Data Reduction

QUANTI 2 · Session 12

François Briatte

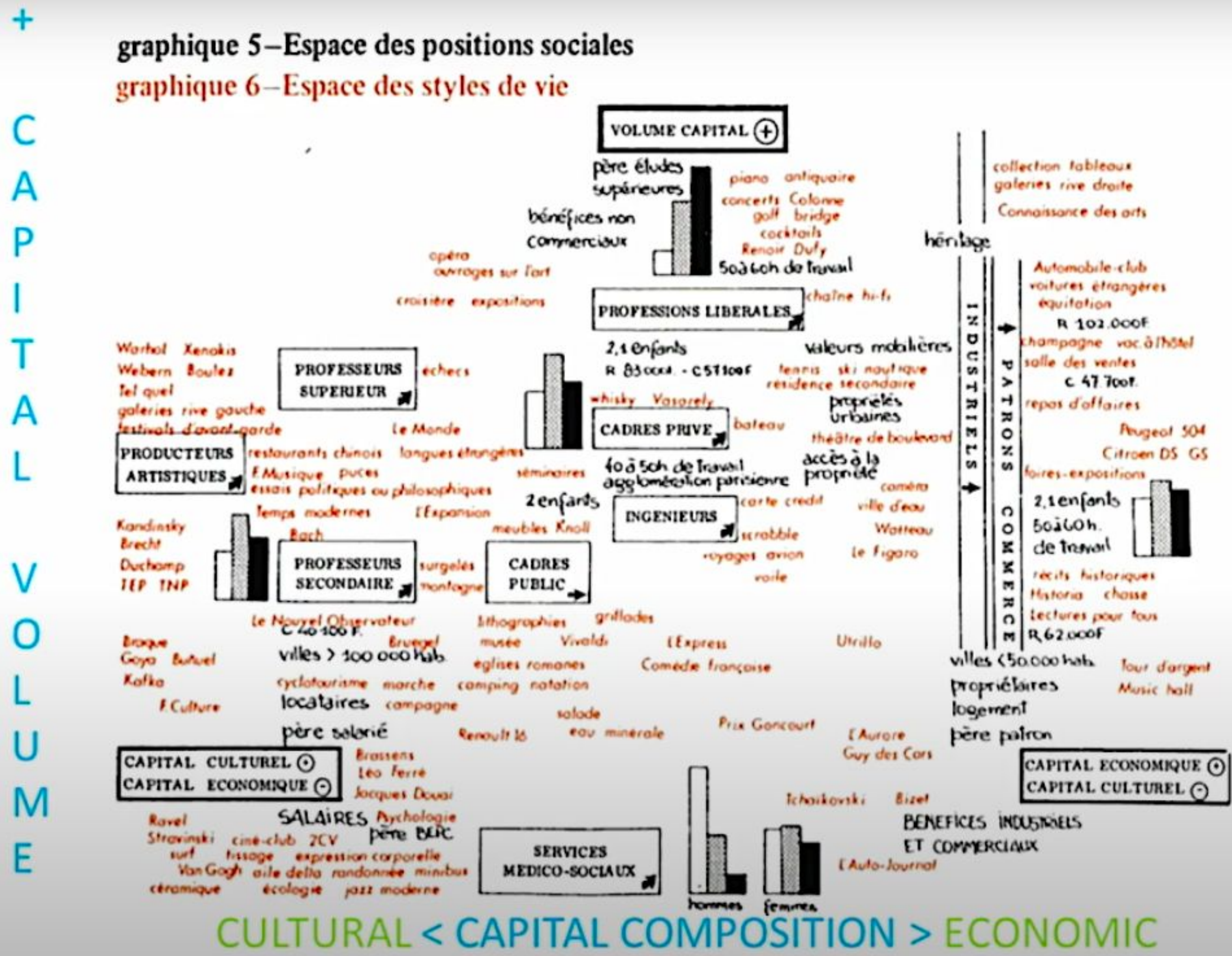
What this is about

- So far, what you have done in this course is a form of **supervised learning** about your data, using a specific set of variables to predict a response
- **Unsupervised learning**, which is often used for exploratory data analysis, lets the data speak for itself by looking for its **latent, non-random structure**
- Two broad approaches: (1) **clustering**, which looks for **partitions** in the data, and (2) **dimension reduction**, which looks for a **simpler version** of the data space

Examples

using **Multiple Correspondence Analysis** (MCA)

Mike Savage: The Importance of Class in an Age of Inequality (Mosse Lecture vom 09.01.2020)



37:48 / 1:37:50

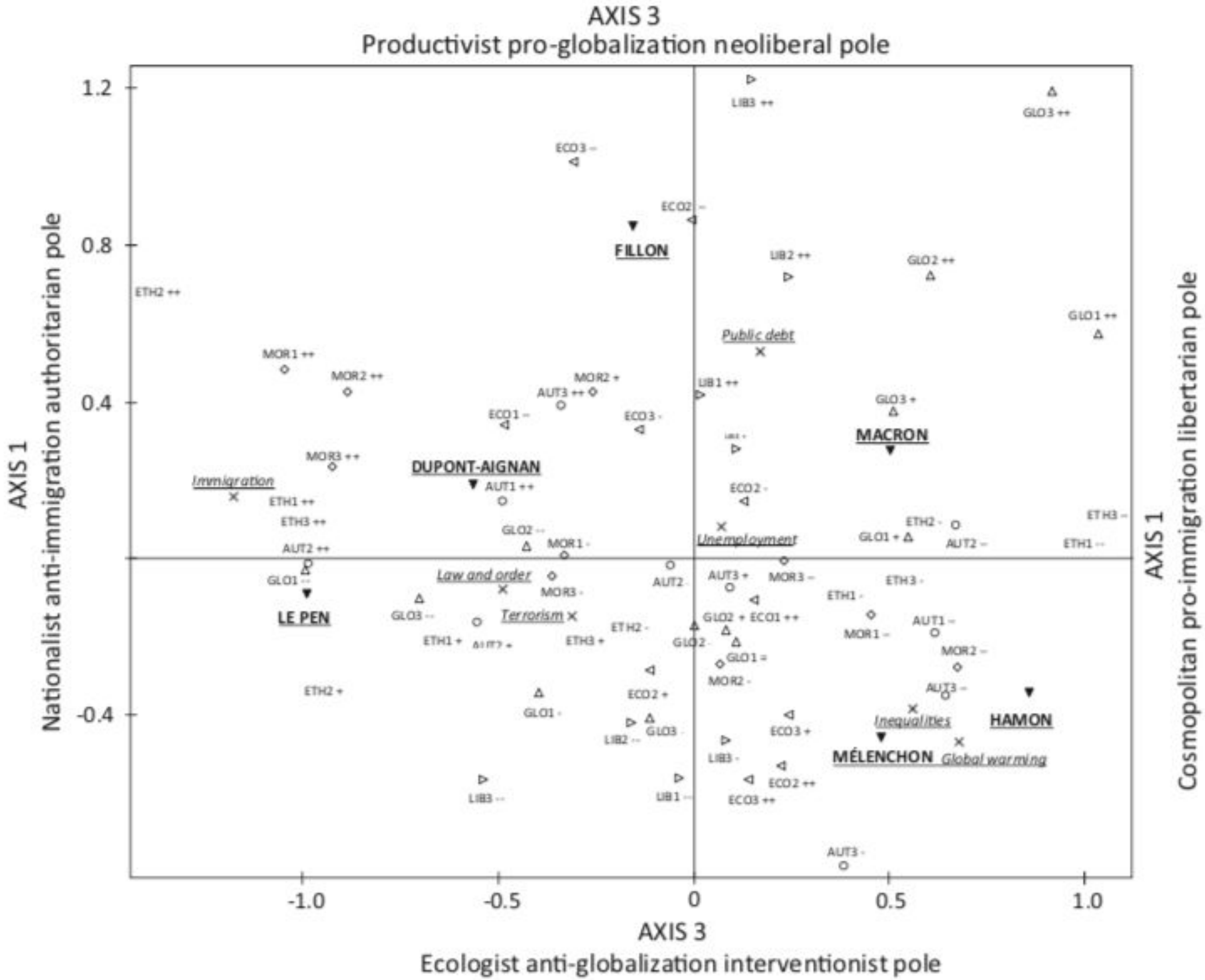


Fig. 2 Multiple correspondence analysis of voters' values (active variables) and vote choices (illustrative variables) in plane 1–3

Clustering
(a.k.a. partitioning)

Measuring distance between data points

Euclidean (*L2*) distance

Manhattan (*L1*) distance

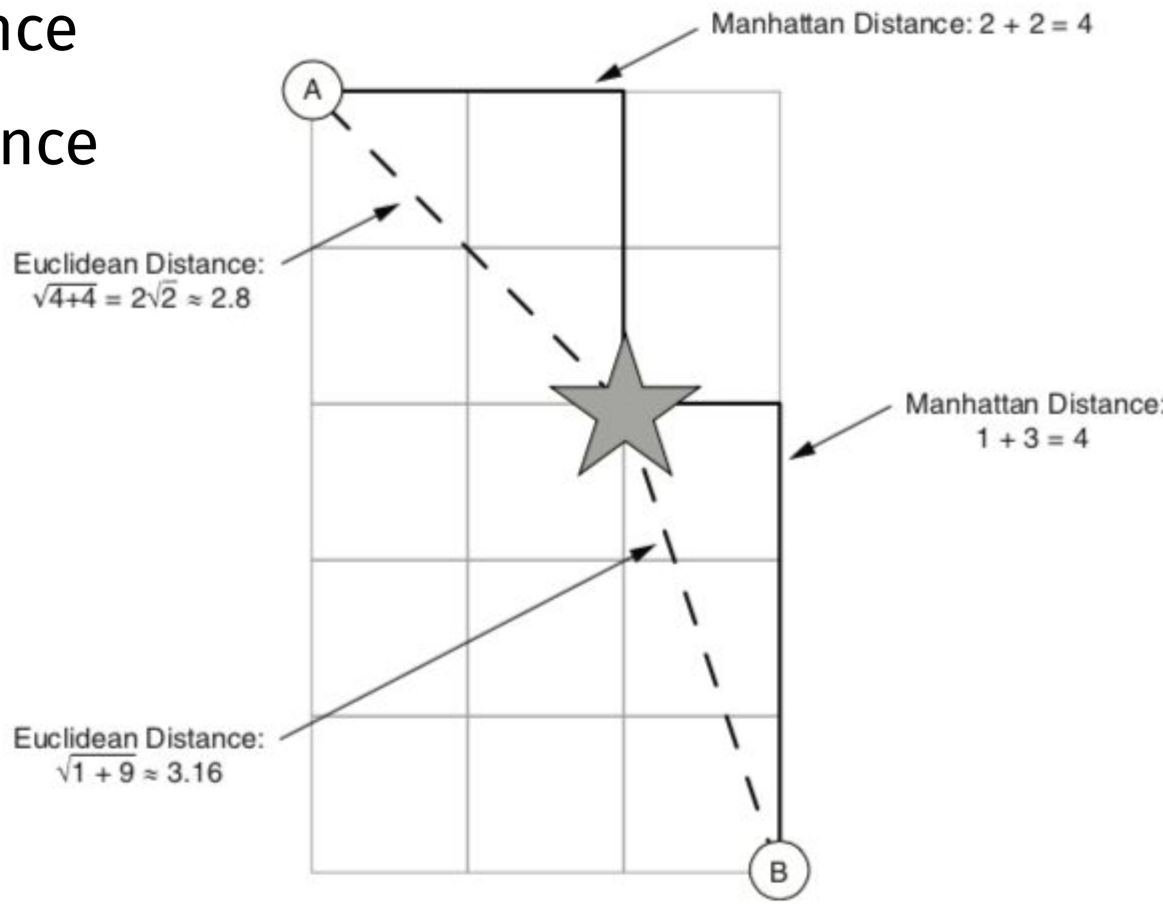
(not shown)

Hamming distance

Pearson distance

Cosine similarity

...



Euclidean distance

(square root of squared distances)

$$d_{euclidean}(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}.$$

Pearson distance

(based on correlation coefficients)

$$d_{pearson}(p,q) = 1 - \frac{\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2 \sum_{i=1}^{n}(q_i - \bar{q})^2}}.$$

Steps to find clusters

- **Use what you know** how to use: correlations, heatmaps, scatterplot matrices… (whatever works for you)
- **Scale your data** to a mean of 0 and standard deviation of 1 to make the variables (roughly) comparable
- **Get a distance matrix** using a metric that makes sense given the nature of the data (use Euclidean as default)
- **Use an algorithm to ‘optimize’ the matrix**, which often means minimizing or maximizing one of its metrics (as in: minimizing the *RSS*, or maximizing the *LL*)

Two example methods

- **Hierarchical clustering**, for when you do not know how many partitions you want

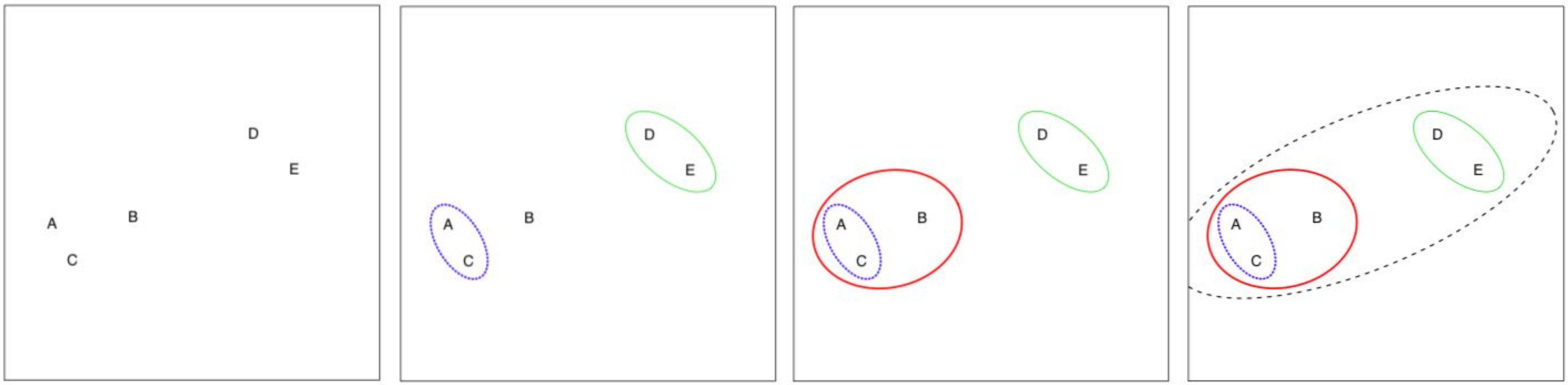
  → Results in a 'tree-like' representation of the data (a dendrogram) that you can 'cut' into clusters

- ***K*-means clustering**, for when you actually know how many partitions you want

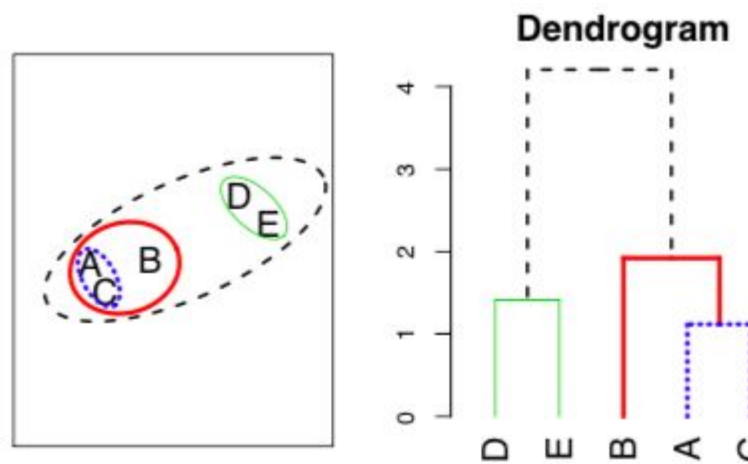
  → Results directly into data partitions

*There are **hundreds** of similar partitioning algorithms*

Hierarchical clustering



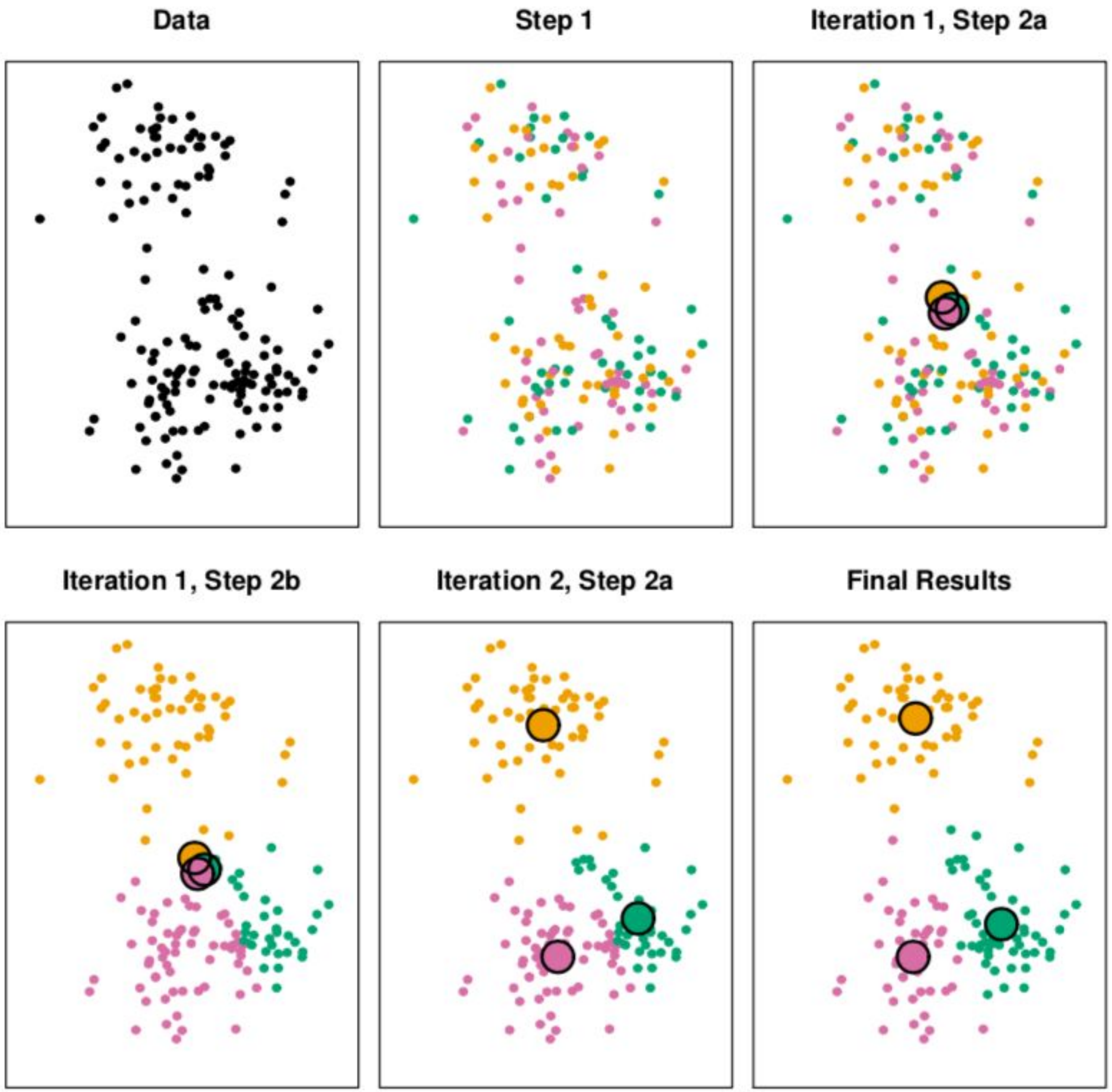
Iteratively group *N* observations from ‘bottom’ (*N* clusters) to ‘top’ (single cluster) using distance (closeness/similarity) measure



Figures from Sanchez (2018)

K-means clustering

Start with k random clusters, find their **centroids** (i.e. their geometric centers), assign observations to nearest centroid (using Euclidean distance), and repeat i times



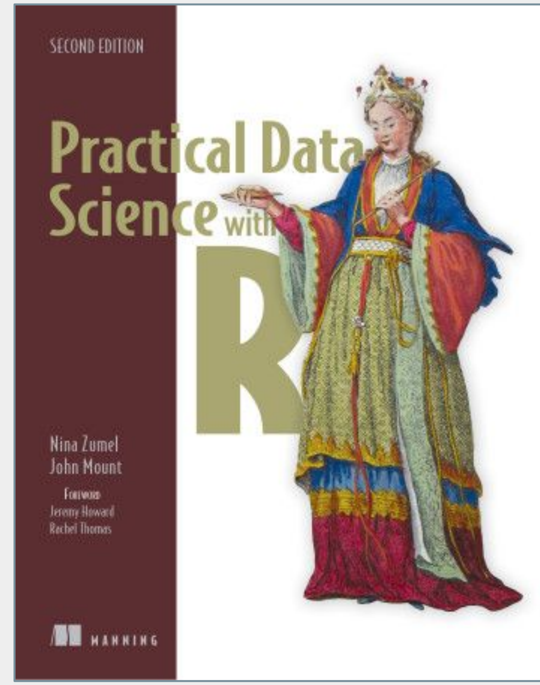
Figures from Sanchez (2018)

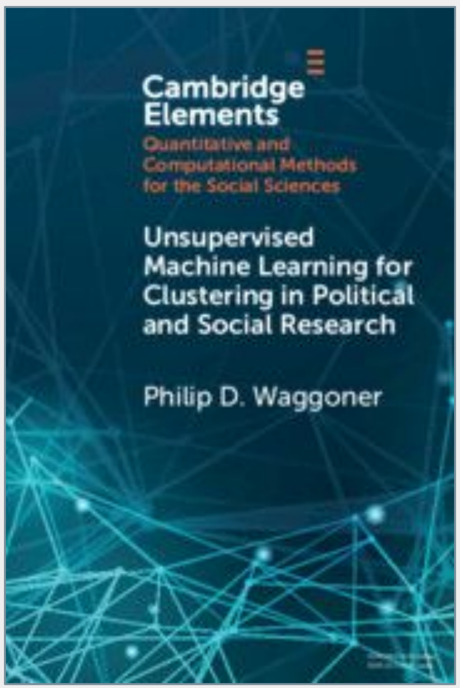
What you will get

- Clustering leaves you with a **small number of partitions** to explore and describe — use that in conjunction with descriptive statistics on each cluster
- Different clustering methods (or different parameters) will result in **different clusters** — compare and select your best result: there is no ‘standard rule’
- **Visualize the results** as much as possible — clustering is a form of exploratory data analysis, so plotting its output is a key aspect of the task

Example script: `01-clustering.r`

Protein consumption data fetched from Zumel and Mount 2019, ch. 9





For full treatment, see Waggoner 2020, ch. 2—4 in particular

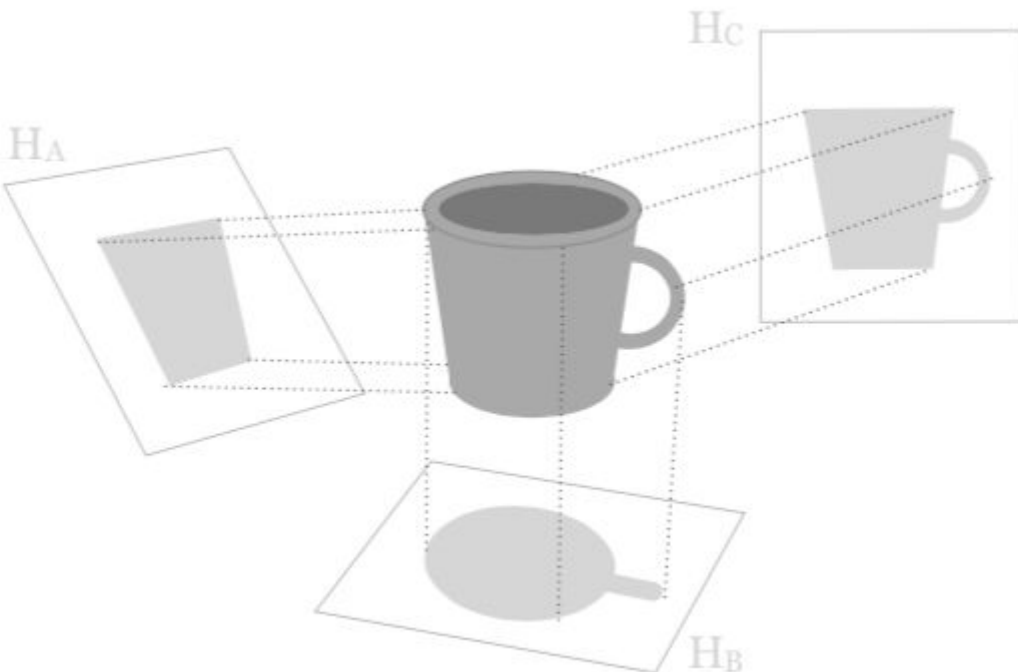
Free version

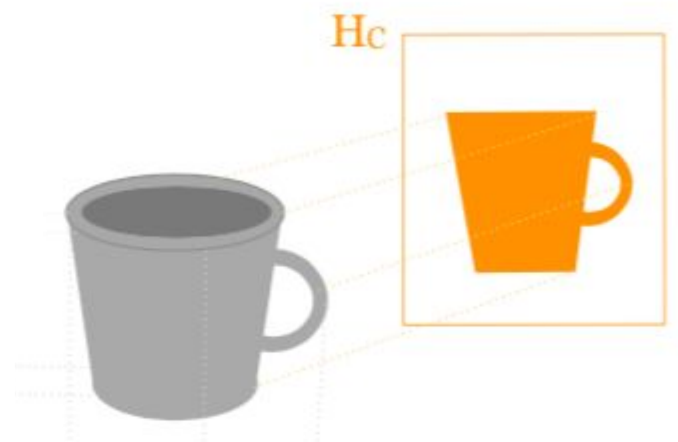
Principal components analysis (PCA)

Principal components (PCs) in brief

- Objective: get **low-dimensional data**, obtained through linear combinations of the variables that have maximal variance (preserve/explain diversity in original data) and are mutually uncorrelated (i.e. orthogonal)
- Method: **singular value decomposition** (SVD) to find the PC loading vectors, which is equivalent to finding the hyperplane that is nearest to all observations (using average squared Euclidean distance)
- As with clustering, (1) the variables should be **rescaled**, and (2) there are myriads of PCA-like methods

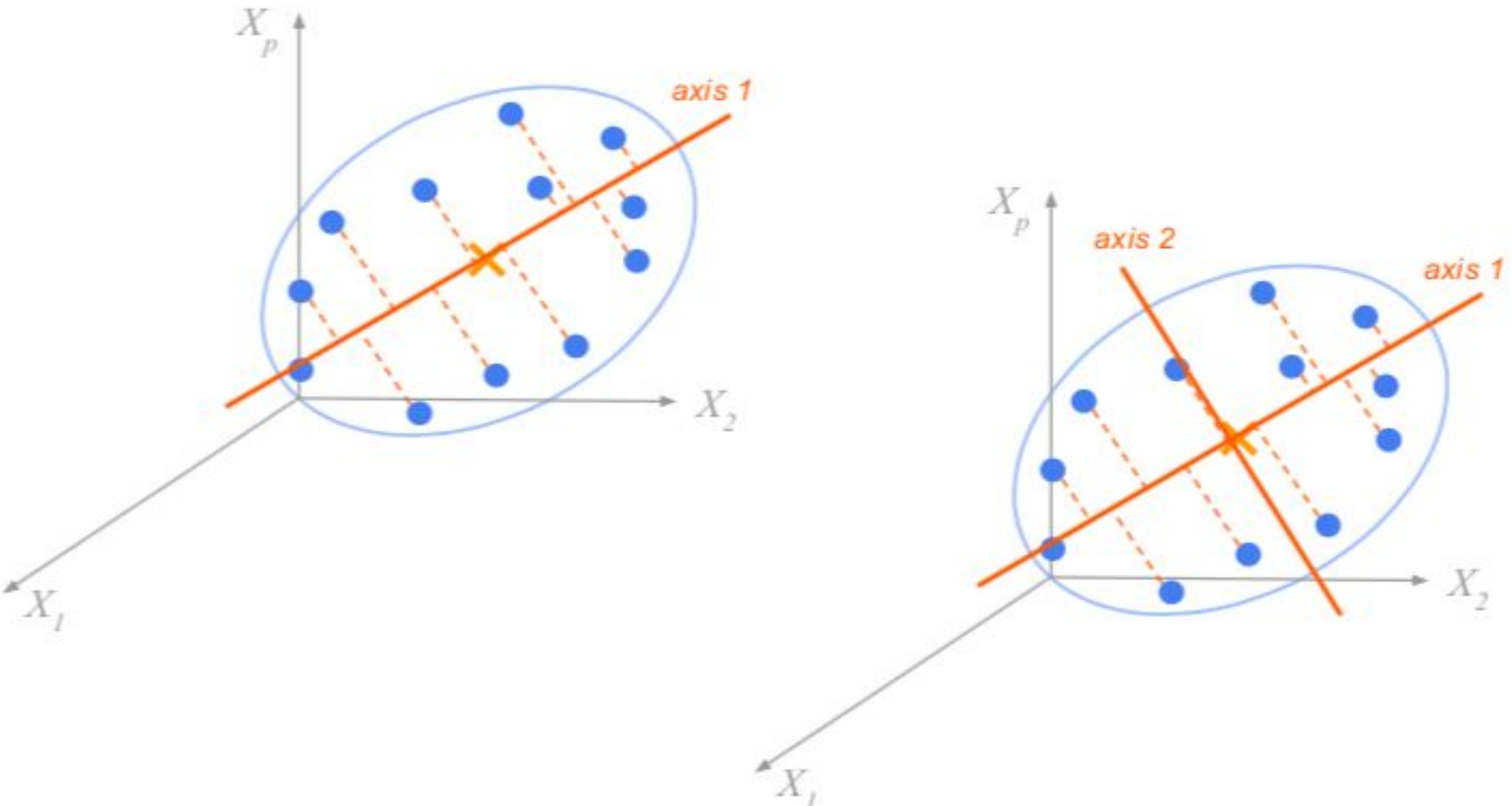
Best low-dimensional subspace among candidates





Figures from Sanchez (2018)

Axes (i.e. PCs) retain as much variance as possible



Figures from Sanchez (2018)

Relationship to linear regression

PCA looks for a 1-dimensional linear component score that describes the (x, y) coords of the data points, capturing as much of the **total variance** (a.k.a. inertia) of x and y

The **proportion of variance explained** (PVE) describes how successful it is at doing so

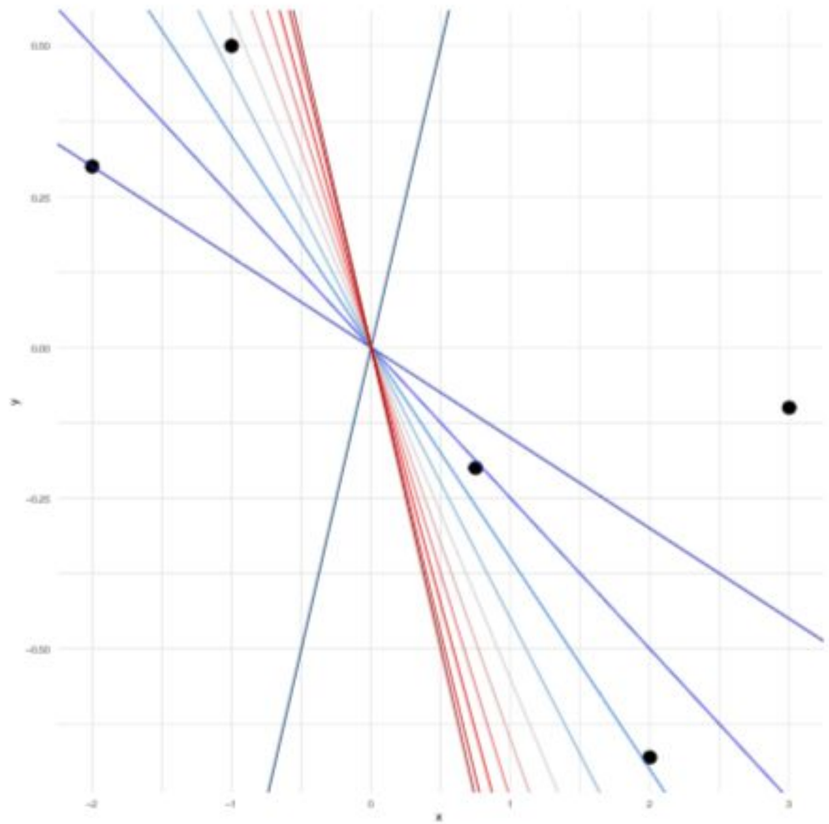


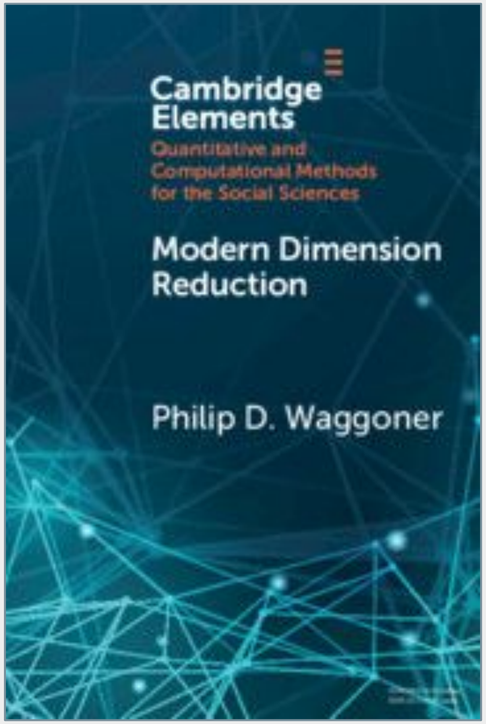
Figure from Waggoner (2020)

Related methods and extensions

- Related methods like **correspondence analysis** and **factor analysis** are very common in some disciplines (e.g. Confirmatory Factor Analysis in psychology)
- Other dimensionality reduction methods (algorithms) can produce even better separated results
  - **t-SNE**, which (unlike PCA) is non-deterministic, and more computationally expensive
  - **UMAP**, which is also non-deterministic, but faster (and arguably better) than t-SNE
  - **DBSCAN** is another well-known, time-tested method

Example script: `02-pca.r`

French electoral survey data from the Comparative National Elections Project (CNEP)



Code based on Waggoner, *Modern Dimension Reduction*, ch. 2

Full code and data

Useful PCA/etc. resources

Useful starting points

- Sanchez and Marzban, *All Models Are Wrong: Concepts of Statistical Learning* (2020), esp. ch. 4

  The book to take a look at — also covers several of the other methods taught through this course

- Documentation pages and tutorials for the FactoMineR and factoextra packages

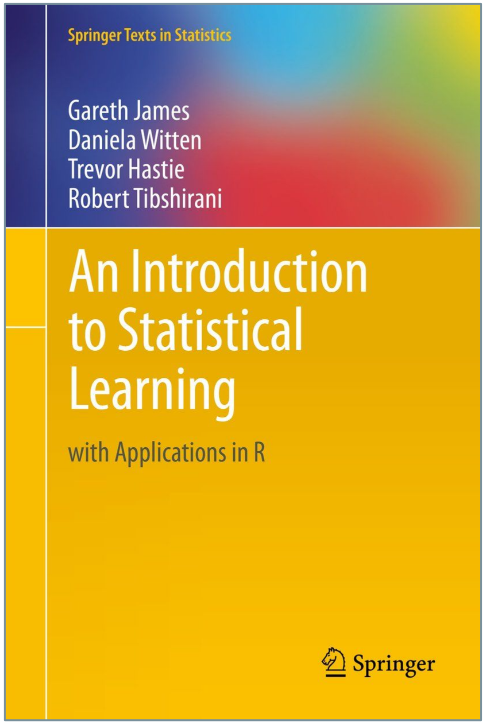
  Available both in English and in French

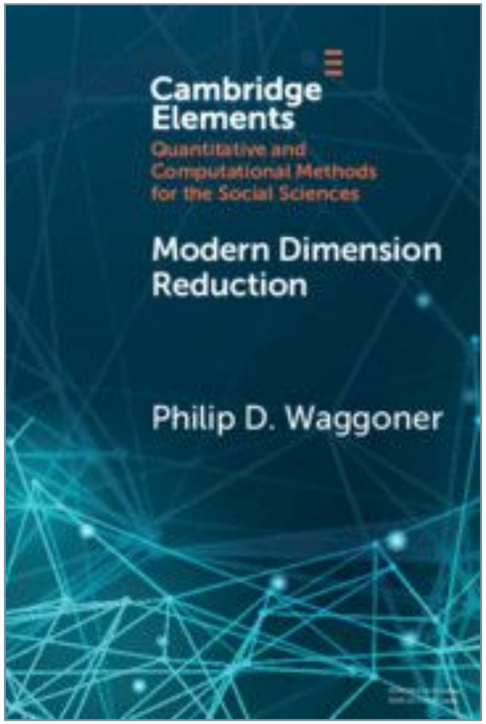
- StatQuest videos on PCA *et al.* (Josh Starmer)

  Video explainers of PCA, *k*-means, UMAP, and more

Useful books

James *et al.* *An Introduction to Statistical Learning* · ch. 10





Waggoner

Modern Dimension Reduction · ch. 2

Free preprint

Full code and data

FACTOMINER

About FactoMineR

FactoMineR is an **R** package dedicated to multivariate Exploratory Data Analysis. It is developed and maintained by François Husson, Julie Josse, Sébastien Lê, d'Agrocampus Rennes, and J. Mazet.

Why Use FactoMineR?

1. It **performs classical principal component methods**: Principal Components Analysis (PCA), Correspondence analysis (CA), Multiple Correspondence Analysis (MCA), clustering
2. as well as advanced methods that take into account a **structure on the data** (groups of variables, hierarchy on the variables, groups of individuals).
3. It allows to **add supplementary informations** such as supplementary individuals and/or variables.
4. It provides a geometrical point of view, a lot of graphical outputs, helps to interpret (automatic description of the dimensions, various indicators, ...).
5. Lot of materials (**MOOC, books**, etc.) is available to explain the methods and the way to implement them in FactoMineR.
6. It **handles missing values** with **missMDA** (see here).
7. It has a **a GUI with a Shiny interface that draws interactive graphs** with **Factoshiny** (see here)
8. It gives **automatic interpretation** of the results with **FactoInvestigate** (see here).

Home Menu

- FactoMineR's description
- News
- Install FactoMineR
- How to cite FactoMineR?
- History of FactoMineR

Authors

- François Husson
- Julie Josse
- Sébastien Lê

Useful Links

AGRO CAMPUS

factoextra : Extract and Visualize the Results of Multivariate Data Analyses

factoextra is an R package making easy to *extract* and *visualize* the output of exploratory **multivariate data analyses**, including:

1. Principal Component Analysis (PCA), which is used to summarize the information contained in a continuous (i.e, quantitative) multivariate data by reducing the dimensionality of the data without loosing important information.
2. Correspondence Analysis (CA), which is an extension of the principal component analysis suited to analyse a large contingency table formed by two *qualitative variables* (or categorical data).
3. Multiple Correspondence Analysis (MCA), which is an adaptation of CA to a data table containing more than two categorical variables.
4. Multiple Factor Analysis (MFA) dedicated to datasets where variables are organized into groups (qualitative and/or quantitative variables).
5. Hierarchical Multiple Factor Analysis (HMFA): An extension of MFA in a situation where the data are organized into a hierarchical structure.
6. Factor Analysis of Mixed Data (FAMD), a particular case of the MFA, dedicated to analyze a data set containing both quantitative and qualitative variables.

There are a number of R packages implementing principal component methods. These packages include: *FactoMineR*, *ade4*, *stats*, *ca*, *MASS* and *ExPosition*.

However, the result is presented differently according to the used packages. To help in the interpretation and in the visualization of multivariate analysis - such as cluster analysis and dimensionality reduction analysis - we developed an easy-to-use R package named factoextra.

More awesome resources

Superb graphical approach to the topic
The Beginner's Guide to Dimensionality Reduction
(Matthew Conlen and Fred Hohman)

Clear write-up on PCA, with code
Understanding PCA using Stack Overflow data (Julia Silge)

Video tutorial, using `tidymodels` code
Dimensionality reduction of UN voting patterns (Julia Silge)

Interactive package for PCA/MCA and more
explor (Julien Barnier)

Thanks for your attention

and best of luck in all your future endeavours