Testing Chi 2 GOF Test

Table of Contents

Load example data

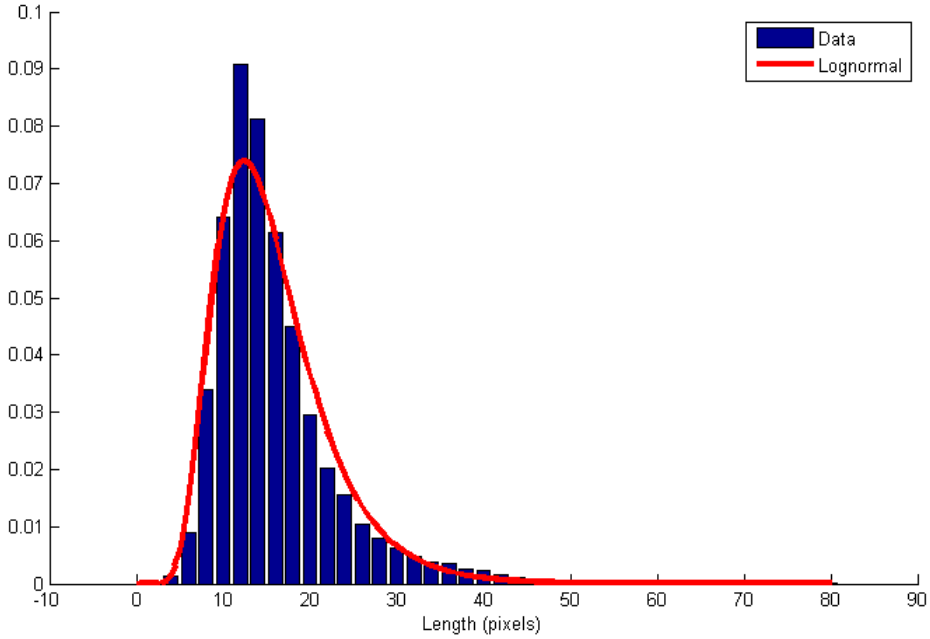
```
load('exampleGOFdata.mat');
```

Matlab's Chi Squared Goodness of Fit (Lognormal)

```
% Shift data towards origin and compute histogram
datax = lengthx;
shift = min(lengths)-1;
datas = abs(lengths - shift)+10^-15;
datan = hist(datas,lengthx);
xout = 0:0.5:80;

% Start figure
figure('Position',[150 150 800 500])
dists = {'lognormal';};
disttext = {'Lognormal';};
colors = [1 0 0];
hold on;

% Plot histogram
bar(datax,datan/trapz(datax,datan));

fprintf('\np-values:\n');

% Compute distribution parameters
params = mle(datas,'distribution',char(dists(1)));

% Goodness of fit test
[h,p,stats] = chi2gof(datas,'cdf',@(z)cdf(char(dists(1)),z,params(1),params(2)),'n

% Finish plotting
plot(xout,pdf(char(dists(1)),xout,params(1),params(2)),'g','LineWidth',3,'Color',c
fprintf('%s distribution: %d\n',char(disttext(1)),p);
hold off;
legend(['Data'; disttext])
xlabel('Length (pixels)')
```

p-values:

Lognormal distribution: 2.824955e-114

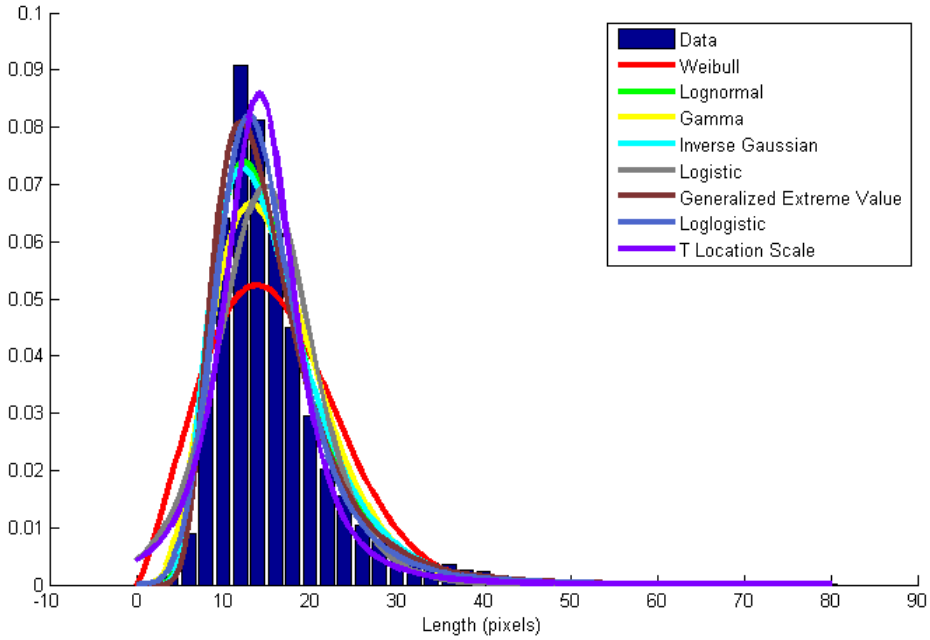


Matlab's Chi Squared Goodness of Fit (Other Distributions)

```
datax = lengthx;
shift = min(lengths)-1;
datas = abs(lengths - shift)+10^-15;
datan = hist(datas,lengthx);
xout = 0:0.5:80;

figure('Position',[150 150 800 500])

dists = {'wbl'; 'lognormal'; 'Gamma'; 'inversegaussian'; 'logistic'; 'gev'; 'loglo
disttext = {'Weibull'; 'Lognormal'; 'Gamma'; 'Inverse Gaussian'; 'Logistic'; ...
            'Generalized Extreme Value'; 'Loglogistic'; 'T Location Scale'};
colors = [1 0 0; 0 1 0; 1 1 0; 0 1 1; 0.5 0.5 0.5; 0.5 0.2 0.2; 0.3 0.4 0.8; 0.5 0

fprintf('\np-values:\n');
hold on;
% Plot histogram
bar(datax,datan/trapz(datax,datan));
for i=1:length(dists)
    params = mle(datas,'distribution',char(dists(i)));
    if length(params)==1
        [h,p] = chi2gof(datas,'cdf',@(z)cdf(char(dists(i)),z,params(1)));
        plot(xout,pdf(char(dists(i)),xout,params(1)),'g','LineWidth',3,'Color',col
    elseif length(params)==2
        [h,p,stats] = chi2gof(datas,'cdf',@(z)cdf(char(dists(i)),z,params(1),param
        plot(xout,pdf(char(dists(i)),xout,params(1),params(2)),'g','LineWidth',3,'
    elseif length(params)==3
        [h,p] = chi2gof(datas,'cdf',@(z)cdf(char(dists(i)),z,params(1),params(2),p
        plot(xout,pdf(char(dists(i)),xout,params(1),params(2),params(3)),'g','Line
```

```
    end
    fprintf('%s distribution: %d\n',char(disttext(i)),p);
end
hold off;
legend(['Data'; disttext])
xlabel('Length (pixels)')
```

```
p-values:
Weibull distribution: NaN
Lognormal distribution: 2.824955e-114
Gamma distribution: 0
Inverse Gaussian distribution: 4.296082e-122
Logistic distribution: NaN
Generalized Extreme Value distribution: 1.462544e-032
Loglogistic distribution: 1.549296e-050
T Location Scale distribution: 0
```

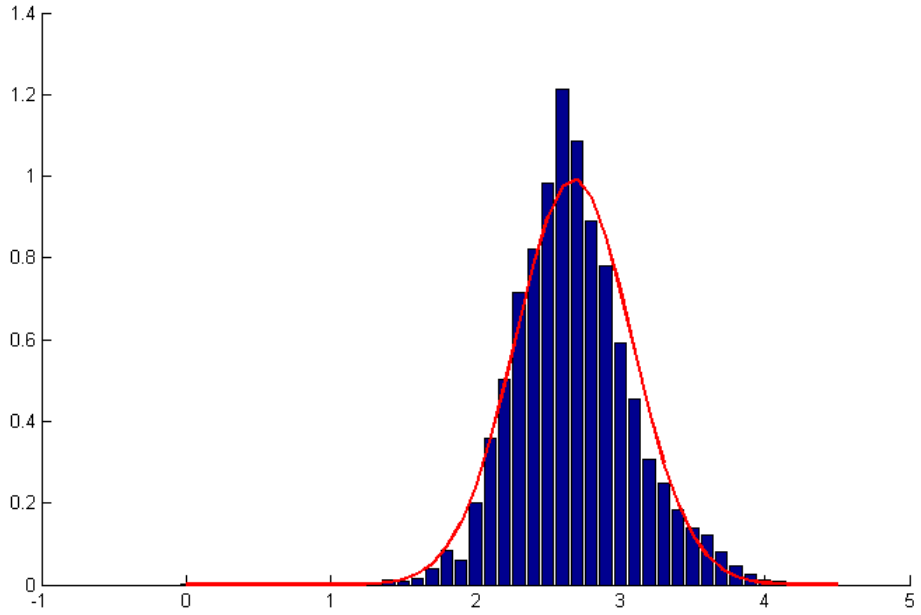


Manual Chi^2 Goodness of Fit Test (With Crack Length Data)

```
% Use example length data
datax = lengthx;
shift = min(lengths)-1;
datas = abs(lengths - shift)+10^-15;
datax = 0:0.1:4.5;

% Convert data to normal
data2 = log(datas);
datan = hist(data2,datax);
mu = mean(data2);
sig = std(data2);
```

```
% Plot histogram with lognormal fit
figure('Position',[150 150 800 500])
hold on
bar(datax,datan/trapz(datax,datan));
plot(datax,normpdf(datax,mu,sig),'r','LineWidth',2)
hold off

% Create bins for GOF test
endpoints = [0 1.5 2 2.5 3 3.5 4];
nbins = length(endpoints)-1;
observed = zeros(nbins,1);
expected = zeros(nbins,1);
totaln = length(data2);
df = nbins-1-2;

% Count observed values
for i=1:nbins
    observed(i) = length(data2(data2>endpoints(i) & data2<endpoints(i+1)));
end

% Compute expected values (area under curve using CDF)
for i=1:nbins
    expected(i) = totaln*(normcdf(endpoints(i+1),mu,sig) - normcdf(endpoints(i),mu
end

% Calculate chi squared statistic
test = sum( ((expected-observed).^2)./expected );

% Compute p-value for test
p = 1-chi2cdf(test,df);

fprintf('\nP-value for Chi2 GOF test for crack length data:\n');
fprintf('p = %f\n',p);
```

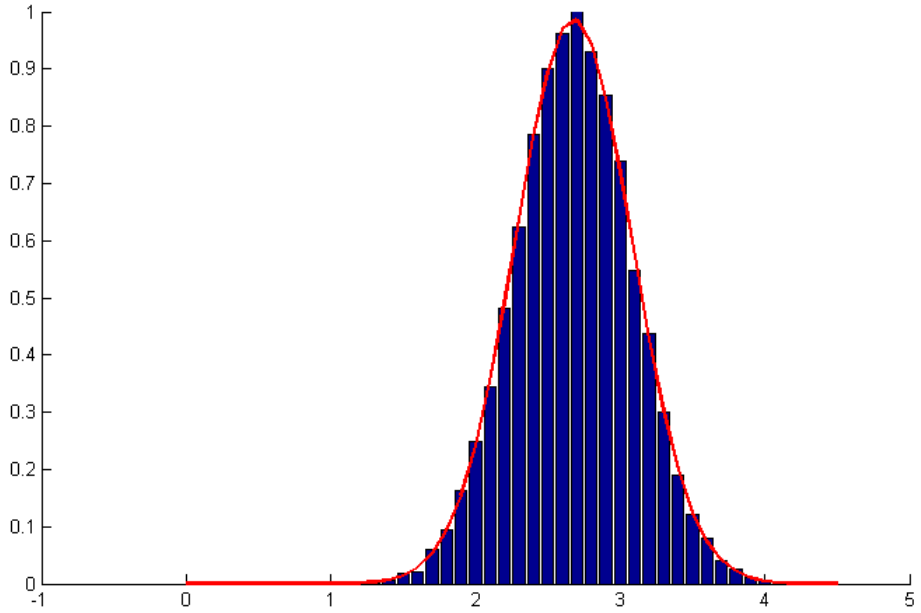
```
P-value for Chi2 GOF test for crack length data:
p = 0.000000
```



Manual Chi^2 Goodness of Fit Test (With Lognormal Data)

```
% Use MATLAB generated lognormal random numbers
datax = lengthx;
datas = lognrnd(2.68,0.4,15000,1);
datax = 0:0.1:4.5;

% Convert data to normal
data2 = log(datas);
datan = hist(data2,datax);
mu = mean(data2);
sig = std(data2);

% Plot histogram with lognormal fit
figure('Position',[150 150 800 500])
hold on
bar(datax,datan/trapz(datax,datan));
plot(datax,normpdf(datax,mu,sig),'r','LineWidth',2)
hold off

% Create bins for GOF test
endpoints = [0 1.5 2 2.5 3 3.5 4];
nbins = length(endpoints)-1;
observed = zeros(nbins,1);
expected = zeros(nbins,1);
totaln = length(data2);
df = nbins-1-2;

% Count observed values
for i=1:nbins
    observed(i) = length(data2(data2>endpoints(i) & data2<endpoints(i+1)));
end

% Compute expected values (area under curve using CDF)
for i=1:nbins
    expected(i) = totaln*(normcdf(endpoints(i+1),mu,sig) - normcdf(endpoints(i),mu
end

% Calculate chi squared statistic
test = sum( ((expected-observed).^2)./expected );

% Compute p-value for test
p = 1-chi2cdf(test,df);

fprintf('\nP-value for Chi2 GOF test for crack length data:\n');
fprintf('p = %f\n',p);
```

```
P-value for Chi2 GOF test for crack length data:
p = 0.562742
```



Published with MATLAB® 7.10