Python is used for computation throughout this homework.

Empirical Exercise 1

1. The series for the RER are constructed as follows.

   Input:

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from stargazer.stargazer import Stargazer

df1 = pd.read_excel("HW2 Price Data.xlsx")

df1 = df1.iloc[4:]
df1.columns = ['Year', 'Quarter', 'P_CHI', 'P_CAN', 'CNY_USD', 'CAD_USD', 'P_US']
df1['Year'] = df1['Year'].astype('int')
df1['Quarter'] = df1['Quarter'].astype('int')
df1['P_CHI'] = df1['P_CHI'].astype('float')
df1['P_CAN'] = df1['P_CAN'].astype('float')
df1['CNY_USD'] = df1['CNY_USD'].astype('float')
df1['CAD_USD'] = df1['CAD_USD'].astype('float')
df1['P_US'] = df1['P_US'].astype('float')

### Question 1 ###

NER_CAN = 1/np.array(df1['CAD_USD'])
P_CAN = np.array(df1['P_CAN'])
NER_CHI = 1/np.array(df1['CNY_USD'])
P_CHI = np.array(df1['P_CHI'])
P_US = np.array(df1['P_US'])

RER_CAN = NER_CAN * P_CAN / P_US
RER_CHI = NER_CHI * P_CHI / P_US

print("RER of Canada against US:")
print(RER_CAN)

print("¥nRER of China against US:")
print(RER_CHI)
```

Output:

```
RER of Canada against US:
[0.85306944 0.80051354 0.79493196 0.81080113 0.77677904 0.79875056
 0.80411048 0.80951394 0.7935749  0.79900401 0.78834241 0.79140295
 0.79467268 0.77191209 0.77795456 0.76825747 0.7460744  0.75212137
 0.71928871 0.69016424 0.7016017  0.71795114 0.71578181 0.71780181
 0.72859197 0.71921482 0.71434794 0.6956592  0.6976373  0.67743912
 0.68698192 0.66643279 0.65734796 0.66552074 0.68481618 0.67034727
 0.68867851 0.72655846 0.76187597 0.7947325  0.81137993 0.78385392
 0.79188681 0.83680796 0.8503111  0.84388932 0.84802885 0.87266273
 0.88795516 0.89750748 0.89930809 0.90290102 0.86464444 0.89497277
 0.95854738 1.02324616 0.98211589 0.98266162 0.97297422 0.84172184]

RER of China against US:
[0.0945573  0.09870715 0.10370546 0.11050564 0.11618861 0.11847148
 0.12065551 0.12272883 0.12544636 0.12702305 0.12624282 0.12725208
 0.12833752 0.12820377 0.12649801 0.12638761 0.12709407 0.12541422
 0.12281688 0.12364973 0.12364717 0.12054423 0.1190686  0.11949389
 0.11986257 0.11681201 0.11535543 0.11657638 0.11670221 0.11485543
 0.11327122 0.11430139 0.11461254 0.11223696 0.11066986 0.11111392
 0.11181708 0.11070039 0.10913669 0.11180199 0.11286204 0.11235545
 0.11179132 0.11148611 0.11255191 0.11100388 0.10975353 0.11153817
 0.11296177 0.11213616 0.1110937  0.11415108 0.11735233 0.11740917
 0.12156136 0.1233414  0.13101876 0.13398336 0.13480253 0.13667208]
```

2. The growth rates of NER are constructed by the following code.

   Input:

```
### Question 2 ###

n = len(NER_CAN)

NER_CAN_1 = NER_CAN[4:n]/NER_CAN[0:(n-4)]
NER_CAN_2 = NER_CAN[8:n]/NER_CAN[0:(n-8)]
NER_CAN_3 = NER_CAN[12:n]/NER_CAN[0:(n-12)]
NER_CAN_4 = NER_CAN[16:n]/NER_CAN[0:(n-16)]
NER_CAN_5 = NER_CAN[20:n]/NER_CAN[0:(n-20)]

NER_CHI_1 = NER_CHI[4:n]/NER_CHI[0:(n-4)]
NER_CHI_2 = NER_CHI[8:n]/NER_CHI[0:(n-8)]
NER_CHI_3 = NER_CHI[12:n]/NER_CHI[0:(n-12)]
NER_CHI_4 = NER_CHI[16:n]/NER_CHI[0:(n-16)]
NER_CHI_5 = NER_CHI[20:n]/NER_CHI[0:(n-20)]
```

3. The results of each regression are shown below.

   Input:

```
### Question 3 ###

LNER_CAN_1 = np.log(NER_CAN_1)
LNER_CAN_2 = np.log(NER_CAN_2)
LNER_CAN_3 = np.log(NER_CAN_3)
```

```
LNER_CAN_4 = np.log(NER_CAN_4)
LNER_CAN_5 = np.log(NER_CAN_5)

LNER_CHI_1 = np.log(NER_CHI_1)
LNER_CHI_2 = np.log(NER_CHI_2)
LNER_CHI_3 = np.log(NER_CHI_3)
LNER_CHI_4 = np.log(NER_CHI_4)
LNER_CHI_5 = np.log(NER_CHI_5)

LRER_CAN = np.log(RER_CAN)
LRER_CHI = np.log(RER_CHI)

d_CAN_1 = {"x": LRER_CAN[0:(n-4)], "y": LNER_CAN_1}
df_CAN_1 = pd.DataFrame(d_CAN_1)
model_CAN_1 = sm.OLS(df_CAN_1['y'], sm.add_constant(df_CAN_1['x']))
results_CAN_1 = model_CAN_1.fit()

d_CAN_2 = {"x": LRER_CAN[0:(n-8)], "y": LNER_CAN_2}
df_CAN_2 = pd.DataFrame(d_CAN_2)
model_CAN_2 = sm.OLS(df_CAN_2['y'], sm.add_constant(df_CAN_2['x']))
results_CAN_2 = model_CAN_2.fit()

d_CAN_3 = {"x": LRER_CAN[0:(n-12)], "y": LNER_CAN_3}
df_CAN_3 = pd.DataFrame(d_CAN_3)
model_CAN_3 = sm.OLS(df_CAN_3['y'], sm.add_constant(df_CAN_3['x']))
results_CAN_3 = model_CAN_3.fit()

d_CAN_4 = {"x": LRER_CAN[0:(n-16)], "y": LNER_CAN_4}
df_CAN_4 = pd.DataFrame(d_CAN_4)
model_CAN_4 = sm.OLS(df_CAN_4['y'], sm.add_constant(df_CAN_4['x']))
results_CAN_4 = model_CAN_4.fit()

d_CAN_5 = {"x": LRER_CAN[0:(n-20)], "y": LNER_CAN_5}
df_CAN_5 = pd.DataFrame(d_CAN_5)
model_CAN_5 = sm.OLS(df_CAN_5['y'], sm.add_constant(df_CAN_5['x']))
results_CAN_5 = model_CAN_5.fit()

stargazer_CAN = Stargazer([results_CAN_1, results_CAN_2,
                          results_CAN_3, results_CAN_4,
                          results_CAN_5])
stargazer_CAN.title('Canada')
stargazer_CAN.custom_columns(['h = 1', 'h = 2', 'h = 3', 'h = 4', 'h = 5'],
                           [1,1,1,1,1])
stargazer_CAN.show_model_numbers(False)
stargazer_CAN.dependent_variable_name('log(gNER)')
stargazer_CAN.rename_covariates({'x': 'log(RER)'})
stargazer_CAN
```

Output:

Canada

					log(gNER)y
	h = 1	h = 2	h = 3	h = 4	h = 5
const	0.008	0.006	-0.096	-0.326^{***}	-0.637^{***}
	(0.026)	(0.045)	(0.070)	(0.087)	(0.069)
log(RER)	-0.042	-0.120	-0.543^{**}	-1.369^{***}	-2.438^{***}
	(0.094)	(0.160)	(0.241)	(0.291)	(0.224)
Observations	56	52	48	44	40
R^2	0.004	0.011	0.100	0.345	0.757
Adjusted R^2	-0.015	-0.009	0.080	0.330	0.751
Residual Std. Error	0.070 (df=54)	0.100 (df=50)	0.128 (df=46)	0.136 (df=42)	0.098 (df=38)
F Statistic	0.197 (df=1; 54)	0.569 (df=1; 50)	5.086^{**} (df=1; 46)	22.166^{***} (df=1; 42)	118.684^{***} (df=1; 38)
Note:					$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Input:

```
d_CHI_1 = {"x": LRER_CHI[0:(n-4)], "y": LNER_CHI_1}
df_CHI_1 = pd.DataFrame(d_CHI_1)
model_CHI_1 = sm.OLS(df_CHI_1['y'], sm.add_constant(df_CHI_1['x']))
results_CHI_1 = model_CHI_1.fit()

d_CHI_2 = {"x": LRER_CHI[0:(n-8)], "y": LNER_CHI_2}
df_CHI_2 = pd.DataFrame(d_CHI_2)
model_CHI_2 = sm.OLS(df_CHI_2['y'], sm.add_constant(df_CHI_2['x']))
results_CHI_2 = model_CHI_2.fit()

d_CHI_3 = {"x": LRER_CHI[0:(n-12)], "y": LNER_CHI_3}
df_CHI_3 = pd.DataFrame(d_CHI_3)
model_CHI_3 = sm.OLS(df_CHI_3['y'], sm.add_constant(df_CHI_3['x']))
results_CHI_3 = model_CHI_3.fit()

d_CHI_4 = {"x": LRER_CHI[0:(n-16)], "y": LNER_CHI_4}
df_CHI_4 = pd.DataFrame(d_CHI_4)
model_CHI_4 = sm.OLS(df_CHI_4['y'], sm.add_constant(df_CHI_4['x']))
results_CHI_4 = model_CHI_4.fit()

d_CHI_5 = {"x": LRER_CHI[0:(n-20)], "y": LNER_CHI_5}
df_CHI_5 = pd.DataFrame(d_CHI_5)
```

```
model_CHI_5 = sm.OLS(df_CHI_5['y'], sm.add_constant(df_CHI_5['x']))
results_CHI_5 = model_CHI_5.fit()

stargazer_CHI = Stargazer([results_CHI_1, results_CHI_2,
                           results_CHI_3, results_CHI_4,
                           results_CHI_5])
stargazer_CHI.title('China')
stargazer_CHI.custom_columns(['h = 1', 'h = 2', 'h = 3', 'h = 4', 'h = 5'],
                             [1,1,1,1,1])
stargazer_CHI.show_model_numbers(False)
stargazer_CHI.dependent_variable_name('log(gNER)')
stargazer_CHI.rename_covariates({'x': 'log(RER)'})
stargazer_CHI
```

Output:

China

					log(gNER)y
	h = 1	h = 2	h = 3	h = 4	h = 5
const	-0.136	-0.502***	-0.665***	-0.731***	-0.850***
	(0.119)	(0.170)	(0.202)	(0.217)	(0.215)
log(RER)	-0.070	-0.244***	-0.323***	-0.356***	-0.414***
	(0.055)	(0.079)	(0.094)	(0.101)	(0.100)
Observations	56	52	48	44	40
R^2	0.029	0.161	0.205	0.228	0.310
Adjusted R^2	0.011	0.144	0.187	0.210	0.291
Residual Std. Error	0.026 (df=54)	0.036 (df=50)	0.043 (df=46)	0.045 (df=42)	0.044 (df=38)
F Statistic	1.622 (df=1; 54)	9.599*** (df=1; 50)	11.839*** (df=1; 46)	12.414*** (df=1; 42)	17.041*** (df=1; 38)
Note:					*p<0.1; **p<0.05; ***p<0.01

4. For both Canada and China, the estimate of $\beta_{i,h}^{NER}$ is larger (in the absolute value) and more significant in the long run than in the short run. These results imply that RER adjusts to shock slowly but does not adjust immediately.
The negative coefficients suggest that NER would change so that RER would converge to a certain level in the long run. Thus, the convergence of RER is caused by the changes in NER rather than the changes in prices.

5. Although the estimates of $\beta_{i,h}^{NER}$ are significantly negative for both Canada and China, the one for Canada is larger than the one for China in the absolute value. Since China is managing the yuan exchange rate, the change in NER is not so large relative to Canada, which has a floating exchange rate regime.

Empirical Exercise 2

1. The predict rates of CPI inflation are summarized in Table 2-1, followed by the source code. It should be noted that the countries with relatively high import content in consumption (e.g., Korea and Thailand) would suffer from a large inflation due to the depreciation of the home currency under the reasonable assumption $\beta^b > \beta^l$.

Table 2-1: Predicted rates of CPI inflation by country

	Argentina	Brazil	Korea	Mexico	Thailand
High & High ($\beta^b = 1, \beta^l = 1$)	50.0%	50.0%	50.0%	50.0%	50.0%
High & Low ($\beta^b = 1, \beta^l = 0.1$)	9.7%	9.0%	14.3%	9.9%	14.3%
Low & Low ($\beta^b = 0.5, \beta^l = 0.1$)	7.1%	6.8%	9.1%	7.2%	9.1%

Input:

```
import numpy as np
import pandas as pd

# Create the dataframe
df2 = pd.DataFrame({'Argentina': [0.530, 0.198], 'Brazil': [0.593, 0.150],
                    'Korea': [0.480, 0.429], 'Mexico': [0.535, 0.203],
                    'Thailand': [0.433, 0.478]})

### Question 1 ###

# Define variables
e = 50
sT = np.array(df2.iloc[[0]])
sM = np.array(df2.iloc[[1]])
sTsM = sT * sM
beta_b = np.array([1, 1, 0.5]).reshape([-1, 1])
beta_l = np.array([1, 0.1, 0.1]).reshape([-1, 1])

# Calculate the inflation
cpi2 = (beta_b @ sTsM + beta_l @ (1 - sTsM)) * e
cpi2_df = pd.DataFrame(cpi2, columns = df2.columns,
                     index = ['High & High', 'High & Low',
                              'Low & Low'])
print(cpi2_df)
```

Output:

```
            Argentina    Brazil     Korea     Mexico   Thailand
High & High   50.0000  50.00000  50.0000  50.000000  50.00000
High & Low     9.7223   9.00275  14.2664   9.887225  14.31383
Low & Low      7.0988   6.77900   9.1184   7.172100   9.13948
```

2. The predict rates of CPI inflation for each income group are summarized in Tables 2-2 and 2-3, followed by the source code. Since low-income group is assumed to consume more import goods than high-income group (which is an empirically reasonable assumption), the low-income group would suffer from a more severe inflation due to the depreciation of the home currency. Our model implies that the depreciation of the home currency could increase the level of income inequality due to the differences in consumption bundles across income levels.

Table 2-2: Predicted rates of CPI inflation *for high-income group* by country

	Argentina	Brazil	Korea	Mexico	Thailand
High & High ($\beta^b = 1, \beta^l = 1$)	50.0%	50.0%	50.0%	50.0%	50.0%
High & Low ($\beta^b = 1, \beta^l = 0.1$)	9.7%	9.0%	14.3%	9.9%	14.3%
Low & Low ($\beta^b = 0.5, \beta^l = 0.1$)	7.1%	6.8%	9.1%	7.2%	9.1%

Table 2-3: Predicted rates of CPI inflation *for low-income group* by country

	Argentina	Brazil	Korea	Mexico	Thailand
High & High ($\beta^b = 1, \beta^l = 1$)	50.0%	50.0%	50.0%	50.0%	50.0%
High & Low ($\beta^b = 1, \beta^l = 0.1$)	14.2%	13.5%	18.8%	14.4%	18.8%
Low & Low ($\beta^b = 0.5, \beta^l = 0.1$)	9.1%	8.8%	11.1%	9.2%	11.1%

Input:

```
### Question 2 ###

# Define the import share
sTsM_H = sTsM          # High income
sTsM_L = sTsM + 0.10   # Low income

# Calculate the inflation for high income group
cpi2_H = (beta_b @ sTsM_H + beta_l @ (1 - sTsM_H)) * e
cpi2_H_df = pd.DataFrame(cpi2_H, columns = df2.columns,
                      index = ['High & High', 'High & Low',
                               'Low & Low'])
```

```python
print('High Income Group:')
print(cpi2_H_df)
```

Output:

```
High Income Group:
             Argentina    Brazil    Korea     Mexico  Thailand
High & High    50.0000  50.00000  50.0000  50.000000  50.00000
High & Low      9.7223   9.00275  14.2664   9.887225  14.31383
Low & Low       7.0988   6.77900   9.1184   7.172100   9.13948
```

Input:

```python
# Calculate the inflation for low income group
cpi2_L = (beta_b @ sTsM_L + beta_l @ (1 - sTsM_L)) * e
cpi2_L_df = pd.DataFrame(cpi2_L, columns = df2.columns,
                         index = ['High & High', 'High & Low',
                                  'Low & Low'])
print('Low Income Group:')
print(cpi2_L_df)
```

Output:

```
Low Income Group:
             Argentina    Brazil    Korea     Mexico  Thailand
High & High    50.0000  50.00000  50.0000  50.000000  50.00000
High & Low     14.2223  13.50275  18.7664  14.387225  18.81383
Low & Low       9.0988   8.77900  11.1184   9.172100  11.13948
```

Theoretical Exercise 1

1.

(a) The profit maximization problem is

$$\max_{P_{in}}(P_{in} - C_{in})Q_{in} = \max_{P_{in}}(P_{in} - C_{in})A_{in}\left(\frac{P_{in}}{P_n}\right)^{-\varepsilon} Q_n.$$

Since A_{in}, P_n and Q_n do not depend on P_{in}, it is equivalent to

$$\max_{P_{in}}(P_{in} - C_{in})P_{in}^{-\varepsilon}.$$

The FOC gives us

$$P_{in}^{-\varepsilon} - \varepsilon(P_{in} - C_{in})P_{in}^{-\varepsilon-1} = 0.$$

Solving it, we obtain

$$\boldsymbol{P_{in} = \frac{\varepsilon}{\varepsilon - 1} C_{in}}.$$

(b)

i. When $E_{in} = 1$, we obtain $\boldsymbol{C_{in}} = W_i E_{in} = \mathbf{20}$. After the appreciation of the Euro, the marginal cost becomes $\boldsymbol{C_{in}} = W_i E_{in} = 20(1.2) = \mathbf{24}$.

ii. Before the Euro appreciation, the price is given by $\boldsymbol{P_{in}} = \frac{\varepsilon}{\varepsilon-1} C_{in} = \frac{4}{3}(20) \simeq \mathbf{26.7}$. After the appreciation, it is $\boldsymbol{P_{in}} = \frac{\varepsilon}{\varepsilon-1} C_{in} = \frac{4}{3}(24) = \mathbf{32}$.

(c)

i. By this definition, the exchange-rate pass-through is given by $\Delta P_{in}/\Delta C_{in} \simeq (32 - 26.7)/(24 - 20) \simeq \mathbf{1.33}$.

ii. By this definition, the exchange-rate pass-through is given by $\%\Delta P_{in}/\%\Delta C_{in} = \Delta \log P_{in}/\Delta \log C_{in} = \mathbf{1}$ since $\log P_{in} = \log\left(\frac{\varepsilon}{\varepsilon-1}\right) + \log C_{in}$ and $\log\left(\frac{\varepsilon}{\varepsilon-1}\right)$ is a constant.

2.

(a) The profit maximization problem becomes

$$\max_{P_{in}}(P_{in} - C_{in})Q_{in} = \max_{P_{in}}(P_{in} - C_{in})A_{in}\left(\frac{P_{in}^r}{P_n}\right)^{-\varepsilon} Q_n.$$

Since A_{in}, P_n and Q_n do not depend on P_{in}, it is equivalent to

$$\max_{P_{in}}(P_{in} - C_{in})(P_{in}^r)^{-\varepsilon} = \max_{P_{in}}(P_{in} - C_{in})(P_{in} + \eta_{in}P_n^d)^{-\varepsilon}.$$

The FOC gives us

$$(P_{in} + \eta_{in}P_n^d)^{-\varepsilon} - \varepsilon(P_{in} - C_{in})(P_{in} + \eta_{in}P_n^d)^{-\varepsilon-1} = 0.$$

Solving it, we obtain

$$\boldsymbol{P_{in} = \frac{\varepsilon}{\varepsilon - 1} C_{in} + \frac{1}{\varepsilon - 1}\eta_{in}P_n^d}.$$

(b) The profit under the optimal price is given by

$$\Pi_{in} = (P_{in} - C_{in})A_{in}\left(\frac{P_{in}^r}{P_n}\right)^{-\varepsilon} Q_n$$
$$= \left(\frac{\varepsilon}{\varepsilon-1}C_{in} + \frac{1}{\varepsilon-1}\eta_{in}P_n^d - C_{in}\right)A_{in}\left(\frac{\frac{\varepsilon}{\varepsilon-1}C_{in} + \frac{1}{\varepsilon-1}\eta_{in}P_n^d + \eta_{in}P_n^d}{P_n}\right)^{-\varepsilon} Q_n$$
$$= \frac{1}{\varepsilon-1}\left(\frac{\varepsilon}{\varepsilon-1}\right)^{-\varepsilon}(C_{in} + \eta_{in}P_n^d)^{-\varepsilon+1}\frac{A_{in}Q_n}{P_n^{-\varepsilon}}.$$

When $Q_n = P_n = A_{in} = 1$, we obtain

$$\Pi_{in} = \frac{1}{\varepsilon-1}\left(\frac{\varepsilon}{\varepsilon-1}\right)^{-\varepsilon}(C_{in} + \eta_{in}P_n^d)^{-\varepsilon+1}.$$

Under the pricing rule in (1a), the profit is given by

$$\Pi_{in}^{(1a)} = \frac{1}{\varepsilon-1}\left(\frac{\varepsilon}{\varepsilon-1}\right)^{-\varepsilon} C_{in}{}^{-\varepsilon+1}.$$

Under the pricing rule in (2a), it is

$$\Pi_{in}^{(2a)} = \frac{1}{\varepsilon-1}\left(\frac{\varepsilon}{\varepsilon-1}\right)^{-\varepsilon}(C_{in} + 15)^{-\varepsilon+1}$$

Since $\varepsilon > 1$, it holds that $C_{in}{}^{-\varepsilon+1} > (C_{in} + 15)^{-\varepsilon+1}$. Hence, $\Pi_{in}^{(2a)} > \Pi_{in}^{(2b)}$, i.e., the profit under the pricing rule in (2a) is smaller than the profit under (1a). Since the retail price includes the price of distribution services under the assumption of (2a), the demand must be lower than (1a), which declines the profit.

(c)

i. The marginal cost does not change from (1b). That is, when $E_{in} = 1$, we obtain $\boldsymbol{C_{in}} = W_i E_{in} = \mathbf{20}$. After the appreciation of the Euro, the marginal cost becomes $\boldsymbol{C_{in}} = W_i E_{in} = 20(1.2) = \mathbf{24}$.

ii. The retail price is given by $P_{in}^r = P_{in} + \eta_{in}P_n^d = \frac{\varepsilon}{\varepsilon-1}C_{in} + \frac{\varepsilon}{\varepsilon-1}\eta_{in}P_n^d$ now. Thus, before the Euro appreciation, the price is $\boldsymbol{P_{in}^r} = \frac{4}{3}(20) + \frac{4}{3}(15) \simeq \mathbf{46.7}$. After the appreciation, it is $\boldsymbol{P_{in}^r} = \frac{4}{3}(24) + \frac{4}{3}(15) = \mathbf{52}$.

(d)

i. By this definition, the exchange-rate pass-through is given by $\Delta P_{in}^{r}/\Delta C_{in} \simeq (52-46.7)/(24-20) \simeq \mathbf{1.33}$.

ii. By this definition, the exchange-rate pass-through is given by $\%\Delta P_{in}^{r}/\%\Delta C_{in} = ((52-46.7)/46.7)/((24-20)/20) \simeq \mathbf{0.57}$. Analytically, it can be computed as $\Delta \log P_{in}/\Delta \log C_{in} = \Delta \log\left(\frac{\varepsilon}{\varepsilon-1} C_{in} + \frac{\varepsilon}{\varepsilon-1}\eta_{in} P_n^d\right)/\Delta \log C_{in} = \left(\frac{\varepsilon}{\varepsilon-1} C_{in}\right)/\left(\frac{\varepsilon}{\varepsilon-1} C_{in} + \frac{\varepsilon}{\varepsilon-1}\eta_{in} P_n^d\right) = C_{in}/(C_{in} + \eta_{in} P_n^d) = 20/(20+15) \simeq 0.57$.

(e) **The *percentage* pass-through rate is lower in (2d) than in (1c)**. As is shown by (2d), the percentage pass-through rate is equal to the ratio of exchange-rate-related marginal production cost (C_{in}) over the total marginal cost ($C_{in} + \eta_{in} P_n^d$). It can be understood intuitively because the exchange rate does not affect the distribution cost $\eta_{in} P_n^d$ but affect only the production cost C_{in}. Thus, the larger is the distribution cost relative to the production cost, the lower will be the effect of the Euro appreciation on the retail price.

Theoretical Exercise 2

1. Since $\theta = 5$ by assumption, the equilibrium prices P_{in} $(i = 1,2)$ satisfy the following system of equations:

$$P_{in} = \frac{s_{in} + 5(1-s_{in})}{s_{in} + 5(1-s_{in}) - 1} C_{in} = \frac{5-4s_{in}}{4-4s_{in}} C_{in} \quad (i = 1,2) \tag{4}$$

$$s_{in} = \frac{P_{in}^{-4}}{P_{1n}^{-4} + P_{2n}^{-4}} = \frac{P_{jn}^{4}}{P_{1n}^{4} + P_{2n}^{4}} \quad \big((i,j) = (1,2),(2,1)\big) \tag{5}$$

Substituting (5) into (4) and rearranging, we obtain

$$\begin{cases} 4P_{1n}^5 = C_{1n}(5P_{1n}^4 + P_{2n}^4) \\ 4P_{2n}^5 = C_{2n}(5P_{2n}^4 + P_{1n}^4) \end{cases} \tag{6}$$

We cannot solve (6) analytically in general, so the solution should be obtained by a numerical method. When $C_{1n} = C_{2n} = 1.5$, we obtain the equilibrium prices $\boldsymbol{P_{1n} = P_{2n} = 2.250}$ as is shown by the following code. Although $P_{1n} = P_{2n} = 0$ satisfies (6) for any C_{1n} and C_{2n}, it cannot be a solution because the right-hand side of (5) cannot be defined.

In this case, we can solve (6) analytically as well because $C_{1n} = C_{2n}$ implies $P_{1n} = P_{2n}$. Substituting $C_{1n} = C_{2n} = 1.5$ and $P_{1n} = P_{2n}$ into (6), we obtain $4P_{1n}^5 = 1.5(6P_{1n}^4)$, which yields $P_{1n} = P_{2n} = 2.25$ under the assumption $P_{1n} > 0$.

Input:

```
from scipy.optimize import fsolve

def equations1(vars):
    P1, P2 = vars
    C1 = 1.5; C2 = 1.5
    eq1 = 4*P1**5 - C1*(5*P1**4 + P2**4)
    eq2 = 4*P2**5 - C2*(5*P2**4 + P1**4)
    return [eq1, eq2]

P1_1, P2_1 =  fsolve(equations1, (2, 2))
print("Price of Product 1: {:.3f}".format(P1_1))
print("Price of Product 2: {:.3f}".format(P2_1))
```

Output:

```
Price of Product 1: 2.250
Price of Product 2: 2.250
```

2. If $C_{1n} = 1$ and $C_{2n} = 2$, the equilibrium price levels are given by $\boldsymbol{P_{1n} \simeq 2.014}$ and $\boldsymbol{P_{2n} \simeq 2.664}$. Thus, the markup is $\boldsymbol{P_{1n}/C_{1n} \simeq 2.014}$ for product 1 and $\boldsymbol{P_{2n}/C_{2n} \simeq 1.332}$ for product 2. The market share of product 1 is $\boldsymbol{s_{1n} \simeq 0.754}$ while the one of product 2 is $\boldsymbol{s_{2n} \simeq 0.246}$.
In the equilibrium, **the levels of markup and market share of firm 1 are higher than those of firm 2**. Intuitively, since firm 1 is relatively competitive in terms of the cost, firm 1 can sell its product at lower price than firm 2, which makes the market share of firm 1 higher than firm 2. Since the high market share leads to the low elasticity of the product, firm 1 can raise the price of product 1 without lowering the demand very much. That explains the reason why the level of markup of firm 1 is higher than that of firm 2 intuitively.

Input:

```
def equations2(vars):
    P1, P2 = vars
    C1 = 1; C2 = 2
    eq1 = 4*P1**5 - C1*(5*P1**4 + P2**4)
    eq2 = 4*P2**5 - C2*(5*P2**4 + P1**4)
    return [eq1, eq2]

P1_2, P2_2 =  fsolve(equations2, (P1_1, P2_1))
print("Price of Product 1: {:.3f}".format(P1_2))
print("Price of Product 2: {:.3f}".format(P2_2))

C1_2 = 1; C2_2 = 2

M1_2 = P1_2/C1_2
M2_2 = P2_2/C2_2
```

```python
print("Markup of Product 1: {:.3f}".format(M1_2))
print("Markup of Product 2: {:.3f}".format(M2_2))

s1_2 = (P2_2**4)/(P1_2**4 + P2_2**4)
s2_2 = (P1_2**4)/(P1_2**4 + P2_2**4)

print("Market Share of Product 1: {:.3f}".format(s1_2))
print("Market Share of Product 2: {:.3f}".format(s2_2))
```

Output:

```
Price of Product 1: 2.014
Price of Product 2: 2.664
Markup of Product 1: 2.014
Markup of Product 2: 1.332
Market Share of Product 1: 0.754
Market Share of Product 2: 0.246
```

3. If $C_{1n} = 1.2$ and $C_{2n} = 2$, the equilibrium price levels are given by $\boldsymbol{P_{1n} \simeq 2.198}$ and $\boldsymbol{P_{2n} \simeq 2.715}$. Thus, the markup is $\boldsymbol{P_{1n}/C_{1n} \simeq 1.832}$ for product 1 and $\boldsymbol{P_{2n}/C_{2n} \simeq 1.357}$ for product 2.
The rate of cost pass-through is defined as the percentage change in the price in response to 1% change in the cost. In this scenario, **the cost of product 1 goes up by 20%, so its rate of cost pass-through is given by** $((2.198 - 2.014)/2.014)/20\% \simeq \mathbf{0.456}$. The rate of cost pass-through is lower than 1 because firm 1 will not increase the price as much as the cost rise so that it would not lose its market share very much. Actually, the market share of firm 1 is about 0.699 after the cost rise—it is lower than before but not that much lower. **Since the cost of product 2 does not rise in this scenario, we cannot compute the rate of firm 2.** We can still see that the markup of firm 2 goes up as the cost of product 1 increases. The cost rise of product 1 will force firm 1 to increase its price, which makes more room for firm 2 to raise the price of its product. Since the cost of product 2 does not change, the price rise will increase the markup of firm 2.

   Let us assume another situation where the cost of both products increases by 20%. As is shown in the following code, the markups of both firms remain exactly the same as before. Thus, the cost pass-through rates are given by 1 for both firms. Since both firms are facing with the identical cost shock and they know that fact, they would not worry about the market share loss and would raise the price as much as the cost rise.

Input:

```python
# New equilibrium

def equations3(vars):
    P1, P2 = vars
    C1 = 1.2; C2 = 2
    eq1 = 4*P1**5 - C1*(5*P1**4 + P2**4)
    eq2 = 4*P2**5 - C2*(5*P2**4 + P1**4)
    return [eq1, eq2]
```

```
P1_3, P2_3 =  fsolve(equations3, (P1_2, P2_2))
print("Price of Product 1: {:.3f}".format(P1_3))
print("Price of Product 2: {:.3f}".format(P2_3))

C1_3 = 1.2; C2_3 = 2

M1_3 = P1_3/C1_3
M2_3 = P2_3/C2_3

print("Markup of Product 1: {:.3f}".format(M1_3))
print("Markup of Product 2: {:.3f}".format(M2_3))

CPT1 = ((P1_3 - P1_2)/P1_2)/((C1_3 - C1_2)/C1_2)

print("Rate of Cost Pass-through of Product 1: {:.3f}".format(CPT1))

s1_3 = (P2_3**4)/(P1_3**4 + P2_3**4)
s2_3 = (P1_3**4)/(P1_3**4 + P2_3**4)

print("Market Share of Product 1: {:.3f}".format(s1_3))
print("Market Share of Product 2: {:.3f}".format(s2_3))
```

Output:

```
Price of Product 1: 2.198
Price of Product 2: 2.715
Markup of Product 1: 1.832
Markup of Product 2: 1.357
Rate of Cost Pass-through of Product 1: 0.456
Market Share of Product 1: 0.699
Market Share of Product 2: 0.301
```

Input:

```
# If both costs go up

def equations4(vars):
    P1, P2 = vars
    C1 = 1.2; C2 = 2.4
    eq1 = 4*P1**5 - C1*(5*P1**4 + P2**4)
    eq2 = 4*P2**5 - C2*(5*P2**4 + P1**4)
    return [eq1, eq2]

P1_4, P2_4 =  fsolve(equations4, (P1_2, P2_2))
print("Price of Product 1: {:.3f}".format(P1_3))
print("Price of Product 2: {:.3f}".format(P2_3))

C1_4 = 1.2; C2_4 = 2.4

M1_4 = P1_4/C1_4
M2_4 = P2_4/C2_4

print("Markup of Product 1: {:.3f}".format(M1_4))
```

```
print("Markup of Product 2: {:.3f}".format(M2_4))

CPT1_2 = ((P1_4 - P1_2)/P1_2)/((C1_4 - C1_2)/C1_2)
CPT2_2 = ((P2_4 - P2_2)/P2_2)/((C2_4 - C2_2)/C2_2)

print("Rate of Cost Pass-through of Product 1: {:.3f}".format(CPT1_2))
print("Rate of Cost Pass-through of Product 2: {:.3f}".format(CPT2_2))

s1_4 = (P2_4**4)/(P1_4**4 + P2_4**4)
s2_4 = (P1_4**4)/(P1_4**4 + P2_4**4)

print("Market Share of Product 1: {:.3f}".format(s1_4))
print("Market Share of Product 2: {:.3f}".format(s2_4))
```

Output:

```
Price of Product 1: 2.198
Price of Product 2: 2.715
Markup of Product 1: 2.014
Markup of Product 2: 1.332
Rate of Cost Pass-through of Product 1: 1.000
Rate of Cost Pass-through of Product 2: 1.000
Market Share of Product 1: 0.754
Market Share of Product 2: 0.246
```