We created this dendrogram with heat map using the Hierarchical Clustering package which you can access by choosing Analysis→Packages→Hierarchical Clustering.

Analysis Programming

This section contains data analysis programming examples. There are many more examples in the WaveMetrics Procedures, Igor Technical Notes, and Examples folders.

Passing Waves to User Functions and Macros

As you look through various examples you will notice two different ways to pass a wave to a function: using a Wave parameter or using a String parameter.

Using a Wave Parameter	Using a String Parameter
`Function Test1(w)` `  Wave w`	`Function Test2(wn)` `  String wn`
Usable in functions, not in macros.	Usable in functions and macros.
w is a "formal" name. Use it just as if it were the name of an actual wave.	Use the $ operator to convert from a string to wave name.

The string method is used in macros and in user functions for passing the name of a wave that may not yet exist but will be created by the called procedure. The wave parameter method is used in user functions when the wave will always exist before the function is called. For details, see **Accessing Waves in Functions** on page IV-82.

Returning Created Waves from User Functions

A function can return a wave as the function result. For example:

```
Function Test()
   Wave w = CreateNoiseWave(5, "theNoiseWave")
   WaveStats w
   Display w as "Noise Wave"
End

Function/WAVE CreateNoiseWave(noiseValue, destWaveName)
   Variable noiseValue
   String destWaveName

   Make/O $destWaveName = gnoise(noiseValue)
   Wave w = $destWaveName
   return w
End
```

If the returned wave is intended for temporary use, you can create it as a free wave:

```
Function Test()
   Wave w = CreateFreeNoiseWave(5)// w is a free wave
   WaveStats w
   // w is killed when the function exist
End

Function/WAVE CreateFreeNoiseWave(noiseValue)
   Variable noiseValue

   Make/O/FREE aWave = gnoise(noiseValue)
   return aWave
End
```