

```
CurveFit gauss gdata_Hist /D
CurveFit gauss gdata_Hist /W=W_SqrtN /I=1 /D
```

Note the addition of `"/W=W_SqrtN /I=1"` addition to the second CurveFit command. This adds the weighting using the weighting wave created by the Histogram operation. Also, `/B=4` was used to have the Histogram operation set the number of bins and bin range appropriately for the input data.

The results from the unweighted fit:

```
y0      = -3.3383 ± 2.98
A        = 133.26 ± 3.51
x0       = 0.024088 ± 0.0252
width    = 1.5079 ± 0.0578
```

And from the weighted fit:

```
y0      = 0.33925 ± 0.804
A        = 135.21 ± 5.25
x0       = 0.0038416 ± 0.031
width    = 1.3604 ± 0.0405
```

The results are clearly and significantly different. Since we created fake data we know what to expect the results to be; the weighted fit also is closer to our expectation.

There are some possible objections to this weighted fit, that may or may not be important to you.

The weighted fit is only an approximate solution to the fact that the actual errors follow a Poisson distribution. The truly correct way to fit count data is to do a maximum likelihood fit. Igor does not directly support maximum likelihood fitting.

When using the square root approximation of the standard deviation of the Poisson distribution, the fitted model is a better approximation of the actual counts than each individual original data points. But Igor doesn't have a way to replace the weighting based on the current iteration, so the only way to do that is to do the fit, re-compute the weighting wave and do the fit again. Repeat long enough that it doesn't change much any more.

The shape of the Poisson distribution is well-approximated by a Gaussian only for large numbers of counts. In practice, "large" may mean "more than 5 or so". In our example, only five points out on the tails have five or fewer counts, but we started with over 1000 points. You may not be so lucky!

Iterative Weighted Fit

This section is for advanced users only. It provides an example of an iterative fit with corrected weighting wave.

```
Function FitGaussHistogram(Wave histwave, Wave InSqrtNwave)
    // Make a copy so that we don't change the input wave
    Duplicate/FREE InSqrtNwave, sqrtNwave

    // Get a first cut at the correct fit using sqrtNwave provided by Histogram
    CurveFit/Q gauss histwave /W=sqrtNwave /I=1
    Wave W_coef

    // Save the fit solution for comparison in the loop
    Duplicate/FREE W_coef, lastCoef

    // Compute the length of the initial solution to use in the stopping criterion
    MatrixOp/FREE/O length = sqrt(sum(magSqr(W_coef)))
    Variable initialLength = length[0]

    // Now loop and re-compute the weighting wave for each iteration
    // Do a new fit with the new weighting
    do
```