NaN

2.535854

2.402959

% nominal GDP growth rate and % change in log GDP

1989

ax = gdp.plot(x ='GDP growth rate', y='change in log GDP', kind = 'scatter', title='GDP growth rate vs change

1999

1979

in log GDP'); ax.set xlabel("% growth rate in GDP "), ax.set ylabel('% change in log GDP ')

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

--- GDP growth rate

2009

Change in log GDP

2019

Data Exercise 1

Intermediate Macroeconomics

import statsmodels.formula.api as smf import statsmodels.api as sm %matplotlib inline

NetID: srh450

Name: Salomon Ruiz

import pandas as pd import numpy as np

Question 1

GDP growth rate log nominal GDP Change in log GDP

5.582526

5.607884

5.631914

NaN

2.568281

2.432063

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

1969

1959

GDP growth rate vs change in log GDP

% growth rate in GDP

If we consider Yt as GDP in a given year t, then Yt+1 = Yt(1 + g), where g is the growth rate.

gdp2 = pd.read csv("GDP.csv"); rgdp = pd.read_csv("GDPC1.csv")

gdp2['GDP Deflator'] = (gdp2['GDP']/gdp2['Real GDP'])*100

Real

GDP

2086.017

2120.450

2132.598

, 4), yticks=([w*1 **for** w **in** range(-2,7)]))

(<matplotlib.lines.Line2D at 0x1c33141aa00>,

1959

gdp2['DATE'], gdp2['Real GDP'] = [pd.to_datetime(gdp['DATE']), rgdp['GDPC1']]

gdp2['Growth rate of nominal GDP (Yn)'] = gdp2['GDP'].pct change()*100 $gdp2['Growth rate of real GDP (Yr)'] = gdp2['Real GDP'].pct_change()*100$

GDP

Deflator

12.739206

12.854205

13.091825

1969

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:

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.

gdp2['Growth rate of GDP deflator (P)'] = gdp2['GDP Deflator'].pct_change()*100; gdp2.head(3)

Growth rate of nominal GDP

 (Y^n)

NaN

2.568281

2.432063

 $ax = gdp2.plot(x = 'DATE', y = ['Growth rate of nominal GDP (Y^n)', 'Growth rate of real GDP (Y^r)', 'Growth rate of$ GDP deflator (P)'], kind = 'line', title = 'Growth rate of nominal GDP, real GDP and GDP deflator', figsize=(15

ax.axhline(y=0, color='k', linewidth= 0.5), ax.set_ylabel("Percent"), ax.set_xlabel("Date"), ax.grid(linestyle=

Growth rate of nominal GDP, real GDP and GDP deflator

Date

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 $gr(Yr P) \approx$

gdp2['natural_log3'] = np.log(gdp2['GDP Deflator']); gdp2['Change in log of nominal GDP'] = gdp2['natural_log1'

gdp2['Change in log of real GDP'] = gdp2['natural log2'].diff()*100; gdp2['Change in log of GDP deflator'] = gd

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

ax2.axhline(y=0, color='k', linewidth= 0.5), ax2.set ylabel("Percent"), ax2.set xlabel("Date"), ax2.grid(linest

% change in log of nominal GDP, real GDP and GDP deflator

Date

cpi['CPI growth rate'] = cpi['CPIAUCSL'].pct change()*100; cpi['GDP deflator growth rate'] = gdp2['Growth rate

ax3 = cpi.plot(x ='DATE', y=['CPI growth rate', 'GDP deflator growth rate'], kind = 'line', title = 'CPI and GD

ax3.axhline(y=0, color='k', linewidth= 0.5), ax3.set ylabel("Percent"), ax3.set xlabel("Date"), ax3.grid(linest

1989

1999

CPI and GDP deflator growth rates

Date

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

gdp2['cpi inflation'] = cpi['% growth rate']; gdp2.corr().loc['Growth rate of real GDP (Yr)', 'cpi inflation']

rgdp2['Trend component from cubic polynomial'] = smf.ols('Log_real_GDP ~ t + tt + ttt',rgdp2).fit().predict() cycle, trend = sm.tsa.filters.hpfilter(rgdp2['Log real GDP'], 1600); rgdp2["Trend component from HP filter (λ =

GDPC1 Log_real_GDP t tt ttt Trend component from cubic polynomial Trend component from HP filter (λ = 1600)

ax = rgdp2.plot(x ='DATE', y=['Log real GDP','Trend component from cubic polynomial','Trend component from HP f

7.653395

7.663052

7.672706

1989

1999

1979

NaN

0.902722

1.848575

1979

produce goods included in the CPI basket, the CPI growth rate was considerably higher than that of the GDP deflator.

gdp2.corr().loc['Growth rate of real GDP (Yr)', 'Growth rate of GDP deflator (P)']

rgdp2= pd.read csv("GDPC1.csv"); rgdp2['DATE'] = pd.to datetime(rgdp2['DATE'])

rgdp2['Log_real_GDP'] = np.log(rgdp2['GDPC1']); rgdp2['t'] = np.arange(start=1 , stop= 289)

1969

NaN

1.594919

1.681023

cpi = pd.read csv("CPIAUCSL.csv"); cpi['DATE'] = pd.to datetime(cpi['DATE'])

GDP deflator growth rate

1989

1999

1979

gdp2['natural_log1'] = np.log(gdp2['GDP']) ; gdp2['natural_log2'] = np.log(gdp2['Real GDP'])

Growth rate of real GDP

 (Y^r)

NaN

1.650658

0.572897

Growth rate of GDP deflator (P)

Growth rate of nominal GDP (Yn) Growth rate of real GDP (Y1) Growth rate of GDP deflator (P)

Change in log of nominal GDP Change in log of real GDP

Change in log of GDP deflator

2019

2009

CPI growth rate

2009

GDP deflator growth rate

7.622613

7.634956

7.647312

2019

NaN

0.902722

1.848575

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

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**

1948-01-01 265.742 1948-04-01 272.567 **2** 1948-07-01 279.196

ax = gdp.plot(x = 'DATE', y=['GDP growth rate', 'Change in log GDP'], title='% nominal GDP growth rate and % cha nge in log GDP' , kind = 'line', figsize=(15, 4), style=['--','-'], alpha= 0.7)

ax.set_ylabel("Percent"), ax.set_xlabel("Date"), ax.grid(linestyle=':')

6

1949

1.2

% change in log GDP

1.3

 $\log(Yt+1) = \log(Yt) + \log(1+g).$

Then, $log(Yt+1) \approx log(Yt) + g$

6 5 4

same.

2.1

Question 2

DATE

1948-01-

1948-04-

01

GDP

272.567

279.196

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

2

None)

4

2

0 -1-2

1949

gr(Yn) + gr(P).

].diff()*100

yle=':')

None)

6

5

1949

1948-01-01

1948-04-01

2 1948-07-01

yle=':')

None)

3

2

0

-1

2.4

1949

0.863149415533173

-0.05766462594665164

-0.059759728122171174

1600)"] = trend; rgdp2.head(3)

DATE

0 1948-01-01 2086.017

1 1948-04-01 2120.450

1948-07-01 2132.598

Log_real_GDP

1959

1969

1979

le = 'Unemployment rate and HP trends', figsize=(13,4))

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

1969

Date

1989

more closely the shape of the log real GDP curve, so it moves above and below the other trend curve.

unr = pd.read csv("UNRATE.csv"); unr['DATE'] = pd.to datetime(unr['DATE'])

ax.set ylabel("Percent"), ax.set xlabel("Date"), ax.grid(linestyle=':')

1999

2009

2019

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

cycle, trend = sm.tsa.filters.hpfilter(unr['UNRATE'], 1600); unr["Trend from HP filter ($\lambda = 1600$)"] = trend cycle2, trend2 = sm.tsa.filters.hpfilter(unr['UNRATE'], 100000); unr["Trend from HP filter (λ = 100000)"] = tre

Unemployment rate and HP trends

Date

unr["detrended UR"] = cycle2; cycle3, trend3 = sm.tsa.filters.hpfilter(rgdp2['GDPC1'], 1600)

Trend component from

cubic polynomial

7.653395

7.663052

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

unr['CPI infl'] = gdp2['cpi inflation']; unr2 = unr.dropna(); unr2['reg'] = smf.ols('CPI infl ~ UNRATE',unr2).f

emp = pd.read_csv("LNS12300060.csv"); emp['unemp'] = unr['UNRATE']; axe = emp.plot.scatter(x = 'unemp', y = 'LNS1

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)

10

10

cycle4, trend4 = sm.tsa.filters.hpfilter(emp['unemp'], 100000); emp["Detrended unemployment"] = cycle4 cycle5, trend5 = sm.tsa.filters.hpfilter(emp['LNS12300060'], 1000000); emp["Detrended employment"] = cycle5; emp.plot.scatter(x = 'Detrended unemployment', y = 'Detrended employment', title = 'Employment vs Unemployment de

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

1979

trend than a curve that follows more closely the volatility of unemployment rate, such as the one with λ = 1600.

ax = unr.plot(x='DATE', y=['UNRATE','Trend from HP filter (λ = 1600)','Trend from HP filter (λ = 100000)'], tit

1989

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

rgdp2['detrended realgdp'] = cycle3; rgdp2["detrended UR"] = unr['detrended UR']; rgdp2["UNRATE"] = unr['UNRAT

1999

Trend component from HP

filter ($\lambda = 1600$)

7.622613

7.634956

2009

detrended

54.571205

60.360660

realgdp

2019

detrended

-0.510423

UR

-0.588799 3.666667

UNRATE

3.733333

9.5

9.0

8.5

8.0

3.2

nd2;

11

10

1949

E']; rgdp2.head(2)

2086.017

2120.450

DATE

1948-

01-01

04-01

-0.7010396105748045

decreases and vice versa.

it().predict()

4

3

2

0

-1

-2

CPI inflation

3.5

82.5 80.0 77.5

75.0 72.5 70.0 67.5 65.0 62.5

0.1479132732948325

Detrended employment

Sources:

Employment rate

3.3

UNRATE

Trend from HP filter ($\lambda = 1600$) Trend from HP filter ($\lambda = 100000$)

1959

GDPC1 Log_real_GDP t tt ttt

7.643012 1 1 1

7.659384 2 4

rgdp2.corr().loc['detrend realg', 'detrended UR']

Unemployment rate vs CPI inflation

Unemployment rate

2300060', title = 'Employment rate vs unemployment rate')

Employment rate vs unemployment rate

Unemployment rate

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

Employment vs Unemployment deviations from a HP trend ($\lambda = 100000$)

Detrended unemployment

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

from IPython.core.display import display, HTML

warnings.filterwarnings('ignore')

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

-0.9244518154484203

https://fred.stlouisfed.org/

import warnings

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

viations from a HP trend ($\lambda = 100000$)',)

axe.set_ylabel("Employment rate"), axe.set_xlabel("Unemployment rate")

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

1949

Question 3

3.1

2.3

p2['natural log3'].diff()*100

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

), yticks=([w*1 for w in range(-2,7)]))

1959

of GDP deflator (P)']; cpi.head(3)

DATE CPIAUCSL CPI growth rate

P deflator growth rates', figsize=(15, 4))

1959

cpi.corr().loc['% growth rate','% growth rate defl']

We can notice the correlation coefficients are almost identical.

rgdp2['tt'] = rgdp2['t']**2; rgdp2['ttt'] = rgdp2['t']**3;

7.643012 1 1 1 7.659384 2 4

7.665096 3 9 27

ax.set ylabel("Percent"), ax.set xlabel("Date")

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

Trend component from cubic polynomial Trend component from HP filter ($\lambda = 1600$)

ilter ($\lambda = 1600$)'], title = 'Log real GDP and trends', figsize=(8,6))

Log real GDP and trends

1969

(<matplotlib.lines.Line2D at 0x1c33103eaf0>,

23.616667

23.993333

24.396667

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

(<matplotlib.lines.Line2D at 0x1c33113b550>,

2.2

3.4

increased in the long run while unemployment rate has been around the same levels, which explains the low correlation.

display(HTML('<style>.prompt{width:0px; min-width:0px; visibility: collapse}</style>'))