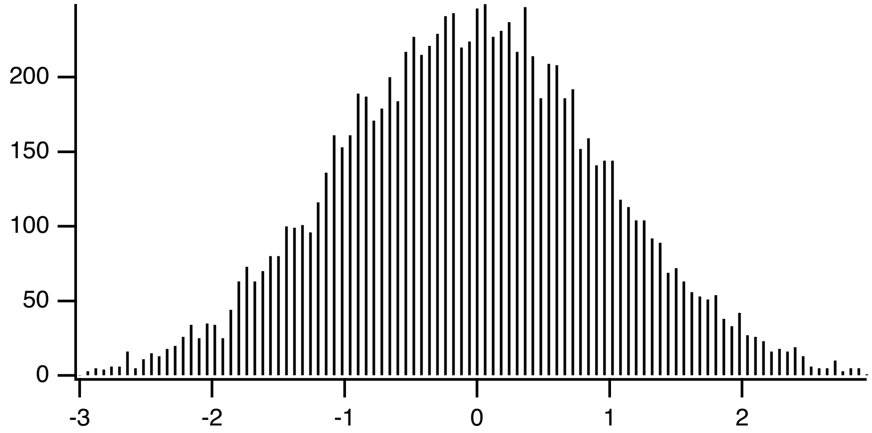


The raw data has 10,000 samples drawn from a Gaussian distribution having standard deviation of 1.0, and mean of zero. The histogram is made with 100 bins with width of 6/100 = 0.06. The amplitude of the histogram is expected to be:

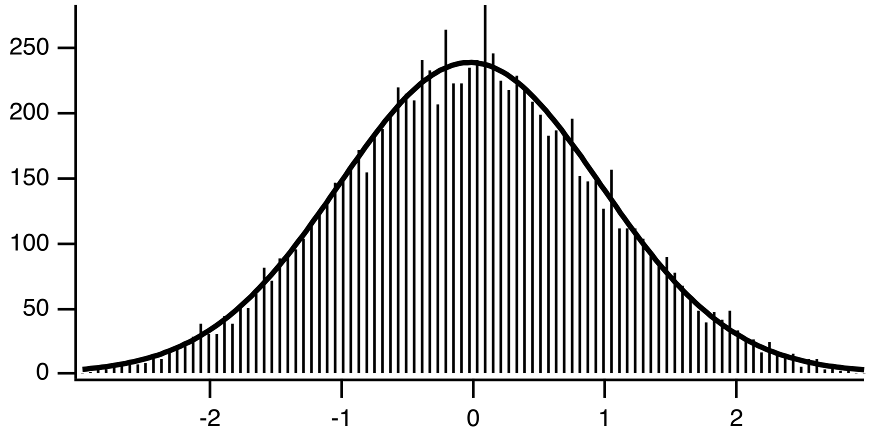
```
A = (N*dx)/sqrt(2*pi)/sigma = (10000*0.06)/sqrt(2*pi)/1 = 239.365
```

The X scaling of the wave hist uses the Histogram default of the left side of the bins.

Curve Fitting to a Histogram

Because the values in the source wave are Gaussian deviates generated by the gnoise function, the histogram should have the familiar Gaussian bell-shape. You can estimate the characteristics of the population the samples were taken from by fitting a Gaussian curve to the data. First try fitting a Gaussian curve to the example histogram:

```
CurveFit gauss hist /D     // Curve fit to histogram
```



The solution from the curve fit is:

```
y0    =  -0.85917 ± 2.74
A     =  238.23 ± 3.07
x0    =  -0.047537 ± 0.0113
width =  1.4337 ± 0.0272
```

Because of the definition of Igor's built-in gauss fit function, we expect the width coefficient to be sqrt(2)*sigma or 1.4142. The coefficients from this fit are within one standard deviation of the expected values, except for peak position x0.

The peak position x0 is shifted approximately half a bin below zero. Since the gnoise() function produces random numbers with mean of zero, we would expect x0 to be close to zero. The shifted value of x0 is a result of Igor's way of storing the X values for histogram bins. Setting the X value to the left edge is good for displaying a bar chart, but bad for curve fitting.

By default, a histogram wave has the X scaling set such that an X value gives the value at the left end of a bin. Usually if you are going to fit a curve to a histogram, you want centered X values. You can change the X scaling to centered values using SetScale. But it is easier to simply use the option to have the Histogram operation produce output with bin-centered X values by adding the /C flag to the Histogram command:

```
Histogram/C/B={-3, (3 - -3)/100, 100} noise, hist  // Perform histogram
CurveFit gauss hist /D                             // Curve fit to histogram
```

The result of the curve fit is the same as before except that x0 is closer to what we expect:

```
y0    =  -0.85917 ± 2.74
A     =  238.23 ± 3.07
x0    =  -0.017537 ± 0.0113
width =  1.4337 ± 0.0272
```

But this shifts the trace showing the histogram by half a bin on the graph. If the trace is displayed using markers or dots, this may be what is desired, but if you have used bars, the display is incorrect.

Another possibility is to make an X wave to go with the histogram data. This X wave would contain X values shifted by half a bin. Use this X wave as input to the curve fit, but don't use it on the graph:

```
Histogram/B={-3, (3 - -3)/100, 100} noise, hist    // Histogram without /C
Duplicate hist, hist_x
hist_x = x + deltax(hist)/2
CurveFit gauss  hist /X=hist_x/D
```

Use this method to graph the original histogram wave without modifying the X scaling, so a graph using bars is correct. It also gives a curve fit that uses the center X values, giving the correct x0. You could also use the Histogram operation twice, once with the /C flag to get bin-centered X values, and once without to get the shifted X scaling appropriate for bars. Both methods have the drawback of creating an extra wave that you must track.

This fit is not statistically correct. Because the histogram represents counts, the values in a histogram should have uncertainties described by a Poisson distribution. The standard deviation of a Poisson distribution is equal to the square root of the mean, which implies that the estimated uncertainty of a histogram bin depends on the magnitude of the value. This in turn implies that the errors are not constant and a curve fit will give a biased solution and possibly poor estimates of the uncertainties of the fit coefficients.

This problem can be solved approximately using a weighting wave. The appropriate weighting wave is generated by the Histogram operation if you add the /N flag or turn on the Create Square Root(N) Wave checkbox in the Histogram dialog.

The next example makes a new data set using gnoise to make gaussian-distributed values, makes a histogram with bin-centered X values and the appropriate weighting wave, and then does two curve fits, one without weighting and one with.

We expect coefficient values from the curve fit to be:

```
y0       =  0
A        =  (1024*deltax(gdata_Hist))/sqrt(2*pi)/1 = 139.863
x0       =  0
width    =  1.4142

SetRandomSeed 0.5             // For reproducible randomness
Make/O/N=1024 gdata = gnoise(1)
Make/O/N=20 gdata_Hist
Histogram/C/N/B=4 gdata,gdata_Hist
Display gdata_Hist
ModifyGraph mode=3,marker=8
```