cluster linkage — Hierarchical cluster analysis

Description

corcluster and clustermat, with a specified linkage method, perform hierarchical agglomerative
corcluster analysis. The following common linkage methods are available: single, complete, average,
Ward’s method, centroid, median, and weighted average.

Quick start

Cluster analysis of data

Single-linkage cluster analysis using Euclidean distance of v1, v2, v3, and v4
corcluster singlelinkage v1 v2 v3 v4

Same as above
  corcluster single v1 v2 v3 v4

As above, and name the cluster analysis singclus
corcluster single v1 v2 v3 v4, name(singclus)

Complete-linkage cluster analysis using absolute-value distance of v1, v2, and v3
corcluster completelinkage v1 v2 v3, measure(absolute)

Same as above
  corcluster complete v1 v2 v3, measure(L1)

Cluster analysis of dissimilarity matrix

Replace current data with solution from single-linkage cluster analysis on matrix D
corclustermat single D, clear

As above, and name the cluster analysis Dclus
  corclustermat single D, clear name(Dclus)

Add solution from average-linkage cluster analysis on matrix D to the existing data
corclustermat averagelinkage D, add
Menu

cluster singlelinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Single linkage

cluster averagelinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Average linkage

cluster completelinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Complete linkage

cluster waveragelinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Weighted-average linkage

cluster medianlinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Median linkage

cluster centroidlinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Centroid linkage

cluster wardslinkage
Statistics > Multivariate analysis > Cluster analysis > Cluster data > Ward's linkage
Syntax

Cluster analysis of data

```
cluster linkage [varlist] [if] [in] [, cluster_options]
```

Cluster analysis of a dissimilarity matrix

```
clustermat linkage matname [if] [in] [, clustermat_options]
```

<table>
<thead>
<tr>
<th>linkage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singlelinkage</td>
<td>single-linkage cluster analysis</td>
</tr>
<tr>
<td>averagelinkage</td>
<td>average-linkage cluster analysis</td>
</tr>
<tr>
<td>completelinkage</td>
<td>complete-linkage cluster analysis</td>
</tr>
<tr>
<td>waveragelinkage</td>
<td>weighted-average linkage cluster analysis</td>
</tr>
<tr>
<td>medianlinkage</td>
<td>median-linkage cluster analysis</td>
</tr>
<tr>
<td>centroidlinkage</td>
<td>centroid-linkage cluster analysis</td>
</tr>
<tr>
<td>wardslinkage</td>
<td>Ward’s linkage cluster analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cluster_options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>measure</strong>(measure)</td>
<td>similarity or dissimilarity measure</td>
</tr>
<tr>
<td><strong>name</strong>(clname)</td>
<td>name of resulting cluster analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>clustermat_options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>shape</strong>(shape)</td>
<td>shape (storage method) of matname</td>
</tr>
<tr>
<td>add</td>
<td>add cluster information to data currently in memory</td>
</tr>
<tr>
<td>clear</td>
<td>replace data in memory with cluster information</td>
</tr>
<tr>
<td>labelvar**(varname)**</td>
<td>place dissimilarity matrix row names in varname</td>
</tr>
<tr>
<td>name**(clname)</td>
<td>name of resulting cluster analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>force</td>
<td>perform clustering after fixing matname problems</td>
</tr>
<tr>
<td>generate**(stub)**</td>
<td>prefix for generated variables; default prefix is clname</td>
</tr>
</tbody>
</table>
Options for cluster linkage commands

**Main**

`measure(measure)` specifies the similarity or dissimilarity measure. The default for `averagelinkage`, `completelinkage`, `singlelinkage`, and `waveragelinkage` is $L_2$ (synonym `Euclidean`). The default for `centroidlinkage`, `medianlinkage`, and `wardslinkage` is $L_2^2$ (Euclidean). This option is not case sensitive. See [MV] `measure_option` for a discussion of these measures.

Several authors advise using the $L_2^2$ measure exclusively with centroid, median, and Ward’s linkage. See Dissimilarity transformations and the Lance and Williams formula and Warning concerning similarity or dissimilarity choice in [MV] `cluster` for details.

`name(clname)` specifies the name to attach to the resulting cluster analysis. If `name()` is not specified, Stata finds an available cluster name, displays it for your reference, and attaches the name to your cluster analysis.

**Advanced**

`generate(stub)` provides a prefix for the variable names created by `cluster linkage`. By default, the variable name prefix will be the name specified in `name()`. Three variables with the suffixes `_id`, `_ord`, and `_hgt` are created and attached to the cluster-analysis results. Users generally will not need to access these variables directly.

Centroid linkage and median linkage can produce reversals or crossovers; see [MV] `cluster` for details. When reversals happen, `cluster centroidlinkage` and `cluster medianlinkage` also create a fourth variable with the suffix `pht`. This is a pseudoheight variable that is used by some postclustering commands to properly interpret the `_hgt` variable.

Options for clustermat linkage commands

**Main**

`shape(shape)` specifies the storage mode of `matname`, the matrix of dissimilarities. `shape(full)` is the default. The following shapes are allowed:

- `full`: `matname` is an $n \times n$ symmetric matrix; the default.
- `lower`: `matname` is a vector of rowwise lower triangle (with diagonal).
- `llower`: `matname` is a vector of rowwise strict lower triangle (no diagonal).
- `upper`: `matname` is a vector of rowwise upper triangle (with diagonal).
- `ulower`: `matname` is a vector of rowwise strict upper triangle (no diagonal).

In the lower triangle of the dissimilarity matrix:

- `D_{11} \ D_{21} \ D_{22} \ D_{31} \ D_{32} \ D_{33} \ \ldots \ D_{n1} \ D_{n2} \ \ldots \ D_{nn}`

- `llower`: `matname` is a vector of rowwise strict lower triangle (no diagonal).
upper specifies that `matname` is a row or column vector of length \( \frac{n(n+1)}{2} \), with the rowwise upper triangle of the dissimilarity matrix including the diagonal of zeros.

\[
\begin{align*}
D_{11} & \quad D_{12} & \ldots & D_{1n} \\
D_{22} & \quad D_{23} & \ldots & D_{2n} \\
D_{33} & \quad D_{34} & \ldots & D_{3n} \\
\vdots & \quad \vdots & \ddots & \vdots \\
D_{n1} & \quad D_{n2} & \ldots & D_{nn}
\end{align*}
\]

upper specifies that `matname` is a row or column vector of length \( \frac{n(n-1)}{2} \), with the rowwise upper triangle of the dissimilarity matrix excluding the diagonal.

\[
\begin{align*}
D_{12} & \quad D_{13} & \ldots & D_{1n} \\
D_{23} & \quad D_{24} & \ldots & D_{2n} \\
D_{34} & \quad D_{35} & \ldots & D_{3n} \\
\vdots & \quad \vdots & \ddots & \vdots \\
D_{n-1,n} & & & D_{nn}
\end{align*}
\]

add specifies that `clustermat`'s results be added to the dataset currently in memory. The number of observations (selected observations based on the `if` and `in` qualifiers) must equal the number of rows and columns of `matname`. Either clear or add is required if a dataset is currently in memory.

clear drops all the variables and cluster solutions in the current dataset in memory (even if that dataset has changed since the data were last saved) before generating `clustermat`'s results. Either clear or add is required if a dataset is currently in memory.

`labelvar(varname)` specifies the name of a new variable to be created containing the row names of matrix `matname`.

`name(clname)` specifies the name to attach to the resulting cluster analysis. If `name()` is not specified, Stata finds an available cluster name, displays it for your reference, and attaches the name to your cluster analysis.

---

Advanced

`force` allows computations to continue when `matname` is nonsymmetric or has nonzeros on the diagonal. By default, `clustermat` will complain and exit when it encounters these conditions. `force` specifies that `clustermat` operate on the symmetric matrix \( \frac{\text{matname} \times \text{matname}'}{2} \), with any nonzero diagonal entries treated as if they were zero.

`generate(stub)` provides a prefix for the variable names created by `clustermat`. By default, the variable name prefix is the name specified in `name()`. Three variables are created and attached to the cluster-analysis results with the suffixes `_id`, `_ord`, and `_hgt`. Users generally will not need to access these variables directly.

Centroid linkage and median linkage can produce reversals or crossovers; see [MV] `cluster` for details. When reversals happen, `clustermat centroidlinkage` and `clustermat medianlinkage` also create a fourth variable with the suffix `_pht`. This is a pseudoheight variable that is used by some of the postclustering commands to properly interpret the `_hgt` variable.
Remarks and examples

cluster and clustermat, with a specified linkage method, perform hierarchical agglomerative
cluster analysis. The following command linkage methods are available: single, complete, average,
Ward’s method, centroid, median, and weighted average. These methods are described in Agglomerative
methods of [MV] cluster. Other methods are mentioned in the literature, but these are the best-known
methods.

The clustermat linkage commands perform hierarchical agglomerative cluster analysis on the
dissimilarity matrix matname. See [MV] clustermat for a general discussion of cluster analysis of
dissimilarity matrices and a description of the other clustermat commands. A general discussion of
agglomerative cluster analysis is provided in Hierarchical cluster-analysis methods of [MV] cluster.

After a cluster linkage or clustermat linkage command, the cluster dendrogram command
(see [MV] cluster dendrogram) displays the resulting dendrogram, the cluster stop or clustermat
stop command (see [MV] cluster stop) helps determine the number of groups, and the cluster
generate command (see [MV] cluster generate) produces grouping variables.

Example 1

As the senior data analyst for a small biotechnology firm, you are given a dataset with four
chemical laboratory measurements on 50 different samples of a particular plant gathered from the
rain forest. The head of the expedition that gathered the samples thinks, based on information from
the natives, that an extract from the plant might reduce the negative side effects associated with your
company’s best-selling nutritional supplement.

While the company chemists and botanists continue exploring the possible uses of the plant and
plan future experiments, the head of product development asks you to look at the preliminary data
and to report anything that might be helpful to the researchers.

Although all 50 plants are supposed to be of the same type, you decide to perform a cluster analysis
to see if there are subgroups or anomalies among them. You arbitrarily decide to use single-linkage
clustering with the default Euclidean distance.

use https://www.stata-press.com/data/r16/labtech
cluster singlelinkage x1 x2 x3 x4, name(sngeuc)
cluster list sngeuc
sngeuc (type: hierarchical, method: single, dissimilarity: L2)
  vars: sngeuc_id (id variable)
   sngeuc_ord (order variable)
   sngeuc_hgt (height variable)
  other: cmd: cluster singlelinkage x1 x2 x3 x4, name(sngeuc)
   varlist: x1 x2 x3 x4
   range: 0.

The cluster singlelinkage command generated some variables and created a cluster object with
the name sngeuc, which you supplied as an argument. cluster list provides details about the
cluster object; see [MV] cluster utility.

What you really want to see is the dendrogram for this cluster analysis; see [MV] cluster
dendrogram.
From your experience looking at dendrograms, two things jump out at you about this cluster analysis. The first is the observations showing up in the middle of the dendrogram that are all close to each other (short vertical bars) and are far from any other observations (the long vertical bar connecting them to the rest of the dendrogram). Next you notice that if you ignore those 10 observations, the rest of the dendrogram does not indicate strong clustering, as shown by the relatively short vertical bars in the upper portion of the dendrogram.

You start to look for clues why these 10 observations are so peculiar. Looking at scatterplots is usually helpful, so you examine the matrix of scatterplots.

Unfortunately, these scatterplots do not indicate what might be going on.

Suddenly, from your past experience with the laboratory technicians, you have an idea of what to check next. Because of past data mishaps, the company started the policy of placing within each dataset a variable giving the name of the technician who produced the measurement. You decide
to view the dendrogram, using the technician’s name as the label instead of the default observation number.

```
. cluster dendrogram sng eucl, labels(labtech) xlabel(), angle(90) labsize(*.75)
```

Your suspicions are confirmed. Sam, one of the laboratory technicians, has messed up again. You list the data and see that all his observations are between zero and one, whereas the other four technicians’ data range up to about 150, as expected. It looks like Sam forgot, once again, to calibrate his sensor before analyzing his samples. You decide to save a note of your findings with this cluster analysis (see [MV] cluster notes for the details) and to send the data back to the laboratory to be fixed.

**Example 2**

The sociology professor of your graduate-level class gives, as homework, a dataset containing 30 observations on 60 binary variables, with the assignment to tell him something about the 30 subjects represented by the observations. You think that this assignment is too vague, but because your grade depends on it, you get to work trying to figure something out.

Among the analyses you try is the following cluster analysis. You decide to use single-linkage clustering with the simple matching binary coefficient because it is easy to understand. Just for fun, though it makes no difference to you, you specify the `generate()` option to force the generated variables to have `zstub` as a prefix. You let Stata pick a name for your cluster analysis by not specifying the `name()` option.

```
. use https://www.stata-press.com/data/r16/homework, clear
. cluster s a1-a60, measure(matching) gen(zstub) name:_clus_1
. cluster list
_clus_1 (type: hierarchical, method: single, similarity: matching)
  vars:  zstub_id (id variable)  
         zstub_ord (order variable)  
         zstub_hgt (height variable)  
  other:  cmd:  cluster singlelinkage a1-a60, measure(matching) gen(zstub)
         varlist:  a1  a2  a3  a4  a5  a6  a7  a8  a9  a10  a11  a12  a13  a14  a15  a16  a17  
               a18  a19  a20  a21  a22  a23  a24  a25  a26  a27  a28  a29  a30  a31  a32  
               a33  a34  a35  a36  a37  a38  a39  a40  a41  a42  a43  a44  a45  a46  a47  
               a48  a49  a50  a51  a52  a53  a54  a55  a56  a57  a58  a59  a60  
  range:  1 0
```
Stata selected _clus_1 as the cluster name and created the variables zstub_id, zstub_ord, and zstub_hgt.

You display the dendrogram by using the `cluster tree` command, which is a synonym for `cluster dendrogram`. Because Stata uses the most recently performed cluster analysis by default, you do not need to type the name.

```
. cluster tree
```

The dendrogram seems to indicate the presence of three groups among the 30 observations. You decide that this is probably the structure your teacher wanted you to find, and you begin to write up your report. You want to examine the three groups further, so you use the `cluster generate` command (see [MV] `cluster generate`) to create a grouping variable to make the task easier. You examine various summary statistics and tables for the three groups and finish your report.

After the assignment is turned in, your professor gives you the same dataset with the addition of one more variable, truegrp, which indicates the groupings he thinks are in the data. You do a cross-tabulation of the truegrp and grp3, your grouping variable, to see if you are going to get a good grade on the assignment.

```
. cluster generate grp3 = group(3)
. table grp3 truegrp
```

<table>
<thead>
<tr>
<th>grp3</th>
<th>truegrp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

Other than the numbers arbitrarily assigned to the three groups, both you and your professor agree. You rest easier that night knowing that you may survive one more semester.

In addition to examining single-linkage clustering of these data, you decide to see what median-linkage clustering shows. As with the single-linkage clustering, you pick the simple matching binary coefficient to measure the similarity between groups. The `name()` option is used to attach the name medlink to the cluster analysis. `cluster list` displays the details; see [MV] `cluster utility`.
You attempt to use the `cluster dendrogram` command to display the dendrogram, but because this particular cluster analysis produced reversals, `cluster dendrogram` refuses to produce the dendrogram. You realize that with reversals, the resulting dendrogram would not be easy to interpret anyway.

You use the `cluster generate` command (see [MV] `cluster generate`) to create a three-group grouping variable, based on your median-linkage clustering, to compare with `truegrp`.

```
. cluster generate medgrp3 = group(3)
. table medgrp3 truegrp

<table>
<thead>
<tr>
<th>medgrp3</th>
<th>truegrp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>
```

Because you were unable to view a dendrogram by using median-linkage clustering, you turn to Ward’s linkage clustering method.

```
. cluster ward a1-a60, measure(match) name(wardlink)
. cluster list wardlink
wardlink (type: hierarchical, method: wards, similarity: matching)
  vars: wardlink_id (id variable)
  wardlink_ord (order variable)
  wardlink_hgt (height variable)
  other: cmd: cluster wardslinkage a1-a60, measure(match) name(wardlink)
  varlist: a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17
  a18 a19 a20 a21 a22 a23 a24 a25 a26 a27 a28 a29 a30 a31 a32
  a33 a34 a35 a36 a37 a38 a39 a40 a41 a42 a43 a44 a45 a46 a47
  a48 a49 a50 a51 a52 a53 a54 a55 a56 a57 a58 a59 a60
  range: 1 0
```
As with single-linkage clustering, the dendrogram from Ward’s linkage clustering seems to indicate the presence of three groups among the 30 observations. However, notice the y-axis range for the resulting dendrogram. How can the matching similarity coefficient range from 1 to less than −2? By definition, the matching coefficient is bounded between 1 and 0. This is an artifact of the way Ward’s linkage clustering is defined, and it underscores the warning mentioned in the discussion of the choice of measure. Also see Dissimilarity transformations and the Lance and Williams formula and Warning concerning similarity or dissimilarity choice in [MV] cluster for more details.

A cross-tabulation of truegrp and wardgrp3, a three-group grouping variable from this cluster analysis, is shown next.

```
. cluster generate wardgrp3 = group(3)
. table wardgrp3 truegrp

<table>
<thead>
<tr>
<th>wardgrp3</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
```

Other than the numbers arbitrarily assigned to the three groups, your teacher’s conclusions and the results from the Ward’s linkage clustering agree. So, despite the warning against using something other than squared Euclidean distance with Ward’s linkage, you were still able to obtain a reasonable cluster-analysis solution with the matching similarity coefficient.

---

### Example 3

The wclub dataset contains answers from 30 women to 35 yes–no questions. The variables are described in example 3 of [MV] clustermat. We are interested in seeing how weighted-average linkage clustering will cluster the 35 variables (instead of the observations).
We use the `matrix dissimilarity` command to produce a dissimilarity matrix equal to one minus the Jaccard similarity; see `[MV] matrix dissimilarity`.

```
. use https://www.stata-press.com/data/r16/wclub, clear
. matrix dissimilarity clubD = , variables Jaccard dissim(oneminus)
. clustermat waverage clubD, name(clubwav) clear labelvar(question)
   number of observations (_N) was 0, now 35
. cluster dendrogram clubwav, labels(question)
> xlabel(, angle(90) labsize(*.75)) title(Weighted-average linkage clustering)
> ytitle(1 − Jaccard similarity, suffix)
```

From these 30 women, we see that the `biog` (enjoy reading biographies) and `hist` (enjoy reading history) questions were most closely related. `bird` (have a bird) seems to be the least related to the other variables. It merges last into the supergroup containing the remaining variables.

Technical note

Cluster commands require a significant amount of memory and execution time. With many observations, the execution time may be significant.

Methods and formulas

`[MV] cluster` discusses and compares the hierarchical clustering methods.

Conceptually, hierarchical agglomerative linkage clustering proceeds as follows. The $N$ observations start out as $N$ separate groups, each of size one. The two closest observations are merged into one group, producing $N − 1$ total groups. The closest two groups are then merged so that there are $N − 2$ total groups. This process continues until all the observations are merged into one large group, producing a hierarchy of groupings from one group to $N$ groups. The difference between the various hierarchical-linkage methods depends on how they define “closest” when comparing groups.

For single-linkage clustering, the closest two groups are determined by the closest observations between the two groups.
In complete linkage, the closest two groups are determined by the farthest observations between the two groups.

For average-linkage clustering, the closest two groups are determined by the average (dis)similarity between the observations of the two groups.

The Lance–Williams formula provides the basis for extending the well-known Ward’s method of clustering into the general hierarchical-linkage framework that allows a choice of (dis)similarity measures.

Centroid linkage merges the groups whose means are closest.

Weighted-average linkage clustering is similar to average-linkage clustering, except that it gives each group of observations equal weight. Average linkage gives each observation equal weight.

Median linkage is a variation on centroid linkage in that it treats groups of unequal size differently. Centroid linkage gives each observation equal weight. Median linkage, however, gives each group of observations equal weight, meaning that with unequal group sizes, the observations in the smaller group will have more weight than the observations in the larger group.

The linkage clustering algorithm produces two variables that together act as a pointer representation of a dendrogram. To this, Stata adds a third variable used to restore the sort order, as needed, so that the two variables of the pointer representation remain valid. The first variable of the pointer representation gives the order of the observations. The second variable has one less element and gives the height in the dendrogram at which the adjacent observations in the order variable join.

When reversals happen, a fourth variable, called a pseudoheight, is produced and is used by postclustering commands with the height variable to properly interpret the ordering of the hierarchy.

See [MV] measure_option for the details and formulas of the available measures, which include (dis)similarity measures for continuous and binary data.

Joe H. Ward, Jr. (1926–2011) was born in Austin, Texas, and obtained degrees in mathematics and educational psychology from the University of Texas. He worked as a personnel research psychologist for the U.S. Air Force Human Resources Laboratory, applying educational psychology, statistics, and computers to a wide variety of procedures, most notably the widely used clustering method named for him. In retirement, he mentored and supported high school students in computers, statistics, science fairs, and basketball.

Also see

[MV] cluster — Introduction to cluster-analysis commands
[MV] cluster dendrogram — Dendrograms for hierarchical cluster analysis
[MV] cluster generate — Generate grouping variables from a cluster analysis
[MV] cluster notes — Cluster analysis notes
[MV] cluster stop — Cluster-analysis stopping rules
[MV] cluster utility — List, rename, use, and drop cluster analyses
[MV] clustermat — Introduction to clustermat commands