chi2gof

Chi-square goodness-of-fit test

[collapse all in page](javascript:void(0);)

**Syntax**

h = chi2gof(x)

h = chi2gof(x,Name,Value)

[h,p] = chi2gof(**\_\_\_**)

[h,p,stats] = chi2gof(**\_\_\_**)

**Description**

[example](http://www.mathworks.com/help/stats/chi2gof.html" \l "btv1j1v-7)

[h](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_h) = chi2gof([x](http://www.mathworks.com/help/stats/chi2gof.html" \l "inputarg_x)) returns a test decision for the null hypothesis that the data in vector x comes from a normal distribution with a mean and variance estimated from x, using the [chi-square goodness-of-fit test](http://www.mathworks.com/help/stats/chi2gof.html" \l "btv1j1v-11). The alternative hypothesis is that the data does not come from such a distribution. The result h is 1 if the test rejects the null hypothesis at the 5% significance level, and 0 otherwise.

[example](http://www.mathworks.com/help/stats/chi2gof.html" \l "btv1kvu-1)

[h](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_h) = chi2gof([x](http://www.mathworks.com/help/stats/chi2gof.html" \l "inputarg_x),[Name,Value](http://www.mathworks.com/help/stats/chi2gof.html" \l "namevaluepairarguments)) returns a test decision for the chi-square goodness-of-fit test with additional options specified by one or more name-value pair arguments. For example, you can test for a distribution other than normal, or change the significance level of the test.

[example](http://www.mathworks.com/help/stats/chi2gof.html" \l "btv1j1v-8)

[[h](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_h),[p](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_p)] = chi2gof(**\_\_\_**) also returns the *p*-value p of the hypothesis test, using any of the input arguments from the previous syntaxes.

[example](http://www.mathworks.com/help/stats/chi2gof.html" \l "btv1j1v-9)

[[h](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_h),[p](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_p),[stats](http://www.mathworks.com/help/stats/chi2gof.html" \l "outputarg_stats)] = chi2gof(**\_\_\_**) also returns the structure stats, containing information about the test statistic.

**Examples**

[collapse all](javascript:void(0);)

**Test for a Normal Distribution**

[Open Script](matlab:openExample('stats/TestForANormalDistributionChi2GofExample'))

Create a standard normal probability distribution object. Generate a data vector x using random numbers from the distribution.

pd = makedist('Normal');

rng default; % for reproducibility

x = random(pd,100,1);

Test the null hypothesis that the data in x comes from a population with a normal distribution.

h = chi2gof(x)

h =

0

The returned value h = 0 indicates that chi2gof does not reject the null hypothesis at the default 5% significance level.

**Test the Hypothesis at a Different Significance Level**

[Open Script](matlab:openExample('stats/TesttheHypothesisataDifferentSignificanceLevelExample'))

Create a standard normal probability distribution object. Generate a data vector x using random numbers from the distribution.

pd = makedist('Normal');

rng default; % for reproducibility

x = random(pd,100,1);

Test the null hypothesis that the data in x comes from a population with a normal distribution at the 1% significance level.

[h,p] = chi2gof(x,'Alpha',0.01)

h =

0

p =

0.3775

The returned value h = 0 indicates that chi2gof does not reject the null hypothesis at the 1% significance level.

**Test for a Weibull Distribution Using a Probability Distribution Object**

[Open Script](matlab:openExample('stats/TestforaWeibullDistributionUsingaProbabilityDistributionExample'))

Navigate to the appropriate folder and load the lightbulb lifetime sample data.

cd(matlabroot);

cd('help/toolbox/stats/examples');

load lightbulb;

Create a vector from the first column of the data matrix, which contains the lifetime in hours of the lightbulbs.

x = lightbulb(:,1);

Test the null hypothesis that the data in x comes from a population with a Weibull distribution. Use fitdist to create a probability distribution object with A and Bparameters estimated from the data.

pd = fitdist(x,'Weibull');

h = chi2gof(x,'CDF',pd)

h =

1

The returned value h = 1 indicates that chi2gof rejects the null hypothesis at the default 5% significance level.

**Test for a Poisson Distribution**

[Open Script](matlab:openExample('stats/TestforaPoissonDistributionExample'))

Create six bins, numbered 0 through 5, to use for data pooling.

bins = 0:5;

Create a vector containing the observed counts for each bin and compute the total number of observations.

obsCounts = [6 16 10 12 4 2];

n = sum(obsCounts);

Fit a Poisson probability distribution object to the data and compute the expected count for each bin. Use the transpose operator .' to transform bins and obsCounts from row vectors to column vectors.

pd = fitdist(bins','Poisson','Frequency',obsCounts');

expCounts = n \* pdf(pd,bins);

Test the null hypothesis that the data in obsCounts comes from a Poisson distribution with a lambda parameter equal to lambdaHat.

[h,p,st] = chi2gof(bins,'Ctrs',bins,...

'Frequency',obsCounts, ...

'Expected',expCounts,...

'NParams',1)

h =

0

p =

0.4654

st =

struct with fields:

chi2stat: 2.5550

df: 3

edges: [-0.5000 0.5000 1.5000 2.5000 3.5000 5.5000]

O: [6 16 10 12 6]

E: [7.0429 13.8041 13.5280 8.8383 6.0284]

The returned value h = 0 indicates that chi2gof does not reject the null hypothesis at the default 5% significance level. The vector E contains the expected counts for each bin under the null hypothesis, and O contains the observed counts for each bin.

**Input Arguments**

[collapse all](javascript:void(0);)

**x — Sample data  
vector**

Sample data for the hypothesis test, specified as a vector.

**Name-Value Pair Arguments**

Specify optional comma-separated pairs of Name,Value arguments. Name is the argument name and Value is the corresponding value. Name must appear inside single quotes (' '). You can specify several name and value pair arguments in any order as Name1,Value1,...,NameN,ValueN.

**Example:**'NBins',8,'Alpha',0.01 pools the data into eight bins and conducts the hypothesis test at the 1% significance level.

[collapse all](javascript:void(0);)

**'NBins' — Number of bins  
10 (default) | positive integer value**

Number of bins to use for the data pooling, specified as the comma-separated pair consisting of 'NBins' and a positive integer value. If you specify a value for NBins, do not specify a value for Ctrs or Edges.

**Example:**'NBins',8

**Data Types:**single | double

**'Ctrs' — Bin centers  
vector**

Bin centers, specified as the comma-separated pair consisting of 'Ctrs' and a vector of center values for each bin. If you specify a value for Ctrs, do not specify a value for NBins or Edges.

**Example:**'Ctrs',[1 2 3 4 5]

**Data Types:**single | double

**'Edges' — Bin edges  
vector**

Bin edges, specified as the comma-separated pair consisting of 'Edges' and a vector of edge values for each bin. If you specify a value for Edges, do not specify a value for NBins or Ctrs.

**Example:**'Edges',[-2.5 -1.5 -0.5 0.5 1.5 2.5]

**Data Types:**single | double

**'CDF' — cdf of hypothesized distribution  
probability distribution object | function handle | cell array**

The cdf of the hypothesized distribution, specified as the comma-separated pair consisting of 'CDF' and a probability distribution object, function handle, or cell array.

* If CDF is a probability distribution object, the degrees of freedom account for whether you estimate the parameters using [fitdist](http://www.mathworks.com/help/stats/fitdist.html) or specify them using [makedist](http://www.mathworks.com/help/stats/makedist.html).
* If CDF is a function handle, the distribution function must take x as its only argument.
* If CDF is a cell array, the first element must be a function handle, and the remaining elements must be parameter values, one per cell. The function must take xas its first argument, and the other parameters in the array as later arguments.

If you specify a value for CDF, do not specify a value for Expected.

**Example:**'CDF',pd\_object

**Data Types:**single | double

**'Expected' — Expected counts  
vector of nonnegative values**

Expected counts for each bin, specified as the comma-separated pair of 'Expected' and a vector of nonnegative values. If Expected depends on estimated parameters, use NParams to ensure that chi2gof correctly calculates the degrees of freedom. If you specify a value for Expected, do not specify a value for CDF.

**Example:**'Expected',[19.1446 18.3789 12.3224 8.2432 4.1378]

**Data Types:**single | double

**'NParams' — Number of estimated parameters  
positive integer value**

Number of estimated parameters used to describe the null distribution, specified as the comma-separated pair consisting of 'NParams' and a positive integer value. This value adjusts the degrees of freedom of the test based on the number of estimated parameters used to compute the cdf or expected counts.

The default value for NParams depends on how you specify the null distribution:

* If you specify [CDF](http://www.mathworks.com/help/stats/chi2gof.html#inputarg_CDF) as a probability distribution object, NParams is equal to the number of estimated parameters used to create the object.
* If you specify CDF as a function name or handle, the default value of NParams is 0.
* If you specify CDF as a cell array, the default value of NParams is the number of parameters in the array.
* If you specify [Expected](http://www.mathworks.com/help/stats/chi2gof.html#inputarg_Expected), the default value of NParams is 0.

**Example:**'NParams',1

**Data Types:**single | double

**'EMin' — Minimum expected count per bin  
5 (default) | nonnegative integer value**

Minimum expected count per bin, specified as the comma-separated pair consisting of 'EMin' and a nonnegative integer value. If the bin at the extreme end of either tail has an expected value less than EMin, it is combined with a neighboring bin until the count in each extreme bin is at least 5. If any interior bins have a count less than 5, chi2gof displays a warning, but does not combine the interior bins. In that case, you should use fewer bins, or provide bin centers or edges, to increase the expected counts in all bins. Specify EMin as 0 to prevent the combining of bins.

**Example:**'EMin',0

**Data Types:**single | double

**'Frequency' — Frequency  
vector of nonnegative integer values**

Frequency of data values, specified as the comma-separated pair consisting of 'Frequency' and a vector of nonnegative integer values that is the same length as the vector [x](http://www.mathworks.com/help/stats/chi2gof.html#inputarg_x).

**Example:**'Frequency',[20 16 13 10 8]

**Data Types:**single | double

**'Alpha' — Significance level  
0.05 (default) | scalar value in the range (0,1)**

Significance level of the hypothesis test, specified as the comma-separated pair consisting of 'Alpha' and a scalar value in the range (0,1).

**Example:**'Alpha',0.01

**Data Types:**single | double

**Output Arguments**

[collapse all](javascript:void(0);)

**h — Hypothesis test result  
1 | 0**

Hypothesis test result, returned as a logical value.

* If h = 1, this indicates the rejection of the null hypothesis at the [Alpha](http://www.mathworks.com/help/stats/chi2gof.html#inputarg_Alpha) significance level.
* If h = 0, this indicates a failure to reject the null hypothesis at the Alpha significance level.

**p — *p*-value  
scalar value in the range [0,1]**

*p*-value of the test, returned as a scalar value in the range [0,1]. p is the probability of observing a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. Small values of p cast doubt on the validity of the null hypothesis.

**stats — Test statistics  
structure**

Test statistics, returned as a structure containing the following:

* chi2stat — Value of the test statistic.
* df — Degrees of freedom of the test.
* edges — Vector of bin edges after pooling.
* O — Vector of observed counts for each bin.
* E — Vector of expected counts for each bin.

**More About**

[collapse all](javascript:void(0);)

**Chi-Square Goodness-of-Fit Test**

The chi-square goodness-of-fit test determines if a data sample comes from a specified probability distribution, with parameters estimated from the data.

The test groups the data into bins, calculating the observed and expected counts for those bins, and computing the chi-square test statistic

*χ*2=*N*Ξ*i*=1(*Oi*−*Ei*)2/*Ei* ,

where *Oi* are the observed counts and *Ei* are the expected counts based on the hypothesized distribution. The test statistic has an approximate chi-square distribution when the counts are sufficiently large.

**Algorithms**

chi2gof compares the value of the test statistic to a chi-square distribution with degrees of freedom equal to *nbins* - 1 - *nparams*, where *nbins* is the number of bins used for the data pooling and *nparams* is the number of estimated parameters used to determine the expected counts. If there are not enough degrees of freedom to conduct the test, chi2gof returns the *p*-value as NaN.

**See Also**

[kstest](http://www.mathworks.com/help/stats/kstest.html) | [lillietest](http://www.mathworks.com/help/stats/lillietest.html)