Consensus Clustering in Stata

Carlo Drago
Università Niccolò Cusano

September 21, 2025

Outline

Clustering: Definition and Importance

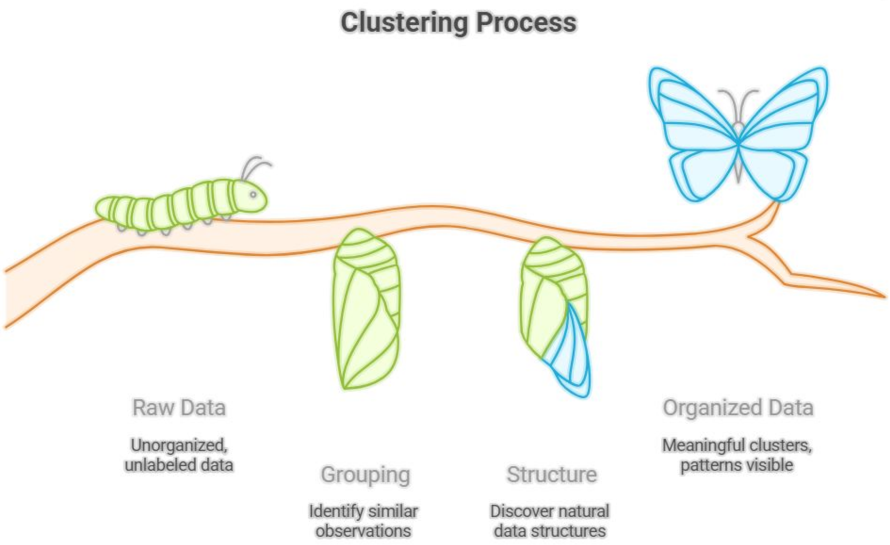
- **Clustering** is a basic method of unsupervised learning that aims to **group data into sets of similar observations without relying on predefined labels**.
- The goal **is to discover natural structures in the data** by forming groups in which the elements are more similar to each other than in other groups.
- This method is important because it helps to **simplify complex data sets, making patterns more visible and easier to interpret**. By organizing data into meaningful clusters, researchers and practitioners can **identify hidden structures, detect subpopulations and form new hypotheses**.
  See Zhang et al. (2023)

Clustering: Definition and Importance

- Clustering also supports decision-making in many fields, such as economics, biology and social sciences, by enabling segmentation, profiling and the identification of **data structures** that would otherwise remain hidden in the raw data.
- As the first step in many analytical pipelines, clustering provides both insight and a basis for further statistical modeling, prediction or policy analysis.

See Jaeger and Banks (2023)

Definition and Importance



Clustering: Problem Statement and Objectives

Given observations $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, unsupervised clustering seeks a partition $\mathcal{C} = \{C_1, \dots, C_K\}$ with $C_c \neq \varnothing$, $C_c \cap C_{c'} = \varnothing$ $(c \neq c')$, and $\bigcup_c C_c = X$, so that within-cluster similarity is large and between-cluster similarity is small. A canonical objective is the minimization of empirical risk built from a dissimilarity $d(\cdot, \cdot)$, with constraints enforcing disjointness and coverage. For **partitioning methods** with prototype $\mu_c \in \mathbb{R}^p$, the classical criterion is

$$\min_{\{\mu_c\}, \mathcal{C}} \sum_{c=1}^{K} \sum_{i \in C_c} \|x_i - \mu_c\|_2^2,$$

whose minimizers under Lloyd's alternation yield the k-means algorithm.

Clustering: Problem Statement and Objectives

More generally, **hierarchical methods** optimize a greedy merge cost, **density-based methods** seek connected high-density regions, and **model-based methods** maximize likelihood under finite mixtures. The choice of K, the metric, and the representation are integral parts of the problem specification rather than post hoc details.

Methodological Families

So **partitioning methods** assign every x_i to one of K clusters by optimizing a global objective; k-means uses Euclidean prototypes while k-medoids minimizes $\sum_c \sum_{i \in C_c} d(x_i,\, m_c)$ for medoids $m_c \in C_c$. **Hierarchical agglomeration** starts from singletons and merges pairs (A, B) minimizing a linkage functional, such as Ward's increase in within-cluster sum of squares (WCSS), or average/complete linkages defined via pairwise distances. **Density-based procedures** like DBSCAN define clusters as ε-connected components of high-density neighborhoods, recovering arbitrarily shaped sets and isolating noise. **Model-based approaches** posit $x_i \overset{\text{i.i.d.}}{\sim} \sum_{c=1}^{K} \pi_c \, \mathcal{N}(\mu_c, \Sigma_c)$ and estimate (π_c, μ_c, Σ_c) by EM; the partition is the MAP allocation. **Graph- and spectral-based methods** embed a similarity graph through the leading eigenvectors of a Laplacian and partition in the reduced space, improving separability when boundaries are non-convex. See Jain, et al. (1999)

Methodological Families

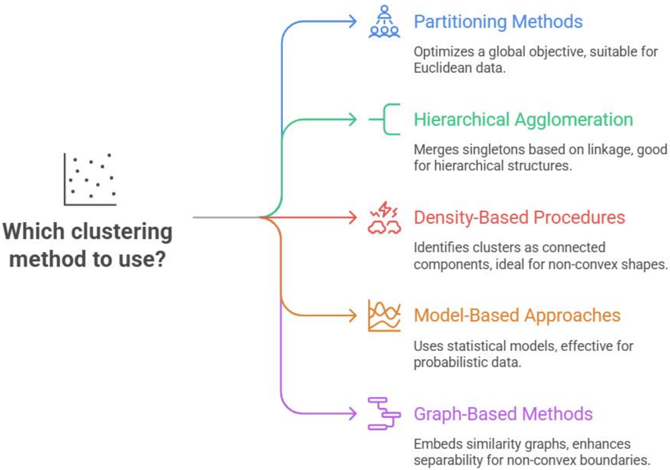
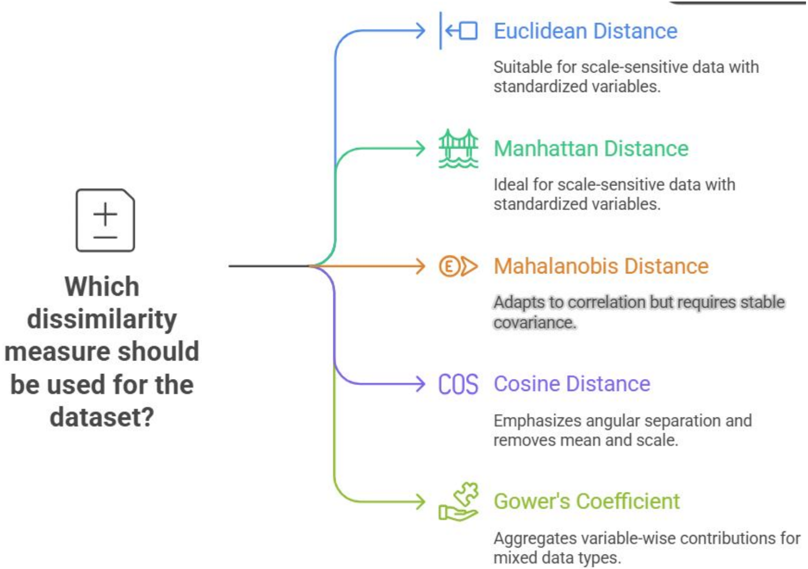


Figure: Methodological Families

Dissimilarities and Data Types

The dissimilarity d controls geometry. **Euclidean** $d_2(x, y) = \|x - y\|_2$ and **Manhattan** $d_1(x, y) = \|x - y\|_1$ are scale-sensitive and typically used with standardized variables $z = (x - \bar{x})/s$. **Mahalanobis** $d_M(x, y) = \|(x - y)\|_{\Sigma^{-1}}$ adapts to correlation but requires stable covariance. **Cosine distance** $1 - \frac{x^\top y}{\|x\|\|y\|}$ emphasizes angular separation; correlation distance removes mean and scale. For mixed types, **Gower's** coefficient aggregates variable-wise contributions with type-aware normalizations; medoid- or hierarchical-based procedures are then appropriate. **Preprocessing choices** (centering, scaling, Box–Cox transforms, robust ranks) have first-order impact on d; justification must be data-driven and reported alongside results to ensure reproducibility. See Kaufman & Rousseeuw (2009), Kumar et al. (2014)

Dissimilarities and Data Types



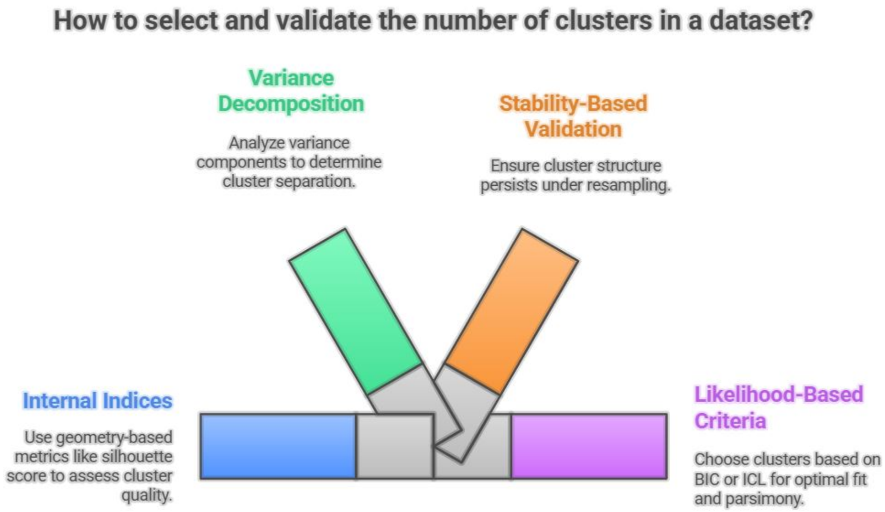
Model Selection and Validation

Selecting K and assessing structure use complementary criteria.
Internal indices exploit geometry: the silhouette of x_i is
$s_i = \dfrac{b_i - a_i}{\max\{a_i, b_i\}} \in [-1, 1]$, where a_i is intra-cluster dissimilarity and b_i the smallest average dissimilarity to another cluster. Variance decomposition on standardized variables yields $\mathrm{TSS} = \mathrm{WCSS} + \mathrm{BCSS}$ and $R^2 = \mathrm{BCSS}/\mathrm{TSS}$, while the Calinski–Harabasz index is

$$\mathrm{CH}(K) = \frac{\mathrm{BCSS}/(K-1)}{\mathrm{WCSS}/(n-K)}.$$

Stability-based validation probes whether structure persists under resampling, perturbations, or random restarts; consensus clustering operationalizes this by aggregating partitions into a co-association matrix. When a likelihood is available (mixtures), K may be chosen by BIC or ICL, inherently trading fit and parsimony. See Rousseeuw (1987) and Caliński & Harabasz (1974).

Model Selection and Validation



High-Dimensional Settings and Interpretability

In high dimensions, distances concentrate and spurious separation arises. Dimensionality reduction via PCA, factor models, or sparse projections can restore contrast, while feature selection guided by variance, mutual information, or domain knowledge improves stability. Interpretability leverages prototypes (means, medoids) and cluster profiles on standardized scales, effect plots along principal directions, and post-clustering inference such as permutation tests on profile differences. **Uncertainty** is summarized by soft assignments (mixtures), posterior credible allocations, or stability scores derived from resampling. It is necessary to take into the account these issues in the analysis. See Steinbach et al. (2004)

Clustering in Stata: Built-ins and Workflow

Stata provides partitioning and hierarchical clustering end-to-end. The command `cluster kmeans` implements k-means with random or user-supplied starts and Euclidean geometry on a specified varlist, while hierarchical procedures such as `cluster wardslinkage` operate directly on variables and `clustermat` variants operate on a user-supplied dissimilarity matrix. A reproducible workflow standardizes variables, fixes the random seed, fits alternative K, and reports `list`/`tabstat` summaries, variance decomposition (WCSS, BCSS, R^2), and dendrograms labeled with human-readable identifiers. When a problem-specific dissimilarity is required, the matrix route `clustermat ... , add` permits hierarchical clustering without discarding the dataset in memory.

Clustering in Stata

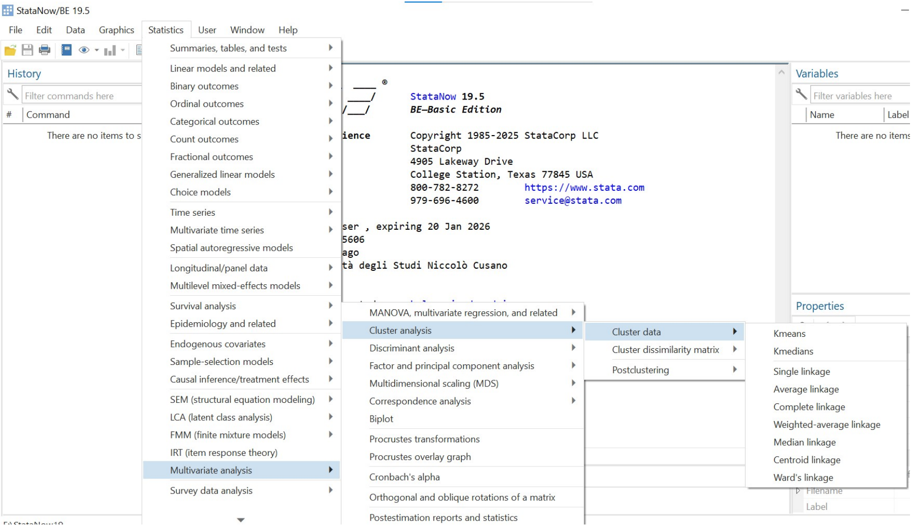


Figure: Clustering in Stata

Clustering in Stata: Minimal Reproducible Example

```
* Standardize, then run k-means and Ward's hierarchical
set seed 12345
cluster kmeans z_murder z_assault z_urbanpop z_rape, ///
    k(4) start(krandom) iterate(100) name(km4) measure(L2)
cluster list km4

cluster wardslinkage z_murder z_assault z_urbanpop z_rape, name(hc)
cluster generate cut4 = groups(4), name(hc)
cluster dendrogram hc, labels(state_name) cutnumber(4) showcount

* Variance decomposition and CH index on standardized vars
tempname W B
scalar `W' = 0
scalar `B' = 0
foreach v in z_murder z_assault z_urbanpop z_rape {
    quietly summarize `v'
    scalar gmean = r(mean)
    forvalues c=1/4 {
        quietly summarize `v' if cut4==`c'
        if r(N)>0 {
            scalar `W' = `W' + (r(N)-1)*r(Var)
            scalar `B' = `B' + r(N)*(r(mean)-gmean)^2
        }
    }
}
scalar TSS = `W' + `B'
scalar R2  = `B'/TSS
scalar CH  = (`B'/(4-1))/(`W'/(c(N)-4))
di as res "R^2=" %5.3f R2 "   CH=" %6.2f CH
```

Introduction to Consensus Clustering

- Consensus clustering aggregates multiple base partitions to stabilize the final solution against initialization noise and sampling variability.
- In the small-n, moderate-d settings typical of socio-economic datasets, the approach reduces instability without imposing strong modeling assumptions.
- The key construct is the co-association (consensus) matrix that records how often two units co-cluster across bootstrap replicates and serves as the basis for a final hierarchical aggregation.

See Fred & Jain (2005), Monti et al. (2003), Strehl & Ghosh (2002) Vega-Pons & Ruiz-Shulcloper (2011)

Introduction to Consensus Clustering

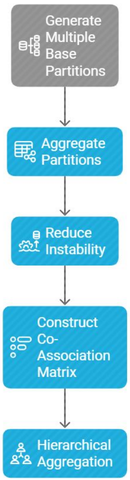


Figure: Consensus Clustering

Motivation: Why Consensus Clustering?

- Classical k-means is sensitive to starting centroids and can yield materially different partitions across runs.
- Bootstrapping the data and aggregating the resulting partitions attenuates this instability and provides an interpretable, data-driven measure of pairwise association that can be converted into a dissimilarity for hierarchical methods, thereby combining the strengths of partitioning and agglomeration.

Consensus Matrix: Correct Definition

Let $X = \{x_1, \ldots, x_n\}$ and let $X^{(b)}$ be the b-th bootstrap sample ($b = 1, \ldots, B$). Denote by $C^{(b)}$ the base partition on $X^{(b)}$. The consensus (co-association) entry for units i and j is

$$M_{ij} = \frac{\sum_{b=1}^{B} \mathbf{1}\{i, j \in X^{(b)}\}\, \mathbf{1}\{C^{(b)}(i) = C^{(b)}(j)\}}{\sum_{b=1}^{B} \mathbf{1}\{i, j \in X^{(b)}\}},$$

that is, the frequency of co-clustering *conditional on co-presence*. If the denominator is zero (the pair is never co-sampled), M_{ij} is undefined at this stage and the subsequent distance convention sets $D_{ij} = 1$ (maximal dissimilarity). We enforce $M_{ii} = 1$.

Consensus Clustering Algorithm (Overview)

- We repeatedly draw bootstrap samples, run k-means on standardized variables, record cluster memberships for the sampled units (collapsing multiplicities to presence), build M with the co-sampling denominator, and convert M into a dissimilarity $D = 1 - M$ with $D_{ii} = 0$ and never co-sampled pairs set to $D_{ij} = 1$.
- Ward's linkage is then applied to the dissimilarity matrix via `clustermat`, and the final cut yields the consensus partition.

Mathematical Formulation

- Given B bootstraps and a base algorithm producing $C^{(b)}$, the final dissimilarity fed to hierarchical clustering is

$$D_{ij} = \begin{cases} 1 - M_{ij}, & \text{if } \sum_b \mathbf{1}\{i, j \in X^{(b)}\} > 0, \\ 1, & \text{otherwise,} \end{cases} \qquad D_{ii} = 0.$$

  Ward's criterion minimizes the increase in total within-cluster sum of squares (WCSS) at each merge.
- After cutting the dendrogram at K groups, we report WCSS, BCSS, TSS, and $R^2 = \mathrm{BCSS}/\mathrm{TSS}$ computed on standardized variables.

Diagnostics for Selecting K in Consensus Clustering

Let $M^{(K)}$ be the consensus matrix obtained from B resamples at a given K. A global diagnostic is the empirical CDF of off-diagonal $M_{ij}^{(K)}$; sharper concentration near $\{0,1\}$ indicates clearer separation. The *delta area* criterion compares the area under the CDF between successive K and selects the smallest K after which gains saturate. Cluster-level stability is summarized by

$$\bar{M}_c = \frac{1}{|C_c|(|C_c|-1)} \sum_{\substack{i,j \in C_c \\ i \neq j}} M_{ij},$$

and item-level stability by $\bar{M}_i = \frac{1}{|C_{c(i)}|-1} \sum_{j \in C_{c(i)}, j \neq i} M_{ij}$. Reporting $\bar{M}_c$ and the distribution of $\bar{M}_i$ allows targeted inspection of weakly attached units and fragile clusters, complementing dendrogram cuts.

Statistical Properties and Caveats

- With bootstrap resampling and randomized base algorithms, the co-association entry M_{ij} estimates the probability that i and j co-cluster *conditional on co-presence*.
- Converting to $D = 1 - M$ and using Ward's linkage yields a Euclidean-consistent aggregation, but M can mask multi-membership structure when data admit overlapping or manifold clusters. Stability should therefore be triangulated with internal indices and, where appropriate, external information. Choices for never co-sampled pairs (set $D_{ij} = 1$) and $D_{ii} = 0$ must be stated explicitly to ensure reproducibility and interpretability.

Implementation Strategy in Stata

- The workflow is purely in Stata. Variables are standardized to z-scores once per run with idempotent safeguards.
- Bootstrap resamples are generated with `bsample`.
- For each draw, k-means is fitted with random initialization, and memberships are stored in wide format as `boot1, ..., bootB` after collapsing multiplicities to binary presence.
- The consensus matrix is computed with the co-sampling denominator, then transformed to $D = 1 - M$ with $D_{ii} = 0$ and missing entries set to one.
- Ward's hierarchical clustering is applied to the *matrix* D via `clustermat ... , add` while the dataset remains in memory, and dendrograms are labeled by the string variable `state_name`.

Algorithm (1/4): Setup and Standardization

Algorithm 1: *

Input: Data $X \in \mathbb{R}^{N \times p}$ (e.g. USArrests); bootstrap iterations $B \in \mathbb{N}$; clusters $K \geq 2$; seed s.

Output: Partition $\widehat{\mathcal{C}}$; consensus $M \in [0,1]^{N \times N}$; distance $D = 1 - M$; Ward dendrogram $\mathcal{H}$; WCSS, BCSS, R^2.

1 **Identifiers and reproducibility.** Ensure unique integer id id_i and readable label (`state_name`); set VERSION and SET SEED s.

2 **Standardization (idempotent).** For each variable $v = 1, \ldots, p$, compute

$$z_{iv} \leftarrow \frac{x_{iv} - \bar{x}_v}{s_v}, \quad \bar{x}_v = \frac{1}{N}\sum_{i=1}^{N} x_{iv}, \quad s_v = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_{iv} - \bar{x}_v)^2}.$$

Persist a clean base dataset $B_0 = (id, \mathtt{name}, Z)$.

3 **Ensemble storage.** Create B label columns `boot1`,…,`bootB` initialized to NA; this table is keyed by id.

4 Notes. Stata correspondence: standardization via SUMMARIZE and GENERATE; reproducible base saved to disk.

Algorithm (2/4): Bootstrap and K-means

Algorithm 2: *

for $b = 1$ **to** B **do**

  Load B_0 and draw a size-N bootstrap sample with replacement using BSAMPLE.

  Run K-means on standardized features Z with K clusters and random starts:

    CLUSTER KMEANS Z, `k(`K`) start(krandom) measure(L2)`.

  For each unit i that appears at least once in draw b (presence-only), set the ensemble label $A_i^{(b)} \leftarrow$ its K-means assignment; otherwise $A_i^{(b)} \leftarrow$ NA.

  Merge the vector $\{A_i^{(b)}\}_{i=1}^{N}$ into the storage table by *id* (one column per b).

Remarks. Presence-only coding removes multiplicity bias from bootstrap re-selections; each draw induces a partial partition over the present units.

Algorithm (3/4): Consensus Matrix and Ward Linkage

Algorithm 3: *

1 **Exact consensus.** For each pair (i, j) define

$$T_{ij} = \#\{\, b : A_i^{(b)} \neq \text{NA} \ \wedge \ A_j^{(b)} \neq \text{NA} \,\}, \qquad C_{ij} = \#\{\, b : A_i^{(b)} = A_j^{(b)} \neq \text{NA} \,\}.$$

Set

$$M_{ij} = \begin{cases} C_{ij}/T_{ij}, & T_{ij} > 0, \\ \text{undef}, & T_{ij} = 0, \end{cases} \quad \text{and enforce } M_{ii} = 1.$$

Implement via nested loops over i, j scanning $b = 1, \ldots, B$.

2 **Distance for linkage.** Let $D_{ij} = 1 - M_{ij}$ where defined; set $D_{ij} = 1$ if $T_{ij} = 0$ (never co-sampled) and $D_{ii} = 0$.

3 Materialize dense D (e.g., MKMAT) and run Ward's method by CLUSTERMAT WARDSLINKAGE D with `, add name(final_hc)`.

4 Obtain a K-group cut with CLUSTER GENERATE `groups(`K`)` on `final_hc`, yielding $\widehat{\mathcal{C}}$.

5 Linkage design. $D = 1 - M$ respects evidence accumulation; pairs never co-sampled are treated as maximally dissimilar for agglomeration.

Algorithm (4/4): Validation and Outputs

Algorithm 4: *

Validation on standardized space. For each variable v compute grand mean $\bar{z}_v$ and, per cluster $c = 1, \ldots, K$, size N_c, mean $\bar{z}_{v,c}$, variance $\mathrm{Var}_c(z_v)$.

Accumulate

$$\mathrm{WCSS} = \sum_{v=1}^{p} \sum_{c=1}^{K} (N_c - 1)\,\mathrm{Var}_c(z_v), \qquad \mathrm{BCSS} = \sum_{v=1}^{p} \sum_{c=1}^{K} N_c \left(\bar{z}_{v,c} - \bar{z}_v\right)^2,$$

set $\mathrm{TSS} = \mathrm{WCSS} + \mathrm{BCSS}$ and $R^2 = \mathrm{BCSS}/\mathrm{TSS}$.

Artifacts. Save the raw consensus table with identifiers and M-columns; export final clusters joined to original and standardized variables; render dendrograms labeled by `name`; print WCSS, BCSS, R^2.

return $(\widehat{\mathcal{C}}, M, D, \mathcal{H}, \mathrm{WCSS}, \mathrm{BCSS}, R^2)$

Complexity. Ensemble K-means $O(BNpKI)$ (Lloyd iterations I) plus consensus accumulation $O(BN^2)$; Ward on dense D uses $O(N^2)$ space.

Case Study: USArrests Dataset

- The USArrests data comprise $n = 50$ US states described by four variables per 100,000 residents: murder, assault, percentage urban population, and rape.
- The question is whether stable and interpretable groups of states emerge under a consensus procedure that aggregates B bootstrap k-means partitions of the standardized variables.

USArrests: Variables and Definitions

- **Coverage.** 50 U.S. states; statistics refer to the early 1970s (primarily **1973**); `UrbanPop` reflects the share of residents in urban areas (1970 Census basis).
- **Variables.**
  - **Murder** (per 100,000): annual rate of willful homicides.
  - **Assault** (per 100,000): annual rate of aggravated assaults.
  - **UrbanPop** (%): percentage of state population living in urban areas.
  - **Rape** (per 100,000): annual rate of reported rapes.

Source: McNeil (1977)

Data Preparation

```
version 17.0
clear all
set more off
set seed 12345

* Load / create USArrests and identifiers
capture confirm variable murder assault urbanpop rape
if _rc {
    clear
    input str14 state float(murder assault urbanpop rape)
    "Alabama" 13.2 236 58 21.2
    "Alaska" 10.0 263 48 44.5
    "Arizona" 8.1 294 80 31.0
    "Arkansas" 8.8 190 50 19.5
    "California" 9.0 276 91 40.6
    "Colorado" 7.9 204 78 38.7
    "Connecticut" 3.3 110 77 11.1
    "Delaware" 5.9 238 72 15.8
    "Florida" 15.4 335 80 31.9
    "Georgia" 17.4 211 60 25.8
    "Hawaii" 5.3 46 83 20.2
    "Idaho" 2.6 120 54 14.2
    "Illinois" 10.4 249 83 24.0
    "Indiana" 7.2 113 65 21.0
    "Iowa" 2.2 56 57 11.3
    "Kansas" 6.0 115 66 18.0
    "Kentucky" 9.7 109 52 16.3
    "Louisiana" 15.4 249 66 22.2
    "Maine" 2.1 83 51 7.8
    "Maryland" 11.3 300 67 27.8
    "Massachusetts" 4.4 149 85 16.3
    "Michigan" 12.1 255 74 35.1
    "Minnesota" 2.7 72 66 14.9
```

The Data Matrix

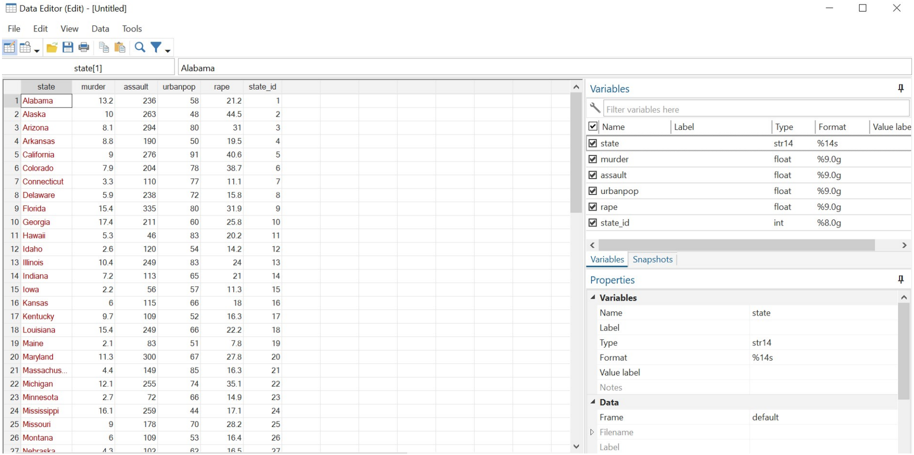
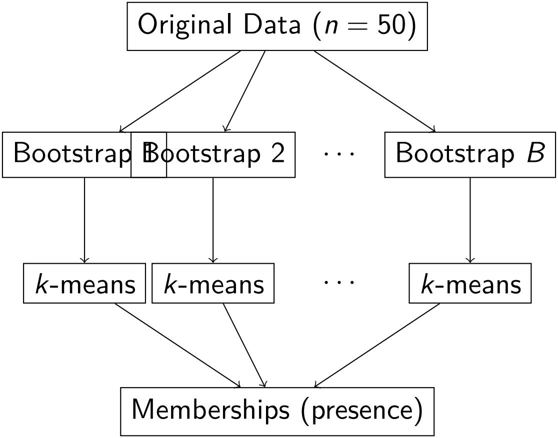


Figure: The Data Matrix

Bootstrap Process Visualization



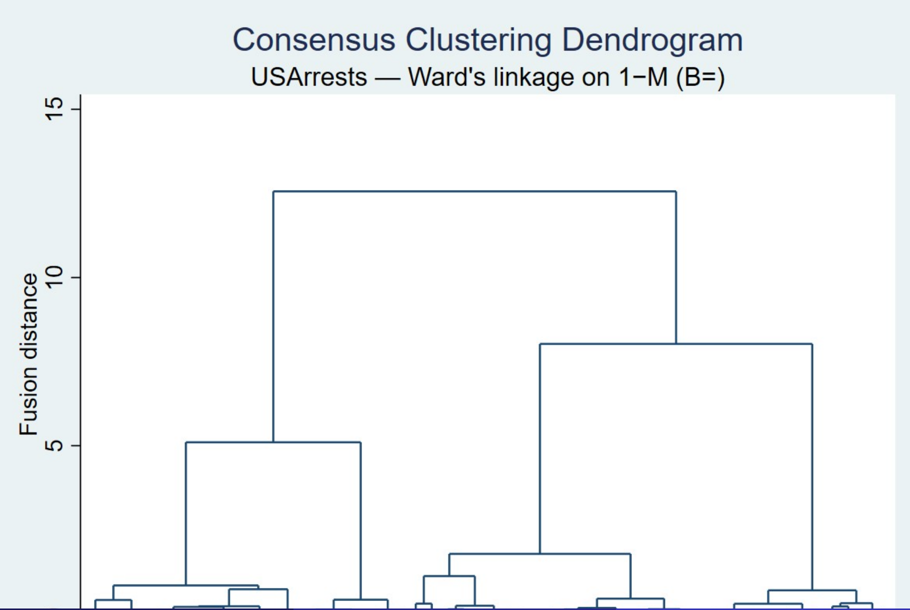
Consensus Matrix: Properties and Convention

- The matrix M is symmetric with entries in $[0, 1]$ and unit diagonal.
- Entries equal to one indicate pairs that always co-cluster when co-sampled, while entries equal to zero indicate pairs that never co-cluster when co-sampled.
- For pairs never co-sampled across the B draws, we leave M_{ij} undefined at construction and subsequently set their dissimilarity to $D_{ij} = 1$, which is a conservative, information-consistent convention.

Ward's Linkage Method

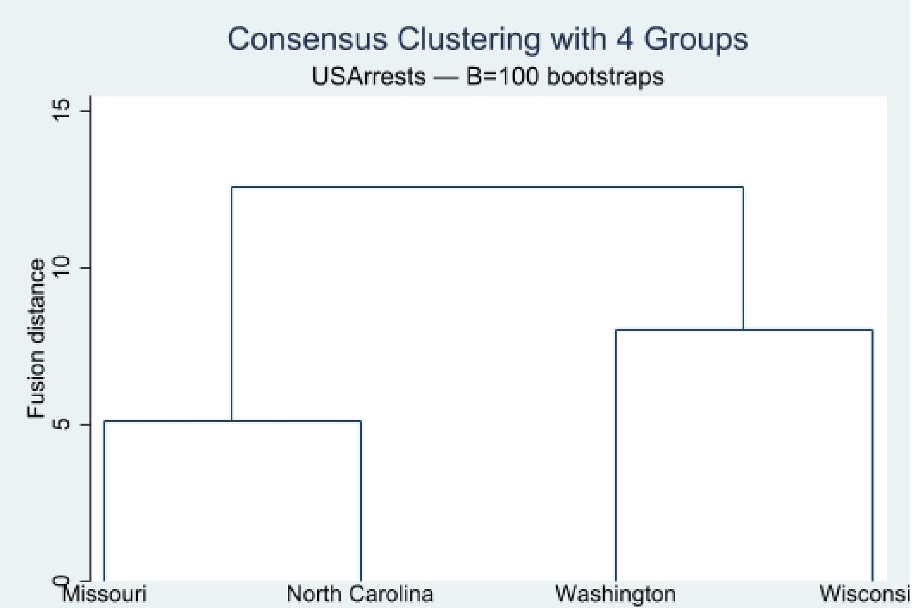
- Ward's method merges clusters to minimize the increase in WCSS, yielding compact groups under Euclidean geometry induced by the consensus dissimilarity.
- It naturally furnishes a dendrogram whose cut at level K defines the final consensus partition, while the full tree visualizes multi-scale structure.

Results



Consensus Clustering Dendrogram
USArrests — Ward's linkage on 1−M (B=)
Fusion distance
15
10
5

Results



Consensus Clustering with 4 Groups
USArrests — B=100 bootstraps
Fusion distance
0
5
10
15
Missouri
North Carolina
Washington
Wisconsi

Results

	state_name	final_~r
1.	Alaska	1
2.	Arizona	1
3.	California	1
4.	Colorado	1
5.	Florida	1
6.	Illinois	1
7.	Maryland	1
8.	Michigan	1
9.	Missouri	1
10.	Nevada	1
11.	New Mexico	1
12.	New York	1
13.	Texas	1
14.	Alabama	2
15.	Georgia	2
16.	Louisiana	2
17.	Mississippi	2
18.	North Carolina	2
19.	South Carolina	2
20.	Tennessee	2
21.	Arkansas	3
22.	Connecticut	3

Figure: Clustering Results

Interpretation

- The consensus partition reveals blocks with high within-block M_{ij} and low between-block M_{ij}, consistent with well-separated groups in the dendrogram.
- Cluster-level summaries on original and standardized scales clarify substantive differences, while the reported R^2 quantifies the share of standardized variance explained by the partition.

Results

-> final_cluster = 1

Variable	Obs	Mean	Std. dev.	Min	Max
murder	13	10.81538	2.083605	7.9	15.4
assault	13	257.3846	43.55942	178	335
urbanpop	13	76	10.77033	48	91
rape	13	33.19231	7.282337	24	46

-> final_cluster = 2

Variable	Obs	Mean	Std. dev.	Min	Max
murder	7	14.67143	1.693826	13	17.4
assault	7	251.2857	48.37601	188	337
urbanpop	7	54.28571	8.538429	44	66
rape	7	21.68571	4.031306	16.1	26.9

Figure: Descriptives by Cluster

Results

-> final_cluster = 3

Variable	Obs	Mean	Std. dev.	Min	Max
murder	18	6.055556	1.941262	3.2	9.7
assault	18	140.0556	41.24029	46	238
urbanpop	18	71.33333	11.19349	50	89
rape	18	18.68333	4.957496	8.3	29.3

-> final_cluster = 4

Variable	Obs	Mean	Std. dev.	Min	Max
murder	12	3.091667	1.557071	.8	6
assault	12	76	25.09256	45	120
urbanpop	12	52.08333	10.53529	32	66
rape	12	11.83333	3.147967	7.3	16.5

Figure: Descriptives by Cluster

Advantages of the Pipeline in Stata

- Users can use results in Stata across multiple runs. This could be very relevant for robustify results of k-means cluster analysis
- Stata is often used in regulation, policy and applied research where verifiability is important: being able to show exactly how the consensus matrix was created is an advantage. This is a general advantage of using programming in Stata.
- Combining k-means and also hierarchical clustering with the Ward method in the context of consensus clustering in Stata offers researchers both the computational speed of partitioning and the structural data exploration of hierarchical clustering.

Advantages of the Pipeline in Stata

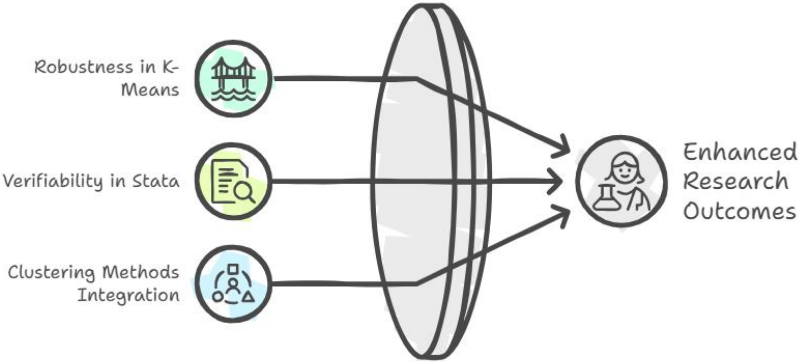


Figure: Advantages of the Pipeline in Stata

Best Practices

- Always standardize inputs, fix the random seed, and save intermediate artifacts.
- Use `clustermat ... , add` when clustering a dissimilarity matrix with a dataset in memory, and label dendrograms with human-readable identifiers such as `state_name`.
- Report WCSS/BCSS/R^2 to quantify explanatory power on standardized variables.

Conclusions

- Consensus clustering based on the co-sampling conditional matrix M and Ward's linkage on yields stable and interpretable groupings. The clustering results can be robustified.
- The pure-Stata implementation allow to implement more the data analysis pipeline with different methodologies (using different methodologies from Stata, Python and R)
- The final results as data labels can be used in other analyses (econometric analyses for instance)

References (1)

1. Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics-theory and Methods, 3(1), 1-27.
2. Drago, C. (2018). MCA-based community detection. In Classification,(big) data analysis and statistical learning (pp. 59-66). Cham: Springer International Publishing.
3. Fred, A. L. N., & Jain, A. K. (2005). Combining Multiple Clusterings Using Evidence Accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 835–850. doi:10.1109/TPAMI.2005.113
4. Jaeger, Adam, and David Banks. "Cluster analysis: A modern statistical review." Wiley Interdisciplinary Reviews: Computational Statistics 15, no. 3 (2023): e1597.
5. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

References (2)

1. Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis. John Wiley & Sons.
2. Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Performance evaluation of distance metrics in the clustering algorithms. *INFOCOMP Journal of Computer Science*, 13(1), 38-52.
3. McNeil. D. R., (1977) Interactive Data Analysis: A Practical Primer — New York: Wiley
4. Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, *52*(1–2), 91–118. https://doi.org/10.1023/A:1023949509487
5. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, 53-65.

References (3)

1. Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. In New directions in statistical physics: econophysics, bioinformatics, and pattern recognition (pp. 273-309). Berlin, Heidelberg: Springer Berlin Heidelberg
2. Strehl, A., & Ghosh, J. (2002). Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, *3*, 583–617.
3. Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A Survey of Clustering Ensemble Algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, *25*(3), 337–372.
4. Zhang, Z., Chen, X., Tang, R., Zhu, Y., Guo, H., Qu, Y., ... & Lo, Y. H. (2023). Interpretable unsupervised learning enables accurate clustering with high-throughput imaging flow cytometry. *Scientific Reports*, 13(1), 20533.