```
DoLineFit("wave0;wave1;")
```

For most purposes, it is better to design the function to take wave reference parameters rather than a string list.

Operating on Qualified Waves

This example illustrates how to operate on waves that match a certain criterion. It is broken into two functions - one that creates the list of qualified waves and a second that operates on them. This organization gives us a general purpose routine (ListOfMatrices) that we would not have if we wrote the whole thing as one function.

```
Function/S ListOfMatrices()
   String list = ""
   Variable index=0
   do
      WAVE/Z w=WaveRefIndexedDFR(:,index)    // Get next wave.
      if (WaveExists(w) == 0)
         break                               // No more waves.
      endif
      if (WaveDims(w) == 2)
         // Found matrix. Add to list with separator.
         list += NameOfWave(w) + ";"
      endif
      index += 1
   while(1)             // Loop till break above.
   return list
End

Function ChooseAndDisplayMatrix()
   String theList = ListOfMatrices()

   String theMatrix
   Prompt theMatrix, "Matrix to display:", popup theList
   DoPrompt "Display Matrix", theMatrix
   if (V_Flag != 0)
      return -1
   endif

   WAVE m = $theMatrix
   NewImage m
End
```

In the preceding example, we needed a list of wave names in a string to use in a Prompt statement. More often we want a list of wave references on which to operate. The next example illustrates how to do this using a general purpose routine that returns a list of wave references in a free wave:

```
// Returns a free wave containing wave references
// for each 2D wave in the current data folder
Function/WAVE GetMatrixWavesInCDF()
   Variable numWavesInCDF = CountObjects(":", 1)
   Make/FREE/WAVE/N=(numWavesInCDF) list

   Variable numMatrixWaves = 0
   Variable i
   for(i=0; i<numWavesInCDF; i+=1)
      WAVE w = WaveRefIndexedDFR(:,i)
      Variable numDimensions = WaveDims(w)
      if (numDimensions == 2)
         list[numMatrixWaves] = w
         numMatrixWaves += 1
      endif
```