

## ECON 706: PROBLEM SET 8

DANIEL PFEFFER

**Problem 1.** Suppose that you have an AR(1) process of the form

$$y_t = \rho y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, 1). \quad (1)$$

Now consider a two-state Markov process with  $s_t \in \{s_1, s_2\}$  with transition matrix

$$P = \begin{pmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{pmatrix}. \quad (2)$$

Here,  $p_{ij}$  is the probability of transitioning from state  $s_1$  to state  $s_2$ . The Markov process has four parameters:  $\theta = (p_{11}, p_{22}, s_1, s_2)$ . We will try to approximate the AR(1) process  $y_t$  in (1) by the Markov process  $s_t$  indexed by  $\theta$ .

- (i) Derive the eigenvalues associated with  $P$  in (2) as a function of  $\theta$ .

*Solution.* The characteristic polynomial is

$$\begin{aligned} \det(P - \lambda I) &= (p_{11} - \lambda)(p_{22} - \lambda) - (1 - p_{11})(1 - p_{22}) = 0 \\ &= \lambda^2 - \lambda(p_{11} + p_{22}) + p_{11}p_{22} - 1 + p_{11}p_{22} - p_{11}p_{22} \\ &= \lambda^2 - \lambda(p_{11} + p_{22}) - (1 - p_{11} - p_{22}), \end{aligned}$$

and has two eigenvalues given by

$$\begin{aligned} \lambda &= \frac{p_{11} + p_{22}}{2} \pm \sqrt{\frac{(p_{11} + p_{22})^2}{4} + 1 - \frac{2(p_{11} + p_{22})}{2}} \\ &= \frac{p_{11} + p_{22}}{2} \pm \left(1 - \frac{p_{11} + p_{22}}{2}\right), \end{aligned}$$

with solutions  $\lambda_1 = 1$  and  $\lambda_2 = p_{11} + p_{22} - 1$ . □

- (ii) Derive the equilibrium distribution associated with  $P$  as a function of  $\theta$ .

*Solution.* Let  $\pi = (\pi_1 \ \pi_2)$  denote the invariant distribution of  $P$ , i.e., the (left) eigenvector corresponding to the eigenvalue  $\lambda_1 = 1$ . Then we know that  $\pi$  must satisfy  $\pi P = \pi$ . i.e.,

$$(\pi_1 \ \pi_2) \begin{pmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{pmatrix} = (\pi_1 \ \pi_2),$$

which is an underdetermined system. Imposing a normalization restriction on  $\pi$  (i.e., the probabilities sum to one) yields

$$\begin{cases} p_{11}\pi_1 + (1 - p_{22})\pi_2 = \pi_1 \\ (1 - p_{11})\pi_1 + p_{22}\pi_2 = \pi_2 \end{cases} \implies \begin{cases} (1 - p_{11})\pi_1 = (1 - p_{22})\pi_2 \\ \pi_1 + \pi_2 = 1 \end{cases},$$

which has solution (equilibrium distribution)

$$\pi = \left( \frac{1 - p_{22}}{2 - p_{11} - p_{22}}, \frac{1 - p_{11}}{2 - p_{11} - p_{22}} \right).$$

Here is a Python representation of a general chain.

---

```
from random import uniform

def sample(pi):
    a = 0.0
    U = uniform(0,1)
    for i in range(len(pi)):
        if a < U <= a + pi[i]:
            return i
        a += pi[i]

class MarkovChain:
    def __init__(self, p=None, X=None):
        self.p, self.X = p, X

    def update(self):
        self.X = sample(self.p[self.X])

    def sample_path(self, n):
        path = []
        for i in range(n):
            path.append(self.X)
            self.update()
        return path
```

---

□

- (iii) For the  $s_t$  process to approximate the  $y_t$  process well, what properties would you like it to satisfy? Another way of asking the question is: think of a discrepancy measure between the processes  $\{s_t\}_{t=-\infty}^{\infty}$  and  $\{y_t\}_{t=-\infty}^{\infty}$ .

*Solution.* We would like  $s_t$  to have the same covariance structure as  $y_t$ , that is, we minimize

$$\sum_{\tau \in \mathbb{Z}} \frac{|\gamma_y(\tau) - \gamma_s(\tau)|}{\gamma_y(0)}$$

with respect to  $\gamma_s$ . approximate the AR(1) process (1) by requiring the  $s_t$  process to have the same first and second order properties as the  $y_t$  process. This provides a reasonable discrepancy measure between the two processes, since the autocovariance functions is a fundamental representation of the  $y_t$  process. This method due to Tauchen (1986). □

- (iv) Based on your notion of discrepancy, explain how you would choose  $\theta$  approximate a continuous AR(1) process characterized by the parameters  $\rho, \sigma^2$ .

*Solution.* The Markov process admits a scalar AR(1) representation of the form

$$s_t = (1 - p_{11}) + (p_{11} + p_{22} - 1)s_{t-1} + \eta_t,$$

where the autoregressive coefficient is  $p_{11} + p_{22} - 1$ , the second eigenvalue  $\lambda_2$ . We will approximate the continuous AR(1) processes (1) as follows. Let us first set  $s_2 = -s_1$ . We now derive two moment conditions; given  $s_{t-1} = s_2$ , the conditional mean and variance of  $s_t$  satisfy

$$\begin{aligned} E(s_t \mid s_{t-1} = s_2) &= (1 - p_{22})s_2 + p_{22}s_2 = \rho s_2 \\ \text{var}(s_t \mid s_{t-1} = s_i) &= (1 - p_{22})(-s_2 - \rho s_2^2)^2 + p_{22}(s_2^2 - \rho s_2^2)^2 = 1. \end{aligned}$$

The first moment condition implies that

$$p_{22} = p_{11} = \frac{1 + \rho}{2},$$

and the second moment condition, together with the previous implication, yields

$$s_2 = -s_1 = \sqrt{\frac{\sigma^2}{1 - \rho^2}}.$$

This implies  $\pi_1 = \pi_2 = 1/2$ . To show that this method makes our discrepancy measure is zero, begin by inducting on  $\tau$  to show that

$$P^\tau = \begin{pmatrix} \frac{1+\rho^\tau}{2} & \frac{1-\rho^\tau}{2} \\ \frac{1-\rho^\tau}{2} & \frac{1+\rho^\tau}{2} \end{pmatrix}.$$

The claim holds for the one-step ahead transition since

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{1+\rho}{2} & \frac{1-\rho}{2} \\ \frac{1-\rho}{2} & \frac{1+\rho}{2} \end{pmatrix} = \begin{pmatrix} \frac{1+\rho}{2} & \frac{1-\rho}{2} \end{pmatrix}.$$

Assume the discrepancy measure is zero for  $\tau$ . Then for  $\tau + 1$ ,

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{1+\rho^\tau}{2} & \frac{1-\rho^\tau}{2} \\ \frac{1-\rho^\tau}{2} & \frac{1+\rho^\tau}{2} \end{pmatrix} = \begin{pmatrix} \frac{1+\rho^\tau}{2} & \frac{1-\rho^\tau}{2} \end{pmatrix}.$$

Next, since by construction  $E s_t = 0$ , the autocovariance function of  $s_t$  is

$$\begin{aligned} \gamma_s(\tau) &= E s_t s_{t+\tau} = E s_t E(s_{t+\tau} \mid s_t) \\ &= \pi_1 s_1 E(s_{t+\tau} \mid s_t = s_1) + (1 - \pi_1) s_2 E(s_{t+\tau} \mid s_t = s_2) \\ &= \frac{1}{2} s_1 \left( \frac{1 + \rho^2}{2} s_1^2 + \frac{1 - \rho^2}{2} s_2^2 \right) + \frac{1}{2} s_2 \left( \frac{1 + \rho^2}{2} s_2^2 + \frac{1 - \rho^2}{2} s_1^2 \right) \\ &= \frac{1}{2} \frac{1 + \rho^2}{2} s_1^2 + \frac{1 - \rho^2}{2} s_1 s_2 + \frac{1}{2} \frac{1 + \rho^2}{2} s_2^2 \\ &= \frac{\sigma^2}{1 - \rho^2} \rho^\tau = \gamma_y(\tau), \end{aligned}$$

the desired result. □

- (v) Choose values for  $\rho$ ,  $\sigma^2$ , compute your proposed  $\theta$ , then simulate the two processes and compare the sample autocovariance functions.

*Solution.* We set  $\rho = 0.5$  and  $\sigma^2 = 1$ . The following code generates a two state Markov chain approximation for this AR(1) process. Then the “true” AR(1) process is simulated. Figure 1 shows the resulting sample autocorrelation functions. Visual inspection suggests that, even with relatively high persistence, we can obtain a reasonable discretized approximation taking values  $s_t \in \{s_1, s_2\}$  to the continuous-valued AR(1) process.

---

```
# Simulation parameters
T = 100
sigma = 1
rho = 0.5

# Tauchen chain
s = sigma / np.sqrt(1-rho**2)
p_tauchen = (1 + rho) / 2
P = [[p_tauchen, 1-p_tauchen], [1-p_tauchen, p_tauchen]]
pi_tauchen = [p_tauchen, 1-p_tauchen]

tauchen_chain = MarkovChain(p=P, X=sample(pi_tauchen))

# Chain sample path
tauchen_path = tauchen_chain.sample_path(T)

# Chain acf
tauchen_acf = acf(tauchen_path)

# AR(1) sample path
y_path = [s]
for t in range(T):
    y_path.append(rho*y_path[t] + np.random.normal())

# AR(1) acf
ar1_acf = acf(y_path)

# Plot results
plt.figure(figsize=(10,4))

ax1 = plt.subplot(221)
plt.scatter(np.linspace(0, T, T), tauchen_path)
plt.setp(ax1.get_xticklabels(), visible=False)

ax2 = plt.subplot(222, sharex=ax1)
plt.plot(y_path)

ax3 = plt.subplot(223)
plt.plot(tauchen_acf)

ax4 = plt.subplot(224)
plt.plot(ar1_acf)

plt.show()
```

---

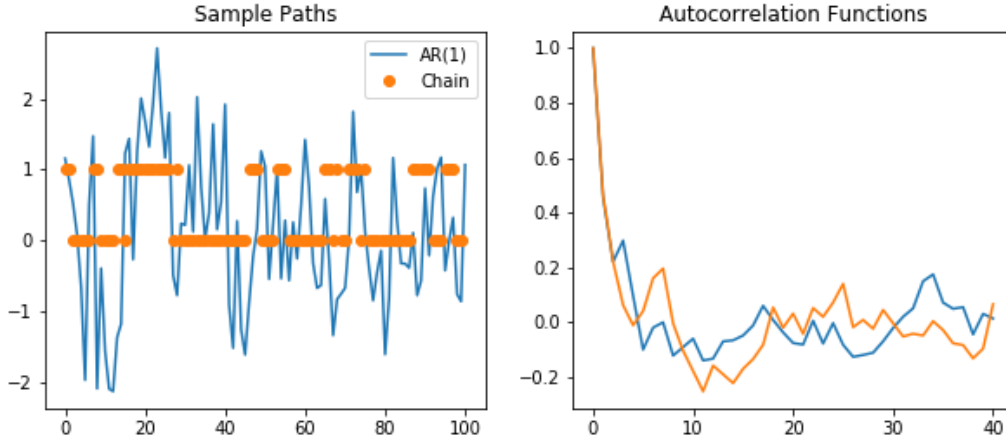


FIGURE 1. Sample autocorrelation functions for simulated sample paths  $\{y_t\}_{t=0}^{100}$  and  $\{s_t\}_{t=0}^{100}$ .

**Problem 2.** Consider the vector error correction model

$$\Delta y_t = \Pi_0 + \alpha\beta' y_{t-1} + \sum_{j=1}^{p-1} \Pi_j \Delta y_{t-j} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma). \quad (3)$$

(i) Rewrite the system as VAR in levels:

$$y_t = \Phi_0 + \sum_{j=1}^p \Phi_j y_{t-j} + \varepsilon_t. \quad (4)$$

What is the relationship between the  $\Phi$  matrices in (4) and the  $\Pi$  matrices in (3)?

*Solution.* We have

$$y_t - y_{t-1} = \Pi_0 + \alpha\beta' y_{t-1} + \Pi_1(y_{t-1} - y_{t-2}) + \cdots + \Pi_{p-1}(y_{t-p+1} - y_{t-p}) + \varepsilon_t,$$

which, after rearranging, becomes a VAR( $p$ ) in levels of the form

$$\begin{aligned} y_t &= \Pi_0 + (\alpha\beta' + \Pi_1 + I)y_{t-1} + (\Pi_2 - \Pi_1)y_{t-2} + \cdots + (\Pi_{p-1} - \Pi_{p-2})y_{t-p} - \Pi_{p-1}y_{t-p} + \varepsilon_t \\ &= \Phi_0 + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + \varepsilon_t \end{aligned}$$

where

$$\begin{aligned} \Phi_0 &= \Pi_0 \\ \Phi_1 &= \alpha\beta' + \Pi_1 + I \\ \Phi_j &= \Pi_j - \Pi_{j-1} \quad \text{for } j = 2, \dots, p-1 \\ \Phi_p &= -\Pi_{p-1}, \end{aligned}$$

□

(ii) Get quarterly data on nominal GDP and nominal investment from the FRED database (there are different types of aggregate investment series, just pick one of them and report which one you chose.)

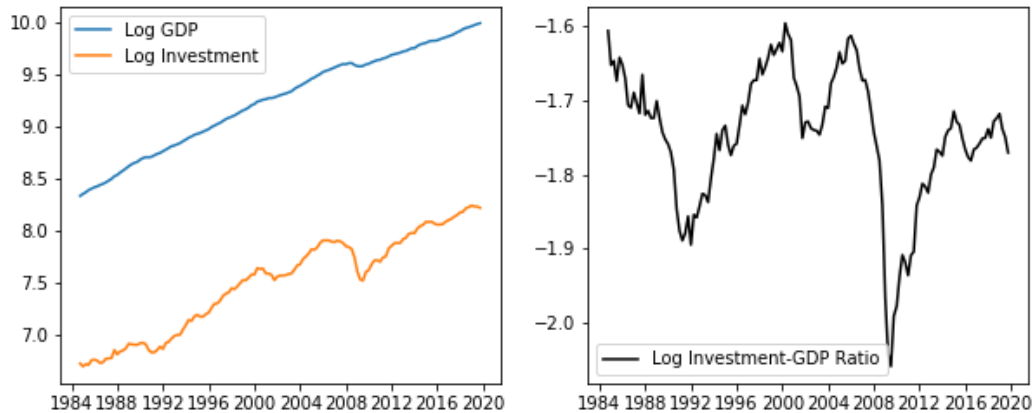


FIGURE 2. All data are nominal.

Plot log GDP and investment, as well as the log investment-to-GDP ratio. Comment on the trends in these three series over different periods of time.

*Solution.* Figure (2) shows log GDP and log gross private domestic investment (left), as well as log investment-to-GDP ratio (right) for 1984:Q1 through 2019:Q4. Both GDP and investment exhibit a linear trend in logs, and hence are nonstationary. The difference in values (i.e., ratio in levels), however, remain relatively constant. Moreover, the somewhat random fluctuations in the log investment-to-GDP ratio have characteristics of white noise, providing evidence of a common stochastic trend component shared by these series'. It should be noted that log investment is more volatile than log GDP, as can be seen around 1992 and 2004 through 2008. This observation is also seen in the log investment-to-GDP ratio at the corresponding time periods. This observed non-constant variance suggests that the log investment-to-GDP ratio may not be white noise. Code is used to generate these plots is presented below.

---

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader.data as web
import datetime

# 1984:Q1 to 2019:Q4
start = datetime.datetime(1984, 10, 1)
end = datetime.datetime(2019, 12, 31)

# Quarterly U.S. real GDP from the FRED
ts = web.DataReader(['GDP', 'GDI'], 'fred', start, end)

y = pd.DataFrame({'log_inv': np.log(ts['GDI']),
                  'log_gdp': np.log(ts['GDP'])})
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,4))

ax1.plot(y['log_gdp'], label='Log GDP')
ax1.plot(y['log_inv'], label='Log Investment')
ax1.legend()

ax2.plot(y['log_inv'] - y['log_gdp'], 'k', label='Log Investment-GDP Ratio')
ax2.legend()

plt.show()
```

□

- (iii) Define  $y_t = (\log \text{Inv}_t, \log \text{GDP}_t)'$ . Set  $p = 4$  and assume that the cointegration vector is  $\beta = (1, -1)'$ . Estimate (3) and (4), by OLS (over the same time period), setting  $p = 4$ .

*Solution.* Under these assumptions, the VECM becomes

$$\Delta y_{it} = \Pi_{i0} + \alpha_i(y_{1,t-1} - y_{2,t-1}) + \sum_{j=1}^3 ((\Pi_j)_{i1} \Delta y_{1,t-j} + (\Pi_j)_{i2} \Delta y_{2,t-j}) + \varepsilon_{it}, \quad i = 1, 2.$$

The following code estimates the VECM and VAR models from 1984:Q1 through 2019:Q4.

```
from statsmodels.tsa.api import VAR, VECM

y = ts[['log_inv', 'log_gdp']]
vecm_mod = VECM(y, k_ar_diff=3, deterministic='co', coint_rank=1, freq='QS')
vecm_res = vecm_mod.fit()

var_mod = VAR(y, freq='QS')
var_res = var_mod.fit(4)
```

The estimated VECM coefficients  $\hat{\Pi}$  are  $\hat{\Pi}_0 = (-0.286, 0.025)'$ ,<sup>1</sup>

$$\alpha\beta' = \begin{pmatrix} -0.080 & 0.093 \\ 0.006 & -0.007 \end{pmatrix} \quad \hat{\Pi}_1 = \begin{pmatrix} -0.113 & 2.684 \\ -0.002 & 0.319 \end{pmatrix}$$

$$\hat{\Pi}_2 = \begin{pmatrix} 0.060 & 1.094 \\ 0.010 & 0.151 \end{pmatrix} \quad \hat{\Pi}_3 = \begin{pmatrix} 0.044 & -0.727 \\ -0.007 & 0.005 \end{pmatrix}.$$

<sup>1</sup>All coefficient matrices throughout are of the form

$$\begin{matrix} & \text{Investment} & \text{GDP} \\ \begin{matrix} \text{Investment} \\ \text{GDP} \end{matrix} & \begin{pmatrix} * & * \\ * & * \end{pmatrix} \end{matrix}.$$

The estimates  $\hat{\Phi}$  are  $\hat{\Phi}_0 = (-0.260, 0.031)'$ ,

$$\begin{aligned}\hat{\Phi}_1 &= \begin{pmatrix} 0.807 & 2.669 \\ 0.004 & 1.285 \end{pmatrix} & \hat{\Phi}_2 &= \begin{pmatrix} 0.177 & -1.562 \\ 0.012 & -0.160 \end{pmatrix} \\ \hat{\Phi}_3 &= \begin{pmatrix} -0.009 & -1.814 \\ -0.016 & -0.145 \end{pmatrix} & \hat{\Phi}_4 &= \begin{pmatrix} -0.080 & 0.818 \\ -0.002 & 0.018 \end{pmatrix}.\end{aligned}$$

□

- (iv) Convert the estimates  $\hat{\Pi}$  into estimates  $\tilde{\Phi}$  obtained using the relationship derived in part (i). Compare the  $\tilde{\Phi}$  estimates obtained indirectly by estimating the VECM, to the  $\hat{\Phi}$  by estimating (4) directly. We did not develop enough econometric theory to conduct a formal test of whether the underlying parameters are identical and the VECM restrictions are satisfied, but please comment on the similarity of the estimates.

*Solution.* We now compute VAR coefficients implied by the above estimates.

---

```
# VAR implied by VECM
vecm_res.var_rep
```

---

The implied estimates  $\tilde{\Phi}$  are

$$\begin{aligned}\tilde{\Phi}_1 &= \begin{pmatrix} 0.807 & 2.777 \\ 0.004 & 1.312 \end{pmatrix} & \tilde{\Phi}_2 &= \begin{pmatrix} 0.172 & -1.590 \\ 0.012 & -0.167 \end{pmatrix} \\ \tilde{\Phi}_3 &= \begin{pmatrix} -0.016 & -1.821 \\ -0.017 & -0.147 \end{pmatrix} & \tilde{\Phi}_4 &= \begin{pmatrix} -0.044 & 0.727 \\ 0.007 & -0.005 \end{pmatrix}.\end{aligned}$$

The VAR estimates implied by the VECM model  $\tilde{\Phi}$  differ only slightly from the  $\hat{\Phi}$  estimates. We get this result since  $\Phi_1$  as defined in (i) captures the cointegrating relation and  $\Phi_2, \Phi_3$ , and  $\Phi_4$  capture the proposed white noise in the cointegrating residuals. □

- (v) Starting from the last  $p$  observations in your sample iterate (3) and (4) forward conditional on their respective OLS parameter estimates. Compare the implied paths of  $\log \text{Inv}_t$  and  $\log \text{GDP}_t$  over the next ten years (forty quarters) by generating plots that show the actual time series until period  $T$  and overlay the model-implied paths for  $t = T + 1$  to  $t = T + 40$ . Note that the VECM gives you a path for  $\Delta y_t$  that you have to convert into a path for  $y_t$  (alternatively, you could work with the result derived in (i)).

*Solution.* Forecasts for nominal GDP and investment are generated below. Figure 3 shows that both models forecast (dashed lines) an upward trend, with the VECM making a slightly more aggressive forecast. The following code forecasts the VECM model and VAR model.

---

```
# Forecasts
vecm_forecast = np.vstack([y.values, vecm_res.predict(40)])
var_forecast = np.vstack([y.values, var_res.forecast(y.values, steps=40)])

x = [i for i in range(len(vecm_forecast))]

plt.figure(figsize=(10,4))

# Historical Data
plt.plot(x[:141], var_forecast[:141,1], label='GDP')
```



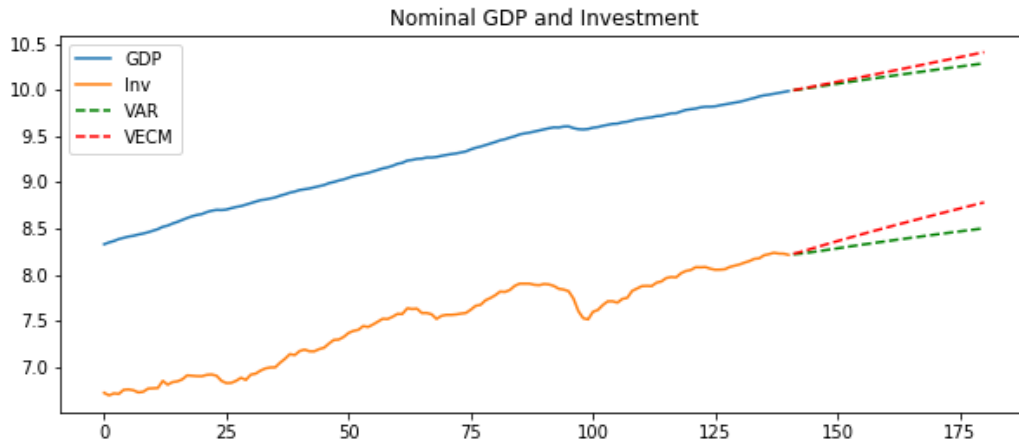


FIGURE 3. VECM and VAR 10 year forecasts for nominal investment and GDP.

```
plt.plot(x[:141], vecm_forecast[:141,0], label='Inv')

# GDP Forecast
plt.plot(x[141:], var_forecast[141:,1], 'g--')
plt.plot(x[141:], vecm_forecast[141:,1], 'r--')

# Investment Forecast
plt.plot(x[141:], var_forecast[141:,0], 'g--', label='VAR')
plt.plot(x[141:], vecm_forecast[141:,0], 'r--', label='VECM')

plt.legend()
plt.title('Nominal GDP and Investment')

plt.show()
```

□

- (vi) Repeat the computations for measures of real investment and real GDP. Why is the ratio of nominal investment and nominal GDP different from the ratio of real investment and real GDP?

*Solution.* Query real investment and GDP data from 1984:Q1 to 2019:Q4 and plot the observed series.

```
# Quarterly U.S. real GDP and real investment from the FRED
ts = web.DataReader(['GPDIC1', 'GDPC1'], 'fred', start, end)

x = pd.DataFrame({'log_inv': np.log(ts['GPDIC1']),
                  'log_gdp': np.log(ts['GDPC1'])})

# Plot data
```

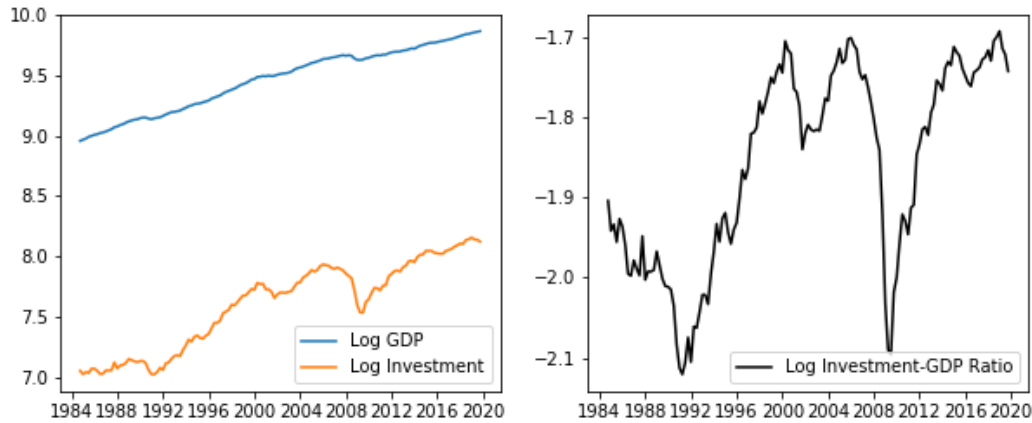


FIGURE 4. Data are in real terms.

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,4))

ax1.plot(x['log_gdp'], label='Log GDP')
ax1.plot(x['log_inv'], label='Log Investment')
ax1.legend()

ax2.plot(x['log_inv'] - x['log_gdp'], 'k', label='Log Investment-GDP Ratio')
ax2.legend()

plt.show()
```

Figure 4 shows log real GDP and log real private domestic investment, as well as the log investment-GDP ratio. Evidently, the aforementioned volatility observed in Figure 2, is less apparent when the series are viewed in real terms. As a result, the log investment-GDP ratio looks less like a white noise process as previously suggested. A possible explanation for this observation could be that inflation is, in some sense, acting as measurement error in the nominal series, resulting in increased noise, and hence making the series look more like a totally random process. Albeit, this speculation is doubtful, especially when viewed in light of the “Slutzky effect” (1937). The following code repeats the above computations for real GDP and real investment.

Fit VECM and VAR model on real investment and GDP.

```
# Real VECM model
rvecm_mod = VECM(x, k_ar_diff=3, deterministic='co',
                  coint_rank=1, freq='QS')
rvecm_res = rvecm_mod.fit()

# Real VAR model
```

```
rvar_mod = VAR(x, freq='QS')
rvar_res = var_mod.fit(4)
```

---

The estimated VECM coefficients  $\hat{\Pi}$  are  $\hat{\Pi}_0 = (-0.286, 0.025)'$ ,

$$\alpha\beta' = \begin{pmatrix} -0.133 & 0.204 \\ 0.003 & -0.005 \end{pmatrix} \quad \hat{\Pi}_1 = \begin{pmatrix} -0.113 & 2.684 \\ -0.002 & -0.319 \end{pmatrix}$$

$$\hat{\Pi}_2 = \begin{pmatrix} 0.060 & 1.094 \\ -0.010 & 0.151 \end{pmatrix} \quad \hat{\Pi}_3 = \begin{pmatrix} 0.044 & -0.727 \\ -0.007 & 0.005 \end{pmatrix}.$$

The estimates  $\hat{\Phi}$  are  $\hat{\Phi}_0 = (-0.2596, 0.0320)'$ ,

$$\hat{\Phi}_1 = \begin{pmatrix} 0.870 & 2.669 \\ 0.004 & 1.285 \end{pmatrix} \quad \hat{\Phi}_2 = \begin{pmatrix} 0.177 & -0.177 \\ 0.013 & -0.160 \end{pmatrix}$$

$$\hat{\Phi}_3 = \begin{pmatrix} -0.009 & -1.814 \\ 0.016 & -0.145 \end{pmatrix} \quad \hat{\Phi}_4 = \begin{pmatrix} -0.080 & 0.818 \\ -0.002 & 0.018 \end{pmatrix}.$$

Compute the VAR coefficients implied by the VECM.

---

```
# VAR representation implied by VECM
rvecm_res.var_rep
```

---

The VAR coefficients implied by the VECM model are

$$\tilde{\Phi}_1 = \begin{pmatrix} 0.069 & 3.172 \\ 0.001 & 1.269 \end{pmatrix} \quad \tilde{\Phi}_2 = \begin{pmatrix} 0.010 & -0.860 \\ 0.004 & -0.005 \end{pmatrix}$$

$$\tilde{\Phi}_3 = \begin{pmatrix} 0.010 & -2.282 \\ -0.001 & -0.023 \end{pmatrix} \quad \tilde{\Phi}_4 = \begin{pmatrix} -0.003 & 0.018 \\ 0.009 & 0.005 \end{pmatrix}.$$

Comparing  $\tilde{\Phi}$  to the estimated VAR coefficients, we can see that the magnitude of differences of these coefficient matrices is less than the previous differences observed in (iii).

Generate 10 year ahead forecast.

---

```
# VECM and VAR forecasts
rvecm_forecast = np.vstack([x.values, rvecm_res.predict(40)])
rvar_forecast = np.vstack([x.values, rvar_res.forecast(x.values, steps=40)])

# Plot results
rx = [i for i in range(len(rvecm_forecast))]

plt.figure(figsize=(10,4))

# Historical data
plt.plot(rx[:141], rvar_forecast[:141,1], label='GDP')
plt.plot(rx[:141], rvecm_forecast[:141,0], label='Inv')

# Investment forecasts
plt.plot(rx[141:], rvar_forecast[141:,0], 'g--', label='VAR')
plt.plot(rx[141:], rvecm_forecast[141:,0], 'r--', label='VECM')

# GDP forecasts
```

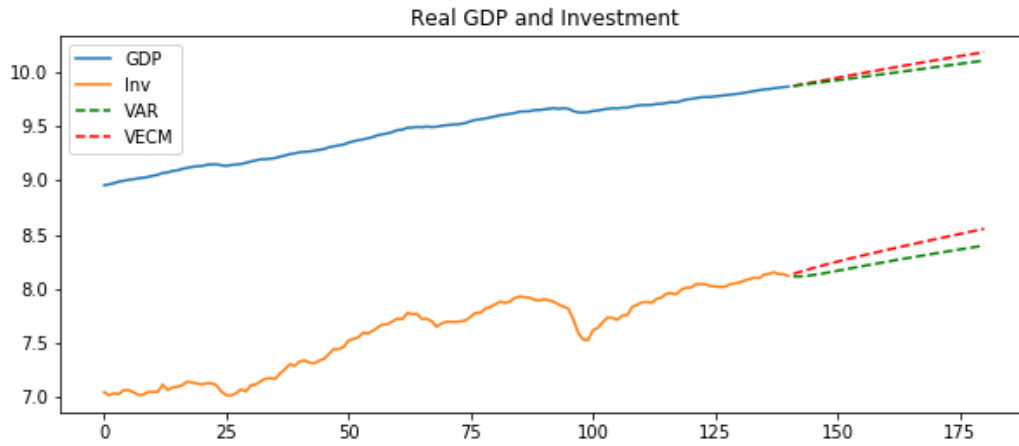


FIGURE 5. VECM and VAR 10 year forecasts for real investment and GDP.

```
plt.plot(rx[141:], rvar_forecast[141:,1], 'r--')
plt.plot(rx[141:], rvecm_forecast[141:,1], 'g--')

plt.legend()
plt.title('Real GDP and Investment')

plt.show()
```

Forecasts for real GDP and investment are given in Figure 5 and both are lower than forecasts for their nominal counterparts. The VECM forecast shows declining growth while the VAR forecasts show constant growth. This again may be related to the earlier speculation that inflation is biasing our nominal estimates.

□