Intermediate Macroeconomics

Data Exercise 1

Name: Salomon Ruiz

NetID: srh450

```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
import statsmodels.api as sm
%matplotlib inline
```

Question 1

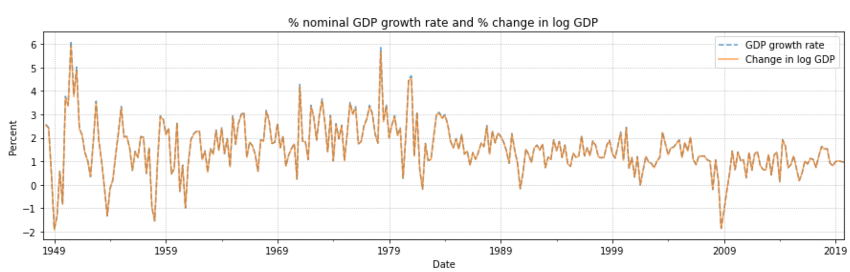
1.1

```
gdp = pd.read_csv("GDP.csv")
gdp['DATE'] = pd.to_datetime(gdp['DATE']); gdp['GDP growth rate'] = gdp['GDP'].pct_change()*100
gdp['log nominal GDP'] = np.log(gdp['GDP']); gdp['Change in log GDP'] = gdp['log nominal GDP'].diff()*100; gdp.head(3)
```

	DATE	GDP	GDP growth rate	log nominal GDP	Change in log GDP
0	1948-01-01	265.742	NaN	5.582526	NaN
1	1948-04-01	272.567	2.568281	5.607884	2.535854
2	1948-07-01	279.196	2.432063	5.631914	2.402959

```
ax = gdp.plot(x ='DATE', y=['GDP growth rate', 'Change in log GDP'], title='% nominal GDP growth rate and % change in log GDP' , kind = 'line', figsize=(15, 4), style=['--','-'], alpha= 0.7)
ax.set_ylabel("Percent"), ax.set_xlabel("Date"), ax.grid(linestyle=':')
```

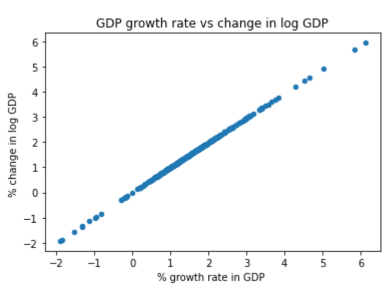
(Text(0, 0.5, 'Percent'), Text(0.5, 0, 'Date'), None)



1.2

```
ax = gdp.plot(x ='GDP growth rate', y='change in log GDP' , kind = 'scatter', title='GDP growth rate vs change in log GDP'); ax.set_xlabel("% growth rate in GDP "), ax.set_ylabel('% change in log GDP ')
```

(Text(0.5, 0, '% growth rate in GDP '), Text(0, 0.5, '% change in log GDP '))



1.3

If we consider Yt as GDP in a given year t, then Yt+1 = Yt(1 + g), where g is the growth rate.
Then, we take the natural log of Yt+1 and by the arithmetic fact that log(xy) = log(x) + log(y), we get the following:
log(Yt+1) = log(Yt) + log(1 + g).
If we calculate the limit of (log(1 + g))/g when g is close to zero, then we realize log(1 + g) is approximately g since the limit is 1.
Then, log(Yt+1) ≈ log(Yt) + g
Finally, we can say g ≈ log(Yt+1) - log(Yt), which explains why the two data series are very close. The growth rate and the change in log are almost the same.

Question 2

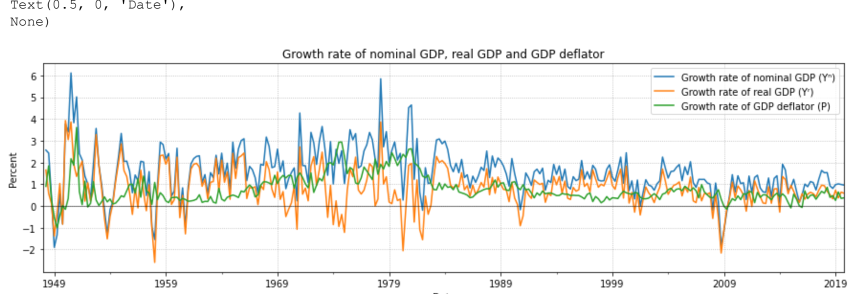
2.1

```
gdp2 = pd.read_csv("GDP.csv"); rgdp = pd.read_csv("GDPC1.csv")
gdp2['DATE'], gdp2['Real GDP'] = [pd.to_datetime(gdp['DATE']), rgdp['GDPC1']]
gdp2['GDP Deflator'] = (gdp2['GDP']/gdp2['Real GDP'])*100
gdp2['Growth rate of nominal GDP (Yⁿ)'] = gdp2['GDP'].pct_change()*100
gdp2['Growth rate of real GDP (Yʳ)'] = gdp2['Real GDP'].pct_change()*100
gdp2['Growth rate of GDP deflator (P)'] = gdp2['GDP Deflator'].pct_change()*100; gdp2.head(3)
```

	DATE	GDP	Real GDP	GDP Deflator	Growth rate of nominal GDP (Yⁿ)	Growth rate of real GDP (Yʳ)	Growth rate of GDP deflator (P)
0	1948-01-01	265.742	2086.017	12.739206	NaN	NaN	NaN
1	1948-04-01	272.567	2120.450	12.854205	2.568281	1.650658	0.902722
2	1948-07-01	279.196	2132.598	13.091825	2.432063	0.572897	1.848575

```
ax = gdp2.plot(x ='DATE', y=['Growth rate of nominal GDP (Yⁿ)', 'Growth rate of real GDP (Yʳ)', 'Growth rate of GDP deflator (P)'], kind = 'line', title = 'Growth rate of nominal GDP, real GDP and GDP deflator',figsize=(15, 4), yticks=([w*1 for w in range(-2,7)]))
ax.axhline(y=0, color='k', linewidth= 0.5), ax.set_ylabel("Percent"), ax.set_xlabel("Date"), ax.grid(linestyle=':')
```

(<matplotlib.lines.Line2D at 0x1c33141aa00>,
 Text(0, 0.5, 'Percent'),
 Text(0.5, 0, 'Date'),
 None)



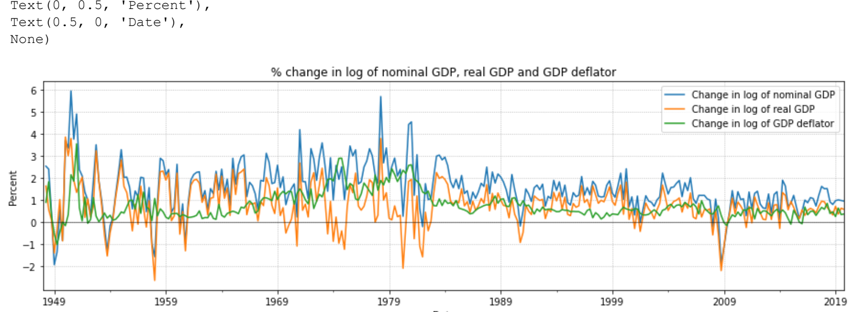
We can see from the graph that it is approximately true that gr(Yn) = gr(Yr) + gr(P). This is explained by the arithmetic fact that the percentage change in the product of two variables is aproximately equal to the sum of the percentage change of each variable. Therefore, since Yn = Yr *P, then gr(Yr P) ≈ gr(Yn) + gr(P).*

2.2

```
gdp2['natural_log1'] = np.log(gdp2['GDP']) ; gdp2['natural_log2'] = np.log(gdp2['Real GDP'])
gdp2['natural_log3'] = np.log(gdp2['GDP Deflator']); gdp2['Change in log of nominal GDP'] = gdp2['natural_log1'].diff()*100
gdp2['Change in log of real GDP'] = gdp2['natural_log2'].diff()*100; gdp2['Change in log of GDP deflator'] = gdp2['natural_log3'].diff()*100
```

```
ax2= gdp2.plot(x ='DATE', y=['Change in log of nominal GDP', 'Change in log of real GDP', 'Change in log of GDP deflator'], kind = 'line', title = '% change in log of nominal GDP, real GDP and GDP deflator',figsize=(15, 4), yticks=([w*1 for w in range(-2,7)]))
ax2.axhline(y=0, color='k', linewidth= 0.5), ax2.set_ylabel("Percent"), ax2.set_xlabel("Date"), ax2.grid(linestyle=':')
```

(<matplotlib.lines.Line2D at 0x1c33113b550>,
 Text(0, 0.5, 'Percent'),
 Text(0.5, 0, 'Date'),
 None)



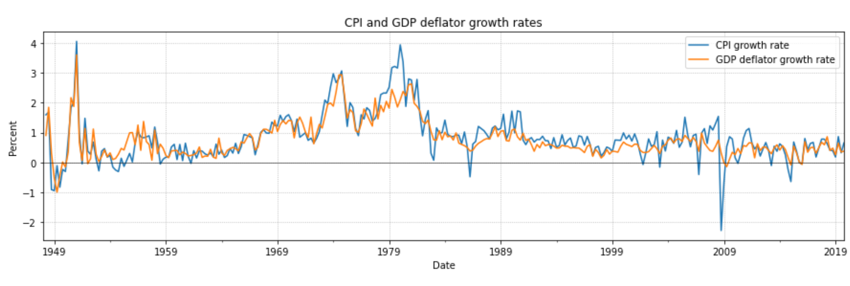
2.3

```
cpi = pd.read_csv("CPIAUCSL.csv"); cpi['DATE'] = pd.to_datetime(cpi["DATE"])
cpi["CPI growth rate"] = cpi['CPIAUCSL'].pct_change()*100; cpi["GDP deflator growth rate"] = gdp2['Growth rate of GDP deflator (P)']; cpi.head(3)
```

	DATE	CPIAUCSL	CPI growth rate	GDP deflator growth rate
0	1948-01-01	23.618667	NaN	NaN
1	1948-04-01	23.993333	1.584619	0.902722
2	1948-07-01	24.396667	1.681023	1.848575

```
ax3 = cpi.plot(x ="DATE", y=["CPI growth rate", "GDP deflator growth rate"], kind = 'line', title = "CPI and GDP deflator growth rates", figsize=(15, 4), yticks=([w*1 for w in range(-2,7)]))
ax3.axhline(y=0, color='k', linewidth= 0.5), ax3.set_ylabel("Percent"), ax3.set_xlabel("Date"), ax3.grid(linestyle=':')
```

(<matplotlib.lines.Line2D at [illegible]>,
 Text(0, 0.5, 'Percent'),
 Text(0.5, 0, 'Date'),
 None)




```
cpi.corr().loc["CPI growth rate","GDP deflator growth rate"]
```

[illegible]

The coefficient we obtain indicates there is a strong positive correlation between the growth rate of the CPI and the growth rate of the GDP deflator. However, we can see in the graph that there are periods of time with big discrepancies between the two series. A potential explanation could be an increase or decrease in price of imported goods. Since the basket of goods and services used to compute CPI can include imported goods, it can be significantly higher or lower than the GDP deflator. For instance, during the oil crisis of 1979, since oil prices increased and the US imported oil to produce goods included in the CPI basket, the CPI growth rate was considerably higher than that of the GDP deflator.

2.4

```
[illegible]
```

[illegible]

```
[illegible]
```

[illegible]

We can notice the correlation coefficients are almost identical.

Question 3

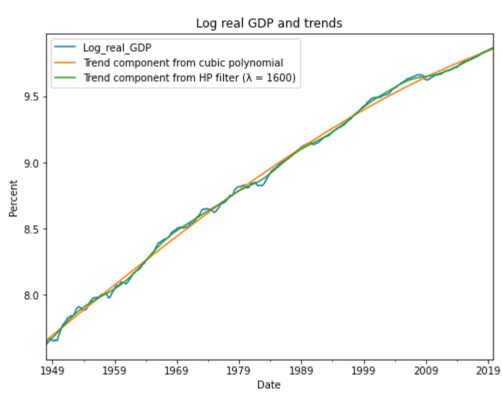
3.1

```
[illegible]
```

	DATE	GDPC1	Log_real_GDP	t	t²	t³	Trend component from cubic polynomial	Trend component from HP filter (λ = 1600)
0	1948-01-01	2086.017	7.643012	1	1	1	[illegible]	[illegible]
1	1948-04-01	2120.450	7.659394	2	4	8	[illegible]	[illegible]
2	1948-07-01	2132.598	7.665090	3	9	27	[illegible]	[illegible]

```
[illegible]
```

(Text(0, 0.5, 'Percent'), Text(0.5, 0, 'Date'))

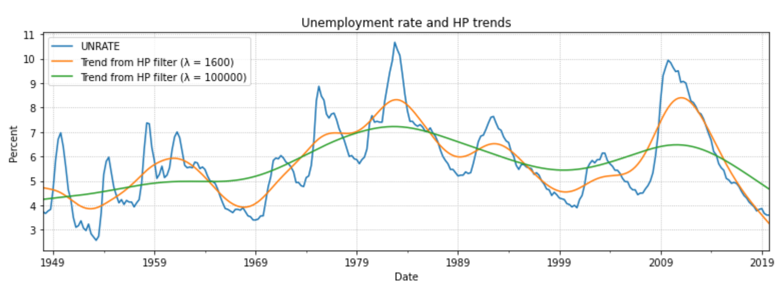


We can see from the plot that the trend from the cubic polynomial is smoother. The trend curve obtained from the HP filter is also smooth, but it follows more closely the shape of the log real GDP curve, so it moves above and below the other trend curve.

3.2

```
[illegible]
```

(Text(0, 0.5, 'Percent'), Text(0.5, 0, 'Date'), None)



I think the HP trend with λ = 100000 captures the idea of a long run trend better since it has less fluctuations. We can say unemployment rate has been between 4 and 7% in the long run. Since the parameter λ determines how smooth the trend is, a smoother curve could better capture the long term trend than a curve that follows more closely the volatility of unemployment rate, such as the one with λ = 1600.

3.3

```
[illegible]
```

[illegible]

```
[illegible]
```

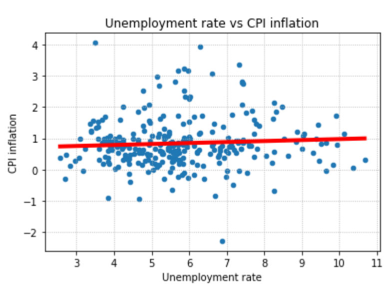
[illegible]

The high negative correlation was expected since the detrended series are countercyclical. That means when unemployment rate rises, real GDP decreases and vice versa.

3.4

```
[illegible]
ax = unr2.plot.scatter(x = 'UNRATE',y ='CPI_infl', title = 'Unemployment rate vs CPI inflation')
unr2.sort_values('UNRATE').set_index('UNRATE')['reg'].plot.line(color='red',linewidth=4)
ax.set_ylabel("CPI inflation"), ax.set_xlabel("Unemployment rate"), ax.grid(linestyle=':')
```

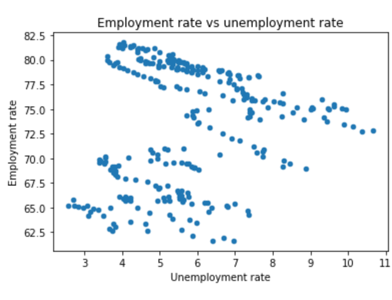
(Text(0, 0.5, 'CPI inflation'), Text(0.5, 0, 'Unemployment rate'), None)



3.5

```
emp = pd.read_csv("LNS12300060.csv"); emp['unemp'] = unr['UNRATE']; axe = emp.plot.scatter(x = 'unemp',y ='LNS12300060', title = 'Employment rate vs unemployment rate')
axe.set_ylabel("Employment rate"), axe.set_xlabel("Unemployment rate")
```

(Text(0, 0.5, 'Employment rate'), Text(0.5, 0, 'Unemployment rate'))

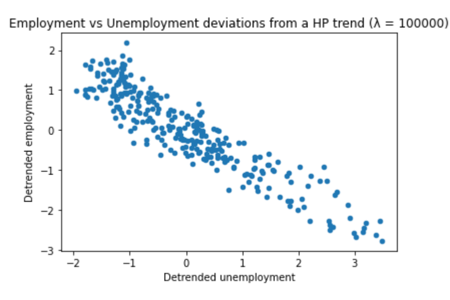



```
emp.corr().loc['unemp', 'LNS12300060']
```

0.1479132732948325

```
cycle4, trend4 = sm.tsa.filters.hpfilter(emp['unemp'], 100000); emp["Detrended unemployment"] = cycle4
cycle5, trend5 = sm.tsa.filters.hpfilter(emp['LNS12300060'], 100000); emp["Detrended employment"] = cycle5;
emp.plot.scatter(x = 'Detrended unemployment',y ='Detrended employment', title = 'Employment vs Unemployment deviations from a HP trend (λ = 100000)', )
```

<matplotlib.axes._subplots.AxesSubplot at 0x1c336191d30>




```
emp.corr().loc['dev unemp', 'dev emplo']
```

-0.9244518154484203

The correlation between employment and unemployment is positive but small. However, there is a strong negative correlation between the employment and unemployment deviations from a HP trend. This is because the deviations represent the cyclical component of the series, and ignores the trend, which means when employment is high, unemployment is low and the opposite is true. However, if we look at the original series, we can observe a growing trend in the employment rate for people aged between 25 and 54 for most of the data time frame. Therefore, the employment rate has increased in the long run while unemployment rate has been around the same levels, which explains the low correlation.

Sources:
https://fred.stlouisfed.org/
https://www.federalreservehistory.org/essays/oil-shock-of-1978-79

```
from IPython.core.display import display,HTML
display(HTML('<style>.prompt{width:0px; min-width:0px; visibility: collapse}</style>'))
import warnings
warnings.filterwarnings('ignore')
```