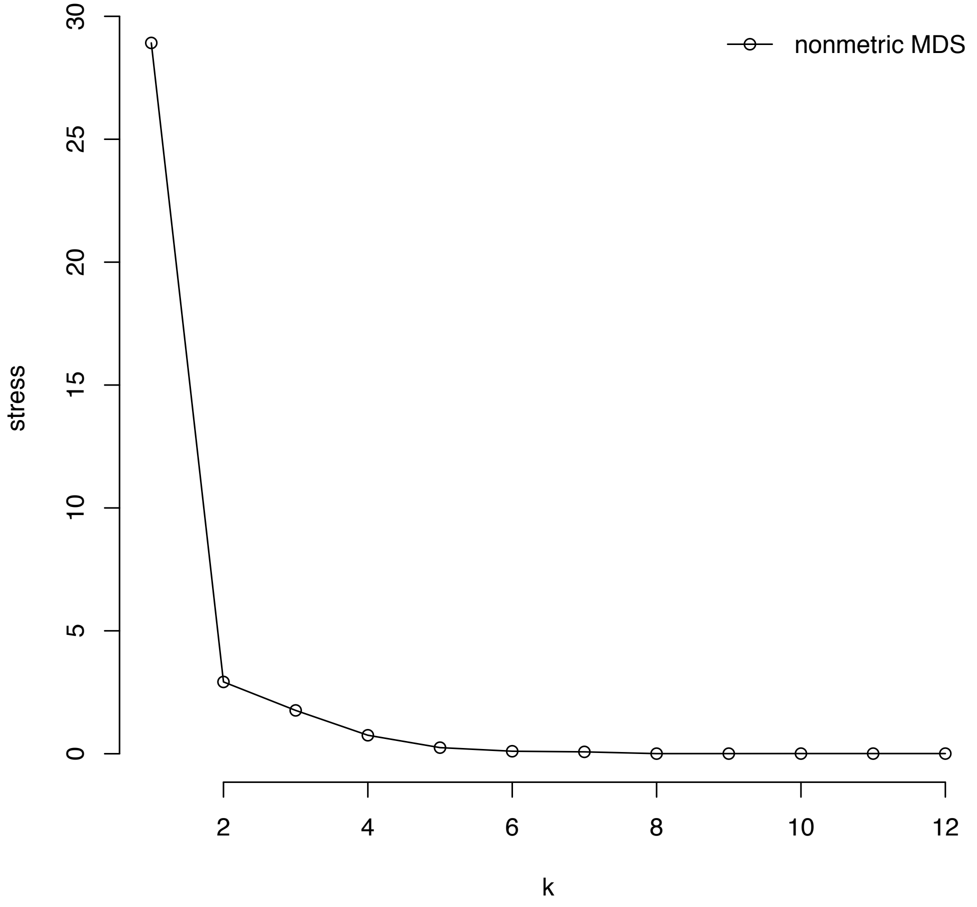

CIELUV



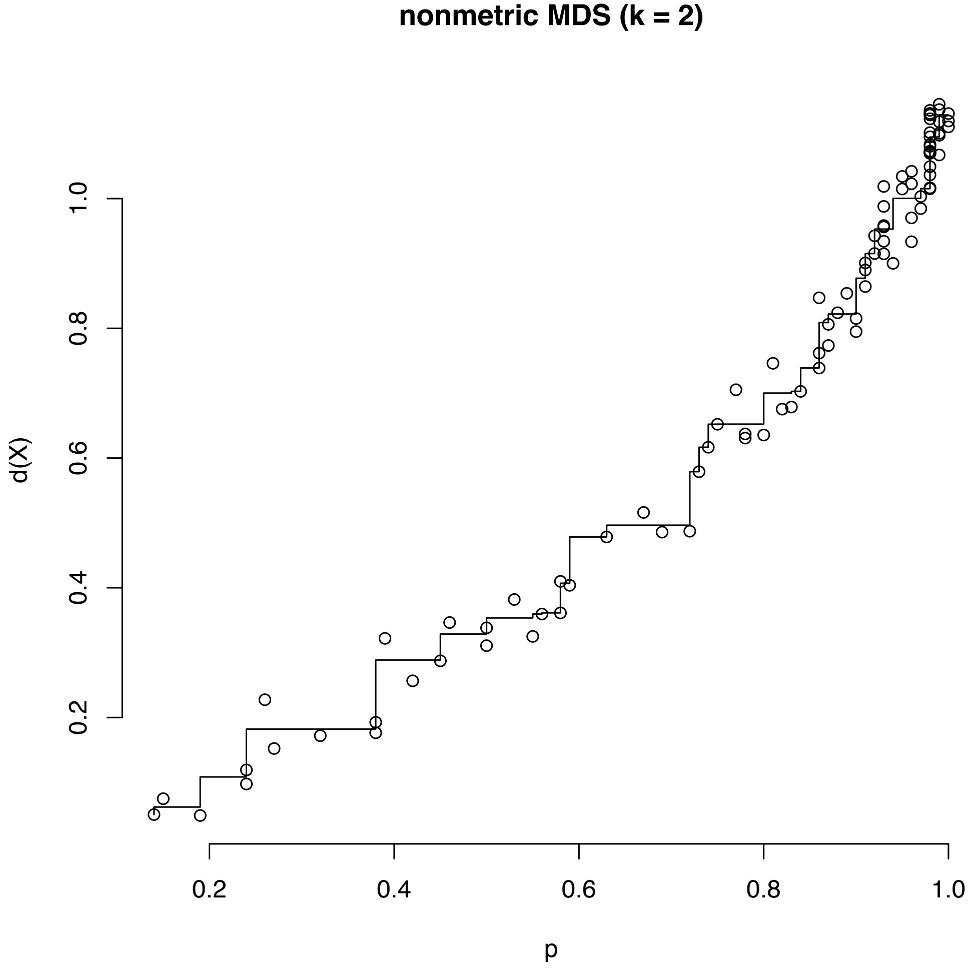




Evaluating the mapping
example: colour



Scree plot gives the stress as a function of k. Here k=1 is too small but k=2 already gives quite a good result.



Shepard plot gives the distances in the embedding as a function of the proximities in the original space.

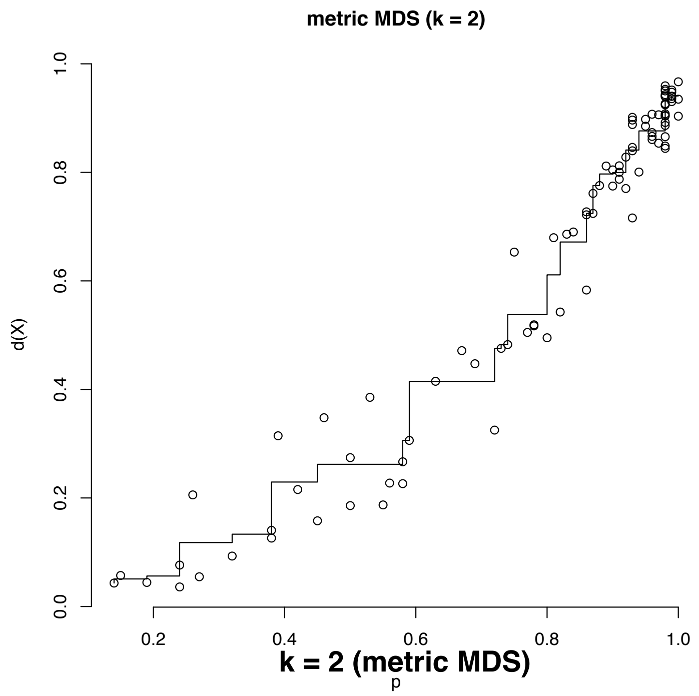
Classical MDS and Sammon mapping

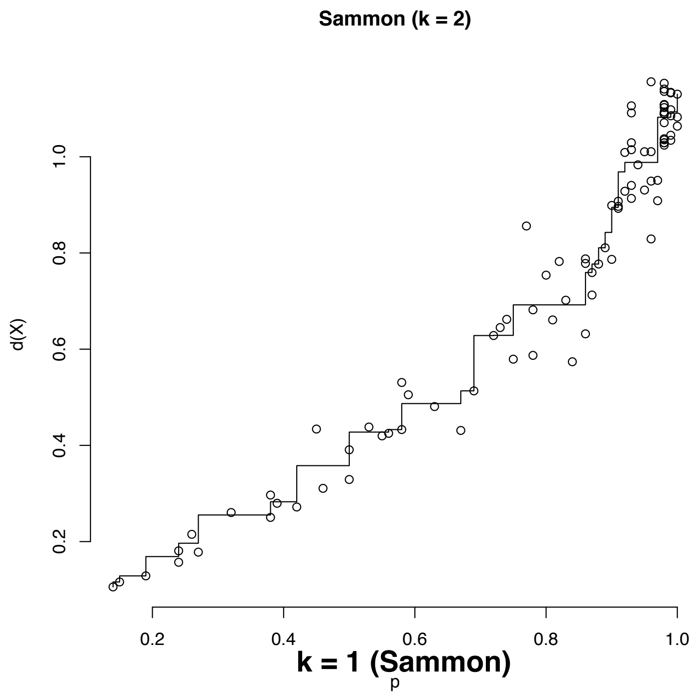
- **Sammon mapping:** given a distance p_{ij} find a representation X that minimises

$$E = \sum_{i<j} \frac{(d_{ij}(X) - p_{ij})^2}{\boxed{p_{ij}}} \longleftarrow \boxed{\text{classical MDS: the same without this}}$$

  - As compared to MDS Sammon mapping should be more accurate for shorter distances but less accurate for longer (why?)
  - Like in nonmetric MDS, solution is found by gradient descent, which may end up in a local minimum

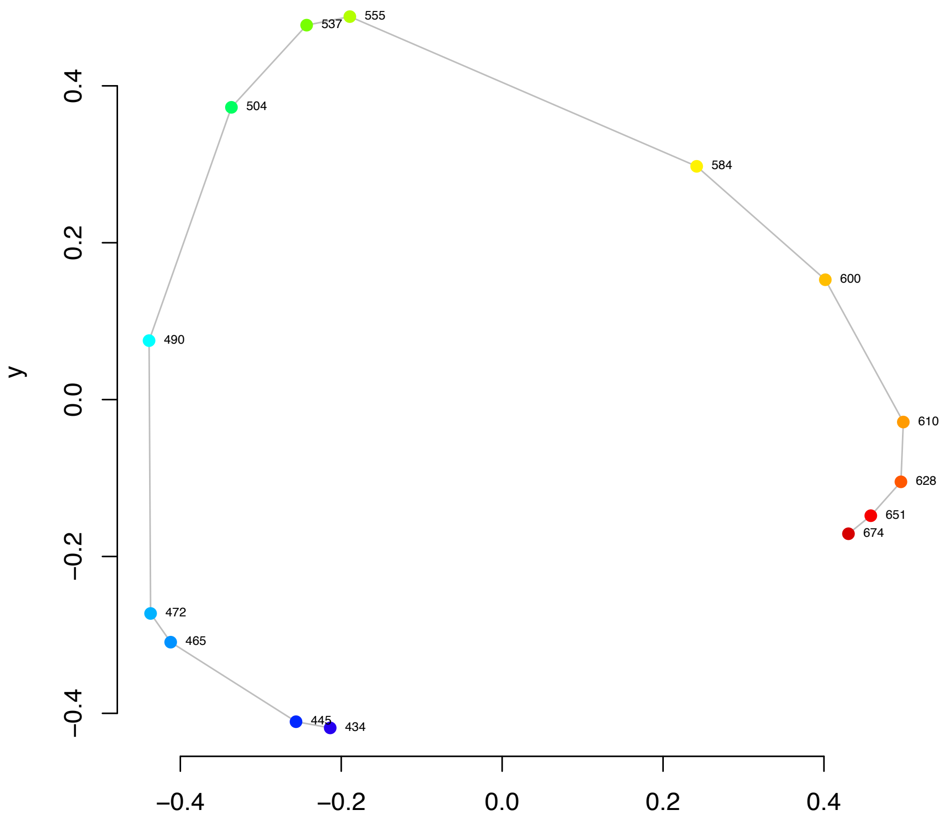
- **Classical MDS** is an instance of metric MDS
  - a.k.a. principal Coordinates Analysis (PCoA), Torgerson Scaling, or Torgerson–Gower scaling.

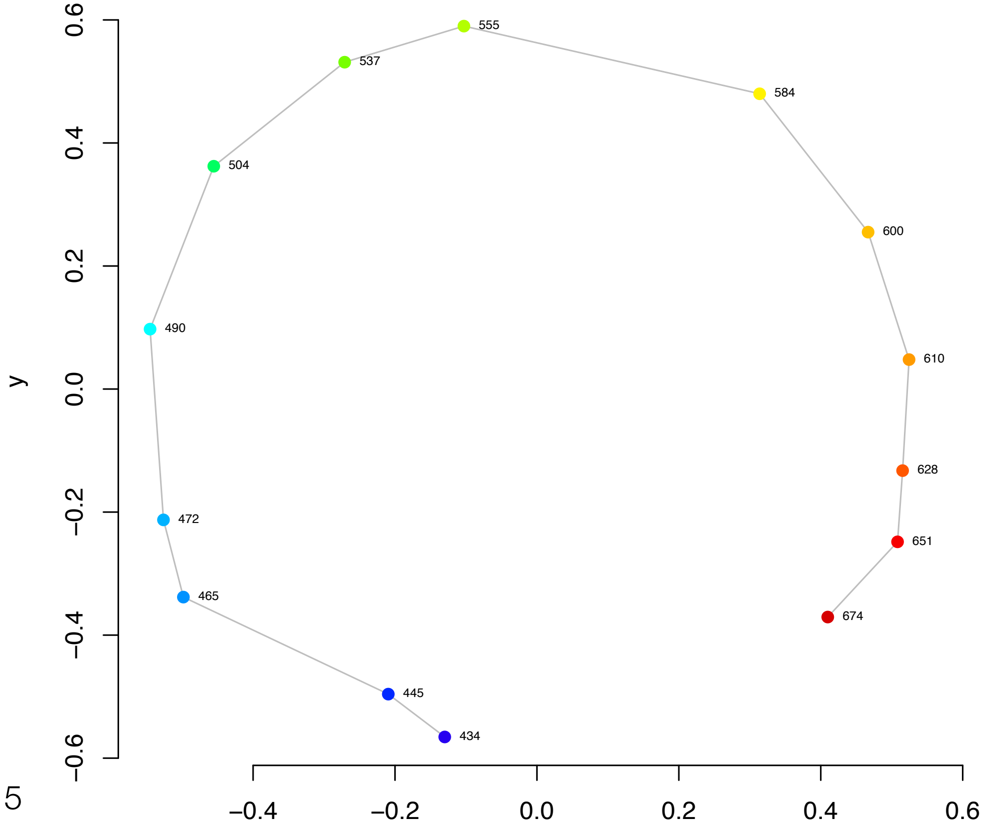




k = 2 (metric MDS)

k = 1 (Sammon)



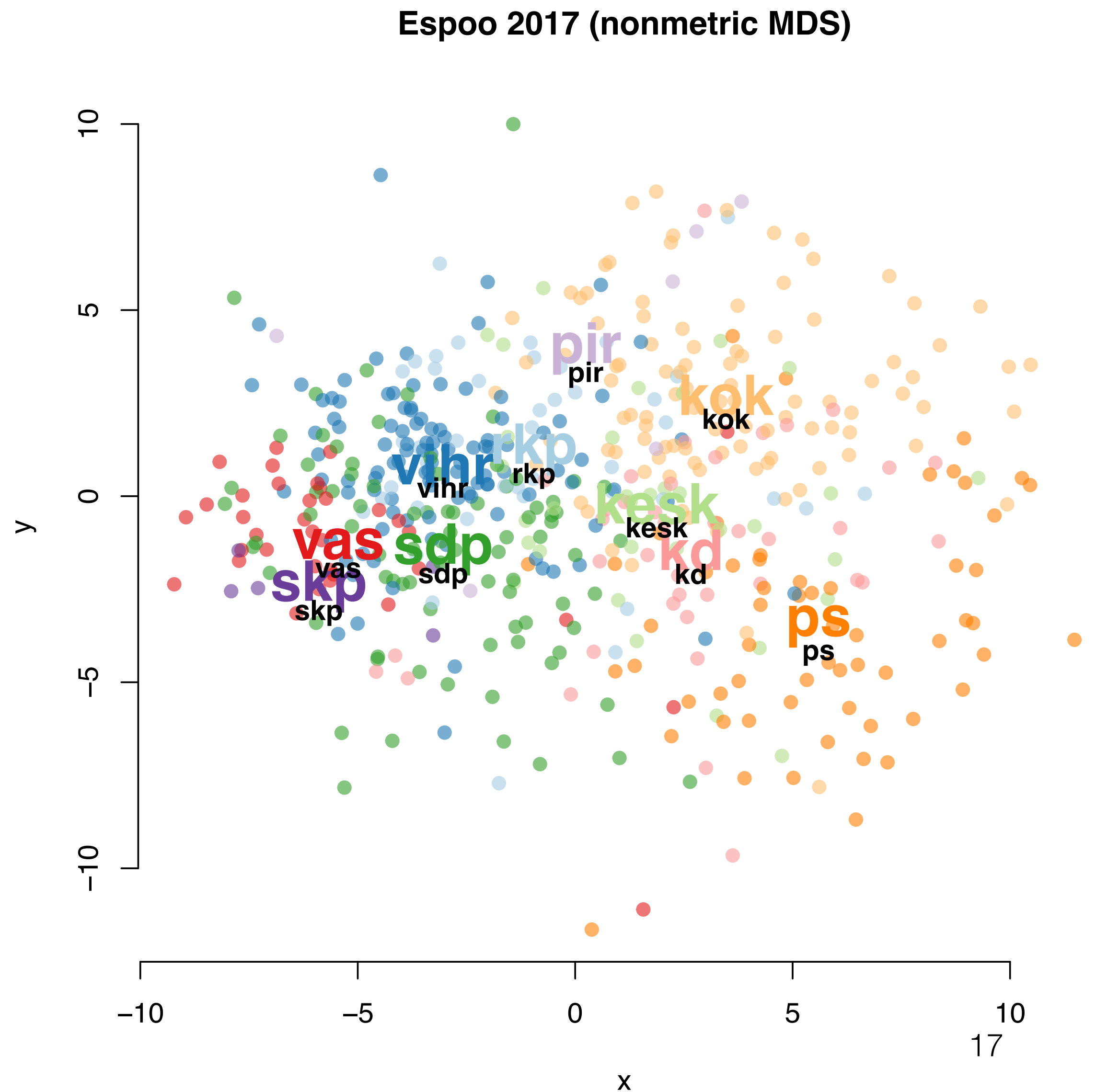


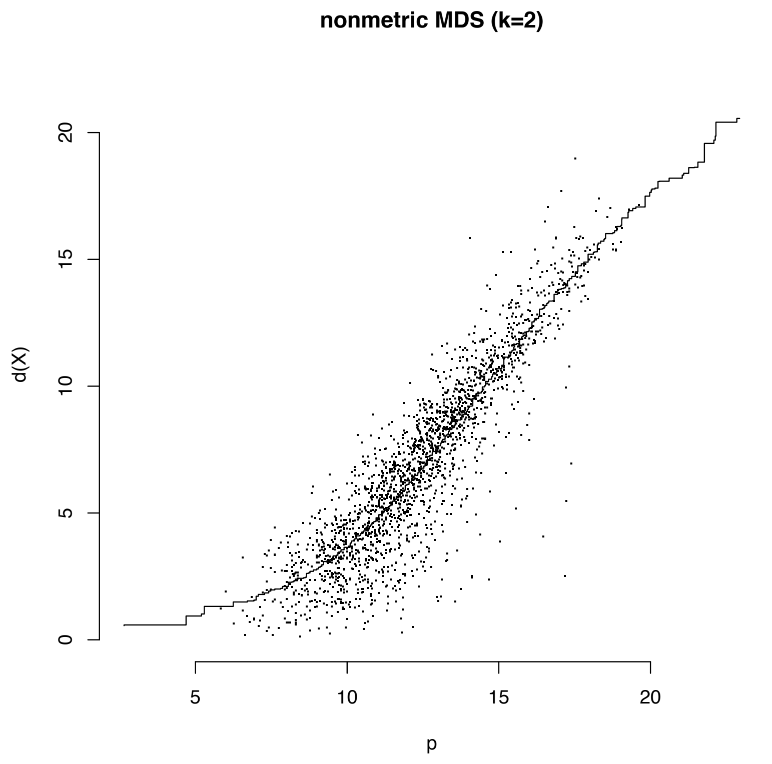
Municipal elections in Espoo in 2017

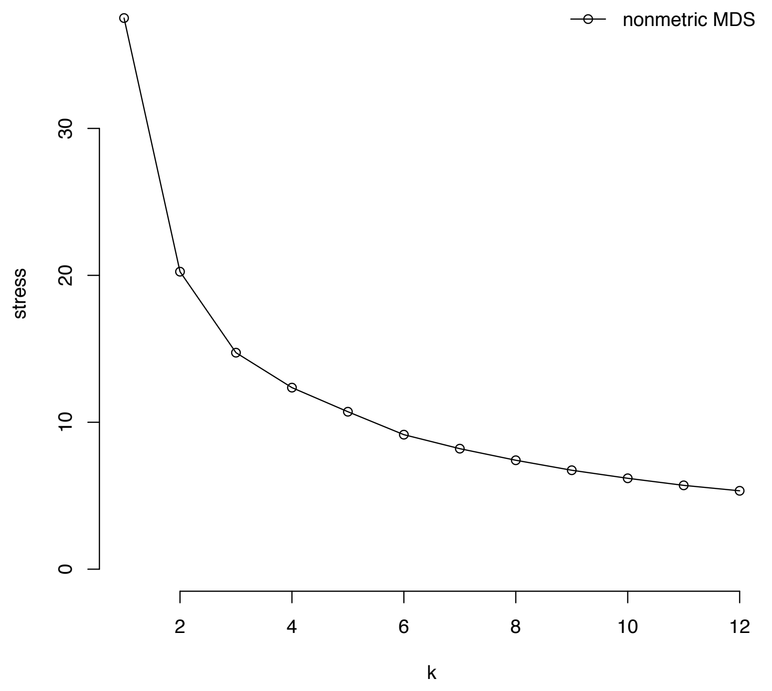
- Survey of candidates done by Helsingin Sanomat
- Here included only 10 parties with largest number of candidates nationally.
- Each candidate rated each of the **m=49** statements on a scale from 1 to 5, where 1=disagree and 5=agree:
    1. *Euthanasia should be allowed.*
    2. *I prefer public instead of the private sector to produce my local health services.*
    3. *Same gender couples should have the same marital and adoptation rights than the different genre couples.*
    4. *Good brother networks influence municipal decision-making.*
    5. *...*
- **n=515** candidates in total, i.e., we have a data 515x49 matrix.

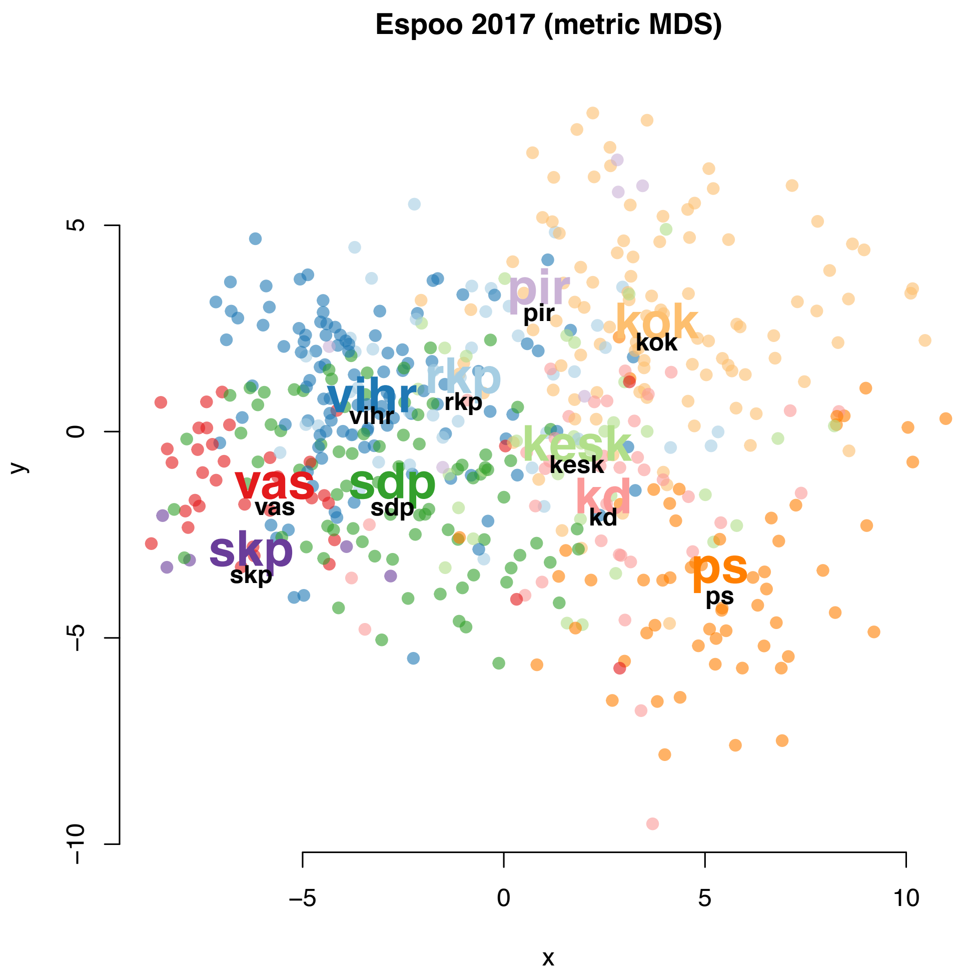
- Distance p_{ij} between candidates i and j is the Euclidean distance of their respective 49-dimensional rating vectors. What is a 2-dimensional representation that preserves these distances faithfully?

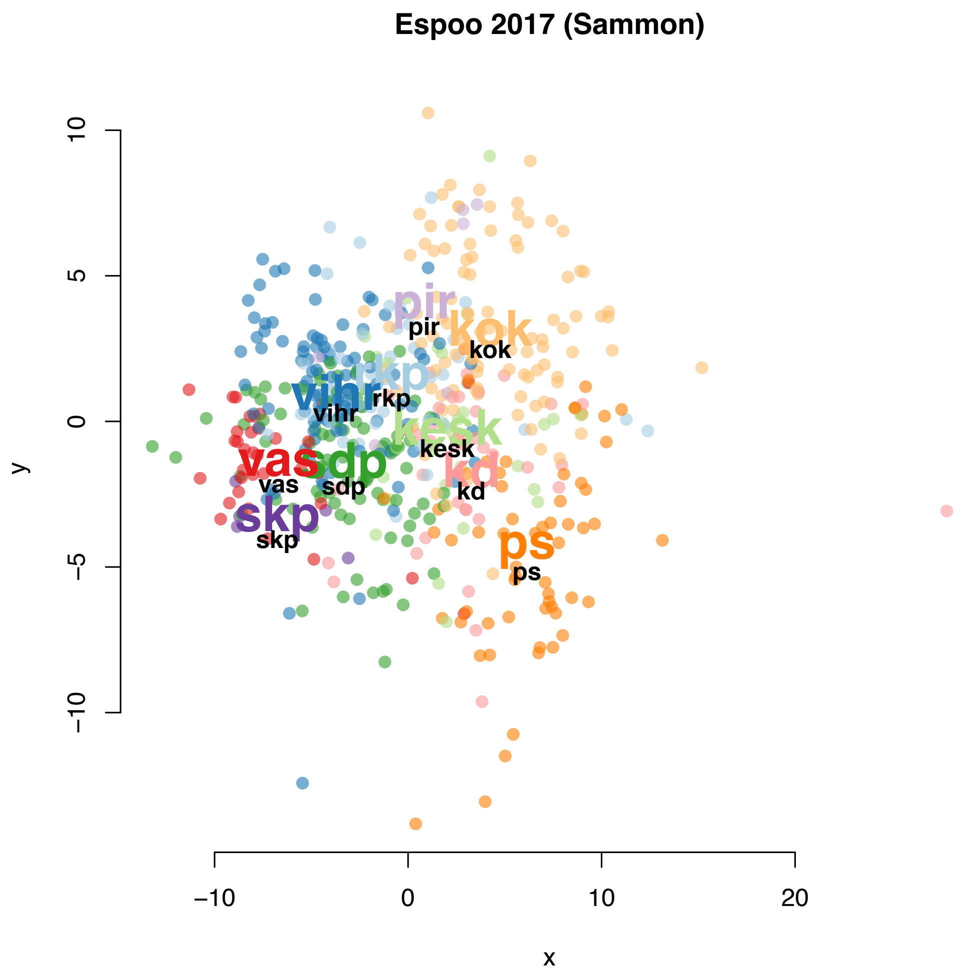
Municipal elections in Espoo in 2017

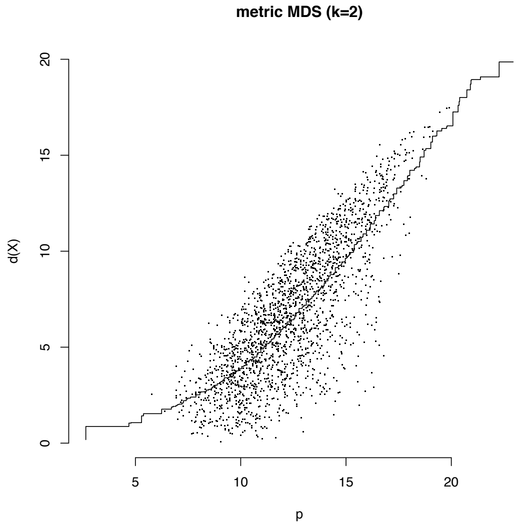


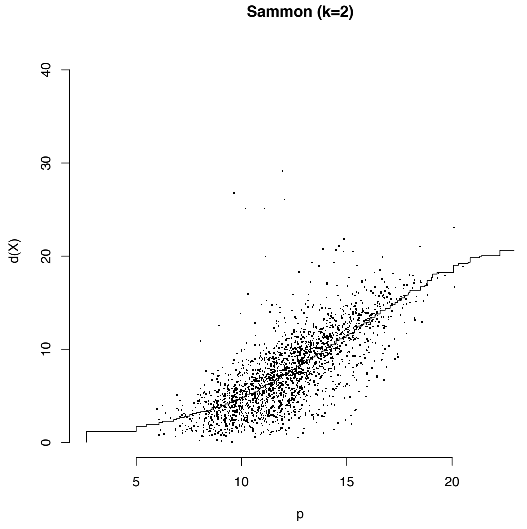





Espoo 2017 (metric MDS)
pir
kok
rkp
vihr
kesk
vas
sdp
kd
skp
ps
y
x

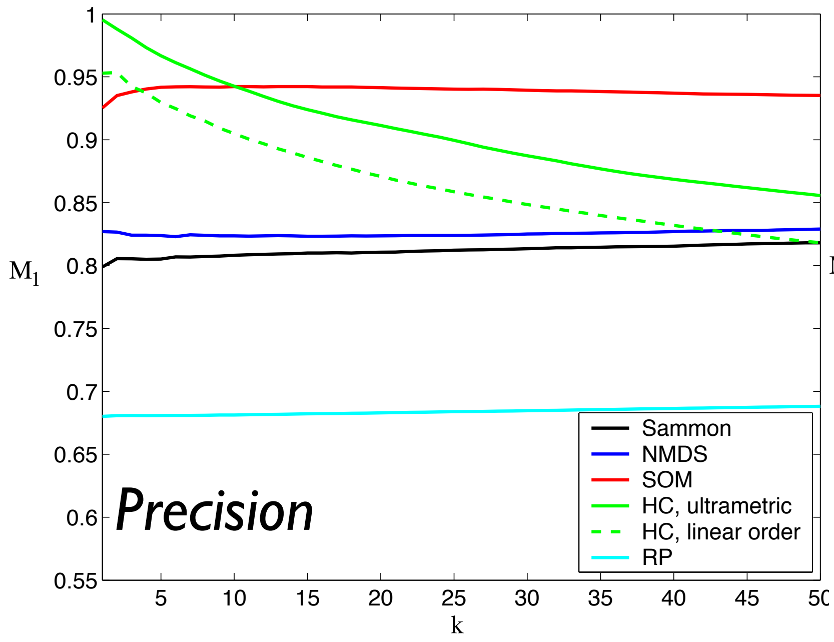

Espoo 2017 (Sammon)
pir
kok
vihr
rkp
kesk
vas
sdp
kd
skp
ps
y
x

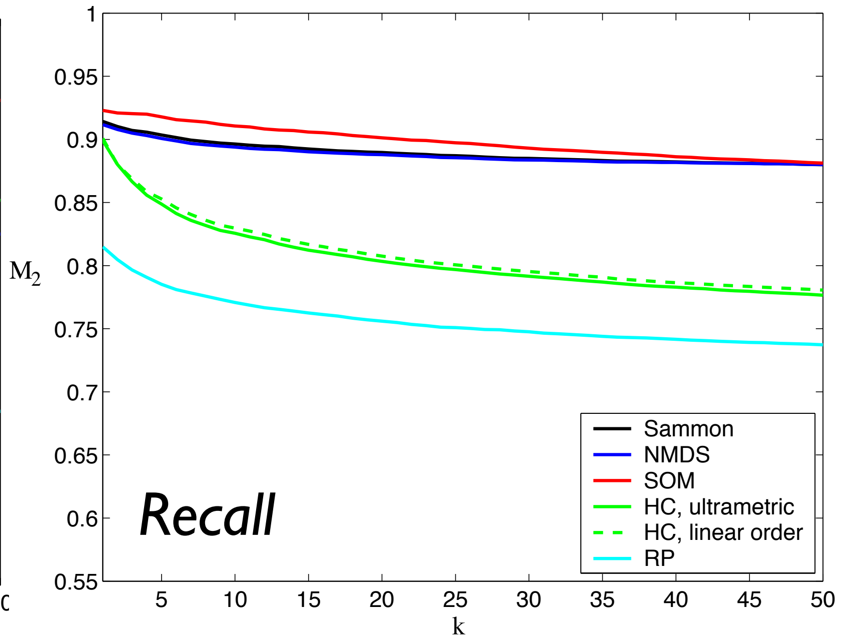

metric MDS (k=2)
d(X)
p


Sammon (k=2)
d(X)
p

Performance of MDS

- MDS is tries to preserve the large distances at the expense of small ones, hence, it can "collapse" some small distances on the expense of preserving large distances
- A projection is *trustworthy (precision)* if *k* closest neighbours of a sample on the projection are also close by in the original space. A projection *preserves the original neighbourhoods* (*recall*) if all *k* closest neighbours of a sample in the original space are also close by in the projection.





Figures are from Kaski, Nikkilä, Oja, Venna, Törönen, Castrén, *Trustworthiness and metrics in visualizing similarity of gene expression*, BMC Bioinformatics 2003, 4:48.

Precision and recall as a function of the neighbourhood size *k* for a yeast data set. Non-metric (ordinal) MDS (NMDS) is shown in blue. Larger precision and recall is better.

Performance of MDS

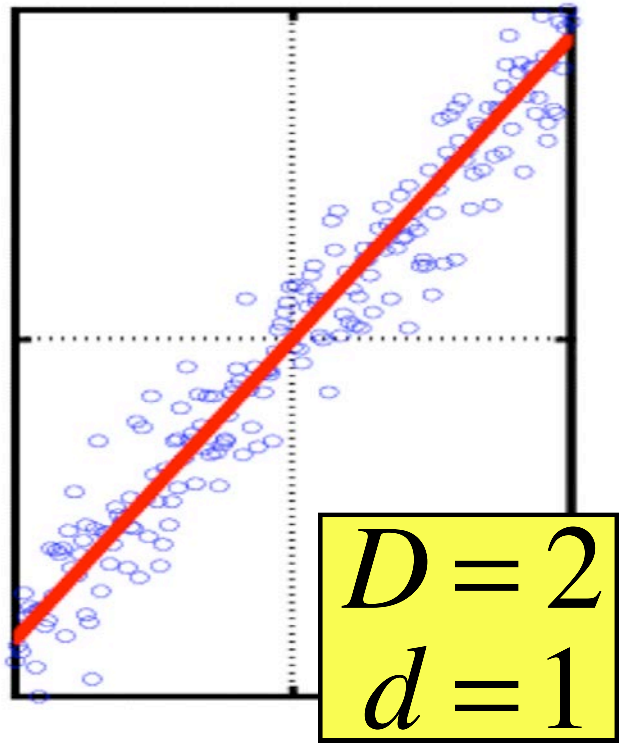
- Relatively good recall, not very good precision
- MDS algorithms typically have running times of the order $O(N^2)$, where N is the number of data items.
- This is not very good: N=1,000 data items are ok, but N=1,000,000 is getting slow.
- Some solutions: use landmark points (i.e., use MDS only on a subset of data points and place the remaining points according to those, use MDS on cluster centroids etc.), use some other algorithm or modification of MDS.
- MDS is not guaranteed to find the global optimum of the stress (cost) function, nor it is guaranteed to converge to the same solution at each run (many of the MDS algorithms are quite good and reliable, though)

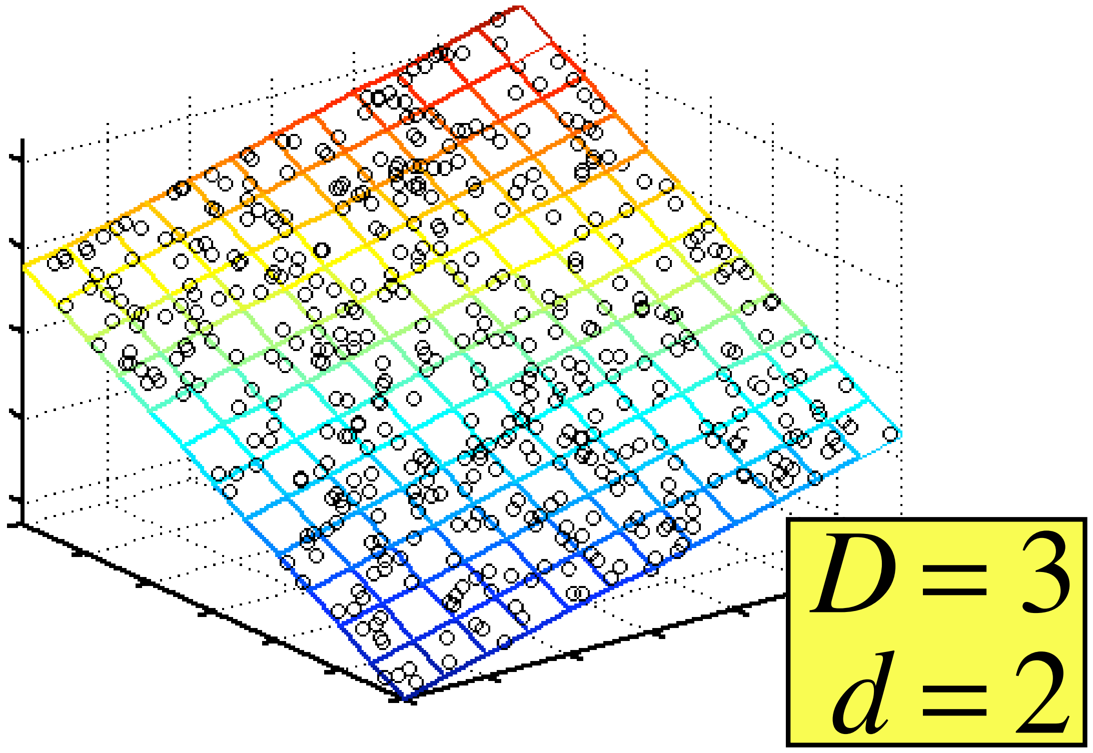
Projection pursuit methods

- MDS (and variants) are based on **distance matrix** between points.
- If data is composed of **vectors** (such as Espoo municipal elections 2017 data) we can use projection pursuit methods.

- Projection pursuit methods try to **find a linear subspace** u that maximises some quantity
- E.g., for the election data let *X* be the 515x49 data matrix and *f* a function. Problem: find 49-dimensional unit vector *u* such that *f(A u)* is maximised.
  - if f is variance we have principal component analysis (PCA)
  - if f is a measure of non-gaussianity we have independent component analysis (ICA)
- Typically, we can find several directions u (possibly with orthogonality conditions).

Principal component analysis (PCA)





- Basic idea: rotate the space such that the data becomes maximally aligned with the coordinate axes

Principal component analysis (PCA)

- The **principal component analysis** (PCA) finds the eigenvalues and -vectors of a matrix
- PCA is an example of the projection pursuit methods. It tries to find a linear subspace that has **maximal variance**.
- Thus, the interesting quality in PCA is variance (distance).
    - you could think PCA as a linearised version of MDS (actually PCA is equivalent to one modification of MDS).
- PCA (unlike MDS) assumes that the data points are vectors in a high-dimensional Euclidean space,
- The data points are projected to d-dimensional Euclidean subspace (d≪D) of the original space.
- The projection to d-dimensional subspace is linear, $A = \sum_{\alpha=1}^{d} e_\alpha e_\alpha^T$

  $y_i = Ax_i$, where e_α are orthogonal unit vectors.
- Goal: nearby points remain nearby, distant points remain distant.

Principal component analysis (PCA)

- Goal, more formally: *find such a projection (matrix A) to d-dimensional subspace that the average error in the squared Euclidean distances between data points is minimised.*

$$\sum_{i,j=1}^{N} \left| \| x_i - x_j \|^2 - \| y_i - y_j \|^2 \right|$$

where || || is the Euclidean distance and $y_i = A\, x_i$.

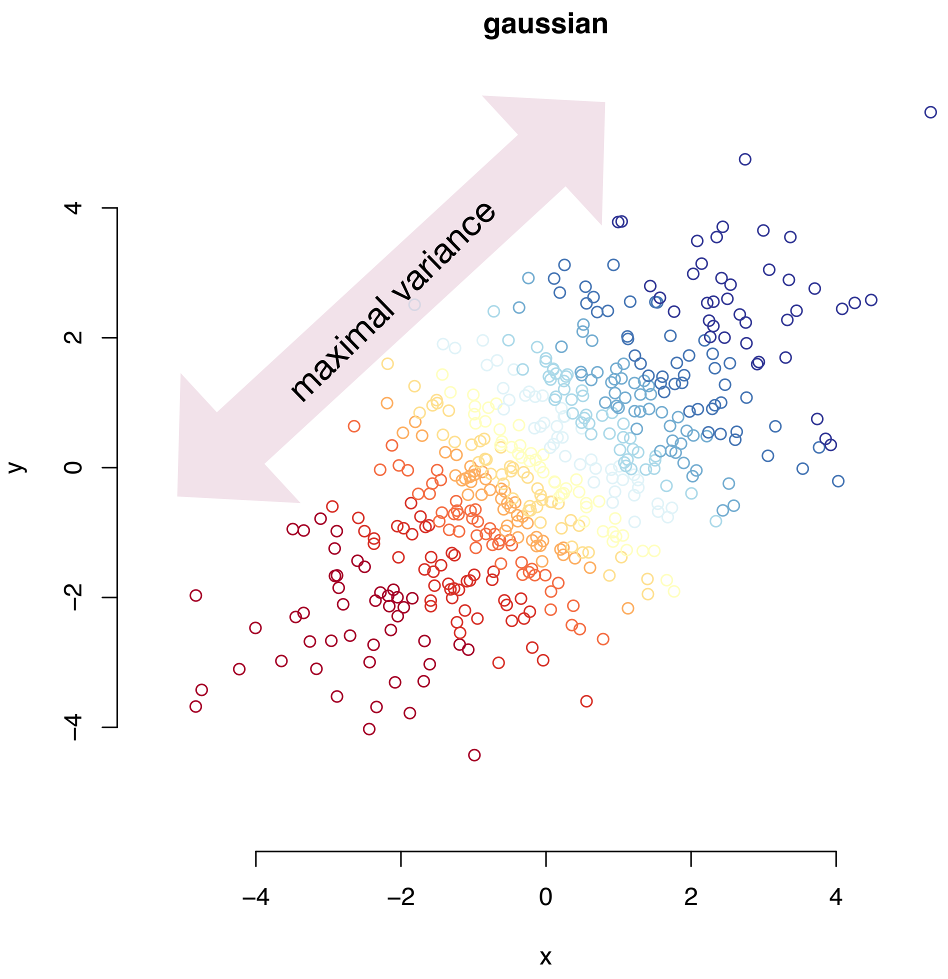
- Denote the mean vector by, $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
- The *covariance matrix* reads then, $C = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T.$
- The covariance matrix can be decomposed (*spectral decomposition*) as

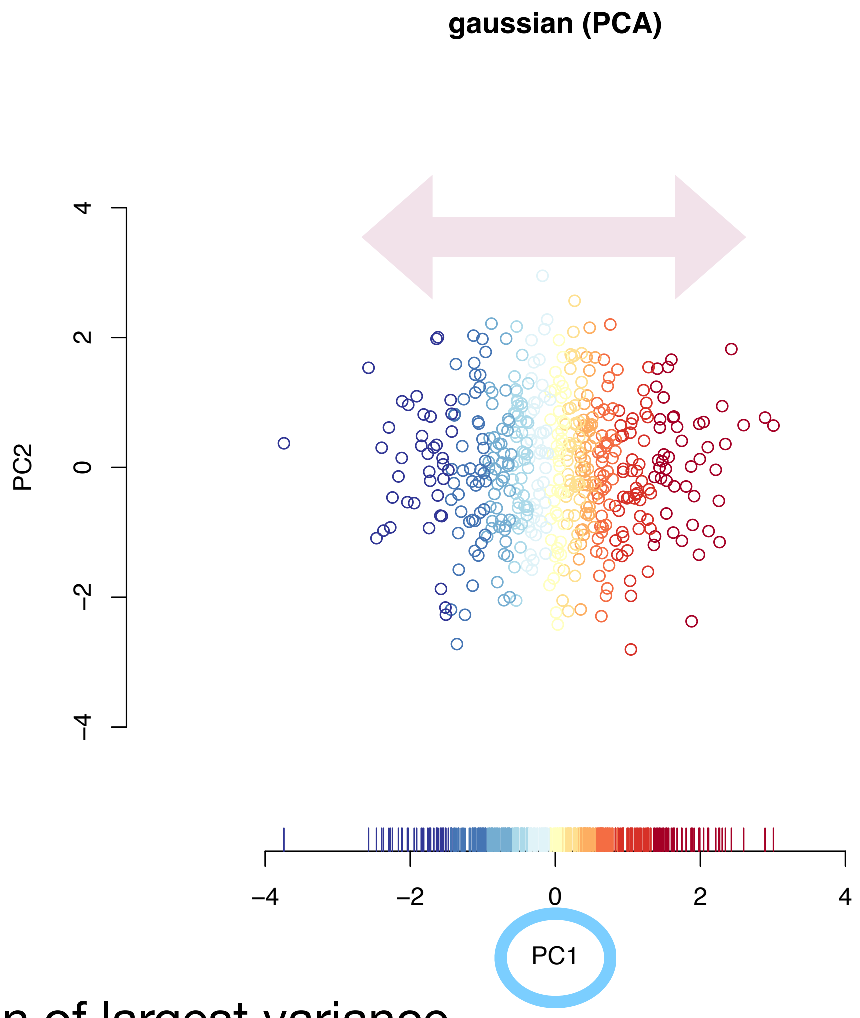
$$C = \sum_{\alpha=1}^{D} \lambda_a e_\alpha e_\alpha^T$$

where λ_α are the *eigenvalues* ($\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$) and e_α are the corresponding orthogonal unit *eigenvectors*.

- The maximum variance projection is then given by $A = \sum_{\alpha=1}^{d} e_\alpha e_\alpha^T$

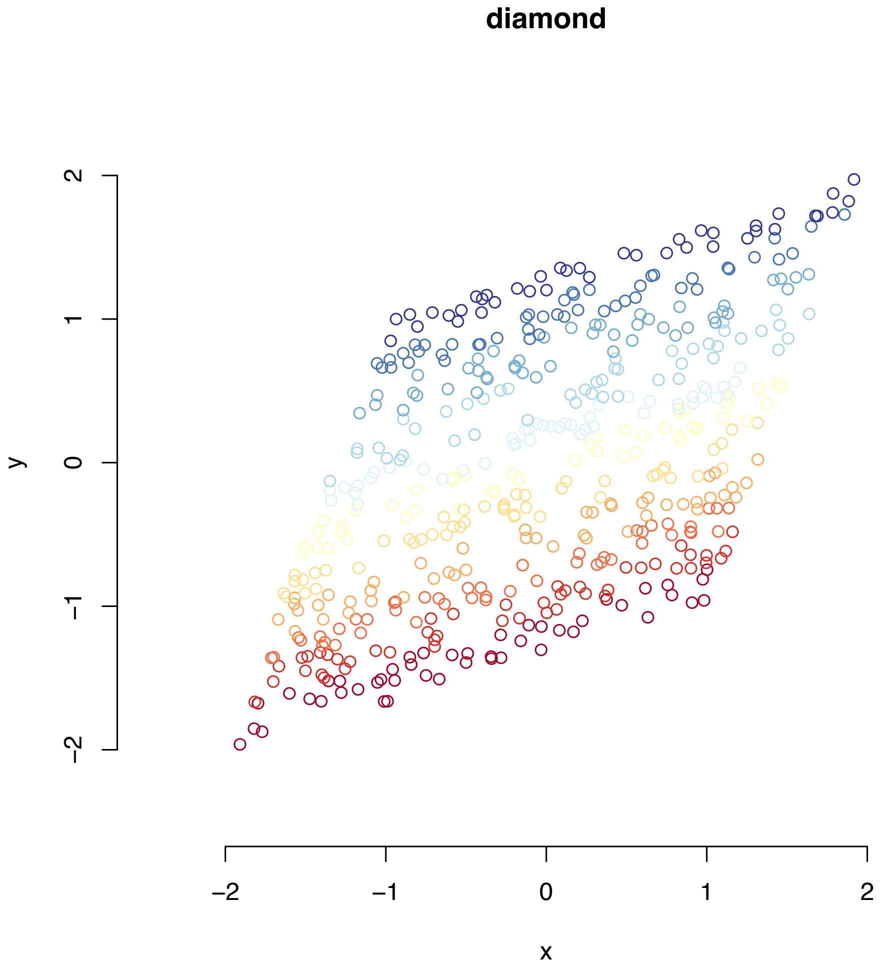
Gaussian data

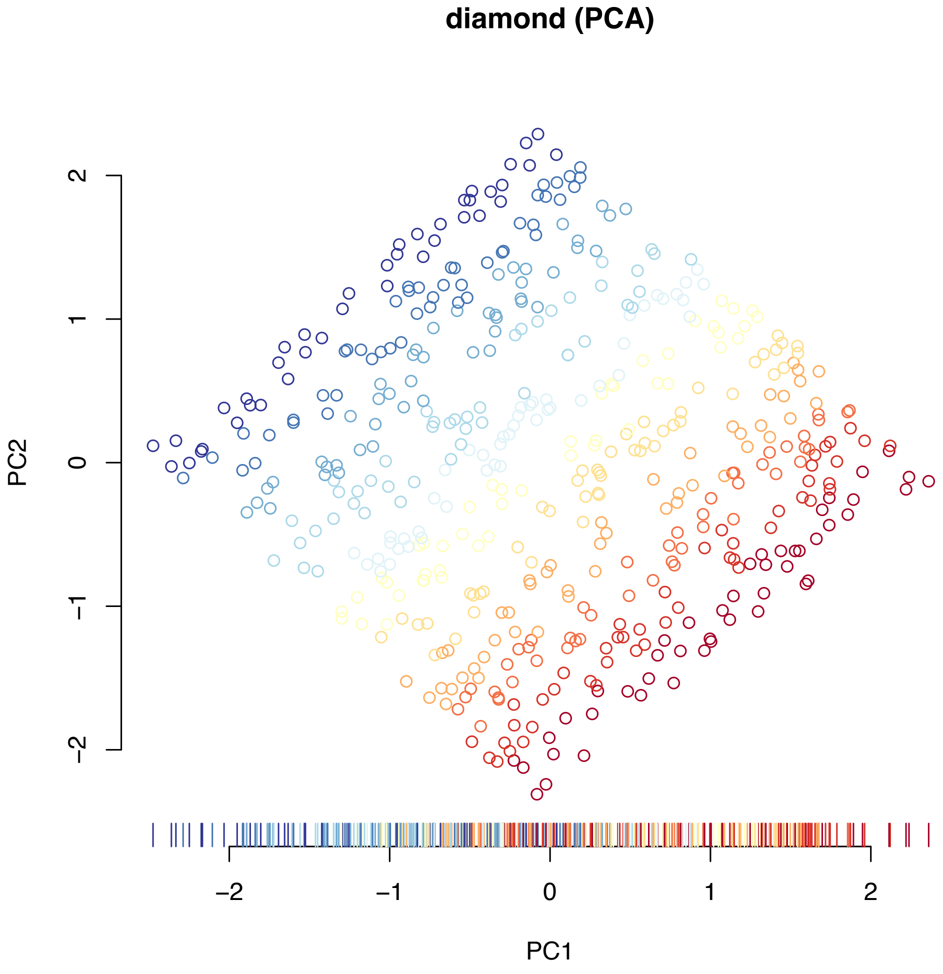




PC1 finds the direction of largest variance.

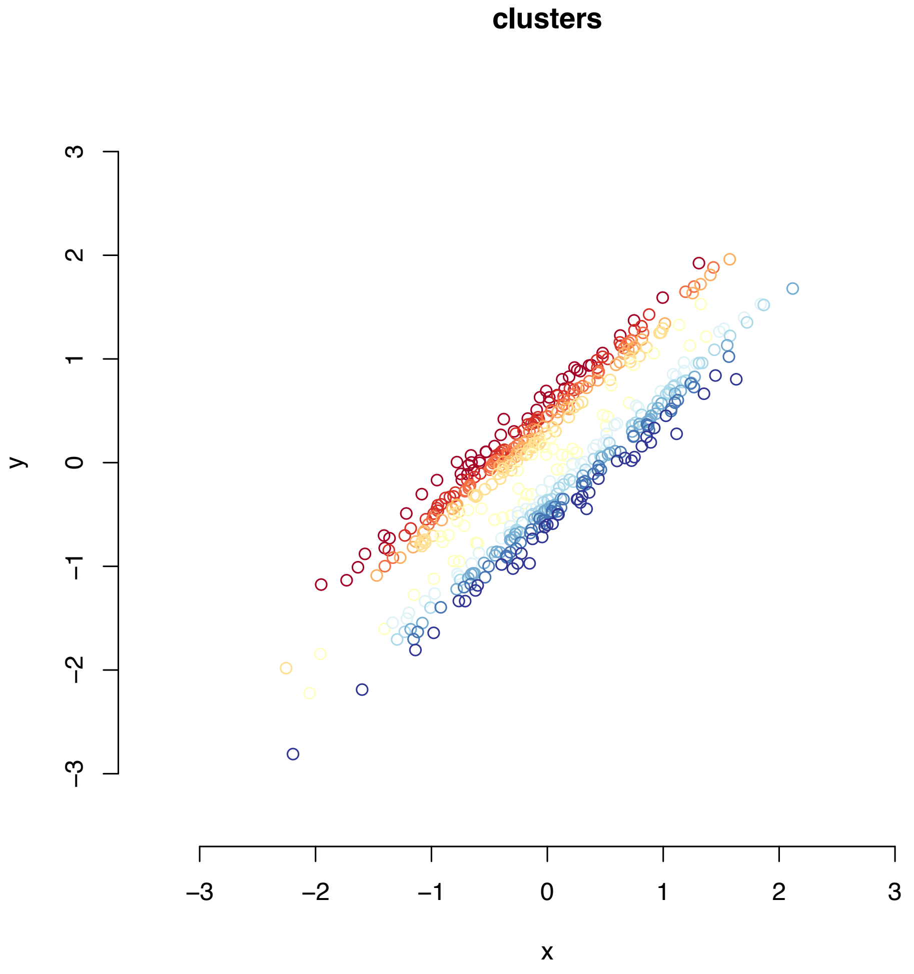
Diamond shaped data

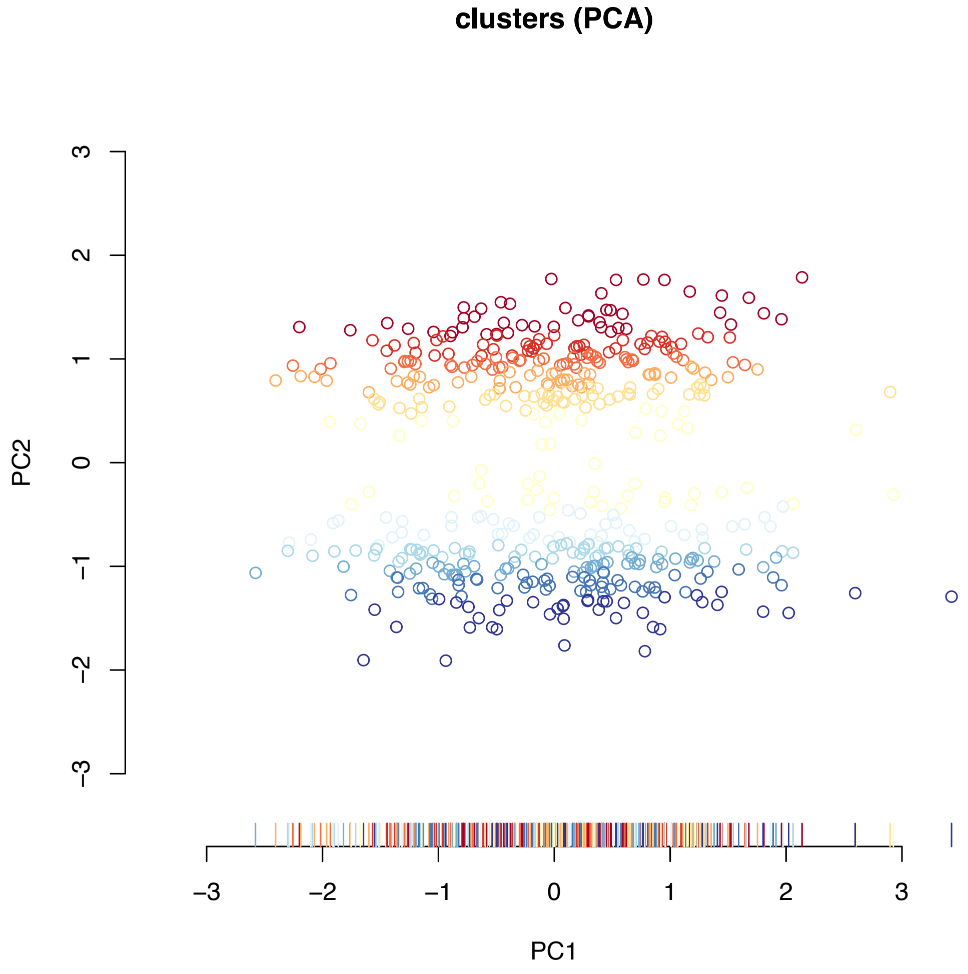




PC1 misses the square structure.

Two clusters





PC1 misses the cluster structure.

Principal component analysis (PCA)

- PCA can be computed easily with (almost) any software that is capable of doing linear algebra.
- PCA is stable, there are no additional parameters, and it is guaranteed always to converge to the same optimum.
- Hence, PCA is usually the first dimension reduction method to try
  (if it doesn't work, then try something more fancy)

- If you find PCA difficult, this may help :-)
  https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

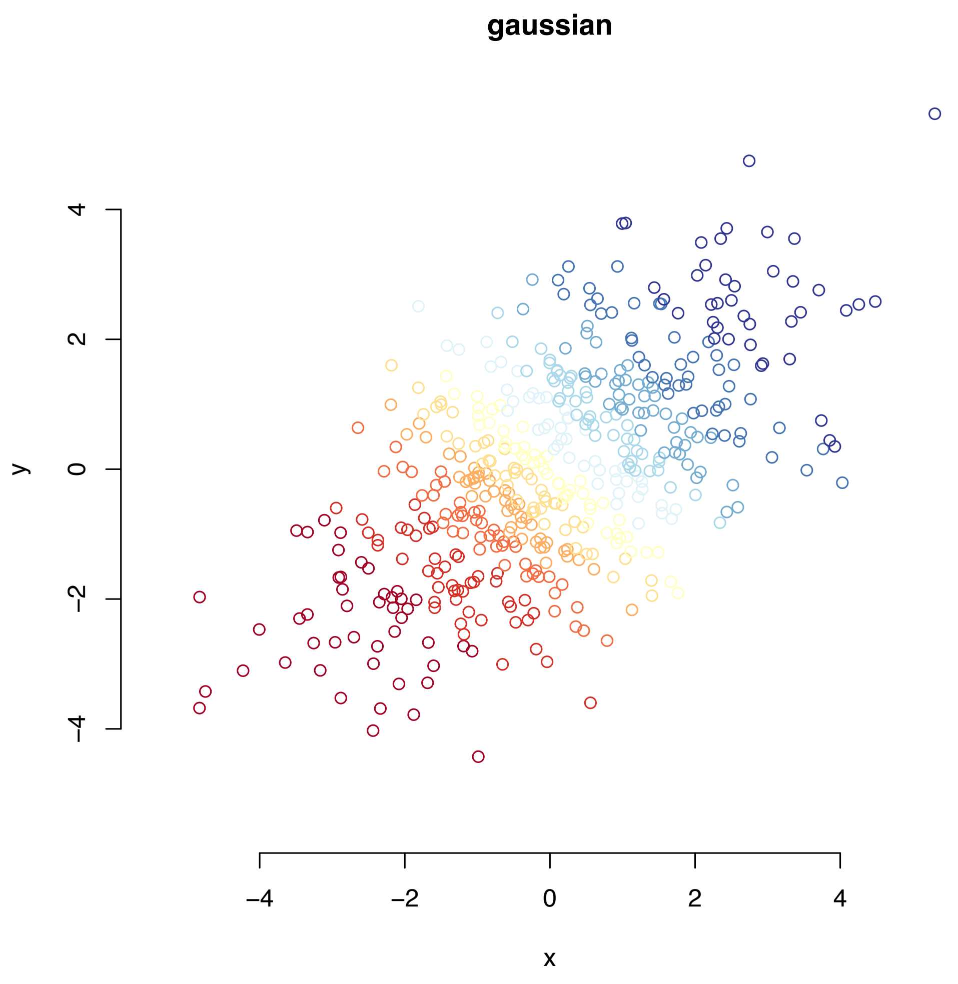
Independent component analysis (ICA)

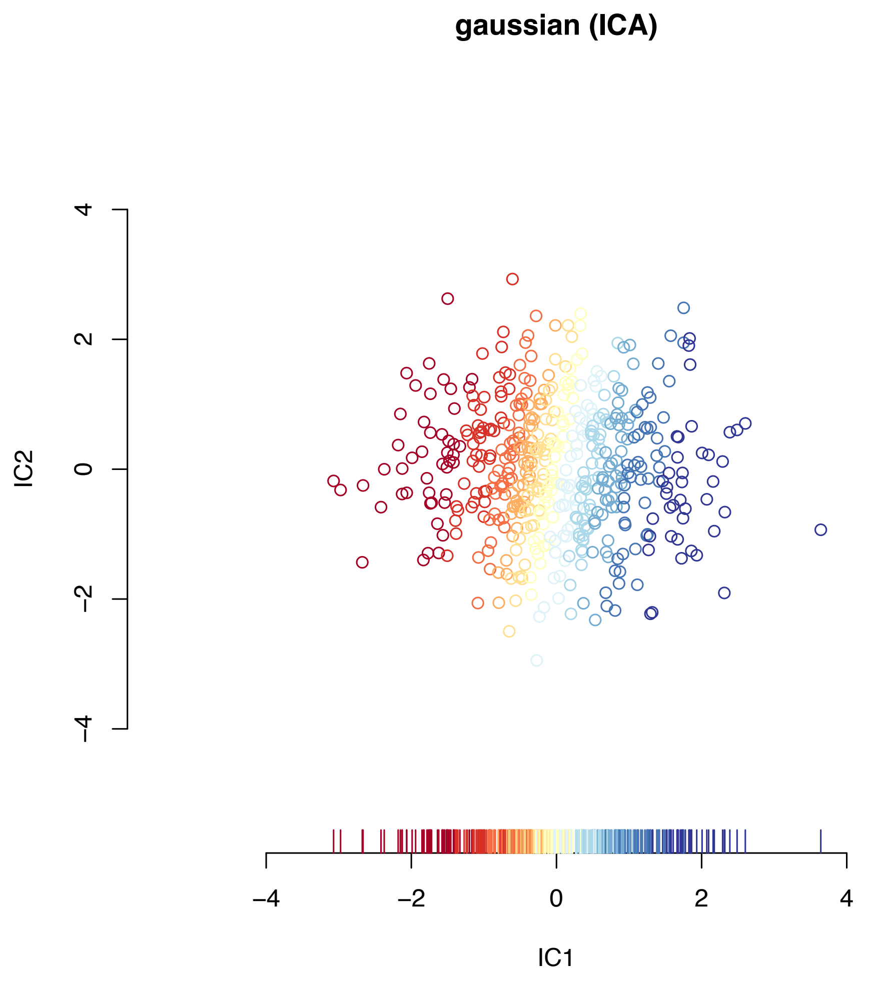
- Goal: function *f* is a measure of non-Gaussianity. Non-Gaussian directions are usually most independent.
- Hence, ICA finds separate processes.

- Like PCA, ICA (usually) makes a linear transformation, but the component directions are not necessarily orthogonal.

- ICA is unstable and may end up to a local minimum.
- There are robust libraries to compute ICA: use the libraries!

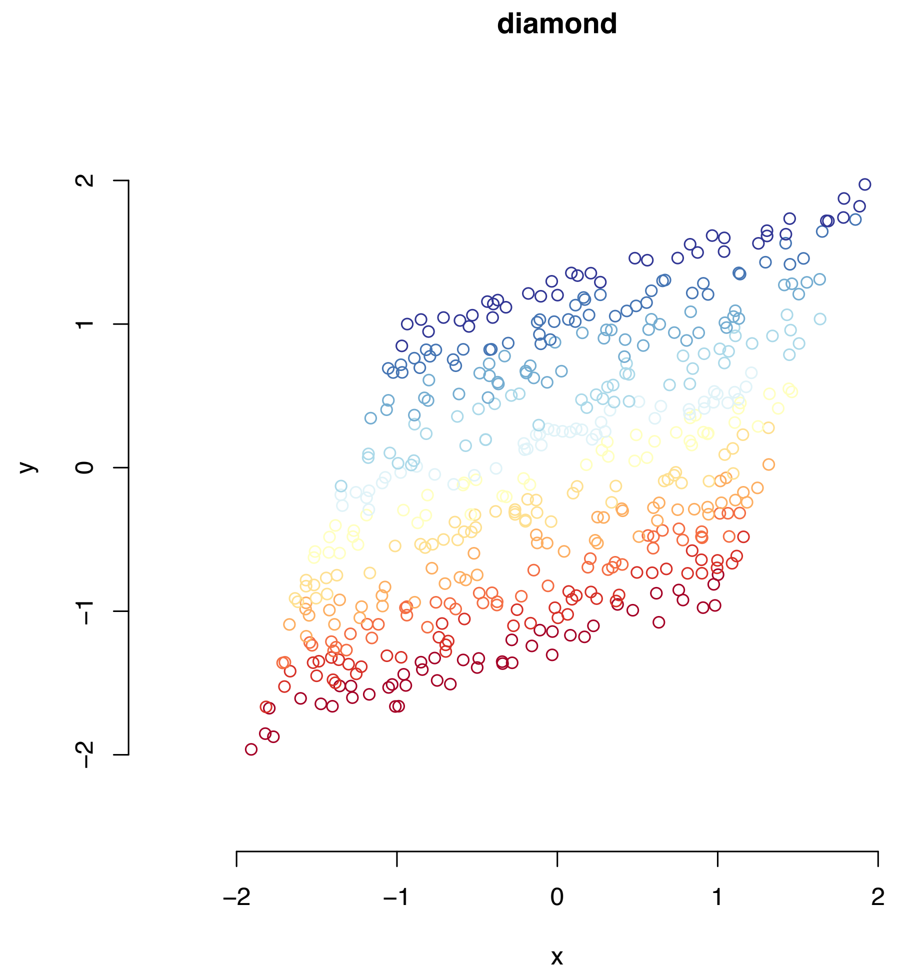
Gaussian data

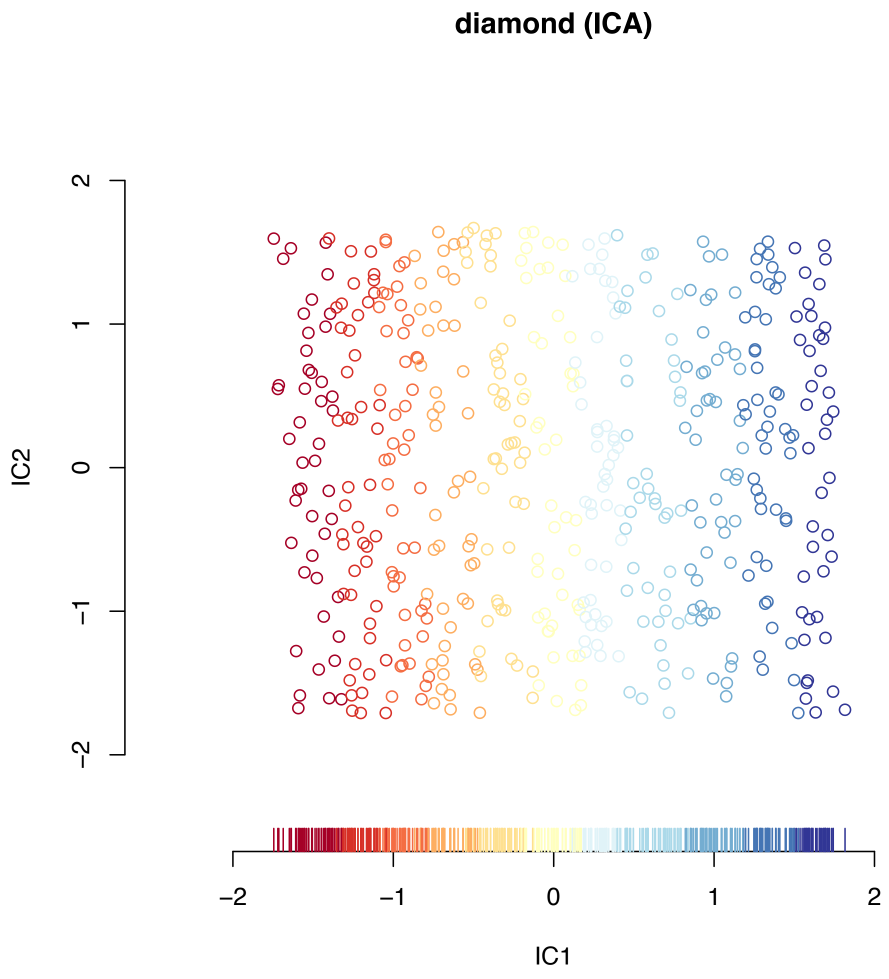




ICA ignores total variance (but finds the maximal variance direction here by co-incidence).

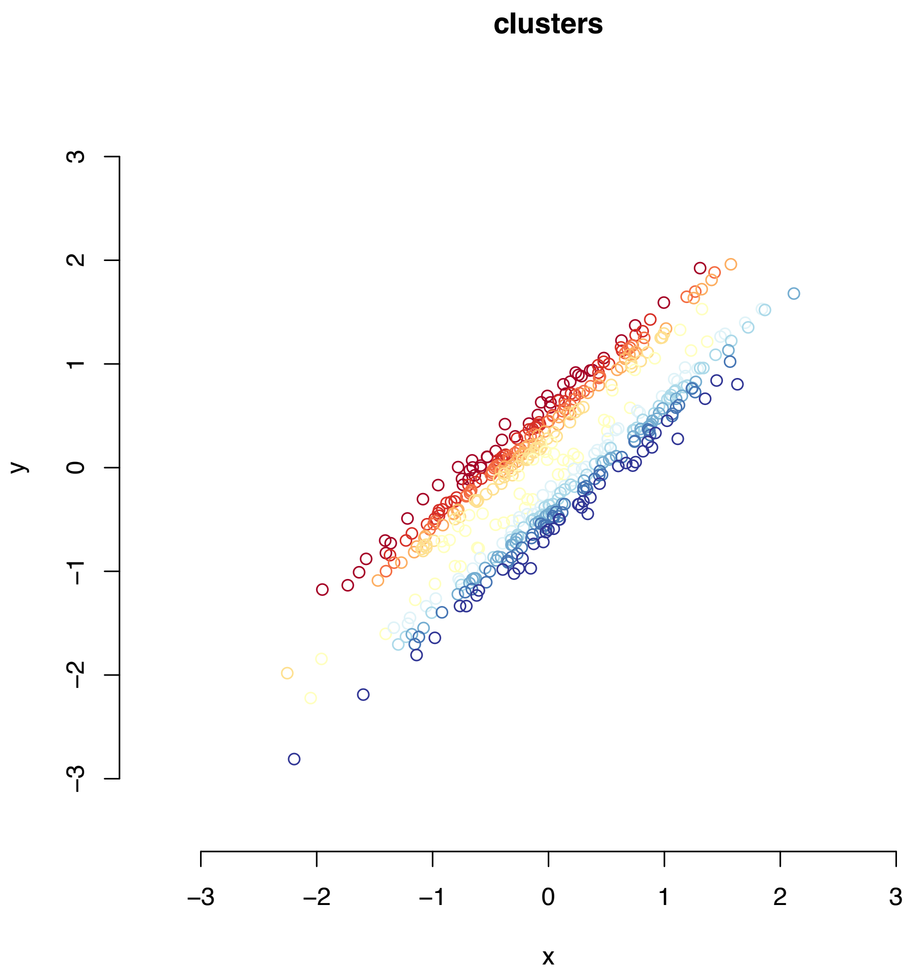
Diamond shaped data

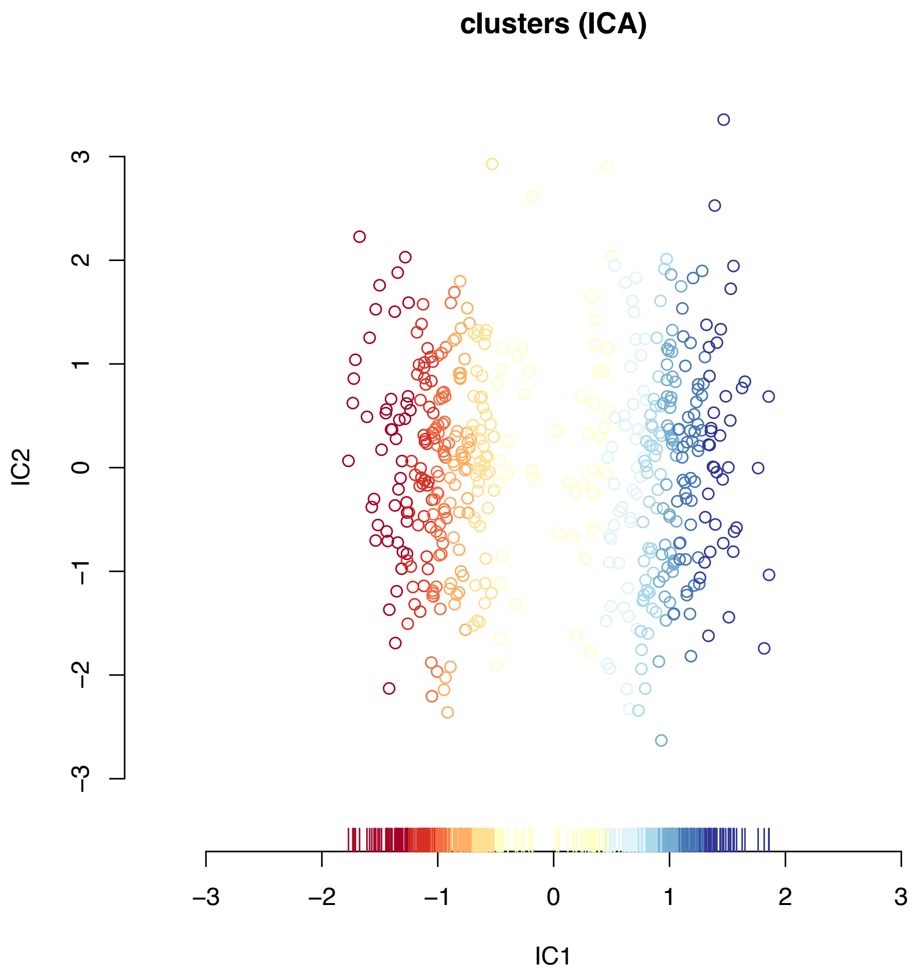




IC1 finds the box in the diamond.

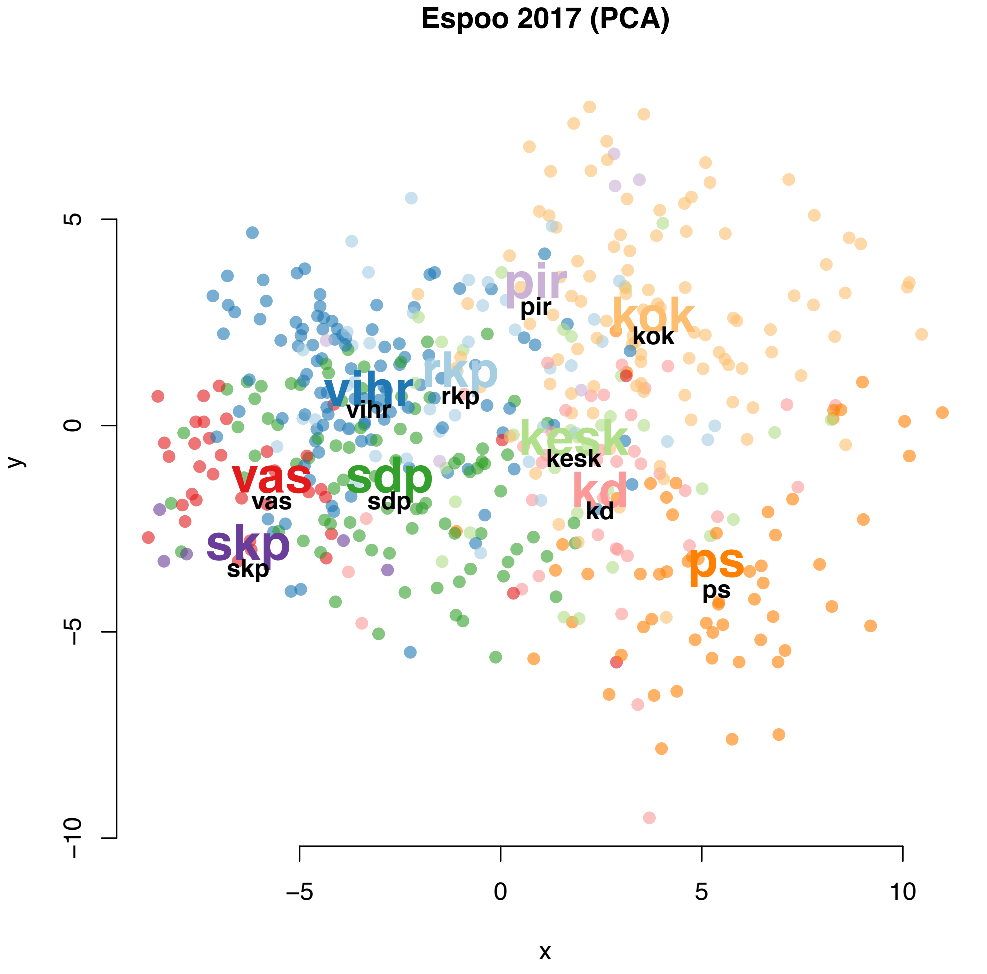
Two clusters

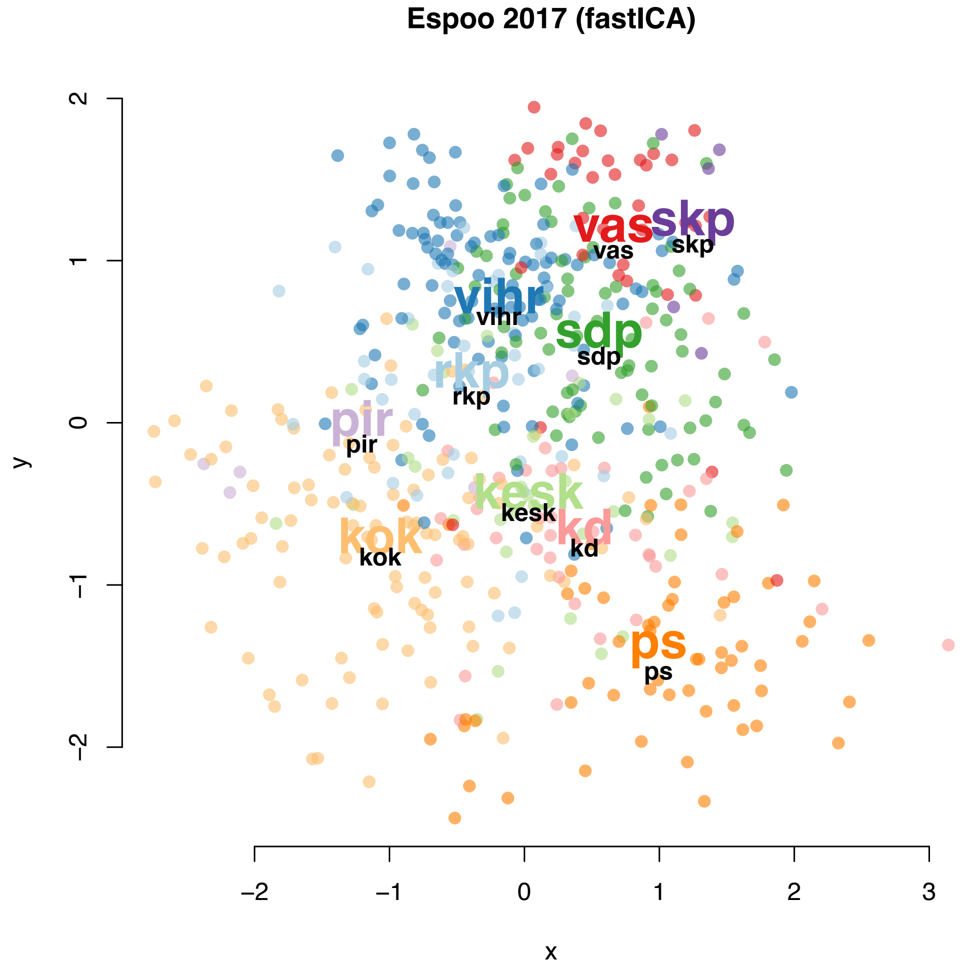




IC1 finds the two clusters.

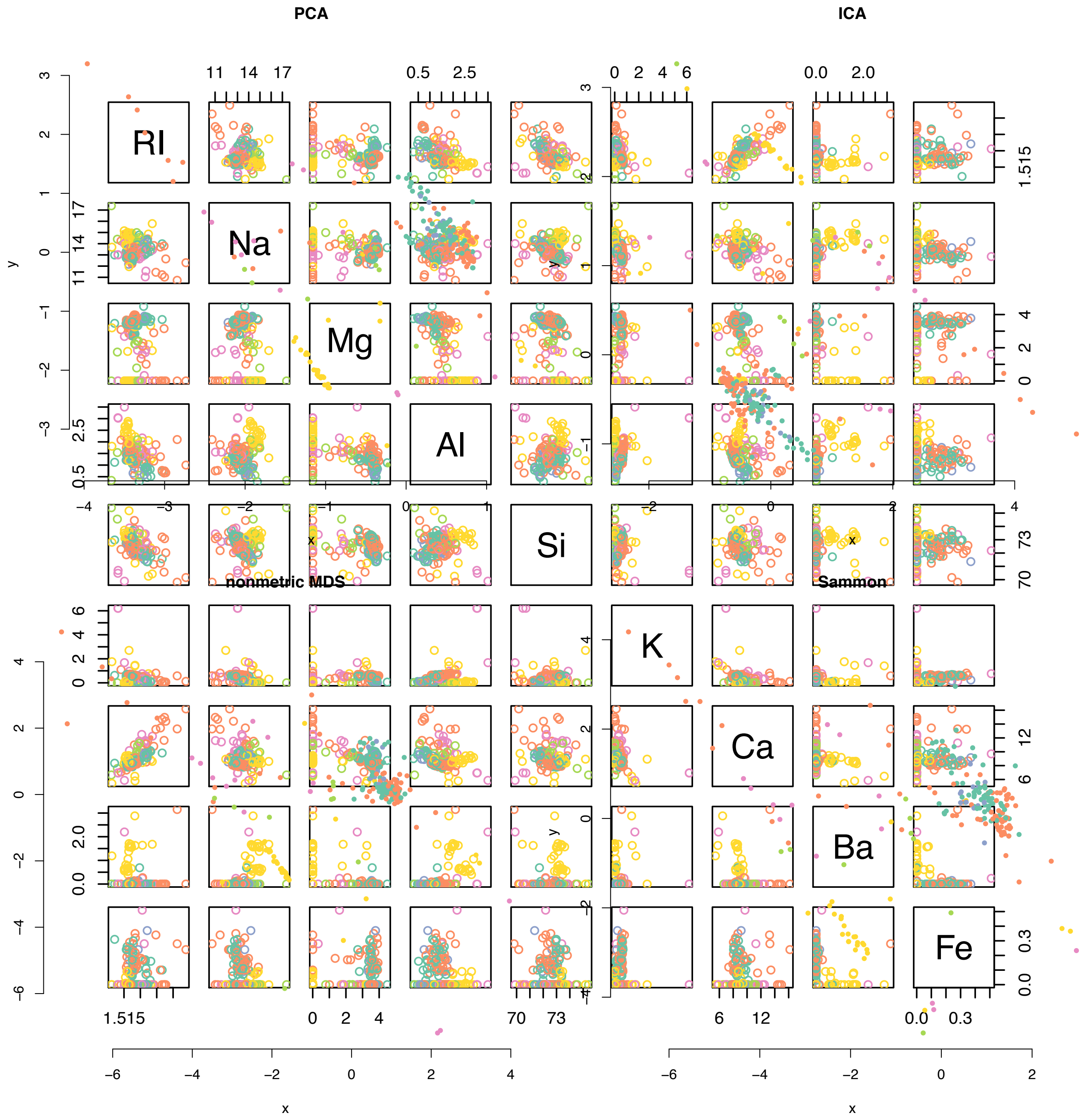
Municipal elections in Espoo in 2017



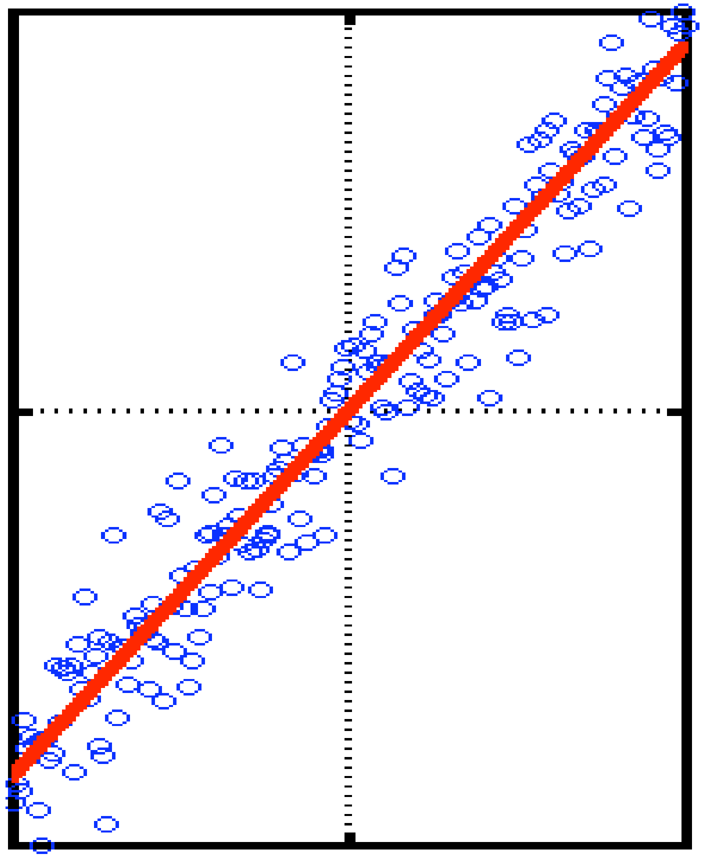


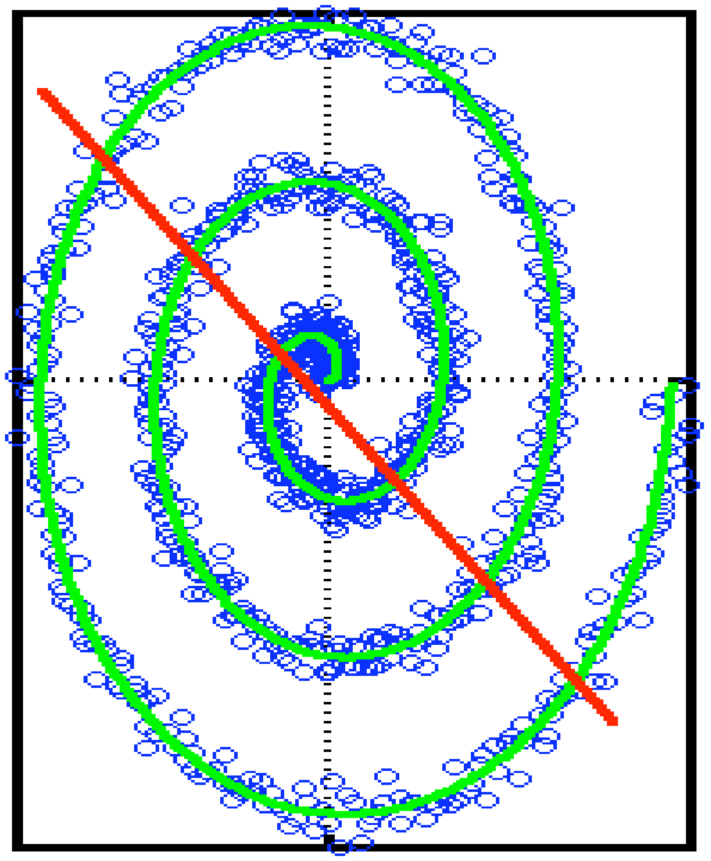
Glass data

- 9D glass identification database
- PCA always projects close-by points close to each other, resulting to reasonable recall
- However, PCA (and MDS) may also “collapse” far away data points into the same location (unless the data lies within low-dimensional linear subspace of the original space), this may lead to not so good precision



Next lecture: visualising manifolds





- The first principal component is given by the red line.
  The green line on the right gives the “correct” non-linear dimension (which PCA or ICA is of course unable to find).