Course 1 - Introduction to stochastic processes

We are going to see some data transformation through an example.

Let us load data on Total share prices for all shares for the US (Not Seasonnally Adjusted) (Source: FRED Federal Reserve of St-Louis)

The series is accessible here https://fred.stlouisfed.org/series/SPASTT01USM661N with a different reference time June 2005.

Initialization

```
clc
close all
clear all

cd('/Users/dvbn/Dropbox/David/University/Teaching/HEC Montreal/Time Series/Code/Matlab/
```

Import the data

```
data = readtable('stock_price.xls');

data(1:10,:)
```

ans = 10×4 table

	year	month	day	sp
1	1957	1	1	0.0374
2	1957	2	1	0.0360
3	1957	3	1	0.0364
4	1957	4	1	0.0374
5	1957	5	1	0.0386
6	1957	6	1	0.0389
7	1957	7	1	0.0396
8	1957	8	1	0.0376
9	1957	9	1	0.0363
10	1957	10	1	0.0341

```
t = datetime(data.year,data.month,data.day);
sp = data.sp;
T = length(t);

table(t(1:10),data.sp(1:10))
```

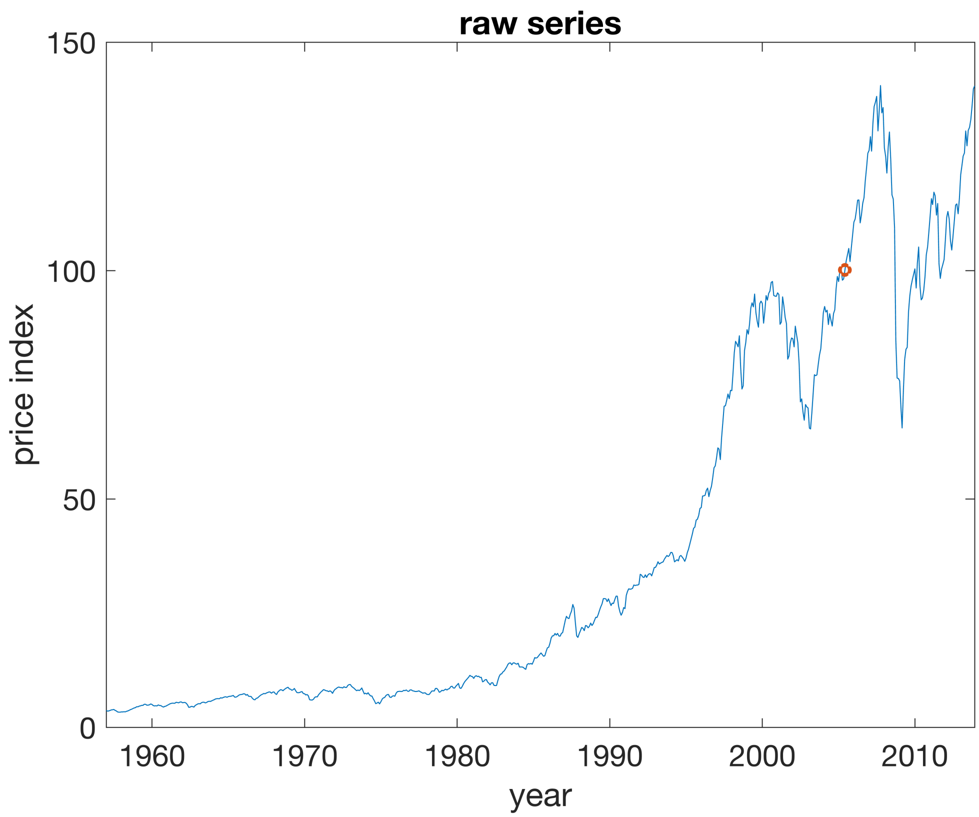
ans = 10×2 table

	Var1	Var2
1	01-Jan-1957	0.0374
2	01-Feb-1957	0.0360
3	01-Mar-1957	0.0364
4	01-Apr-1957	0.0374
5	01-May-1957	0.0386
6	01-Jun-1957	0.0389
7	01-Jul-1957	0.0396
8	01-Aug-1957	0.0376
9	01-Sep-1957	0.0363
10	01-Oct-1957	0.0341

Data vizualization

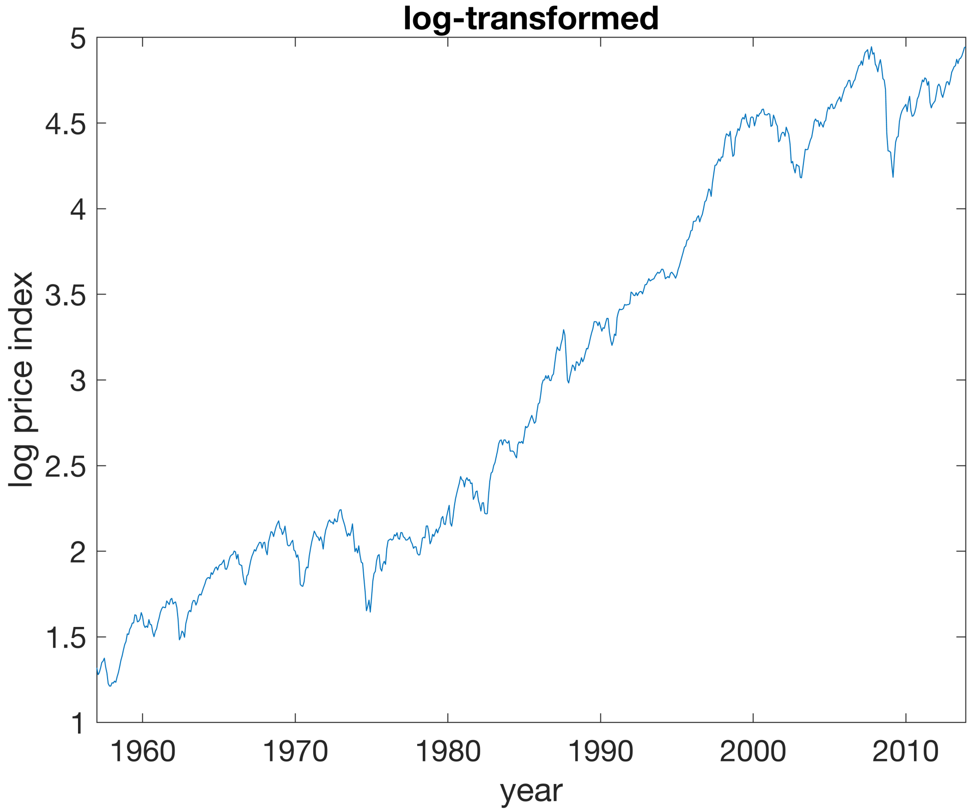
Let us look at the raw data series. The marker indicates the reference time for the price index.

```
figure(1)
plot(t,sp*100)
hold on
plot(t(582),sp(582)*100,'o','MarkerSize',5,'LineWidth',2)
hold off
title('raw series')
xlabel('year')
ylabel('price index')
set(gca,'FontSize',15)
```



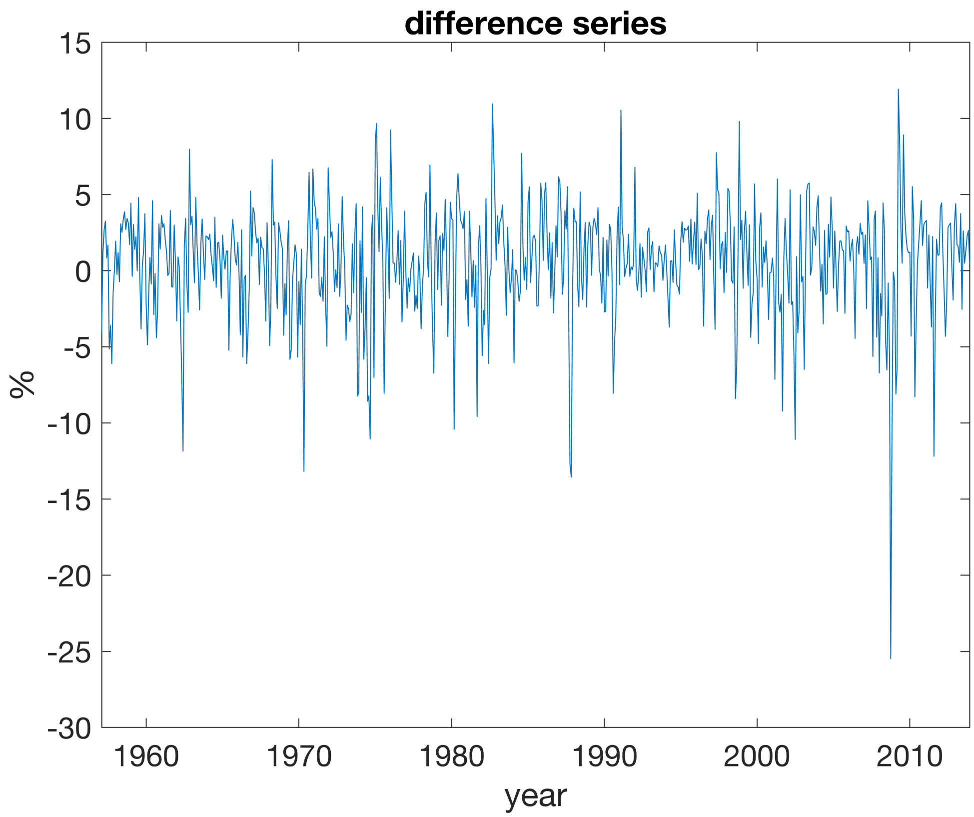
Now, we look at data transformations. First, log transformation of the price index.

```
%% DATA TRANSFORMATIONS

l_sp = log(sp*100); %sp*100 is the price index

plot(t,l_sp)
title('log-transformed')
xlabel('year')
ylabel('log price index')
set(gca,'FontSize',15)
```



It's much more "linear" now. Is it stationary? How about the first-difference series?

```
d_sp = diff(l_sp)*100; %diff gives first-difference, hence growth rate, x100 gives perc

figure(1)
plot(t(2:end),d_sp)
title('difference series')
xlabel('year')
ylabel('%')
set(gca,'FontSize',15)
```



The trend has disappeared, but is it stationary?

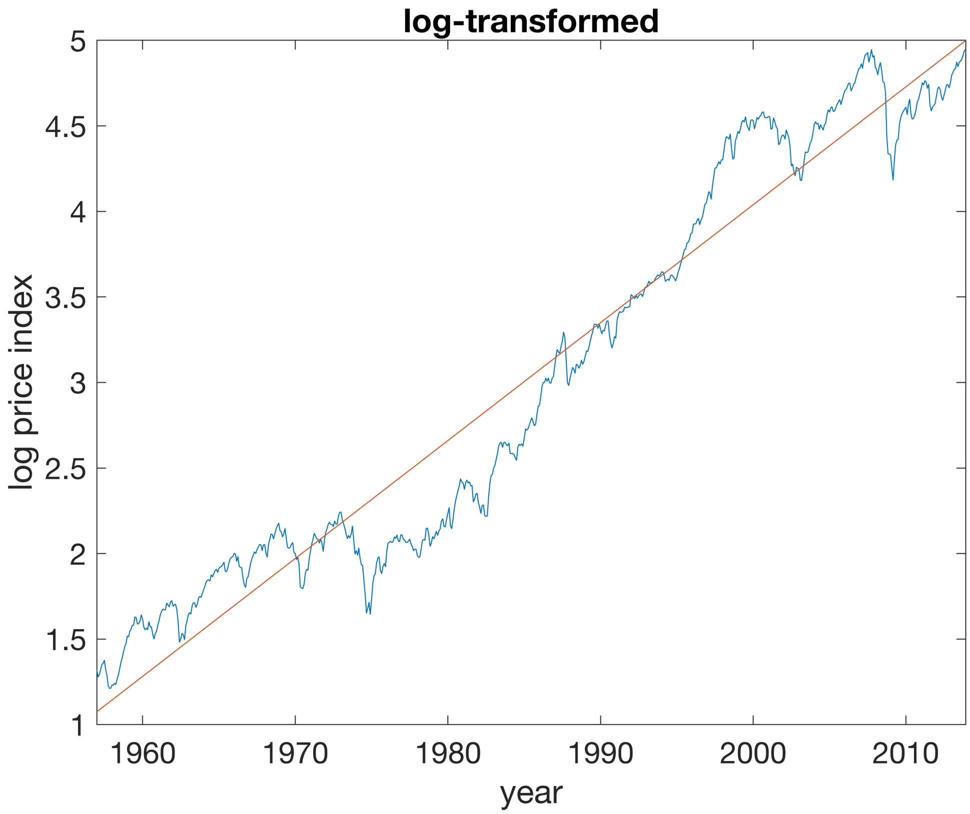
Let us look at the cyclical component of the de-trended log-linearized series.

```
%% LINEAR REGRESSION

trend = 1 : T; %generate a linear time trend
X = [ones([length(l_sp),1]),trend(:)]; %covariates
Y = l_sp; %dependent var
B = X\Y; %MATLAB shortcut to solve A*X = B: the same as OLS solution (X'X)^-1 X'Y

B
```

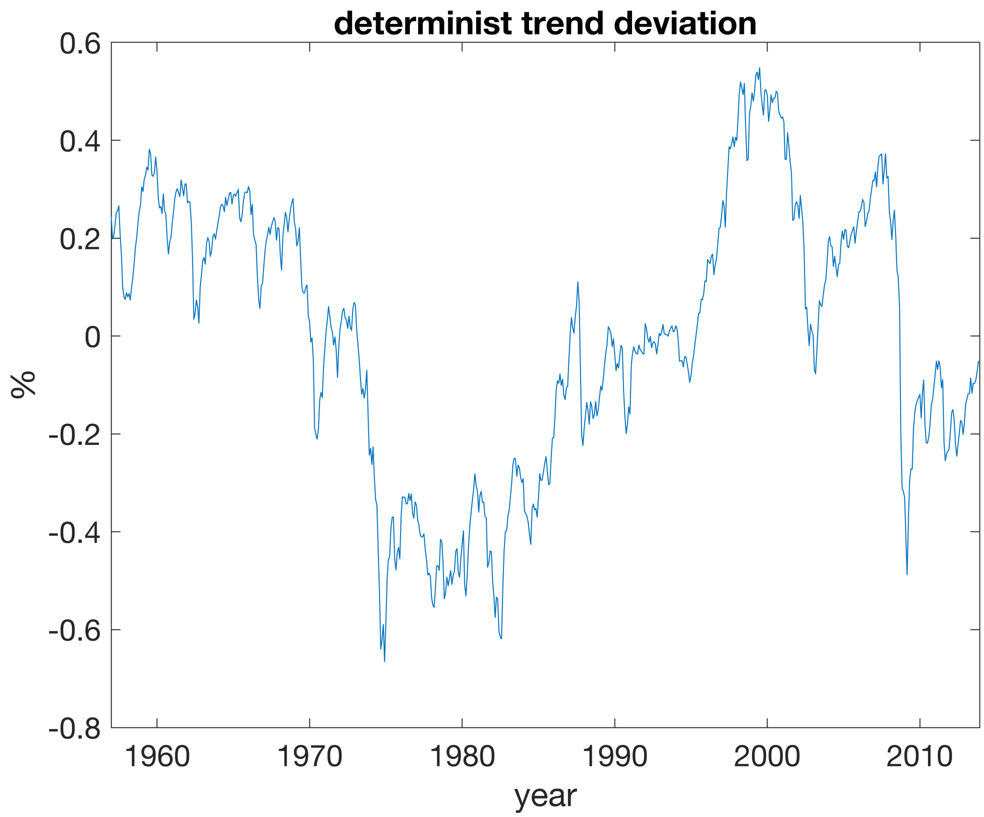
```
B = 2×1
    1.0694
    0.0057
```

```
%look at trend
plot(t,l_sp)
title('log-transformed')
xlabel('year')
ylabel('log price index')
set(gca,'FontSize',15)
hold on
```

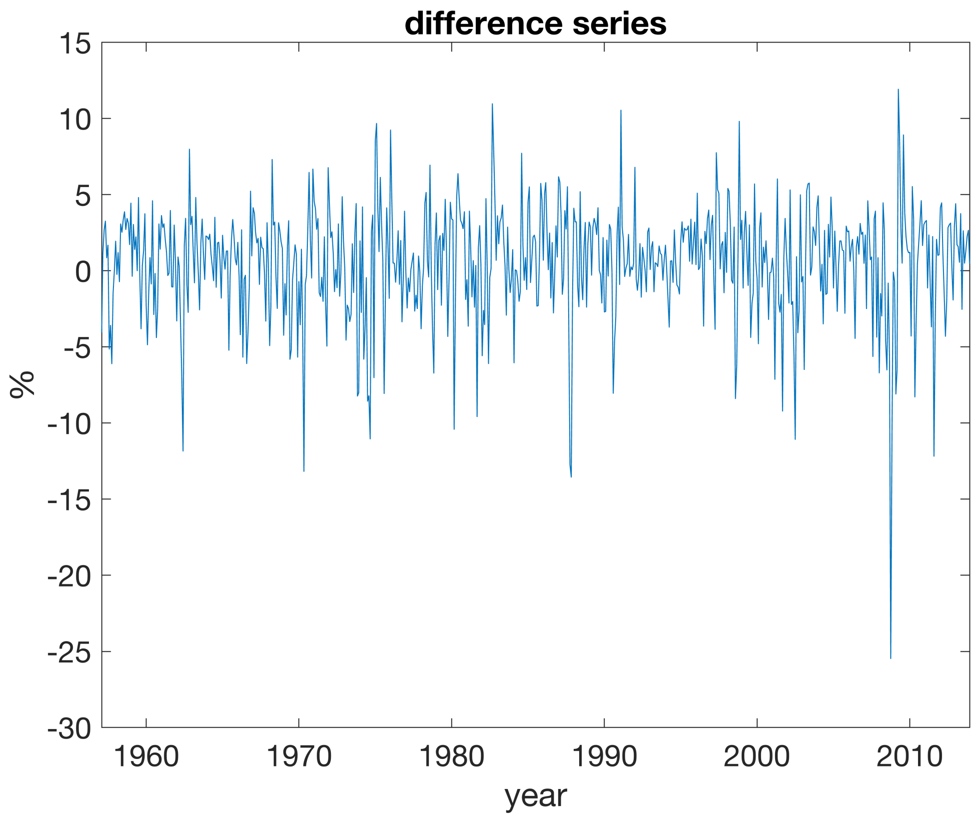
```
plot(t,B(1) + B(2)*trend)
hold off
```




```
%detrend series
c_sp = Y - X*B;

plot(t,c_sp)
title('determinist trend deviation')
xlabel('year')
ylabel('%')
set(gca,'FontSize',15)
```




```
plot(t(2:end),d_sp)
title('difference series')
xlabel('year')
ylabel('%')
set(gca,'FontSize',15)
```




```
% Notes: 1) d_sp corresponds to the growth rate of the stock price index, or stock retu
%        2) c_sp is the percentage deviation of the stock price index relative to the
%        3) X\Y computes the OLS: hat beta = (X'X)^{-1} (X'Y)
```

Autocorrelations

We can use the matlab function to calculate autocorrelation coefficients for the two series.

```
%% AUTOCORRELATIONS

[acf1] = autocorr(d_sp,18)
```

```
acf1 = 19×1
    1.0000
    0.2787
   -0.0009
    0.0265
    0.0661
    0.0657
   -0.0807
   -0.0807
   -0.0016
   -0.0253
      ⋮
```

```
[acf2] = autocorr(c_sp,18)
```

```
acf2 = 19×1
    1.0000
    0.9908
    0.9770
    0.9632
    0.9488
    0.9331
    0.9163
    0.9008
    0.8870
    0.8734
      ⋮
```

```
%% LJUNG-BOX TEST (ECONOMETRICS TOOLBOX)

[h1,pValue1,stat1] = lbqtest(d_sp,'lags',1:18)
```

```
h1 = 1×18 logical array
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
pValue1 = 1×18
10^-7 ×
    0.0000    0.0000    0.0001    0.0001    0.0001    0.0001    0.0000    0.0001 ···
stat1 = 1×18
   53.2789   53.2794   53.7613   56.7721   59.7474   64.2440   68.7498   68.7516 ···
```

```
[h2,pValue2,stat2] = lbqtest(c_sp,'lags',1:18)
```

```
h2 = 1×18 logical array
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
pValue2 = 1×18
     0     0     0     0     0     0     0     0     0     0     0     0     0 ···
stat2 = 1×18
10^3 ×
    0.6744    1.3312    1.9704    2.5916    3.1933    3.7745    4.3369    4.8830 ···
```

```
% h = 1 indicates rejection of the no residual autocorrelation null hypothesis in favo
% h = 0 indicates failure to reject the no residual autocorrelation null hypothesis.

% Notes: 1) The Ljung-Box statistics indicate that d_sp and c_sp are not white noise.

%        2) The autocorrelations suggest that d_sp is stationary (short
%        memory), but that c_sp is not stationary (long memory)

%           For this reason, we use d_sp
```

The Ljung-Box test can also be performed with our implementated function (see Lbox.m)

```
%% LJUNG-BOX TEST (IMPLEMENTED FUNCTION)

T = length(t);
n = 18;
Q1 = zeros(1,18);
```

```
Chi1 = zeros(1,18);
Q2 = zeros(1,18);
Chi2 = zeros(1,18);

for i = 1:18
    [Q1(i) Chi1(i)]  = Lbox(acf1, T-1, i);
    [Q2(i) Chi2(i)]  = Lbox(acf2, T, i);
end

% Q is the stat, Chi is the 5% threshold

table(Q1',Chi2',Q2',Chi1')
```

ans = 18×4 table

	Var1	Var2	Var3	Var4
1	53.2789	3.8415	674.4188	3.8415
2	53.3577	5.9915	1.3322e+03	5.9915
3	53.9180	7.8147	1.9734e+03	7.8147
4	57.0082	9.4877	2.5975e+03	9.4877
5	60.0676	11.0705	3.2030e+03	11.0705
6	64.6529	12.5916	3.7888e+03	12.5916
7	69.2543	14.0671	4.3569e+03	14.0671
8	69.3588	15.5073	4.9095e+03	15.5073
9	69.9053	16.9190	5.4470e+03	16.9190
10	71.8932	18.3070	5.9706e+03	18.3070
11	72.0553	19.6751	6.4816e+03	19.6751
12	72.2699	21.0261	6.9799e+03	21.0261
13	72.6085	22.3620	7.4659e+03	22.3620
14	73.7509	23.6848	7.9399e+03	23.6848
15	74.3448	24.9958	8.4030e+03	24.9958
16	74.6250	26.2962	8.8558e+03	26.2962
17	75.2524	27.5871	9.2980e+03	27.5871
18	75.3694	28.8693	9.7299e+03	28.8693

```
%% GRAPH OF THE DATA

figure(2)
subplot(2,2,1)
plot(t,c_sp)
title('Hypothesis: determinist trend')
set(gca,'FontSize',15)
subplot(2,2,2)
autocorr(c_sp,18)
```

```
title('autocorrelations: c_sp')
set(gca,'FontSize',15)
dt = t(1:T-1);

subplot(2,2,3)
plot(dt,d_sp)
title('Hypothesis: stochastic trend')
set(gca,'FontSize',15)
subplot(2,2,4)
autocorr(d_sp,18)
title('autocorrelations: d_sp')
set(gca,'FontSize',15)
```