

cluster analysis in stata

Published: September 27, 2026 | Index ID: #-

Introduction

In the world of data science and econometrics, the ability to uncover hidden patterns within a dataset can transform raw numbers into actionable insights. One of the most powerful tools for this purpose is cluster analysis—a statistical technique that groups observations into clusters based on similarity. When working with large survey data, economic indicators, or any multivariate dataset, researchers often turn to Stata for its robust statistical engine and user-friendly interface. This article offers a comprehensive guide to performing cluster analysis in Stata, from data preparation to interpreting results and visualizing clusters. Whether you're a seasoned analyst or just starting out, you'll find clear, step by step instructions, practical examples, and best practice tips that will help you harness the full potential of clustering in Stata.

Understanding Cluster Analysis

Cluster analysis, also known as clustering, is an unsupervised learning technique that partitions a set of observations into groups—or clusters—such that observations within the same cluster are more similar to each other than to those in other clusters. This similarity is typically measured by a distance metric (e.g., Euclidean, Manhattan) or a similarity coefficient. Clustering serves many purposes: identifying market segments, detecting outliers, simplifying complex datasets, or creating homogeneous groups for further analysis.

Why Use Clustering?

- **Pattern Discovery:** Reveal natural groupings that may not be obvious.
- **Data Reduction:** Summarize large datasets into a few representative clusters.
- **Feature Engineering:** Generate cluster based variables for predictive models.
- **Targeted Interventions:** Tailor policies or marketing strategies to distinct segments.

Getting Started with Stata for Cluster Analysis

Stata offers a suite of commands for clustering, including `cluster kmeans`, `cluster ward` (hierarchical), and `cluster two` (two step). Before diving into commands, ensure your dataset is clean, variables are appropriately scaled, and missing values are handled.

Data Preparation

Proper preparation is critical. Begin by inspecting your data:

1. **Check for Missing Values:** Use `misstable summarize` to identify gaps. Impute or drop as needed.
2. **Variable Selection:** Choose variables that capture the dimensions of similarity you're interested in. Avoid redundant or highly correlated variables to prevent distortion.
3. **Standardization:** Since distance metrics are sensitive to scale, standardize continuous variables with `egen zvar = std(var)`. Categorical variables should be transformed into dummy variables if they're to be included in distance calculations.

Choosing the Right Clustering Algorithm

Stata supports several clustering methods, each suited to different data characteristics and research objectives.

K Means Clustering

Ideal for large, numeric datasets where you want a predefined number of clusters. The algorithm partitions observations into k clusters by minimizing within cluster variance.

```
cluster kmeans varlist, k(3) distance(euclidean)
```

Key parameters: k (number of clusters), distance (distance metric), iterations (maximum iterations), and seed (random seed for reproducibility).

Hierarchical Clustering

Useful when you want a dendrogram that illustrates nested cluster relationships. Stata implements Ward's method by default but also offers single, complete, and average linkage.

```
cluster ward varlist, distance(euclidean)
```

After running, use cluster dendrogram to visualize the hierarchy.

Two Step Clustering

Stata's cluster two command is efficient for very large datasets and can handle both continuous and categorical variables. It first pre-clusters observations, then applies hierarchical clustering.

```
cluster two varlist, maxclusters(5) distance(euclidean)
```

Implementing K Means Clustering in Stata

Below is a detailed walkthrough of a typical k-means workflow, from data scaling to interpreting the output.

Step 1: Standardize Variables

```
egen z_income = std(income)
```

```
egen z_age = std(age)
```

```
egen z_expenditure = std(expenditure)
```

Step 2: Run K Means

```
cluster kmeans z_income z_age z_expenditure, k(4) seed(123)
```

The seed option ensures reproducibility of cluster assignments.

Step 3: Examine Cluster Centers

```
cluster center
```

This displays the mean values of each variable within each cluster, allowing you to label clusters (e.g., "High income, young").

Step 4: Visualize Cluster Profiles

Use graph matrix or twoway scatter to plot key variables colored by cluster assignment.

```
scatter z_income z_expenditure, by(cluster) mlabel(id)
```

Step 5: Validate the Solution

Cluster validity can be assessed with internal metrics like the silhouette coefficient or the elbow method. Stata

doesn't compute these directly, but you can approximate them:

- Silhouette: Calculate the average silhouette width by measuring within cluster cohesion and between cluster separation.
- Elbow Method: Plot the sum of squared distances (SSD) versus number of clusters and look for the point where SSD decreases sharply.

Hierarchical Clustering and Dendrograms

Hierarchical clustering is especially valuable when you're uncertain about the number of clusters. The dendrogram provides a visual representation of how clusters merge at different distance thresholds.

Running Hierarchical Clustering

```
cluster ward z_income z_age z_expenditure
```

Plotting the Dendrogram

```
cluster dendrogram
```

Inspect the dendrogram to decide where to cut the tree. For instance, cutting at a height that yields four clusters may be appropriate if the dendrogram shows a clear separation at that level.

Two Step Clustering for Mixed Data Types

When your dataset contains both continuous and categorical variables, two step clustering offers a flexible solution. It automatically selects the optimal number of clusters based on criteria like the Bayesian Information Criterion (BIC).

Example Workflow

```
cluster two income age gender education, maxclusters(10) distance(euclidean)
```

After execution, Stata provides cluster centroids, cluster sizes, and model fit statistics. Use cluster predict to assign observations to clusters.

Interpreting and Reporting Cluster Results

Once clusters are defined, the next step is to interpret them meaningfully. This involves examining cluster centroids, sizes, and variable distributions.

Cluster Profiles

Generate summary statistics for each cluster:

```
bysort cluster: summarize income age expenditure
```

These statistics help you label clusters (e.g., "Young, high spending" or "Senior, moderate income").

Statistical Tests Between Clusters

Use ANOVA or Kruskal Wallis tests to determine whether differences across clusters are statistically significant.

```
anova income cluster
```

```
kwallis expenditure, by(cluster)
```

Exporting Cluster Assignments

Save cluster labels as a new variable for use in subsequent analyses:

```
cluster predict cluster_id
```

Now you can merge `cluster_id` with other datasets or include it as a predictor in regression models.

Visualizing Clusters in Stata

Effective visualization turns numbers into stories. Stata offers several graphing options tailored for clustering.

Scatter Plots with Cluster Colors

```
twoway (scatter z_income z_expenditure, mcolor(cluster))
```

Parallel Coordinates Plot

Displays multivariate cluster profiles:

```
parallel z_income z_age z_expenditure, cluster(cluster)
```

Heatmaps of Cluster Centroids

Use the `heatplot` package or build a matrix plot to compare centroids across clusters.

Case Study: Segmenting Household Consumption

Let's walk through a realistic example using a household survey dataset. The goal: identify distinct consumer segments based on income, age, expenditure, and education level.

Step 1: Load and Inspect Data

```
use household_survey.dta, clear
misstable summarize
describe
```

Step 2: Standardize Continuous Variables

```
egen z_income = std(income)
egen z_expenditure = std(expenditure)
```

Step 3: Convert Categorical Variables

Generate dummy variables for education categories:

```
tab education, gen(edu_)
```

Step 4: Run Two-Step Clustering

```
cluster two z_income z_expenditure edu_1 edu_2 edu_3, maxclusters(6) distance(euclidean)
```

Step 5: Evaluate the Model

Stata outputs the BIC and cluster sizes. Suppose the BIC suggests four clusters. Extract cluster assignments:

```
cluster predict segment_id
```

Step 6: Profile Each Segment

bysort segment_id: summarize income age expenditure education

Interpretation might reveal:

- Segment 1: High income, young, high expenditure.
- Segment 2: Moderate income, older, moderate expenditure.
- Segment 3: Low income, young, low expenditure.
- Segment 4: Moderate income, older, high expenditure.

Step 7: Visualize Segments

Create a scatter plot with cluster colors:

```
twoway (scatter z_income z_expenditure, mcolor(segment_id))
```

Troubleshooting Common Issues

- High Dimensionality: Too many variables can dilute meaningful distances. Apply dimensionality reduction techniques (e.g., PCA) before clustering.
- Non-Gaussian Data: Distance metrics assume certain data distributions. Consider using Gower distance for mixed data.
- Outliers: Outliers can distort cluster centers. Use robust scaling or remove extreme observations.
- Choosing k: Use domain knowledge, silhouette analysis, or the elbow method to decide the number of clusters.

Best Practices for Cluster Analysis in Stata

- Document Every Step: Record variable transformations, scaling choices, and command options for reproducibility.
- Cross-Validate: Run clustering on a training subset and test on a hold-out sample to assess stability.

Baca Juga (Artikel Terkait)

- [cloud computing gartner inc](http://basicgravity.com/cloud-computing-gartner-inc.pdf)

URL: <http://basicgravity.com/cloud-computing-gartner-inc.pdf>

- [cloud computing for dummies](http://basicgravity.com/cloud-computing-for-dummies.pdf)

URL: <http://basicgravity.com/cloud-computing-for-dummies.pdf>

- [cloud computing exam questions answers](http://basicgravity.com/cloud-computing-exam-questions-answers.pdf)

URL: <http://basicgravity.com/cloud-computing-exam-questions-answers.pdf>

- [cloud computing concepts technology architecture the prentice hall service technology series from thomas erl](http://basicgravity.com/cloud-computing-concepts-technology-architecture-the-prentice-hall-service-technology-series-from-thomas-erl.pdf)

URL: <http://basicgravity.com/cloud-computing-concepts-technology-architecture-the-prentice-hall-service-technology-series-from-thomas-erl.pdf>

Tags: [#cluster](#) [#analysis](#) [#stata](#)

