```
clear
```

1.Loading data

```
% Specify the filename of the Excel file to be read.
filename = 'crabsTitle.xlsx'
```

```
filename = crabsTitle.xlsx
```

```
[a, name, aCell] = xlsread(filename);
```

'a' - numeric data, 'name'- the text data (like headers), 'aCell' will contain both text and numeric data in cell format

```
dataTable = a;
Y = dataTable(:, 1);          % Y - groupping variable
widths = dataTable(:, 4);     % widths
sattelts = dataTable(:, 5);   % sattelts
weights = dataTable(:, 6);    % 'weights'
catwidths = dataTable(:,7);   % 'catwidths'
```

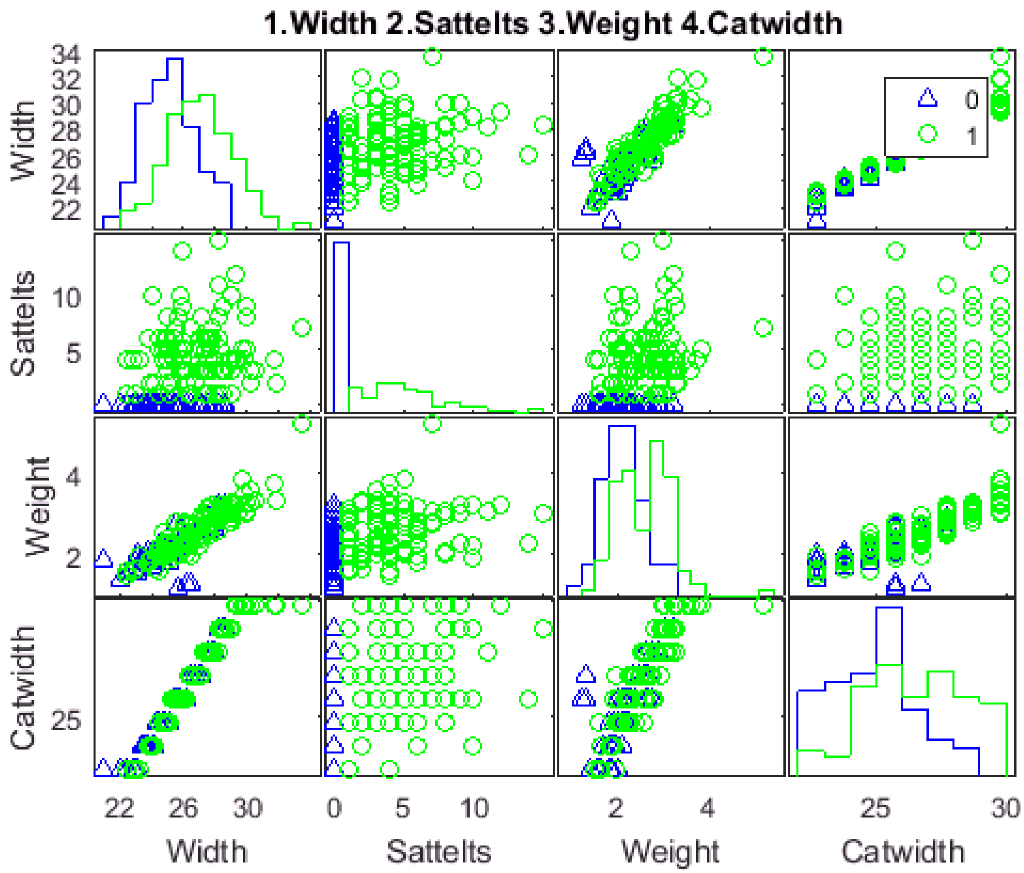
2.Data presentation

```
% Y - groupping variable.
X = Y;
% Combine matrix 'Z' with multiple attributes for each entry in the dataset
Z = [widths sattelts weights catwidths];
XZ=[X,Z]; % it's useful for preprocessing
```

gplotmatrix: *a matrix of scatter plots and histograms for data (including heterogeneous data)*

Create an object table for given data vectors:

```
tblmy = table(Y, widths, sattelts, weights, catwidths, ...
       'VariableNames', {'Y', 'Width', 'Sattelts', 'Weight', 'Catwidth'});
gplotmatrix(Z, [], Y, [], '^o*x+', [], 'on', '', ...
   {'Width', 'Sattelts', 'Weight', 'Catwidth'}, []);
title(['1.Width ', '2.Sattelts ', '3.Weight ','4.Catwidth']);
```



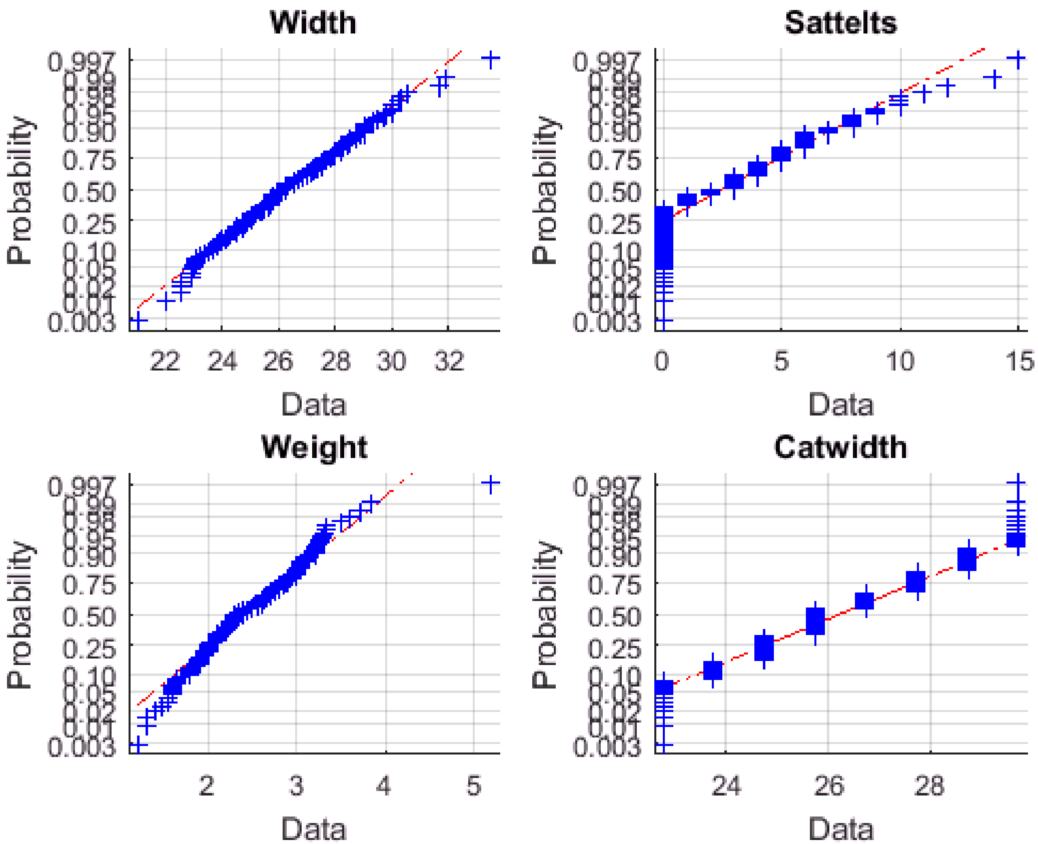
3.Estimate the normality of the data distribution

a) by normplot

```
figure
nameT={'Width', 'Sattelts', 'Weight', 'Catwidth'}
```

```
nameT =
    'Width'    'Sattelts'    'Weight'    'Catwidth'
```

```
for i = 1:4
    subplot(2,2,i);
    normplot(Z(:,i));title(nameT{i}) % sgtitle - header for total subplot
end
```



For normally distributed data points should form an approximate straight line. Empirical distribution function ecdf is constructed in accordance with empirical data, inverse of theoretical normal distribution function is determined analytically, based on an effective sample parameters and their superposition close to a straight line, since the empirical distribution is close to the theoretical one)

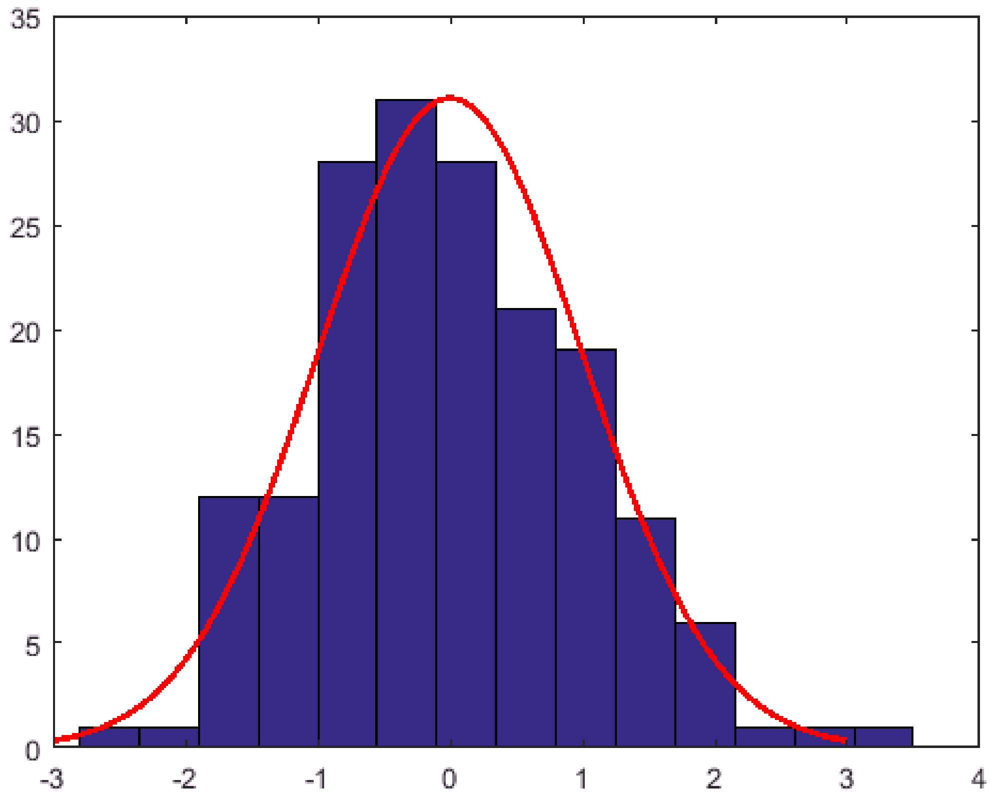
b) by Statistical test (Kolmogorov-Smirnov) - is used for Standard Normal distribution, control Width, but before let's do the standardization:

```
widthsS=(widths-mean(widths))/std(widths);
[h,p,KSstat,critValue] = kstest(widthsS)
```

```
h =
   0

p = 0.3865
KSstat = 0.0678
critValue = 0.1022
```

```
figure
histfit(widthsS)
```



4.Preprocess the matrix form

```
XZinitial = [X, Z];

% Initialize an empty array 'NaNInd' to store indices of rows with NaNs.
NaNInd = [];
[n, k] = size(XZ);

% Loop through each column of 'XZ'.
for col = 1:k

        [i] = find(isnan(XZ(:,col)));  % indices of rows with NaNs
    if ~isempty(i)
        % 'union' use because we can find the same row for other column
        NaNInd = union(NaNInd, i);
    end
end

% Remove the rows with NaNs from 'XZ'.
XZ(NaNInd,:)=[];
```

Standardization:

```
[p, q] = size(XZ);
m = mean(XZ); % vector of means
s = std(XZ); % the standard deviation of each column
XZs = (XZ - repmat(m, p, 1)) ./ repmat(s,p,1);
```

Two ways to get correlation coefficient matrix: (You cat compare using witout sign ;)

```
RhoXZs=XZs'*XZs/(p-1);
[rho,P]=corr(XZs);
```

It can be seen that these five variables are related to each other, and the correlation between variables 'Width', 'Weight' and 'Catwidth' is the most significant.

5.Linear regression model fitting (matrix form); Weights - response:

```
% Rearrange the elements of the mean and standard deviation vector
m = [m(1), m(2:5)];
s = [s(1), s(2:5)];

% The 4-th column of 'XZs' is the dependent variable in the regress.
X = XZs(:, 4);
XZs(:, 4) = [];
% Z now contains only the independent variables:
Z = XZs;
[n, k] = size(Z);
```

Regression model is obtained with directly matrix approach, taking into account only variables with standard Normal distribution as response - weights and idependent variable - x2 (widths) - according to normplot ~ Z(:,2):

```
ZEd=Z(:,2); disp(' b2 (compare with  fitlm  0.83922) is equal:' );
```

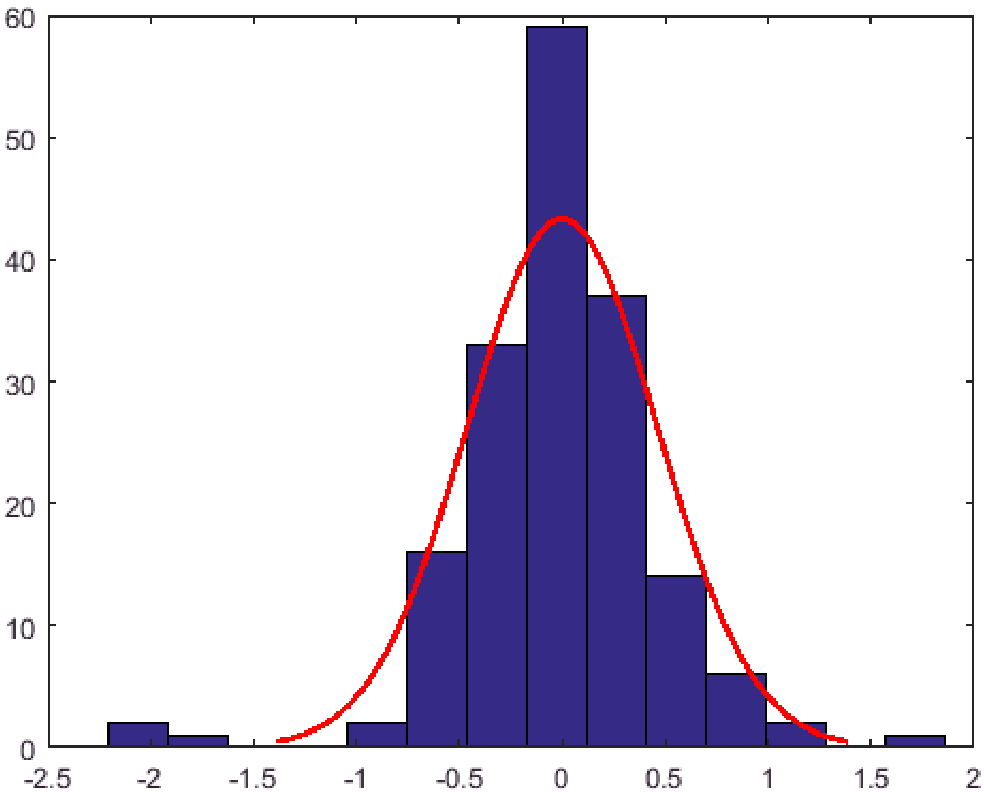
```
 b2 (compare with  fitlm  0.83922) is equal:
```

```
bEd=Z(:,2)\X % XEd=ZEd*bEd; - our system; \ - such a division
```

```
bEd = 0.8869
```

```

XEd=ZEd*bEd;  % XEd predicted by regression; XEd(i) weight of i-th crab
err=(X-XEd); figure; histfit(err);
```




```
disp('You can see outliers on histogramm!')
```

```
You can see outliers on histogramm!
```

```
SqError=sqrt(var(X-XEd)) % standard deviation
```

```
SqError = 0.4620
```

```
disp('SqError compare with result in fitlm');
```

```
SqError compare with result in fitlm
```

```

VX=var(X);  disp('corrected R2 istead 0.787 (the same) is equal:');
```

```
corrected R2 istead 0.787 (the same) is equal:
```

```
R2matrixApproach=1-var(X-XEd)/VX
```

```
R2matrixApproach = 0.7865
```

6. Fitting regression - fitlm

Fit a linear regression model: ***fitlm*** is used with Z as the independent variables and X as respnse (dependent variable) here.

```
disp('1-Intersept, x1-Y, x2-widths, x3 - sattelts, x4 - catwidths')
```

```
1-Intersept, x1-Y, x2-widths, x3 - sattelts, x4 - catwidths
```

```
lrm = fitlm(Z,X,'linear')
```

```
lrm = 

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4

Estimated Coefficients:
                   Estimate        SE         tStat        pValue  
                   __________    ________    __________    __________

    (Intercept)    1.9285e-15     0.03508    5.4974e-14             1
    x1              -0.023458    0.050527      -0.46427       0.64306
    x2                0.83922     0.15936        5.2661    4.2188e-07
    x3               0.091306    0.049213        1.8553      0.065304
    x4               0.026701     0.15844       0.16852       0.86638


Number of observations: 173, Error degrees of freedom: 168
Root Mean Squared Error: 0.461
R-squared: 0.792,  Adjusted R-Squared 0.787
F-statistic vs. constant model: 160, p-value = 3.45e-56
```

7.Function *regress*; display model coefficient and confidence intervals:

Perform linear regression using ***regress***. This function estimates the linear regression coefficients ('b'),their confidence intervals (CI) ('bINT'),the residuals ('R'), the CI for residual ('RINT') and other statistics ('STATS') from the data. The confidence level is set to 95%

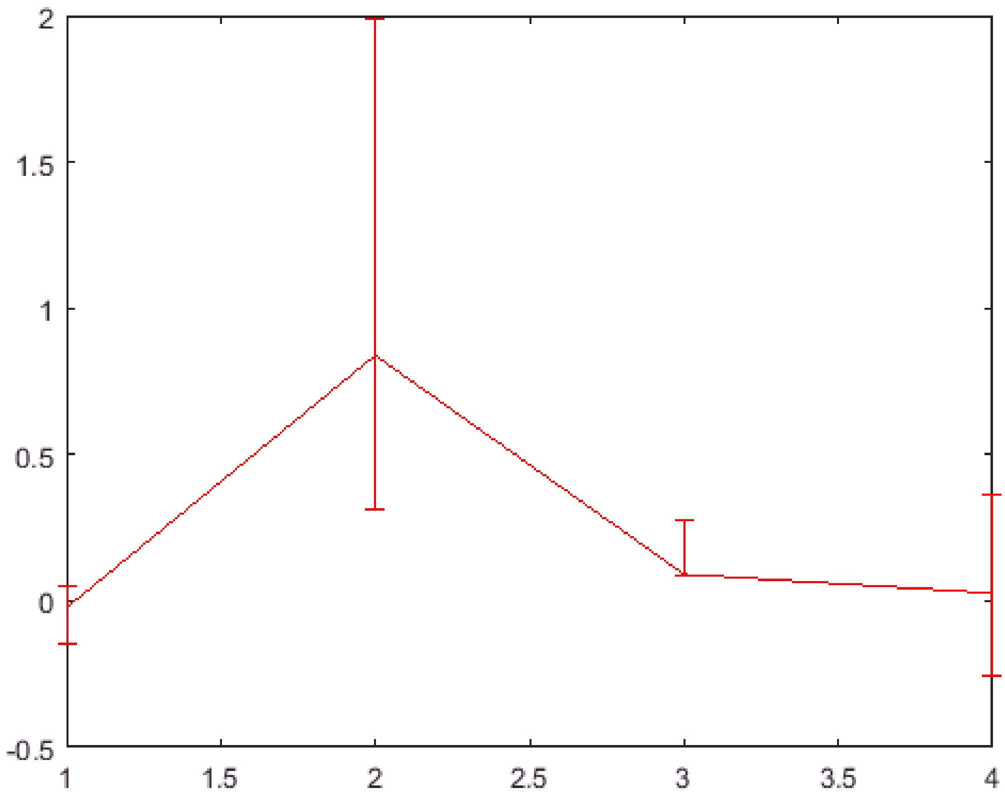
```
[b, bINT, R, RINT, STATS] = regress(X, Z, 0.05);
```

```
Warning: R-square and the F statistic are not well-defined unless X has a column of ones.
Type "help regress" for more information.
```

STATS=[R2 stat, F-stat, F p-value, error variance].

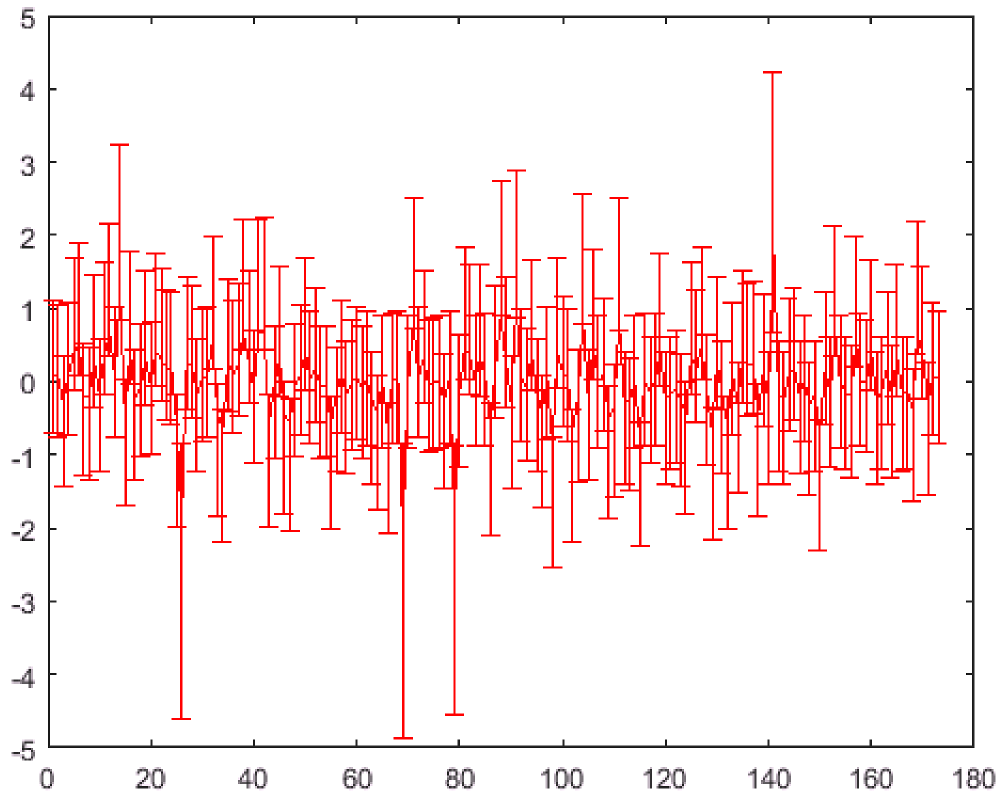
```

% 'errorbar' plots the estimated coefficients ('b') with their CI
% The points and error bars are colored red.
figure;
errorbar(1:4, b, bINT(:, 1), bINT(:, 2), 'color', 'red');
```

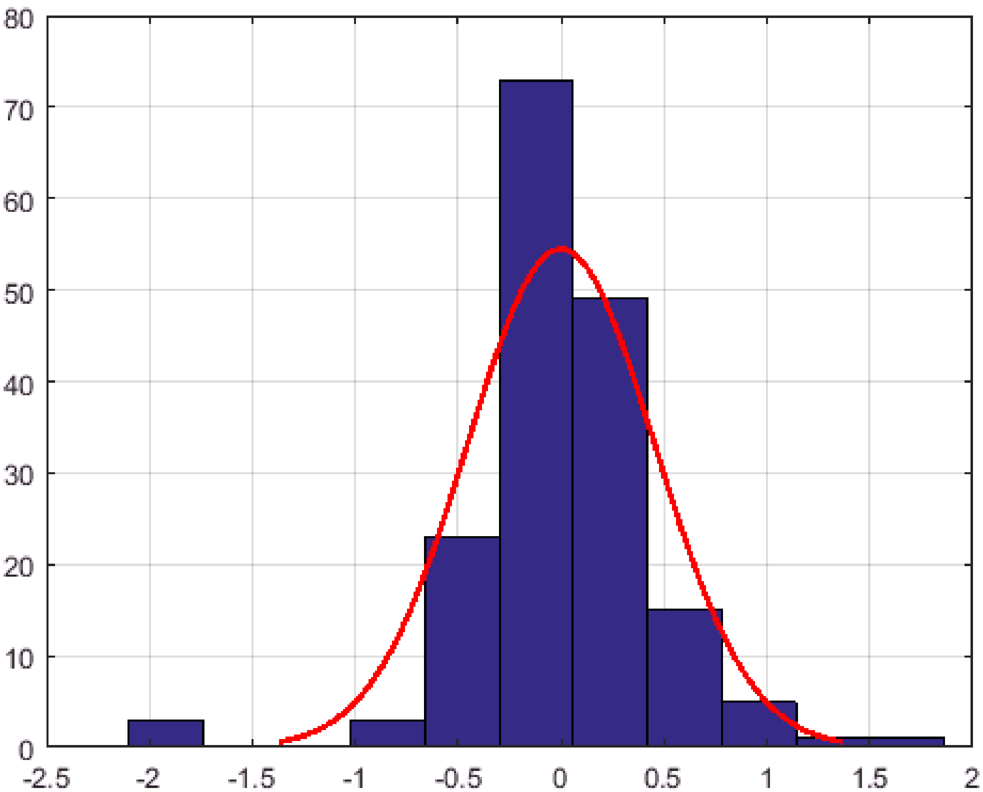


R-error of regression:

```
figure;
errorbar(1:length(RINT(:, 1)), R, RINT(:, 1), RINT(:, 2), 'color', 'red');
```




```
err=R;
histfit(err, 11);
% Turn on the grid for the current figure.
grid on;
```



The error distribution is close to the standard normal distribution, and the absolute value of most errors is less than 0.5

If RINT(i,:) does not contain zero, then the i-th residual is larger than would be expected, at the 5% significance level. It means that the I-th observation is an outlier.

7.Remove outliers and compare original data with new data.

```
Xnew = X;
Znew = Z;

% Remove outliers:
  indicesRows = [];
```

As before:

```
for row = 1:length(RINT(:,1))
    maxvalue = R(row) + RINT(row, 2);
    minvalue = R(row) + RINT(row, 1); % RINT(row, 1)-negative
    if maxvalue < 0 || minvalue > 0
        indicesRows(end+1) = row;     % outlier
    end
end
Xnew(indicesRows) = [];
Znew(indicesRows, :) = [];

% Reconstruct a new standardized matrix:
XZsnew = [Znew(:,1), Xnew, Znew(:,2), Znew(:,3),Znew(:,4)];
```

Comparation:

```
lrmnew = fitlm(Znew, Xnew, 'linear')
```

```
lrmnew = 

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4

Estimated Coefficients:
                   Estimate        SE         tStat       pValue  
                   _________    ________    ________    __________

    (Intercept)     0.013931    0.018862     0.73859       0.46151
    x1             -0.011598     0.02713    -0.42752       0.66972
    x2               0.65416     0.12188      5.3674    3.6255e-07
    x3              0.062315    0.025363      2.4569      0.015354
    x4               0.20952     0.11586      1.8084      0.072885


Number of observations: 133, Error degrees of freedom: 128
Root Mean Squared Error: 0.217
R-squared: 0.942,  Adjusted R-Squared 0.94
F-statistic vs. constant model: 517, p-value = 6.21e-78
```

Coeffitient of determination - is the best!