Chi-square goodness-of-fit test

Syntax

h = chi2gof(x)
[h,p] = chi2gof(...)
[h,p,stats] = chi2gof(...)
[...] = chi2gof(X,name1,val1,name2,val2,...)

Description

h = chi2gof(x) performs a chi-square goodness-of-fit test of the default null hypothesis that the data in vector x are a random sample from a normal distribution with mean and variance estimated from x, against the alternative that the data are not normally distributed with the estimated mean and variance. The result h is 1 if the null hypothesis can be rejected at the 5% significance level. The result h is 0 if the null hypothesis cannot be rejected at the 5% significance level.

The null distribution can be changed from a normal distribution to an arbitrary discrete or continuous distribution. See the syntax for specifying optional argument name/value pairs below.

The test is performed by grouping the data into bins, calculating the observed and expected counts for those bins, and computing the chi-square test statistic

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}$$

where O_i are the observed counts and E_i are the expected counts. The statistic has an approximate chi-square distribution when the counts are sufficiently large. Bins in either tail with an expected count less than 5 are pooled with neighboring bins until the count in each extreme bin is at least 5. If bins remain in the interior with counts less than 5, chi2gof displays a warning. In this case, you should use fewer bins, or provide bin centers or edges, to increase the expected counts in all bins. (See the syntax for specifying optional argument name/value pairs below.)

chi2gof sets the number of bins, nbins, to 10 by default, and compares the test statistic to a chi-square distribution with nbins - 3 degrees of freedom to take into account the two estimated parameters.

[h,p] = chi2gof(...) also returns the p-value of the test, p. The p-value is the probability, under assumption of the null hypothesis, of observing the given statistic or one more extreme.

[h,p,stats] = chi2gof(...) also returns a structure stats with the following fields:

- chi2stat — The chi-square statistic
- df — Degrees of freedom
- edges — Vector of bin edges after pooling
- O — Observed count in each bin
- E — Expected count in each bin

[...] = chi2gof(X,name1,val1,name2,val2,...) specifies optional argument name/value pairs chosen from the following lists. Argument names are case insensitive and partial matches are allowed.

The following name/value pairs control the initial binning of the data before pooling. You should not specify more than one of these options.

- 'nbins' — The number of bins to use. Default is 10.
- 'ctrs' — A vector of bin centers
- 'edges' — A vector of bin edges

The following name/value pairs determine the null distribution for the test. You should not specify both 'cdf' and...
The following name/value pairs determine the null distribution for the test. You should not specify both ‘cdf’ and ‘expected’.

- **‘cdf’** — A fully specified cumulative distribution function. This can be a function name or function handle, and the function must take $x$ as its only argument. Alternately, you can provide a cell array whose first element is a function name or handle, and whose later elements are parameter values, one per cell. The function must take $x$ as its first argument, and other parameters as later arguments.
- **‘expected’** — A vector with one element per bin specifying the expected counts for each bin
- **‘nparams’** — The number of estimated parameters; used to adjust the degrees of freedom to be $n_{bins} - 1 - n_{params}$, where $n_{bins}$ is the number of bins

If your ‘cdf’ or ‘expected’ input depends on estimated parameters, you should use ‘nparams’ to ensure that the degrees of freedom for the test is correct. If ‘cdf’ is a cell array, the default value of ‘nparams’ is the number of parameters in the array; otherwise the default is 0.

The following name/value pairs control other aspects of the test.

- **‘emin’** — The minimum allowed expected value for a bin; any bin in either tail having an expected value less than this amount is pooled with a neighboring bin. Use the value 0 to prevent pooling. The default is 5.
- **‘frequency’** — A vector the same length as $x$ containing the frequency of the corresponding $x$ values
- **‘alpha’** — Significance level for the test. The default is 0.05.

### Examples

**Example 1**
Equivalent ways to test against an unspecified normal distribution with estimated parameters:

```matlab
x = normrnd(50,5,100,1); [h,p] = chi2gof(x)
h = 0 p = 0.7532 [h,p] = chi2gof(x,'cdf',@(z)normcdf(z,mean(x),std(x)),'nparams',2) h = 0 p = 0.7532 [h,p] = chi2gof(x,'cdf',{@normcdf,mean(x),std(x)}) h = 0 p = 0.7532```

**Example 2**
Test against the standard normal:

```matlab>x = randn(100,1); [h,p] = chi2gof(x,'cdf',@normcdf) h = 0 p = 0.9443```

**Example 3**
Test against the standard uniform:

```matlab
x = rand(100,1);

n = length(x);
edges = linspace(0,1,11);
expectedCounts = n * diff(edges);
[h,p,st] = chi2gof(x,'edges',edges,...
    'expected',expectedCounts)
```

```
    h = 0
    p = 0.3191
    st =
        chi2stat: 10.4000
        df: 9
        edges: [1x11 double]
          O: [6 11 4 12 15 8 14 9 11 10]
          E: [1x10 double]
```

**Example 4**

Test against the Poisson distribution by specifying observed and expected counts:

```matlab
bins = 0:5;
obsCounts = [6 16 10 12 4 2];
n = sum(obsCounts);
lambdaHat = sum(bins.*obsCounts)/n;
expCounts = n*poisspdf(bins,lambdaHat);

[h,p,st] = chi2gof(bins,'ctrs',bins,...
    'frequency',obsCounts,...
    'expected',expCounts,...
    'nparams',1)
```

```
h = 0
p = 0.4654
st =
    chi2stat: 2.5550
    df: 3
    edges: [1x6 double]
      O: [6 16 10 12 6]
```

**See Also**

crosstab, chi2cdf, ktest, lillietest