The most common method for calculating vector dissimilarity is the Euclidean metric, the familiar square root of the summed squared differences of the vector elements. See **HCluster Vector Dissimilarity Calculation Methods** on page III-163 for a description of the vector dissimilarity metrics offered by the HCluster operation.

If you use /ITYP=DMatrix, you can prepare your own dissimilarity matrix using whatever method you wish for measuring the dissimilarity between vectors. Your dissimilarity metric must return a positive number. Identical vectors, such as comparing a vector with itself, have a dissimilarity of zero.

- The dissimilarity between a vector and a previously-determined cluster or between two previously-determined clusters

  We call this a "linkage" calculation.

  You specify how to calculate linkage using the HCluster /LINK flag.

  See **HCluster Linkage Calculation Methods** on page III-166 dfor a description of the linkage calculation methods offered by the HCluster operation. The linkage method that you choose can have a very strong effect on the resulting dendrogram.

HCluster Vector Dissimilarity Calculation Methods

You use the HCluster /DISS=dm flag to specify the dissimilarity metric between two data vectors. Our definitions of dissimilarity follows Python scipy.spatial.distance.pdist.

The following values are supported for the *dm* keyword. If you omit /DISS, HCluster defaults to the Euclidean method.

***dm* = Euclidean**

This is the usual way to measure the dissimilarity between two vectors, the two-norm or L_2 norm. It is simply the Euclidean distance. This is the default.

$$d(u,v) = \|u - v\|_2 = \sqrt{\sum_j (u_j - v_j)^2}$$

***dm* = SquaredEuclidean**

Just like Euclidean, but omits taking the square root. May be needed to reproduce some results from R or Python. Results in the same clustering as Euclidean, but exaggerates larger differences.

$$d(u,v) = \sum_j (u_j - v_j)^2$$

***dm* = SEuclidean**

Standardized Euclidean. Euclidean distance in which the dimensions are scaled by V_j, which is usually the variance of the j-th element of all the vectors.

$$d(u,v) = \sqrt{\sum_j (u_j - v_j)^2 / V_j}$$

Specify a wave giving the V_j vector using the /VARW flag.

***dm* = Cityblock**

Manhattan distance or L_1 norm.

$$d(u,v) = \sum_j |u_j - v_j|$$

Cityblock gives the same value of 2 for vectors (0,2), (2,0), and (1,1). Euclidean distance gives a smaller value, sqrt(2), for the vector (1,1). This can affect the resulting clusters.

dm **= Chebychev**

Supremum or L_∞ norm.

$$d(u,v) = \max_j |u_j - v_j|$$

dm **= Minkowski**

The L_p norm.

$$d(u,v) = \left(\sum_j |u_j - v_j|^p \right)^{1/p}$$

The value of p is specified using the HCluster /P flag.

p = 1 makes Minkowski equivalent to Cityblock.

p = 2 makes Minkowski equivalent to Euclidean.

p = Inf makes Minkowski equivalent to Chebychev.

dm **= Cosine**

$$d(u,v) = 1 - \frac{\langle u,v \rangle}{\|u\| \cdot \|v\|} = 1 - \frac{\sum_j u_j v_j}{\sqrt{\sum_j u_j^2 \cdot \sum_j v_j^2}}$$

dm **= Canberra**

$$d(u,v) = \sum_j \frac{|u_j - v_j|}{|u_j| + |v_j|}$$

Terms in which uj = vj = 0 contribute 0 to the sum.

dm **= BrayCurtis**

$$d(u,v) = \sum_j \frac{|u_j - v_j|}{|u_j + v_j|}$$

Terms in which uj = vj = 0 contribute 0 to the sum.

In the following, the notation |{...}| indicates the count of true boolean values.

dm **= Hamming**

$$d(u,v) = |\{j | u_j \neq v_j\}|$$

Hamming is actually intended to be used with binary data, but the definition will test a "1" and "2" as being different. See Matching below, which tests each vector element for u_j != 0. For data that is all ones or zeroes, Hamming and Matching give the same results.

dm **= Jaccard**