The Hanning operation is not multidimensional aware. See Chapter II-6, **Multidimensional Waves**, particularly **Analysis on Multidimensional Waves** on page II-95 for details.

**See Also**

The **WindowFunction** operation implements the Hanning window as well as other forms such as Hamming, Parzen, and Bartlet (triangle).

**ImageWindow**, **DPSS**

# Hash

**Hash(*inputStr*, *method*)**

The Hash function returns a cryptographic hash of the data in *inputStr*.

**Parameters**

*inputStr* is string of length up to 2^31 bytes. *inputStr* can contain binary or text data.

*method* is a number indicating the hash algorithm to use:

| | |
|---|---|
| 1 | SHA-256 (SHA-2) |
| 2 | MD4 |
| 3 | MD5 |
| 4 | SHA-1 |
| 5 | SHA-224 (SHA-2) |
| 6 | SHA-384 (SHA-2) |
| 7 | SHA-512 (SHA-2) |

Prior to Igor Pro 7.00, only method 1 was supported.

**See Also**

**WaveHash**, **StringCRC**, **WaveCRC**

# HCluster

**HCluster [ flags ] *sourceWave***

The HCluster operation computes the information needed to create a cluster dendrogram using an agglomerative hierarchical clustering algorithm. "HCluster" stands for "hierarchical clustering". The HCluster operation was added in Igor Pro 9.00.

For background information, see **Hierarchical Clustering** on page III-162.

The input *sourceWave* represents either vectors in some data space or a square vector dissimilarity matrix (also called a "distance" matrix). You indicate which type of input you are providing using the /ITYP flag.

HCluster creates an output vector dissimilarity matrix wave or an output dendrogram wave or both, depending on the /OTYP flag. The output wave names default to M_HCluster_Dissimilarity and M_HCluster_Dendrogram but you can override the default using /DEST.

**Flags**

| | |
|---|---|
| /ITYP=*it* | *it* is a keyword specifying the kind of data in *sourceWave*: |
| | *it*=`Vectors`: *sourceWave* rows represent data vectors (default). |
| | *it*=`DMatrix`: *sourceWave* contains a square vector dissimilarity matrix. |

/OTYP=*ot*

*ot* is a keyword specifying what type of output to be produced:

*ot*=`DMatrix`: The output is a vector dissimilarity matrix. You can use /OTYP=DMatrix only if /ITYP=Vectors or if you omit /ITYP.

*ot*=`Dendrogram`: The output is a multi-column wave describing the nodes in a dendrogram illustrating the way original data is joined into clusters. This is the default if you omit /OTYP.

*ot*=`Both`: The output is both the vector dissimilarity matrix and a dendrogram.

See the /DEST flag for further discussion of the output wave or waves.

/LINK=*linkMethod*

*linkMethod* is a keyword specifying the method used to determine the dissimilarity between nodes in the dendrogram that represent more than one data vector. This is also referred to as the "linkage" method. Our definitions of node dissimilarities follows Python scipy.cluster.hierarchy.linkage.

The available keywordds for linkMethod are listed and described under **HCluster Linkage Calculation Methods** on page III-166.

If you omit /LINK, HCluster defaults to the average method.

/DISS=*dm*

*dm* is a keyword specifying the vector dissimilarity metric for calculating the dissimilarity between two data vectors. Our definitions of vector dissimilarity follows Python scipy.spatial.distance.pdist.

The available /DISS keywords are listed and described under **HCluster Vector Dissimilarity Calculation Methods** on page III-163.

If you omit /DISS, HCluster defaults to the Euclidean metric.

/P=*pow*

*pow* is the power for the Minkowski vector dissimilarity metric. The value of *pow* must be positive. The default is 2.0, equivalent to the Euclidean vector dissimilarity metric. Values that are too large can lead to floating-point overflow. Values less than 1.0 may give surprising results, as this can cause an inversion of the usual distance ordering. If the vector dissimilarity metric is not Minkowski this flag is ignored.

/VARW=*varWave*

Specifies the normalizing values Vj for use with the SEuclidean vector dissimilarity metric. Usually, the wave elements are variances of the vector elements over all the vectors. Thus, if you have a multi-column wave in which rows represent individual vectors, *varWave* should be filled with variances of the wave's columns. If your vectors have length of M, then varWave should be a 1D wave with M elements. This wave can be conveniently created using the MatrixOP operation, like this:

```
MatrixOp/O varWave = VarCols(rowVectorMatrix)^t
```

If the vector dissimilarity matrix is not SEuclidean, the /VARW flag is ignored.

/DEST=*outWaveName*

Specifies the output waves when you have specified /OTYP=`DMatrix` or /OTYP=`Dendrogram`.

If you specified /OTYP=`DMatrix`, *outWaveName* is the name of the output vector dissimilarity matrix wave to be created or overwritten, optionally preceded by a data folder path. If you omit /DEST, HCluster creates an output vector dissimilarity matrix named M_HCluster_Dissimilarity in the current data folder.

If you specified /OTYP=`Dendrogram`, *outWaveName* is the name of the output dendrogram wave to be created or overwritten, optionally preceded by a data folder path. If you omit /DEST, HCluster creates an output dendrogram named M_HCluster_Dendrogram in the current data folder.