Case study: Dow Jones and Standard & Poor variations in 2017

Preparing data

Getting Values of SP500 and DJIA indicators for the year 2017 from Fred website

```
SP=getFREDindicator('SP500','2017-01-01','2017-12-31');
DJ=getFREDindicator('DJIA','2017-01-01','2017-12-31');
```

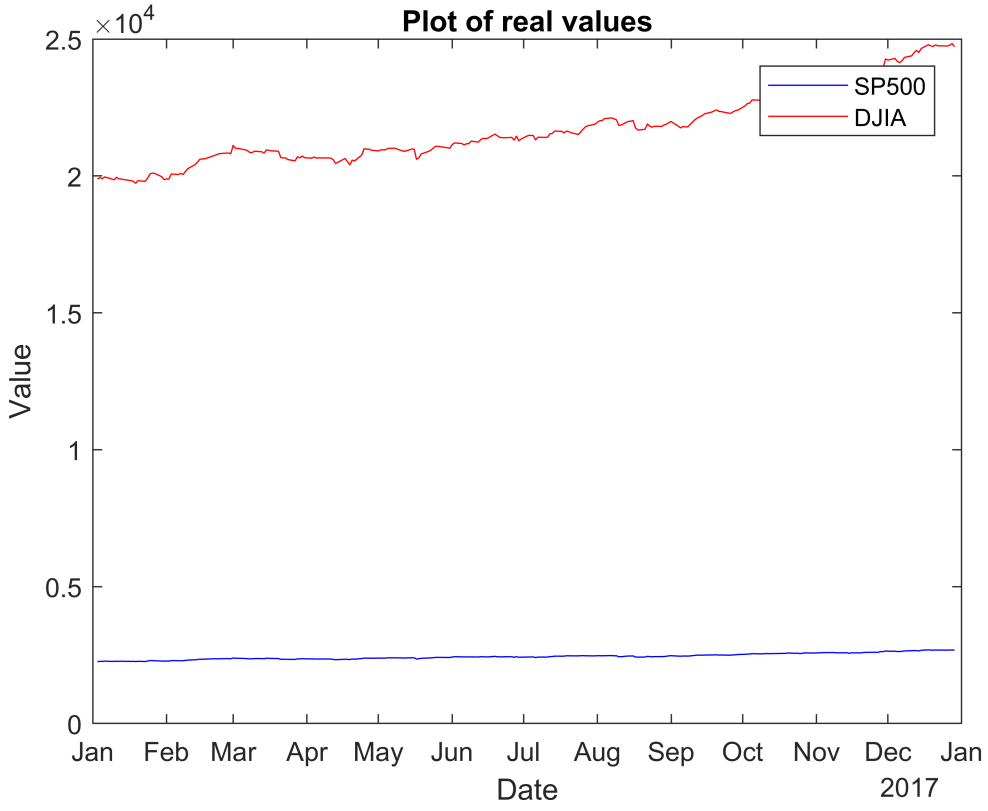
Parsing char arrays

```
dtsp=datetime(SP.Date,'InputFormat','yyyy-MM-dd');
dtdj=datetime(SP.Date,'InputFormat','yyyy-MM-dd');
```

Preliminar study

Plotting indicator values

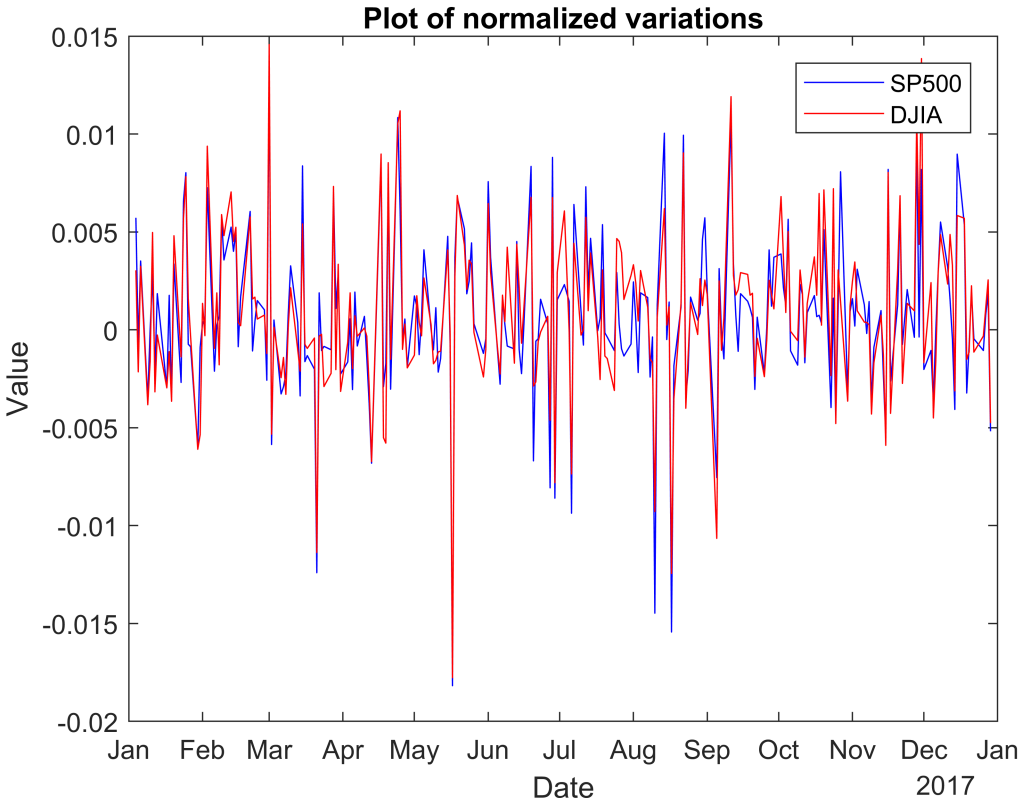
```
plot(dtsp,SP.Value,'b',dtdj,DJ.Value,'r')
title('Plot of real values')
xlabel('Date')
ylabel('Value')
legend('SP500','DJIA')
```



The value of Dow Jones is far larger than the Standard & Poor one, but it would be more convenient to study only the variations, rescaling indexes for a fair comparison; so we need to perform a normalization.

Calculating and plotting normalized variations

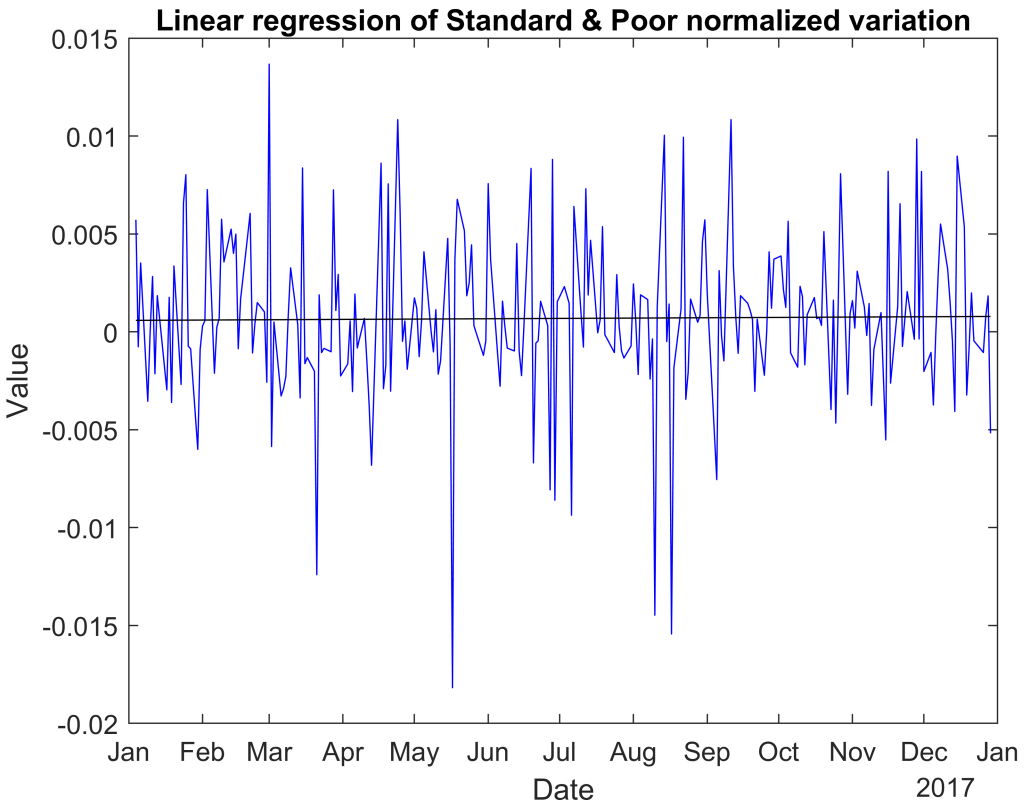
```
wsp=normVar(SP.Value);
wdj=normVar(DJ.Value);
plot(dtsp(2:end),wsp,'b',dtdj(2:end),wdj,'r')
title('Plot of normalized variations')
xlabel('Date')
ylabel('Value')
legend('SP500','DJIA')
```



On a first look it seems that the indexes fluctuations are very similar, but we can perform a linear regression of the variations to better compare them.

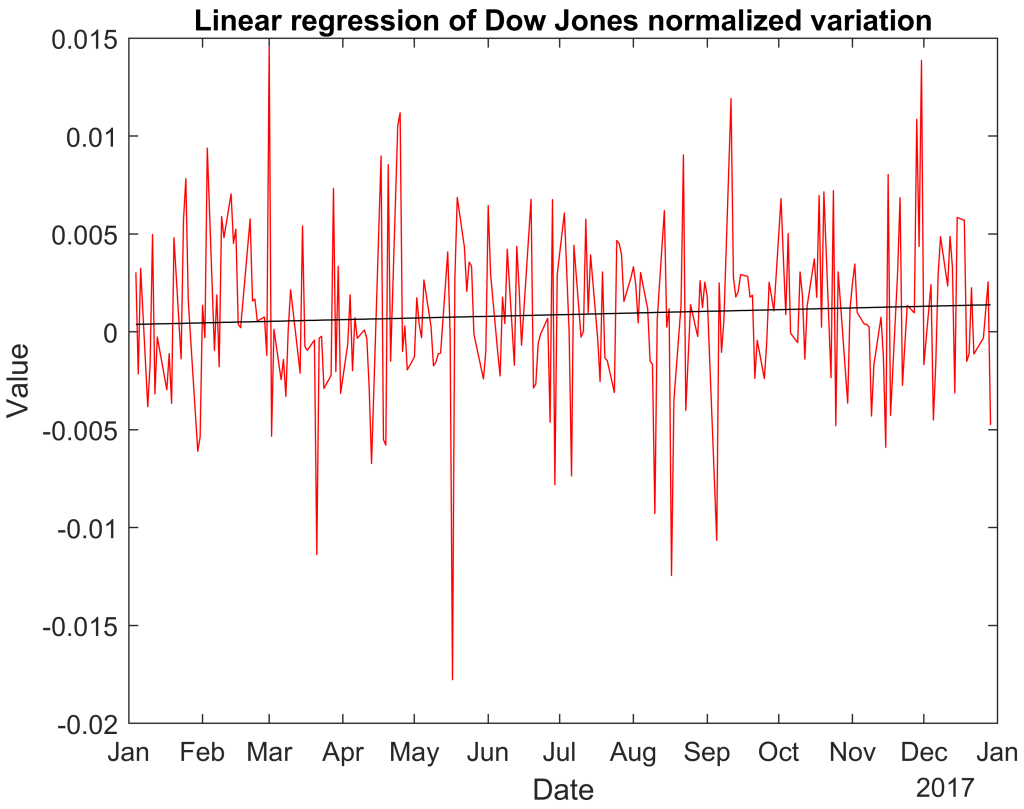
Linear regression of SP500

```
x=1:numel(wsp);
p=polyfit(x,transpose(wsp),1);
f=polyval(p,x);
plot(dtsp(2:end),wsp,'b',dtsp(2:end),f,'k')
title('Linear regression of Standard & Poor normalized variation');
xlabel('Date');
ylabel('Value');
```



Linear regression of DJIA

```
x=1:numel(wdj);
p=polyfit(x,transpose(wdj),1);
f=polyval(p,x);
plot(dtdj(2:end),wdj,'r',dtdj(2:end),f,'k')
title('Linear regression of Dow Jones normalized variation');
xlabel('Date');
ylabel('Value');
```



It seems that Dow Jones is increasing more than Standard & Poor.

Let's check the correlation between SP500 and DJIA during year 2017

```
c=corrcoef(SP.Value,DJ.Value);
Correlation=c(1,2)
```

```
Correlation =
   0.990736704885293
```

It turns out a big correlation between DJ and SP, this explain the similar fluctuations.

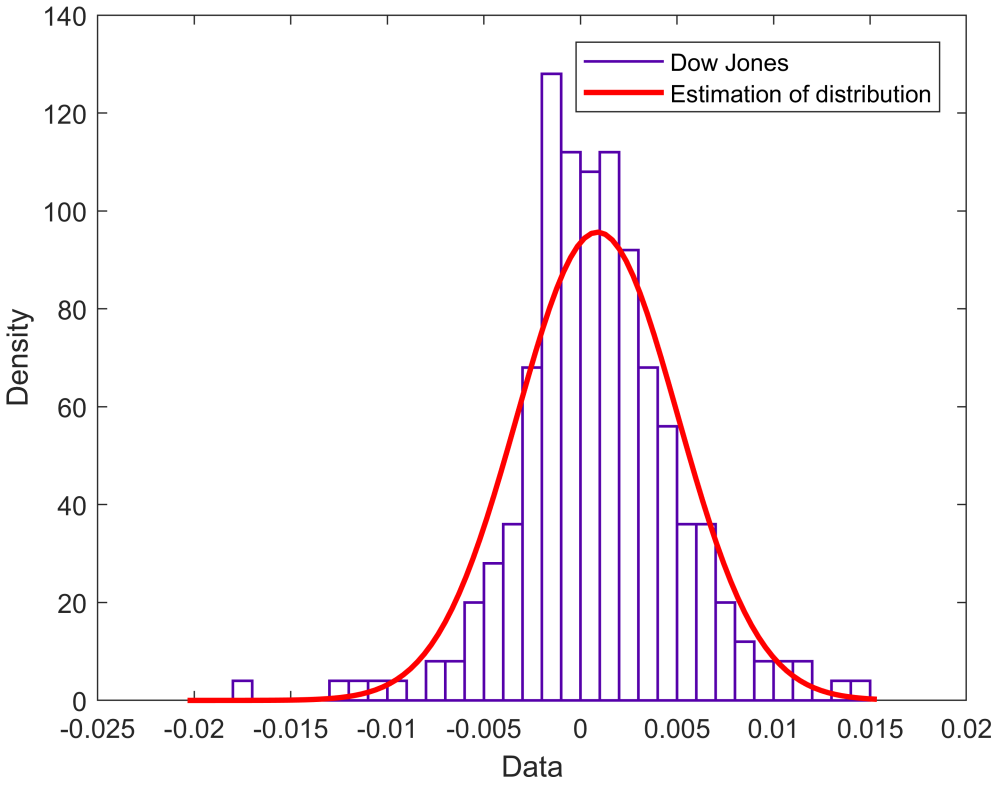
Going deeper

The study of distributions will allow us to better understand the situation.

Dow Jones and Standard & Poor distribution of normalized variations

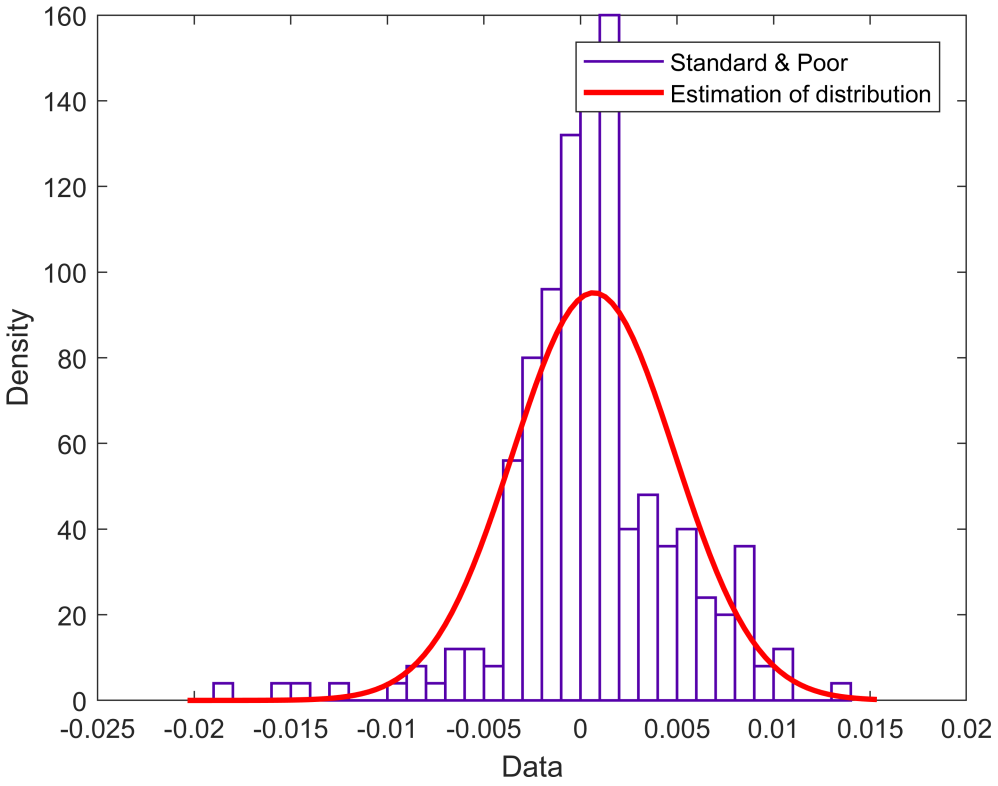
```
Wdjfit=FitWdj(wdj)
```

```
Wdjfit =
  NormalDistribution

  Normal distribution
       mu = 0.000880149   [0.000360721, 0.00139958]
    sigma =  0.00416995   [0.00383366, 0.00457142]
```




```
Wsp=FitWsp(wsp)
```

```
Wsp =
  NormalDistribution

  Normal distribution
       mu = 0.000685083   [0.000163025, 0.00120714]
    sigma =  0.00419107   [0.00385308, 0.00459457]
```



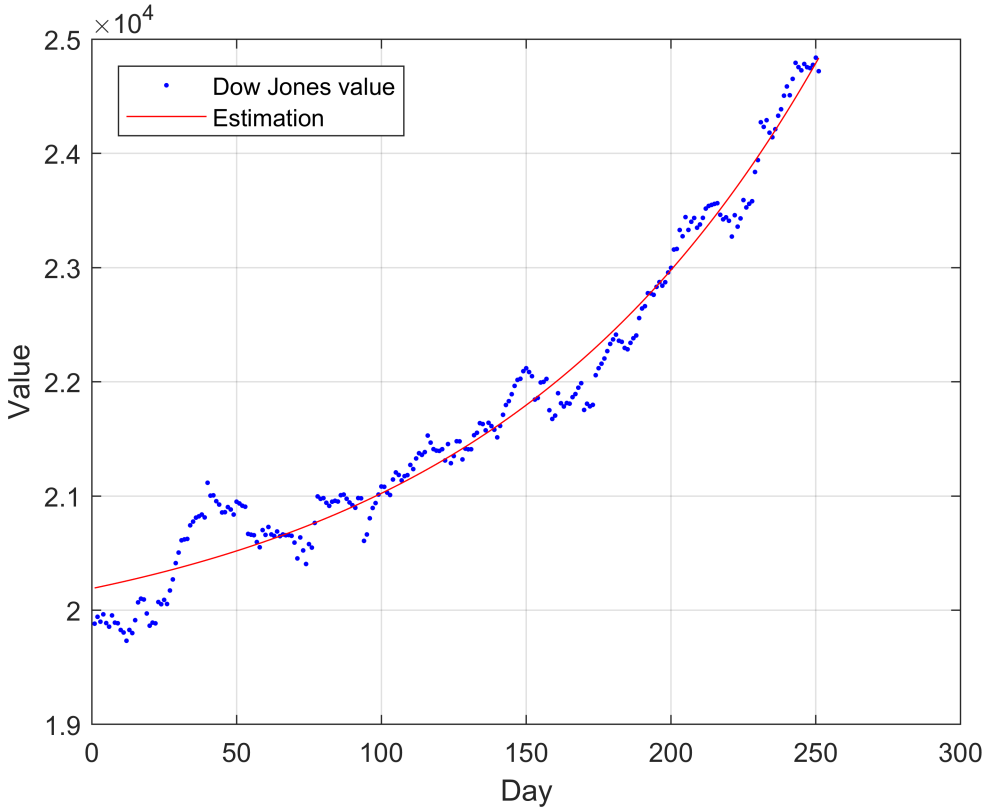
As we can see both indexes follow a normal distribution with positive mean, but the Dow Jones one is greater (0,00088 vs 0,00068), this confirm the preliminary study made with linear regression.

Let's make an estimation of our two indexes:

Dow Jones trend

```
DJval=DJ.Value;
xdj=transpose(1:numel(DJ.Value));
dj=FitDj(xdj,DJval)
```

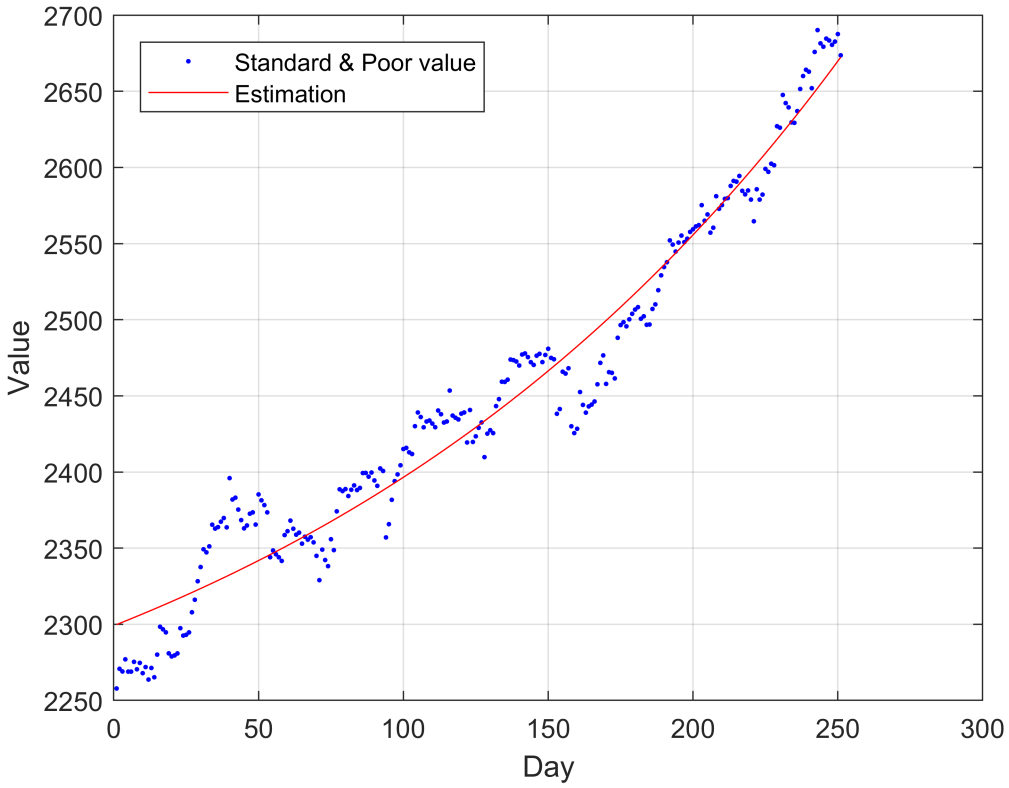
```
dj =
     General model:
     dj(x) = a*exp(-b*x)+c
     Coefficients (with 95% confidence bounds):
       a =       619.8  (490.4, 749.2)
       b =   -0.008524  (-0.009273, -0.007775)
       c =   1.957e+04  (1.938e+04, 1.976e+04)
```



Standard & Poor trend

```
SPval=SP.Value;
xsp=transpose(1:numel(SP.Value));
sp=FitSp(xsp,SPval)
```

```
sp =
     General model:
     sp(x) = a*exp(-b*x)+c
     Coefficients (with 95% confidence bounds):
       a =       153.8  (106.7, 200.8)
       b =   -0.004908  (-0.005778, -0.004037)
       c =        2145  (2092, 2199)
```



Both indexes follow an exponential trend; as we expected the Dow Jones increases faster (ca. exp(0.0085x) vs exp(0.0049x)).

To conclude our analysis we can compute the most important descriptors of the studied indexes:

Dow Jones descriptors

```
[Max, Min, ~, ~, Mean, Std]=getDescriptors(DJ);
format long;
Max, Min, Mean, Std
```

```
Max =
     2.483751000000000e+04
Min =
     1.973240000000000e+04
Mean =
     2.175020374501992e+04
Std =
     1.322007851861826e+03
```

Standard & Poor descriptors

```
[Max, Min, ~, ~, Mean, Std]=getDescriptors(SP);
Max, Min, Mean, Std
```

```
Max =
     2.690160000000000e+03
```

```
Min =
     2.257830000000000e+03
Mean =
     2.449076374501991e+03
Std =
     1.094172354138195e+02
```

```
function T=getFREDindicator(indicator, start_date, end_date)
% getFREDIndicator download an indicator from Fred
%   This function connects to Fred site and downloads informations
%   about the indicator in the specified period
    if ~exist('start_date','var')
        start_date='1776-07-04';
    end
    if ~exist('end_date','var')
        end_date='9999-12-31';
    end
    site='https://api.stlouisfed.org/fred/series/observations?';
    id=['series_id=' indicator];
    api='&api_key=                                ';
    file='&file_type=json';
    observation=['&observation_start=' start_date '&observation_end=' end_date];
    url=[site id api file observation];
    json=webread(url);
    Date=repmat(' ',numel(json.observations),10);
    Value=zeros(numel(json.observations),1);
    for i=1:numel(json.observations)
       Date(i,:)=json.observations(i).date;
       Value(i)=str2double(json.observations(i).value);
    end
    Tnan=table(Date,Value,'VariableNames',{'Date','Value'});
    T=Tnan(~isnan(Tnan.Value),:);
end

function [Max, Min, WMax, WMin, Mean, Std] = getDescriptors(T)
%getDescriptors calculate statistical informations about the input serie
%   calculates max, min, year of max, year of min, mean and standard deviation
%   of the input serie
    [Max, indexmax]=max(T.Value);
    app=strsplit(T(indexmax,1).Date,'-');
    WMax=str2double(app(1));
    [Min, indexmin]=min(T.Value);
    app=strsplit(T(indexmin,1).Date,'-');
    WMin=str2double(app(1));
    Mean=mean(T.Value);
    Std=std(T.Value);
end

function W = normVar(v)
```

```
%NormVar computes normalized variation of value v
%   output vector has one element less than input one
    W=zeros(numel(v)-1,1);
    for i=2:numel(v)
        W(i-1)=(v(i)-v(i-1))/v(i-1);
    end
end

function pd1 = FitWdj(wdj)
%CREATEFIT    Create plot of datasets and fits
%   PD1 = CREATEFIT(WDJ)
%   Creates a plot, similar to the plot in the main distribution fitter
%   window, using the data that you provide as input.  You can
%   apply this function to the same data you used with dfittool
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of datasets:  1
%   Number of fits:  1
%
%   See also FITDIST.

% This function was automatically generated on 19-Feb-2018 17:59:19

% Output fitted probablility distribution: PD1

% Data from dataset "Dow Jones":
%    Y = wdj

% Force all inputs to be column vectors
wdj = wdj(:);

% Prepare figure
clf;
hold on;
LegHandles = []; LegText = {};


% --- Plot data originally in dataset "Dow Jones"
[CdfF,CdfX] = ecdf(wdj,'Function','cdf');  % compute empirical cdf
BinInfo.rule = 1;
[~,BinEdge] = internal.stats.histbins(wdj,[],[],BinInfo,CdfF,CdfX);
[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);
hLine = bar(BinCenter,BinHeight,'hist');
set(hLine,'FaceColor','none','EdgeColor',[0.333333 0 0.666667],...
    'LineStyle','-', 'LineWidth',1);
xlabel('Data');
ylabel('Density')
LegHandles(end+1) = hLine;
LegText{end+1} = 'Dow Jones';

% Create grid where function will be computed
XLim = get(gca,'XLim');
```

```
XLim = XLim + [-1 1] * 0.01 * diff(XLim);
XGrid = linspace(XLim(1),XLim(2),100);


% --- Create fit "Estimation of distribution"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd1 = ProbDistUnivParam('normal',[ 0.0008766423835715, 0.004161976503422])
pd1 = fitdist(wdj, 'normal');
YPlot = pdf(pd1,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 0 0],...
    'LineStyle','-', 'LineWidth',2,...
    'Marker','none', 'MarkerSize',6);
LegHandles(end+1) = hLine;
LegText{end+1} = 'Estimation of distribution';

% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend = legend(LegHandles,LegText,'Orientation', 'vertical', ...
'FontSize', 9, 'Location', 'northeast');
set(hLegend,'Interpreter','none');
end

function pd1 = FitWsp(wsp)
%CREATEFIT    Create plot of datasets and fits
%   PD1 = CREATEFIT(WSP)
%   Creates a plot, similar to the plot in the main distribution fitter
%   window, using the data that you provide as input.  You can
%   apply this function to the same data you used with dfittool
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of datasets:  1
%   Number of fits:  1
%
%   See also FITDIST.

% This function was automatically generated on 19-Feb-2018 18:26:23

% Output fitted probablility distribution: PD1

% Data from dataset "Standard & Poor":
%    Y = wsp

% Force all inputs to be column vectors
wsp = wsp(:);

% Prepare figure
clf;
```

```
hold on;
LegHandles = []; LegText = {};


% --- Plot data originally in dataset "Standard & Poor"
[CdfF,CdfX] = ecdf(wsp,'Function','cdf');  % compute empirical cdf
BinInfo.rule = 1;
[~,BinEdge] = internal.stats.histbins(wsp,[],[],BinInfo,CdfF,CdfX);
[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);
hLine = bar(BinCenter,BinHeight,'hist');
set(hLine,'FaceColor','none','EdgeColor',[0.333333 0 0.666667],...
    'LineStyle','-', 'LineWidth',1);
xlabel('Data');
ylabel('Density')
LegHandles(end+1) = hLine;
LegText{end+1} = 'Standard & Poor';

% Create grid where function will be computed
XLim = get(gca,'XLim');
XLim = XLim + [-1 1] * 0.01 * diff(XLim);
XGrid = linspace(XLim(1),XLim(2),100);


% --- Create fit "Estimation of distribution"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd1 = ProbDistUnivParam('normal',[ 0.0006823538090482, 0.004182905136672])
pd1 = fitdist(wsp, 'normal');
YPlot = pdf(pd1,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 0 0],...
    'LineStyle','-', 'LineWidth',2,...
    'Marker','none', 'MarkerSize',6);
LegHandles(end+1) = hLine;
LegText{end+1} = 'Estimation of distribution';

% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend = legend(LegHandles,LegText,'Orientation', 'vertical',...
'FontSize', 9, 'Location', 'northeast');
set(hLegend,'Interpreter','none');
end

function [fitresult, gof] = FitDj(xdj, DJval)
%CREATEFIT(XDJ,DJVAL)
%  Create a fit.
%
%  Data for 'FitDj' fit:
%      X Input : xdj
%      Y Output: DJval
```

```
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 21-Feb-2018 18:46:55


%% Fit: 'FitDj'.
[xData, yData] = prepareCurveData( xdj, DJval );

% Set up fittype and options.
ft = fittype( 'a*exp(-b*x)+c', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = [0.403912145588115 0.0964545251683886 0.131973292606335];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'FitDj' );
h = plot( fitresult, xData, yData );
legend( h, 'Dow Jones value', 'Estimation', 'Location', 'NorthWest' );
% Label axes
xlabel Day
ylabel Value
grid on
end

function [fitresult, gof] = FitSp(xsp, SPval)
%CREATEFIT(XSP,SPVAL)
%  Create a fit.
%
%  Data for 'FitSp' fit:
%      X Input : xsp
%      Y Output: SPval
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 21-Feb-2018 19:09:30


%% Fit: 'FitSp'.
[xData, yData] = prepareCurveData( xsp, SPval );

% Set up fittype and options.
ft = fittype( 'a*exp(-b*x)+c', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
```

```
opts.Display = 'Off';
opts.StartPoint = [0.0597795429471558 0.234779913372406 0.353158571222071];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'FitSp' );
h = plot( fitresult, xData, yData );
legend( h, 'Standard & Poor value', 'Estimation', 'Location', 'NorthWest' );
% Label axes
xlabel Day
ylabel Value
grid on
end
```