Advanced Macroeconomics WS18/19 - First Assigment

Table of Contents

This script is part of the first assignment of Advanced Macroeconomics course offered in the Winter Term of 2018/2019. The authors of this script are Kaan Can Özipek (4920088), Marcelo Avila (4679876), Nicholas Borsotto (4907195) and Silvia Meletti (5275867). This script requires MATLAB 2018b and the Datafeed Toolbox. The HP-Filter.m and customPlot.m functions should also be found in the working directory or added paths.

00 Settings - Clear Environment and change directory

```
clear; close all; clc;

% Changing working directory to current script dir
tmp = matlab.desktop.editor.getActive;
cd(fileparts(tmp.Filename));
% add /functions and /data path
addpath('./functions/');
addpath('./data/');
```

Q. 01 - Fetch Data

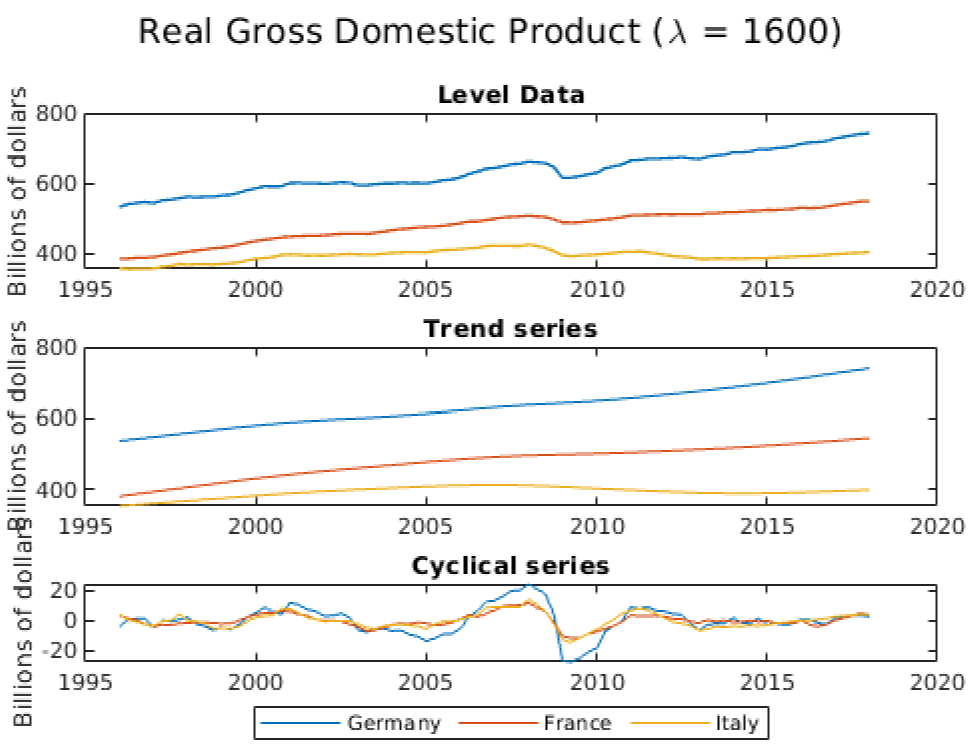
```
startDate = "01/01/1996";
endDate = "01/01/2018";

series = ["FRAPFCEQDSNAQ" "DEUPFCEQDSNAQ" "ITAPFCEQDSNAQ" ...
"CLVMNACSCAB1GQFR", "CLVMNACSCAB1GQDE", "CLVMNACSCAB1GQIT"];
```

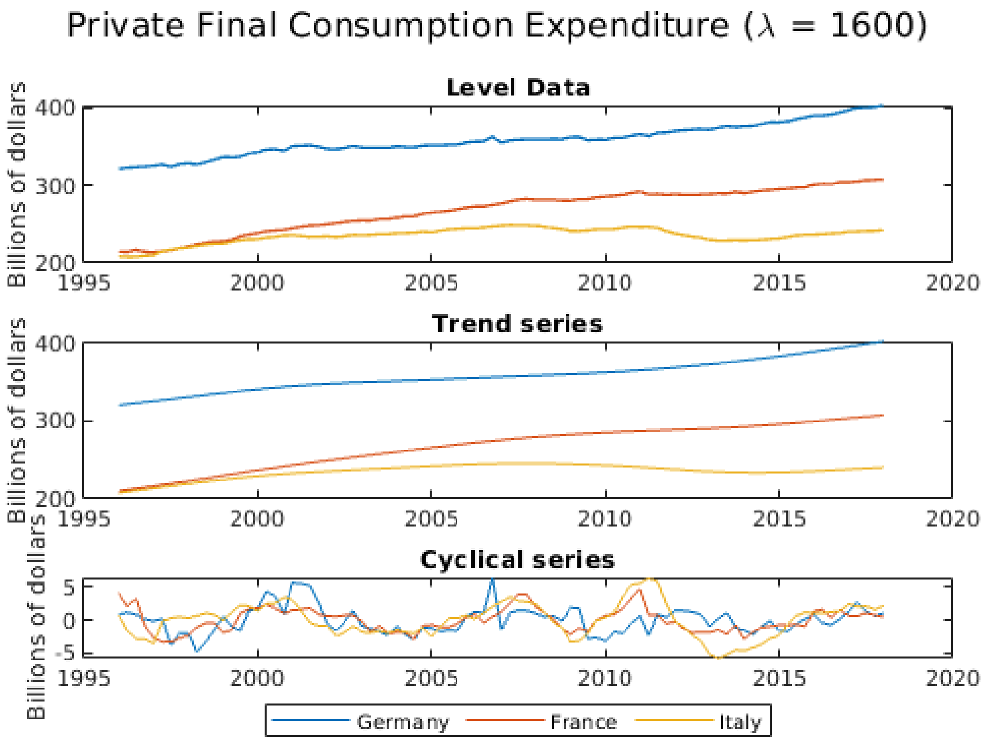
Using `fred` and `fetch` function to retrieve data from FRED this function requires the Datafeed Toolbox. If this toolbox is not available try to load the data that should be stored in the working directory with the following command:

```
load('dataStruct.mat')

% using fred.stloisfed connection
url = "https://fred.stlouisfed.org/";

% retrieve data struct of the 6 time series
dataStruct = fetch(fred(url), series, startDate, endDate);

% consumption data
con_fr = dataStruct(1).Data(:,2) / 10^9;
con_de = dataStruct(2).Data(:,2) / 10^9;
con_it = dataStruct(3).Data(:,2) / 10^9;

% gdp data
gdp_fr = dataStruct(4).Data(:,2) / 10^3;
gdp_de = dataStruct(5).Data(:,2) / 10^3;
gdp_it = dataStruct(6).Data(:,2) / 10^3;
```

Q. 02 and 03 - Apply HP Filter

```
lambda = 1600;
customPlot('gdp', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)

customPlot('con', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)
```





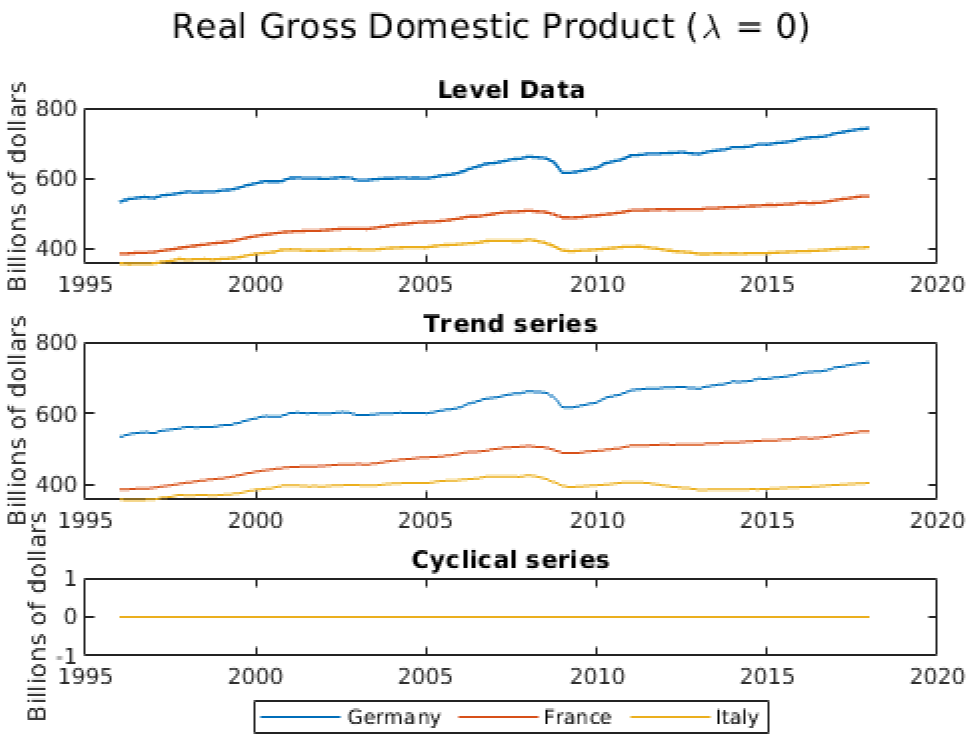
Q. 04 - Repeat for lambda -> 0

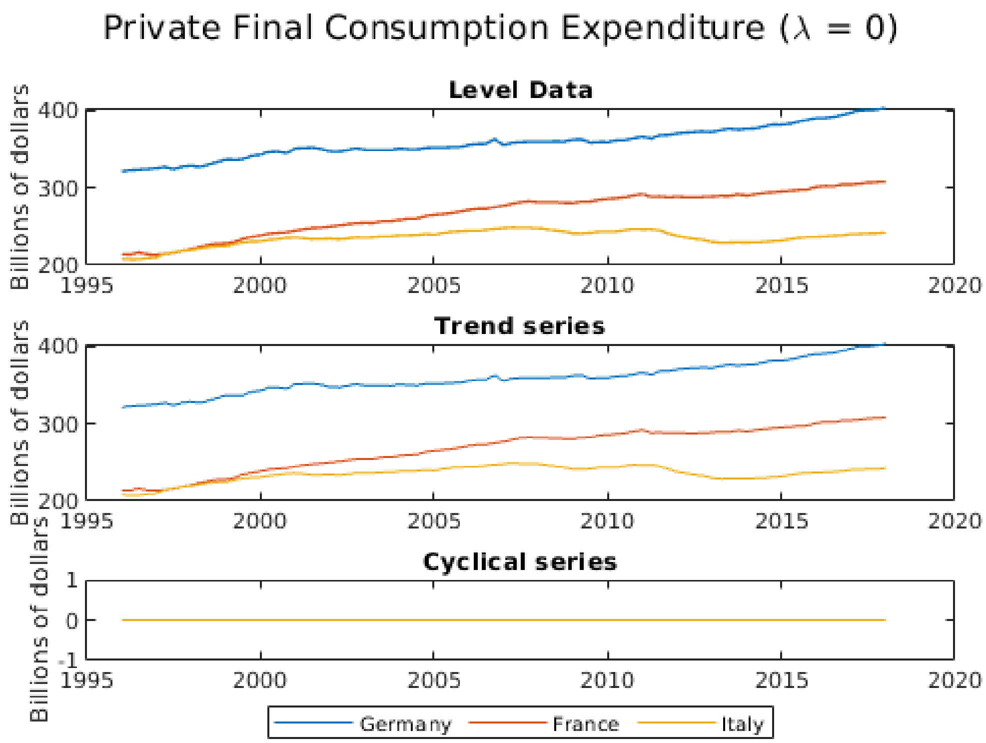
ANSWER: HP-Filter minimization problem:

$$\min_{\tau}\left(\sum_{t=1}^{T}(y_t-\tau_t)^2+\lambda\sum_{t=2}^{T-1}[(\tau_{t+1}-\tau_t)-(\tau_t-\tau_{t-1})]^2\right)$$

In the case lambda tends to 0, we only minimize the first term of the HP-filter minimization problem. So the trend component is exactly the same as the observed data, and the cyclical component is, therefore, zero.

```
% lambda -> 0
lambda = 0;
customPlot('gdp', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)
customPlot('con', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)
```

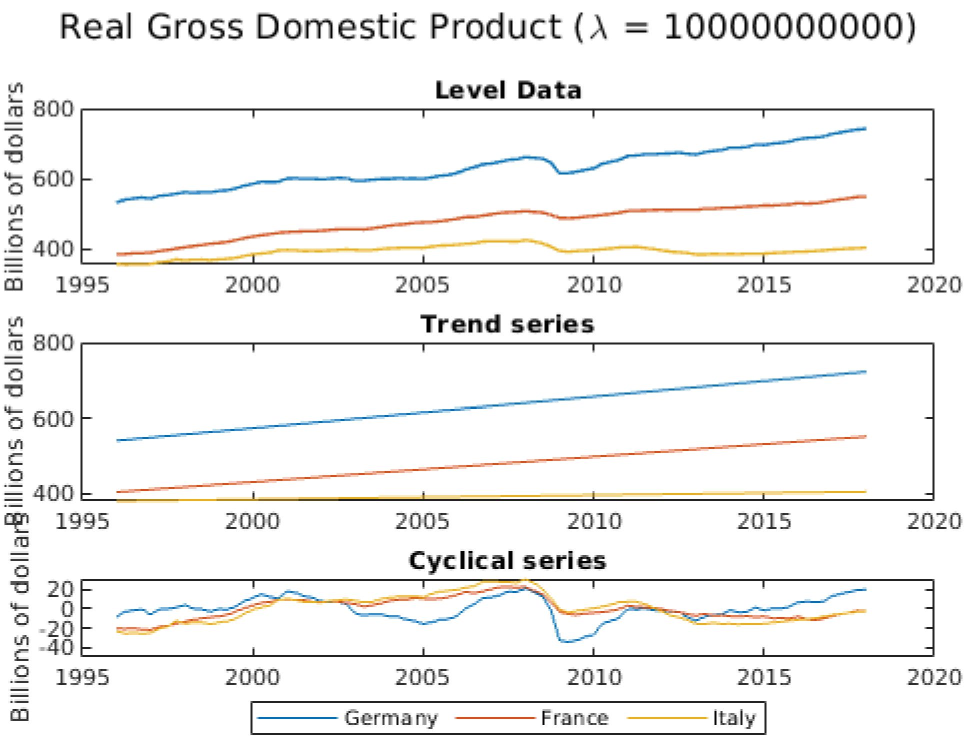

Real Gross Domestic Product (λ = 0)
Level Data
Trend series
Cyclical series
Billions of dollars
Germany
France
Italy

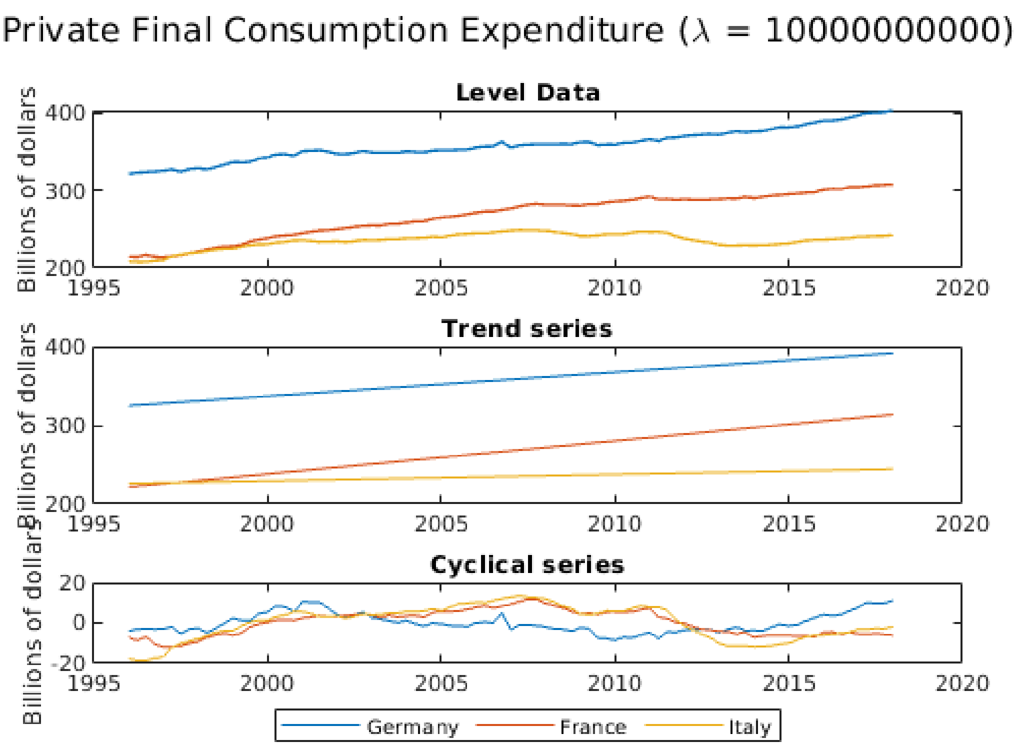

Private Final Consumption Expenditure (λ = 0)
Level Data
Trend series
Cyclical series
Billions of dollars
Germany
France
Italy

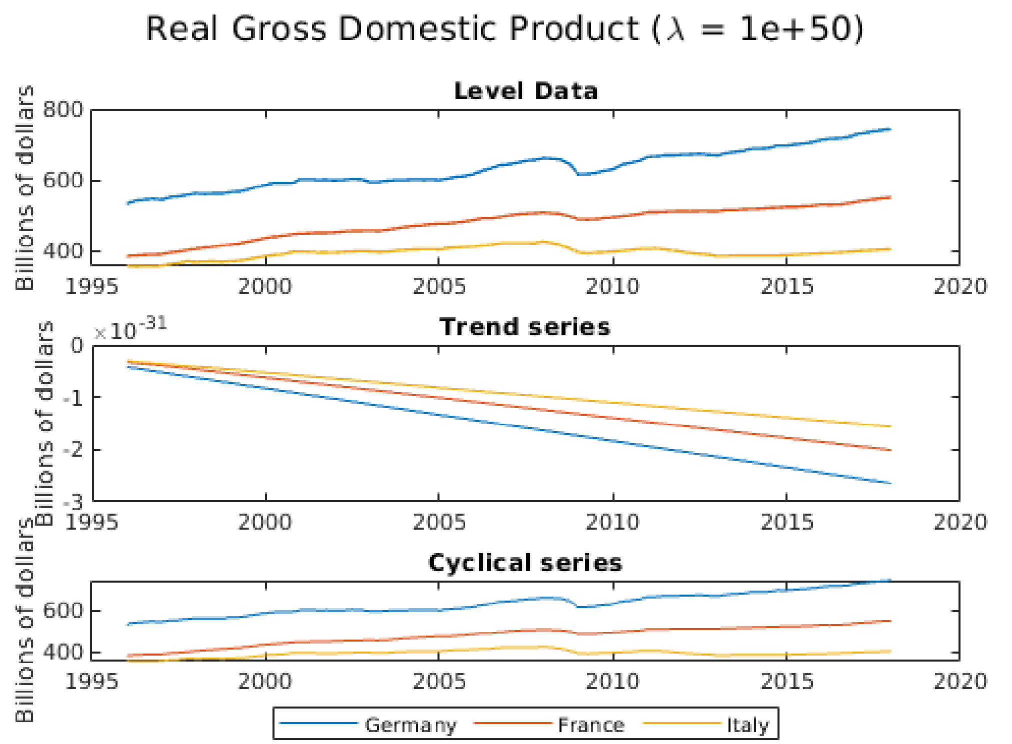
Q. 04 - Repeat for lambda -> inf

ANSWER: In the case of $\lambda \to \infty$, the second component in the HP-Filter gets the greatest weight and, therefore, we have a linear trend where the second term of the minization problem is zero, because the change in the trend is constant. The cyclical componente shows the difference between the observed data and the linear trend. Due to (near-) singularity problems, the matrix A from the HP-Filter function is not invertible for very big lambdas. The results are, therefore, nonsensical and for Infitiy not computable.

```
lambda = 10^10;
customPlot('gdp', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)
customPlot('con', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)

lambda = 10^50;
customPlot('gdp', gdp_fr, gdp_de, gdp_it, con_fr, con_de, con_it,
 lambda)
```




Private Final Consumption Expenditure (λ = 10000000000)
Level Data
Trend series
Cyclical series
Billions of dollars
400
300
200
20
0
-20
1995
2000
2005
2010
2015
2020
Germany
France
Italy


Real Gross Domestic Product (λ = 1e+50)
Level Data
Trend series
Cyclical series
Billions of dollars
800
600
400
×10^-31
0
-1
-2
-3
1995
2000
2005
2010
2015
2020
Germany
France
Italy

Q. 05 - Apply Log

```
lgdp_de = log(gdp_de);
lgdp_fr = log(gdp_fr);
lgdp_it = log(gdp_it);

lcon_de = log(con_de);
lcon_fr = log(con_fr);
lcon_it = log(con_it);

%%%% hp filter on log data
lambda = 1600;

[lgdp_T_fr, lgdp_C_fr] = hp_filter(lgdp_fr, lambda);
[lcon_T_fr, lcon_C_fr] = hp_filter(lcon_fr, lambda);

%%%% germany
[lgdp_T_de, lgdp_C_de] = hp_filter(lgdp_de, lambda);
[lcon_T_de, lcon_C_de] = hp_filter(lcon_de, lambda);

%%%% italy
[lgdp_T_it, lgdp_C_it] = hp_filter(lgdp_it, lambda);
[lcon_T_it, lcon_C_it] = hp_filter(lcon_it, lambda);
```

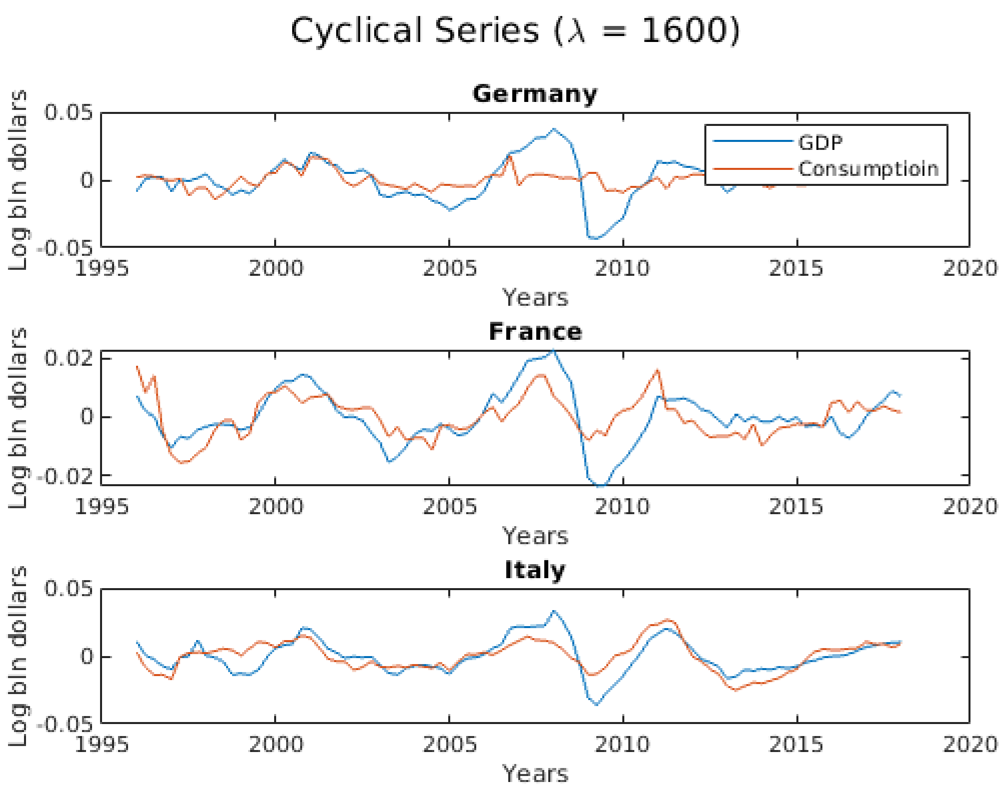
Q. 05 - Repeat for lambdas inf

ANSWER: The cyclical component is obtained through the difference between the actual data and the trend series generated by the HP-Filter. Since we applied logarithm to the original data, the cyclical fluctions are obtained through a log-difference, and therefore, can be approximately interpreted as a percentage difference from the trend. The application of the logarithm is also indicated when dealing with exponentially growing series in order to reduce possible heteroskedaticity and achieve a more well-behaved series.

From the cyclical component above we can notice that GDP has a peak right before and declines sharply during the financial crises of 2007~08. Especially regarding Italy, one can notice the effects of the Euro sovereign debt crises around 2012. The cyclical series of consumption from Italy and France seem to closely follow the one of GDP. On the other hand, consumption in Germany is less affected by the strong changes in GDP. This could suggest that households in Germany were less affecterd by both crises than in the other two countries.

```
dates=1996.0:0.25:2018.0;

figure;
titleOfGraph = 'Cyclical Series (\lambda = ' + string(lambda) + ')';
sgtitle(titleOfGraph)
% working on 3 by 1 plots, plot 01
subplot(3,1,1);      % GERMANY
plot(dates, lgdp_C_de); hold on;
plot(dates, lcon_C_de); hold on;
xlabel('Years');
ylabel('Log bln dollars');
title('Germany');
legend('GDP', 'Consumptioin');
```

```
% working on 3 by 1 plots, plot 02
subplot(3,1,2);
plot(dates, lgdp_C_fr); hold on;
plot(dates, lcon_C_fr); hold on;
title('France');
ylabel('Log bln dollars');
xlabel('Years');

% working on 3 by 1 plots, plot 03
subplot(3,1,3)
plot(dates, lgdp_C_it); hold on;
plot(dates, lcon_C_it); hold on;
ylabel('Log bln dollars');
xlabel('Years');
title('Italy');
```



Q. 06 - Std Deviation

ANSWER: The results show that consumption is less volatile than GDP in all three countries. This could be due to the fact that GDP absorbs also the volatility of its other components, that is, Government expenditure, Investments and Net Exports.

The very low standard deviation in consumption for Germany corroborates the hypothesis advanced in the previous answer.

```
std(lgdp_C_de); %      0.0150
```

```
std(lgdp_C_fr); %     0.0092
std(lgdp_C_it); %     0.0128

std(lcon_C_de); %     0.0060
std(lcon_C_fr); %     0.0070
std(lcon_C_it); %     0.0110
```

Q. 07 - Cyclical Series

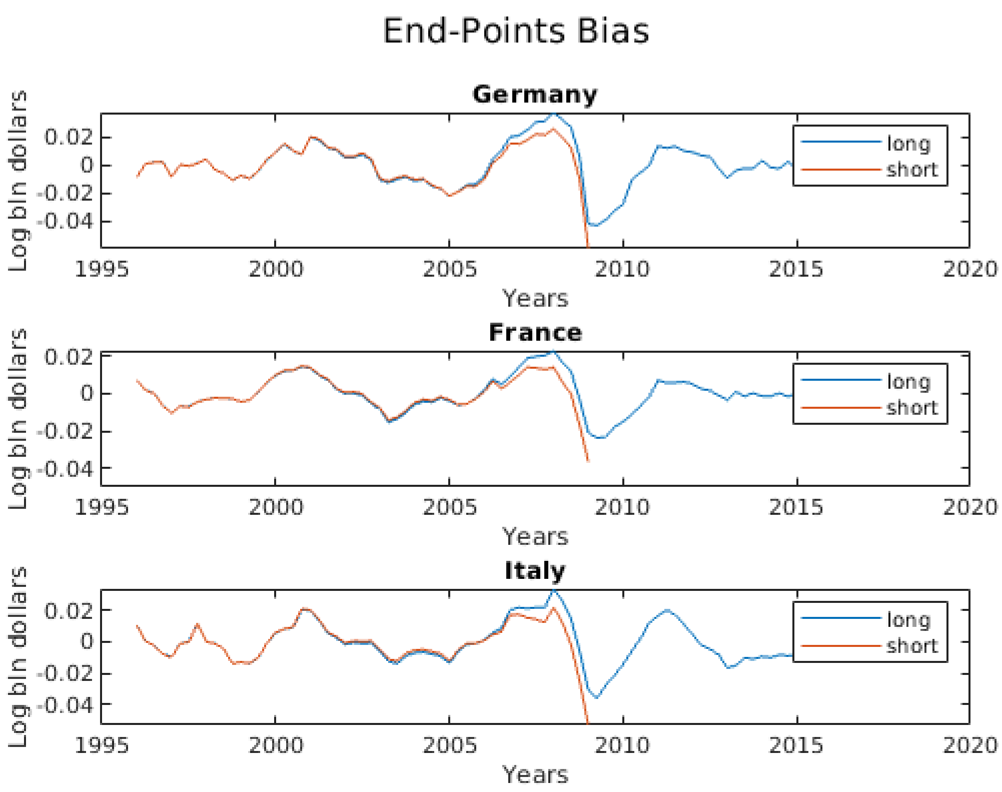
ANSWER: How one can take from the HP-Filter minimization problem, the first and the final trend points, that is, the end-points, are not smoothed by the change in growth trend. That is, the second term is computed only from t=2 to T-1, whereas the first term is computed for the whole time series. This results in an exaggerated estimation for the trend at the extremes, that is exaggerated towards the actual data points.

In our case, one can see the bias only at the right end-point, because here we use the same starting date for the short and long series.

Using the HP-Filter can be problematic when the (right) endpoint is the focus of the analysis and can lead to wrong inferences or predictions. When dealing with real-time date, one is always susceptible to this bias. Moreover, it is problematic because the endpoint bias reverberates not only on the very last points of the trend, but on a longer span, with diminishing impact towards the middle of the trend series. Furthermore, the bias tends to be larger the further the data point lies from the longer term trend.

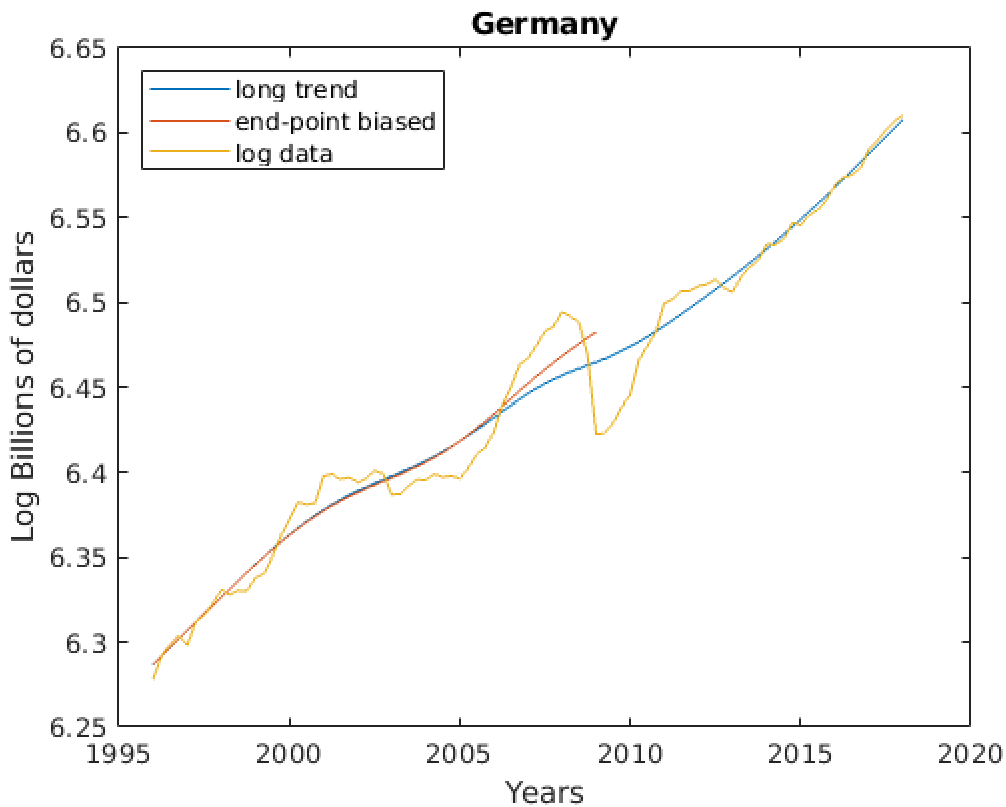
```
% Q.7 - a) slice timeseries

lambda = 1600;
startDate = 1996;
endDate = 2009;

lgdp_it_cut = timeseries(lgdp_it,dates);
lgdp_it_cut = getsampleusingtime(lgdp_it_cut, startDate, endDate);
lgdp_de_cut = timeseries(lgdp_de,dates);
lgdp_de_cut = getsampleusingtime(lgdp_de_cut, startDate, endDate);
lgdp_fr_cut = timeseries(lgdp_fr,dates);
lgdp_fr_cut = getsampleusingtime(lgdp_fr_cut, startDate, endDate);

[lgdp_it_cut_T, lgdp_it_cut_C] =  hp_filter(lgdp_it_cut.Data, lambda);
[lgdp_de_cut_T, lgdp_de_cut_C] =  hp_filter(lgdp_de_cut.Data, lambda);
[lgdp_fr_cut_T, lgdp_fr_cut_C] =  hp_filter(lgdp_fr_cut.Data, lambda);

%%%% Q.7 - a) plot cyclical series

datesCut = startDate:0.25:endDate;

figure;
sgtitle('End-Points Bias')
% working on 3 by 1 plots, plot 01
subplot(3,1,1);
plot(dates, lgdp_C_de)      ; hold on;
plot(datesCut, lgdp_de_cut_C)  ; hold on;

xlabel('Years');
ylabel('Log bln dollars');
title('Germany');
```

```
legend('long', 'short');

% working on 3 by 1 plots, plot 01
subplot(3,1,2);
plot(dates, lgdp_C_fr)        ; hold on;
plot(datesCut, lgdp_fr_cut_C)  ; hold on;

xlabel('Years');
ylabel('Log bln dollars');
title('France');
legend('long', 'short');


subplot(3,1,3);
plot(dates, lgdp_C_it)        ; hold on;
plot(datesCut, lgdp_it_cut_C)  ; hold on;

xlabel('Years');
ylabel('Log bln dollars');
title('Italy');
legend('long', 'short');
```



The direct effect of the endpoint bias can be more easily assessed when comparing the plot of the two trends, rather than the one of the resulting cyclical fluctuations. As one can see in the next graph, when dealing with the HP-Filter on shorter time-series, one overestimates the trend when compared to the filter applied over the full sample.

```
figure;
plot(dates, lgdp_T_de) ; hold on;
plot(datesCut, lgdp_de_cut_T) ; hold on;
plot(dates, lgdp_de); hold on;

xlabel('Years');
ylabel('Log Billions of dollars');
title('Germany');
legend('Location', 'northwest')
legend('long trend', 'end-point biased', 'log data');
```



Q. 08 - 0 Download new data

```
series = ["MKTGDPDEA646NWDB", "PCAGDPDEA646NWDB", ...
    "MKTGDPITA646NWDB", "PCAGDPITA646NWDB"];
startDate = "01/01/1970";
endDate = "01/01/2016";

% retrieve data struct of the 6 time series
url = "https://fred.stlouisfed.org/";
dataStruct2 = fetch(fred(url), series, startDate, endDate);

gdpDE    = dataStruct2(1).Data(:,2);
gdpDEpc  = dataStruct2(2).Data(:,2);
gdpIT    = dataStruct2(3).Data(:,2);
gdpITpc  = dataStruct2(4).Data(:,2);
```

Q. 08 - a) Apply HP Filter and comment

ANSWER: We picked $ad\lambda = 6.25$, because it is the conventional value for annual data, whereas for quarterly it is 1600 and 129 600 for monthly data. However, this is still a debated topic in the literature and some authors defend the usage of significantly different values, such as \lambda = 100 for yearly data.

```
lambda = 6.25;
[gdpDE_T, gdpDE_C] =  hp_filter(gdpDE, lambda);
[gdpDE_T_pc, gdpDEpc_C] =  hp_filter(gdpDEpc, lambda);

[gdpIT_T, gdpIT_C] =  hp_filter(gdpIT, lambda);
[gdpIT_T_pc, gdpITpc_C] =  hp_filter(gdpITpc, lambda);
```

Q. 08 - b) Normalizing the trend series

```
%germany
gdpDE_T_n = gdpDE_T / gdpDE_T(1);
gdpDE_T_n_pc = gdpDE_T_pc / gdpDE_T_pc(1);

% italy
gdpIT_T_n = gdpIT_T / gdpIT_T(1);
gdpIT_T_n_pc = gdpIT_T_pc / gdpIT_T_pc(1);
```

Q. 08 - c) compute annual growth rates

ANSWER: Germany has a 6.260% compound average growth rate of GDP and 6.158% of GDP per capita, whereas Italy has 6.258% and 5.968%, respectively. The growth rates between the two countries are similar, but Italy grew less than Germany in nominal terms, with a bigger difference when accounting for the population growth. We were expecting a bigger difference when comparing the growth of GDP and GDP per capita, however, from the 1970 onwards, population grew at a very little rate.

Here, we computed the compound average growth rate in order to account for the compouding growth effect, which can be substantial in the longer run.

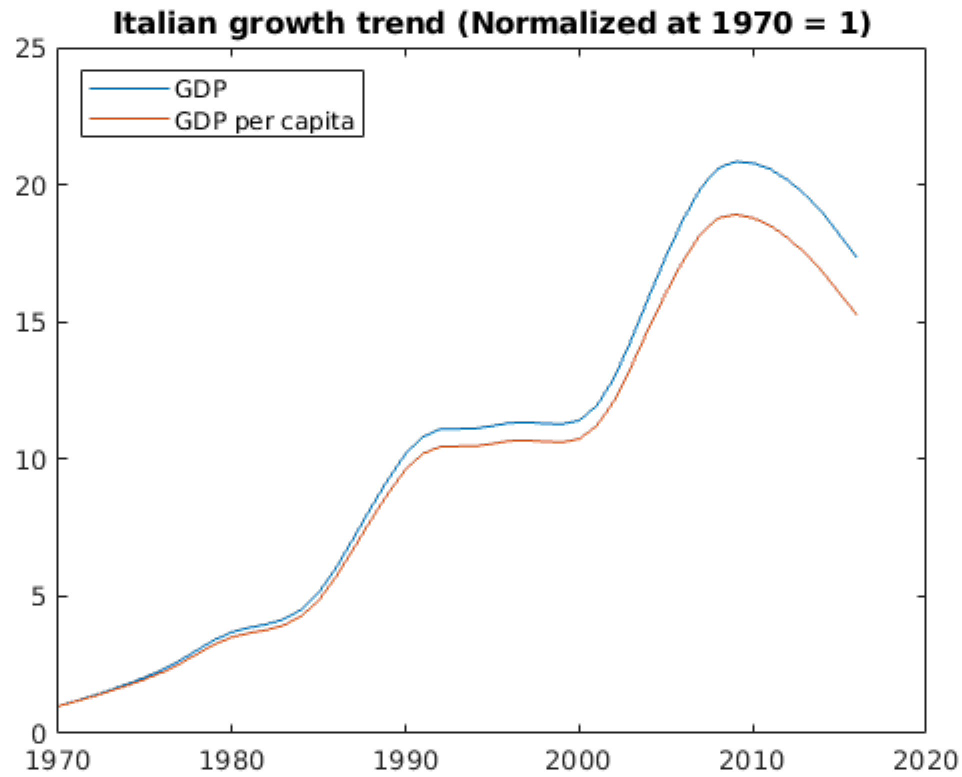
```
%germany
DEgrowth = (gdpDE_T_n(2:end)./gdpDE_T_n(1:end-1)-1);
DEgrowth_pc = (gdpDE_T_n_pc(2:end)./gdpDE_T_n_pc(1:end-1)-1);

%italy
ITgrowth = (gdpIT_T_n(2:end)./gdpIT_T_n(1:end-1)-1);
ITgrowth_pc =(gdpIT_T_n_pc(2:end)./gdpIT_T_n_pc(1:end-1)-1);

% Germay
%CAGR = (Ending value / Beginning value) ^ (1/n) - 1
CAGR_DE = (gdpDE_T_n(end) / gdpDE_T_n(1)) ^ (1 / length(gdpDE_T_n)) -
 1
CAGR_DE_pc = (gdpDE_T_pc(end) / gdpDE_T_pc(1)) ^ (1 /
 length(gdpDE_T_pc)) - 1

% Italy
CAGR_IT = (gdpIT_T_n(end) / gdpIT_T_n(1)) ^ (1 / length(gdpIT_T_n)) -
 1
```

```
CAGR_IT_pc = (gdpIT_T_n_pc(end) / gdpIT_T_n_pc(1)) ^ (1 /
 length(gdpIT_T_n_pc)) - 1
```

```
CAGR_DE =

    0.0626


CAGR_DE_pc =

    0.0616


CAGR_IT =

    0.0626


CAGR_IT_pc =

    0.0597
```

Q. 08 - d) Plot normalized trend series for GDP and GDP per capita

```
figure;
plot(1970:2016, [gdpIT_T_n, gdpIT_T_n_pc]); hold on;
title('Italian growth trend (Normalized at 1970 = 1)')
legend('GDP', 'GDP per capita')
legend('Location','northwest', 'Orientation','vertical')
snapnow;
```



Published with MATLAB® R2018b