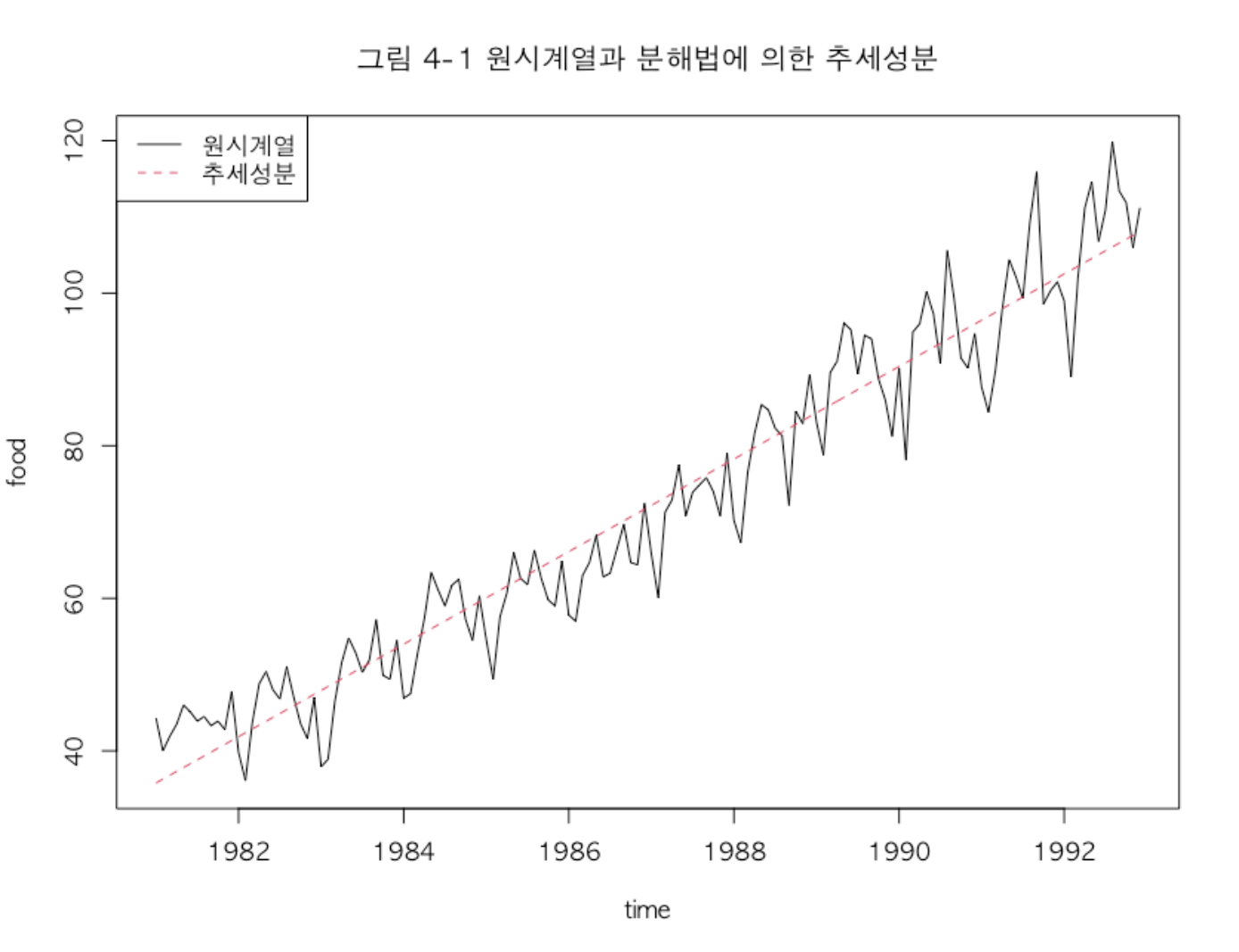
HW 04

2021234640 이종현

Table of Contents

Figure 4-1

아래 원시계열이 선형 추세를 따르고 있기 때문에 선형 추세를 먼저 fiting한 결과는 아래와 같다. 이후 계절 성분을 fitting하면 될 듯하다.




```
> anova(fit)
Analysis of Variance Table

Response: food
           Df Sum Sq Mean Sq F value    Pr(>F)
t           1  63600   63600  1954.9 < 2.2e-16 ***
Residuals 142   4620      33
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

선형 추세가 잘 fitting 되었음을 볼 수 있다. 이제 잔차에 대해 계절 성분을 피팅해보자. 또한 분산이 증가하는 경향을 보이고 있기 때문에 가법 모형 보다는 승법 모형이 적절하다. 때문에 추세를 조정한 시계열은 기존 시계열에서 선형 추세를 나누는 형태로 계산된다.

Figure 4-2

계절 성분을 추정함에 있어 bias 없는 12개의 계절 성분을 dummy 변수로 하여 추정한 모델은 DW test 에서 자기 상관이 있음이 확인되었다.

```
> temp_lm = lm(adjtrend ~ y + 0)
> dwtest(temp_lm)

    Durbin-Watson test

data:  temp_lm
DW = 0.58482, p-value = 2.416e-16
alternative hypothesis: true autocorrelation is greater than 0
```

때문에 자기회귀오차모형을 이용하여 계절 성분을 fitting한다.

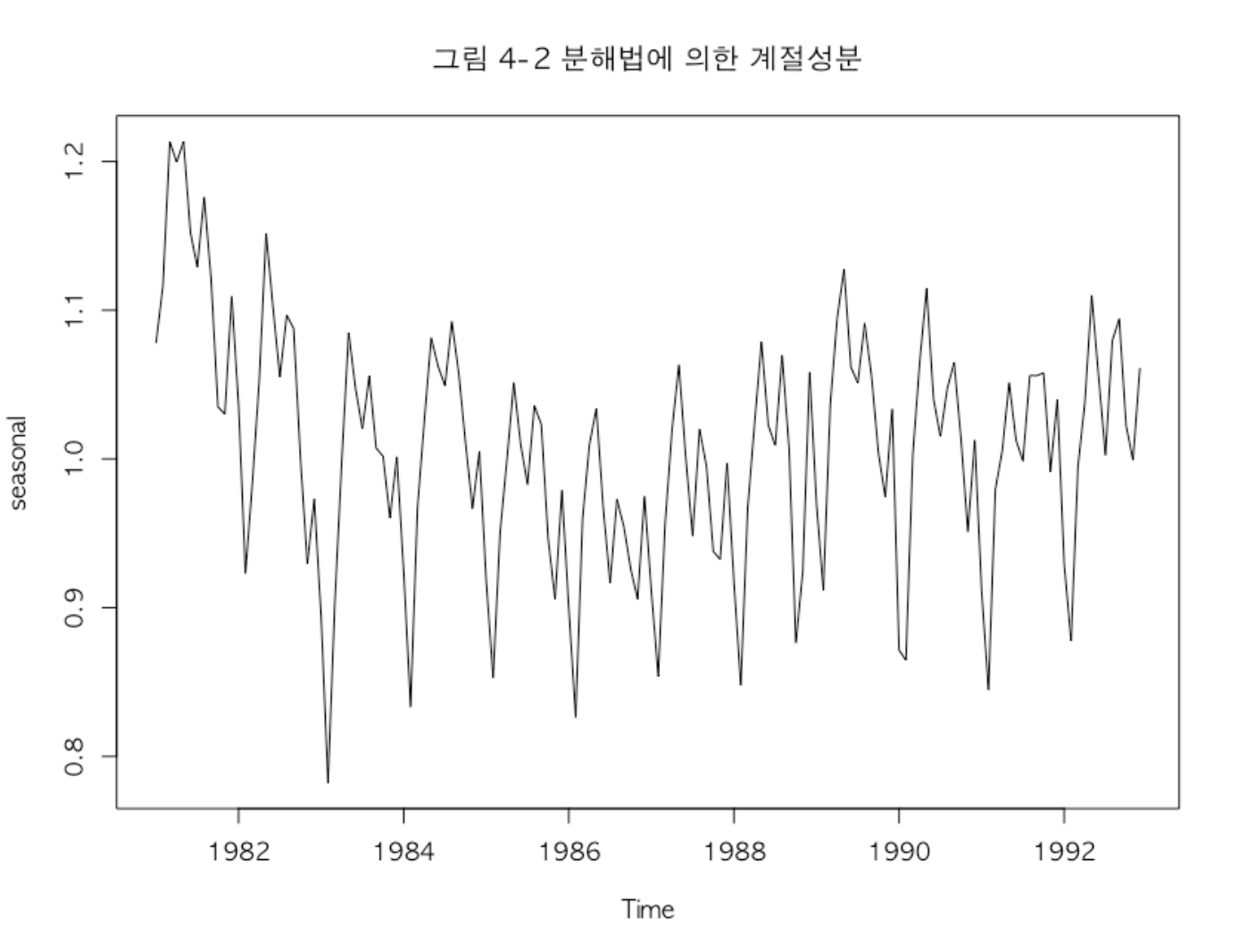
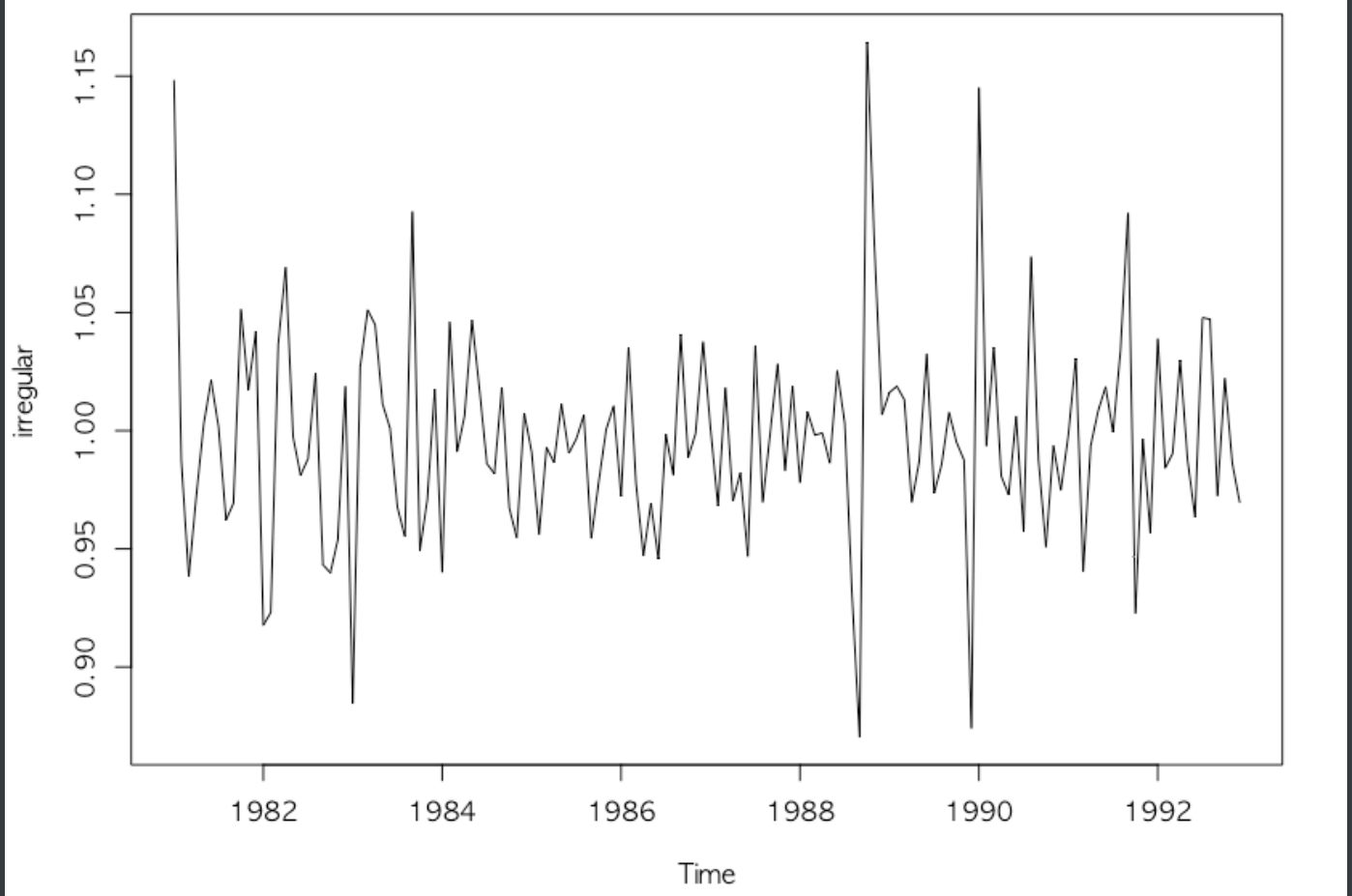


Figure 4-3

승법 모형이기 때문에 잔차는 Z / (trend * seaonal) 로 계산할 수 있다.

그림 4-3 분해법에 의한 불규칙성분



잔차에 대해 ACF 를 확인해본 결과, 자기상관성이 크게 제거되었음을 확인하였다.

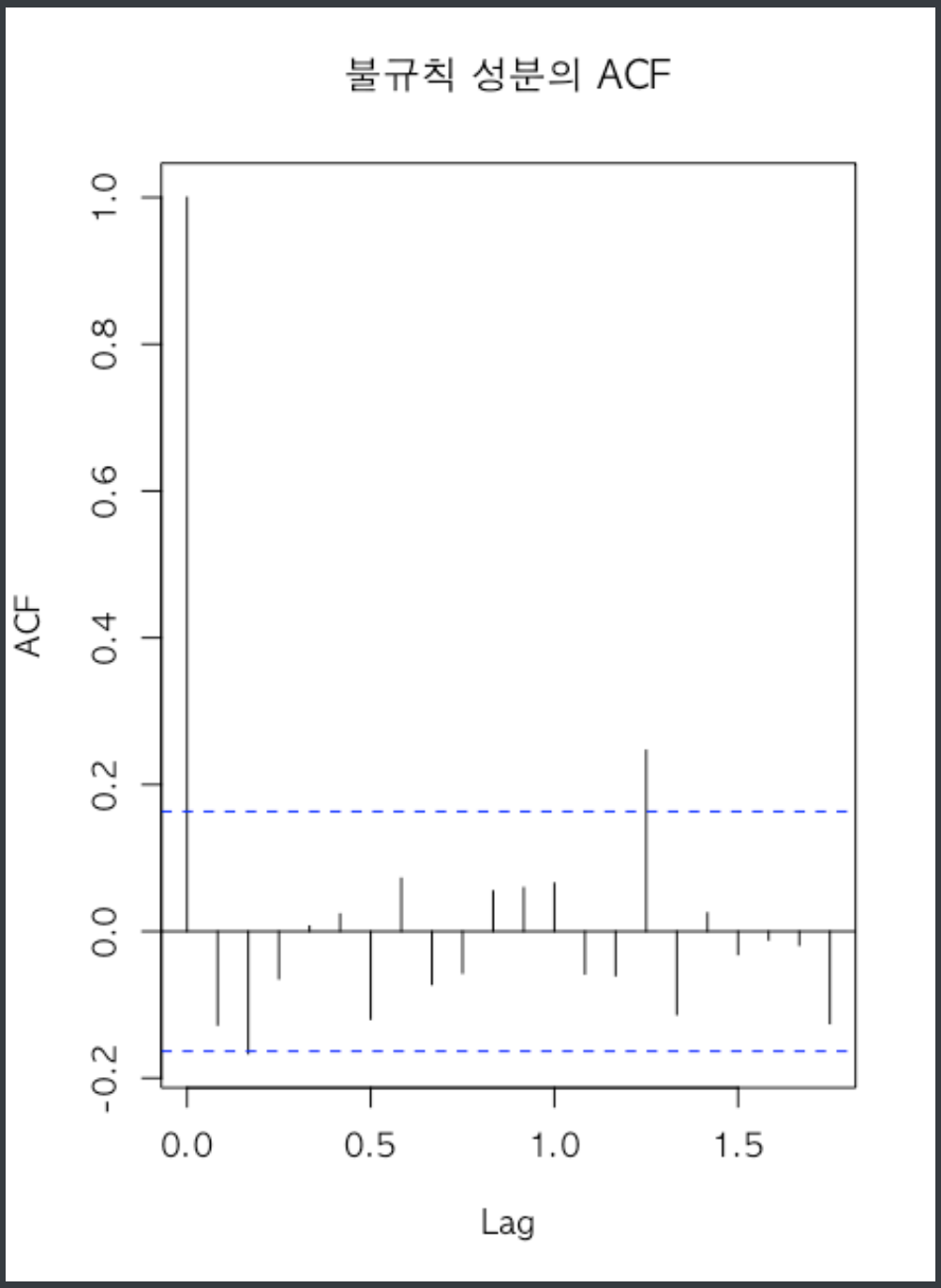
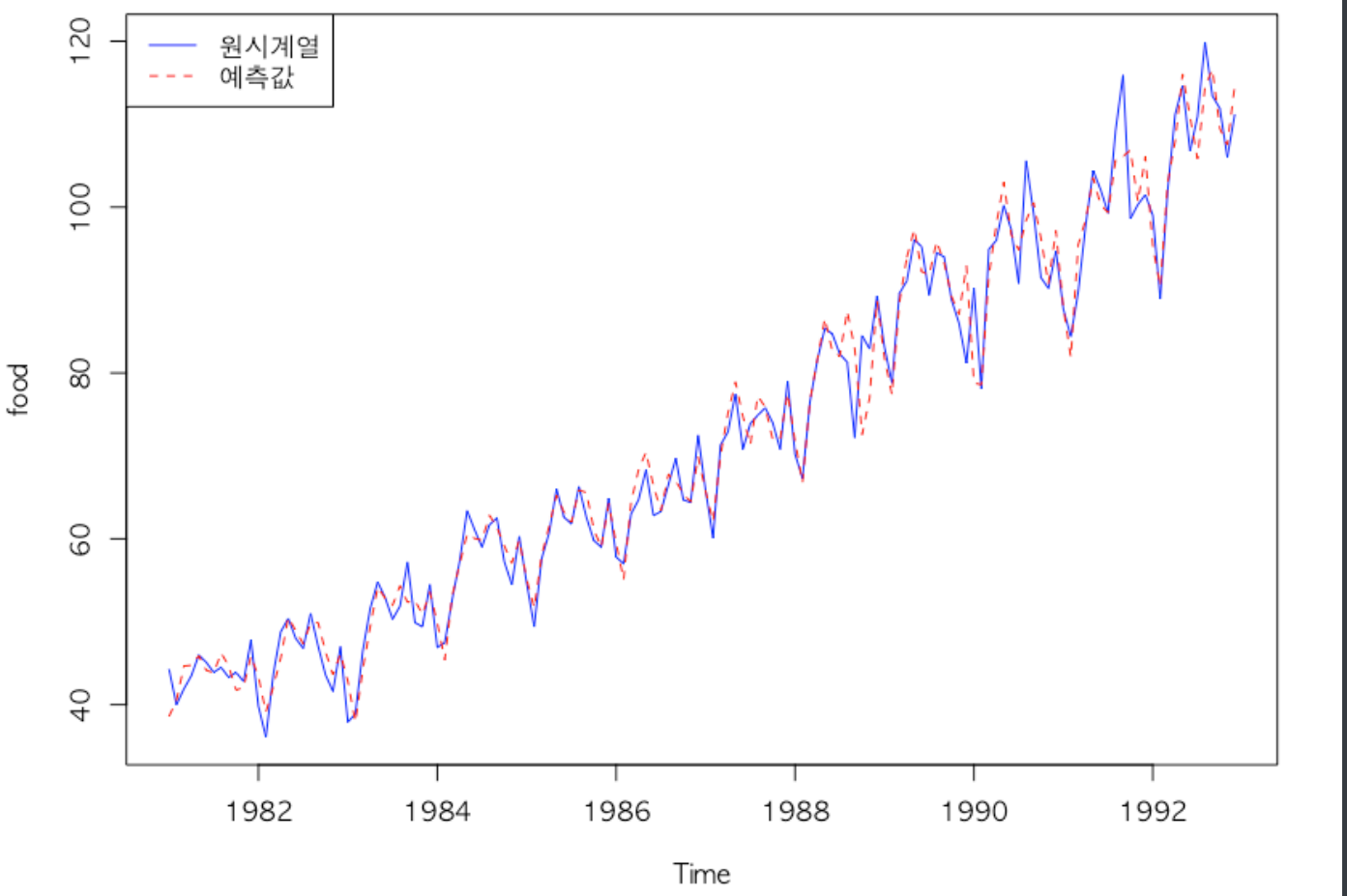


Figure 4-4

따라서 Trend 와 seasonal 의 곱을 이용하여 최종 예측된 값은 아래와 같다.

그림 4-4 원시계열과 예측값




```
> table4 # 최종 결과
          date  food     trend  seasonal irregular
1   1981-01-01  44.3  35.79171 1.0778057 1.1483671
2   1981-02-01  40.0  36.29729 1.1161309 0.9873489
3   1981-03-01  41.9  36.80286 1.2134718 0.9382158
4   1981-04-01  43.5  37.30844 1.1995519 0.9719930
5   1981-05-01  46.0  37.81401 1.2132945 1.0026257
6   1981-06-01  45.1  38.31959 1.1523512 1.0213413
7   1981-07-01  43.9  38.82516 1.1290919 1.0014332
8   1981-08-01  44.5  39.33073 1.1759323 0.9621563
9   1981-09-01  43.3  39.83631 1.1213139 0.9693522
10  1981-10-01  43.9  40.34188 1.0351174 1.0512808
11  1981-11-01  42.8  40.84746 1.0300158 1.0172668
12  1981-12-01  47.8  41.35303 1.1092092 1.0420945
13  1982-01-01  39.8  41.85861 1.0361581 0.9176399
```

```
14  1982-02-01  36.1  42.36418 0.9231016 0.9231215
15  1982-03-01  43.7  42.86975 0.9834988 1.0364697
16  1982-04-01  48.8  43.37533 1.0524058 1.0690396
17  1982-05-01  50.4  43.88090 1.1517230 0.9972566
18  1982-06-01  48.0  44.38648 1.1021423 0.9811895
19  1982-07-01  46.8  44.89205 1.0550194 0.9881342
20  1982-08-01  51.0  45.39763 1.0966539 1.0243949
...
```

Figure 4-5

다음은 이동 평균 3을 이용하여 추정한 값과 원시계열에 대한 그림이다.

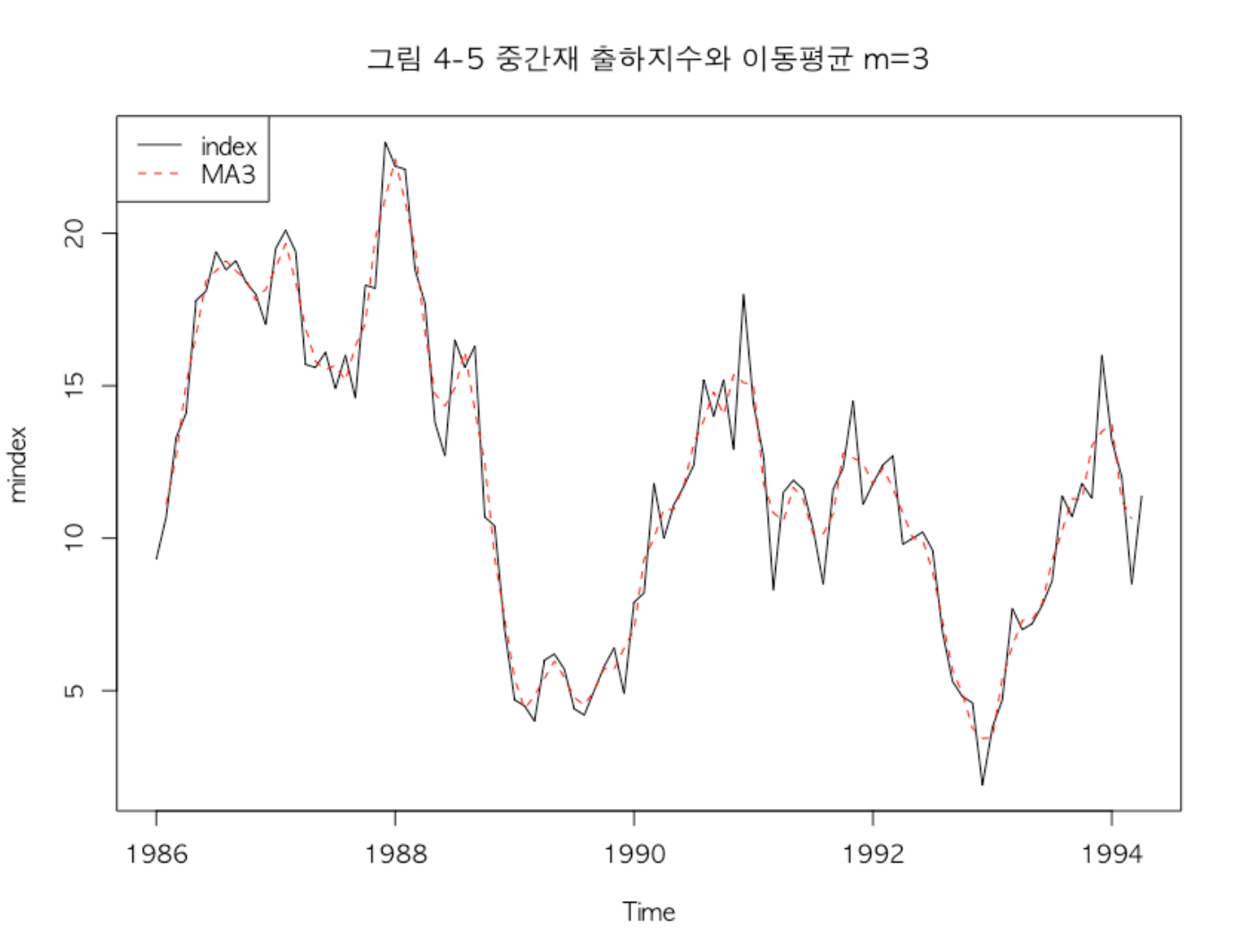


Figure 4-6

Figure 4-5에서는 m=3을 사용하였으나, 4-6에서는 m=7로 하였다. 앞선 예시보다 평활이 더 강하게 적용된 것을 볼 수 있다. 자료의 손실이 발생하는 대신, 추세를 잘 확인할 수 있다는 trade-off가 있다.

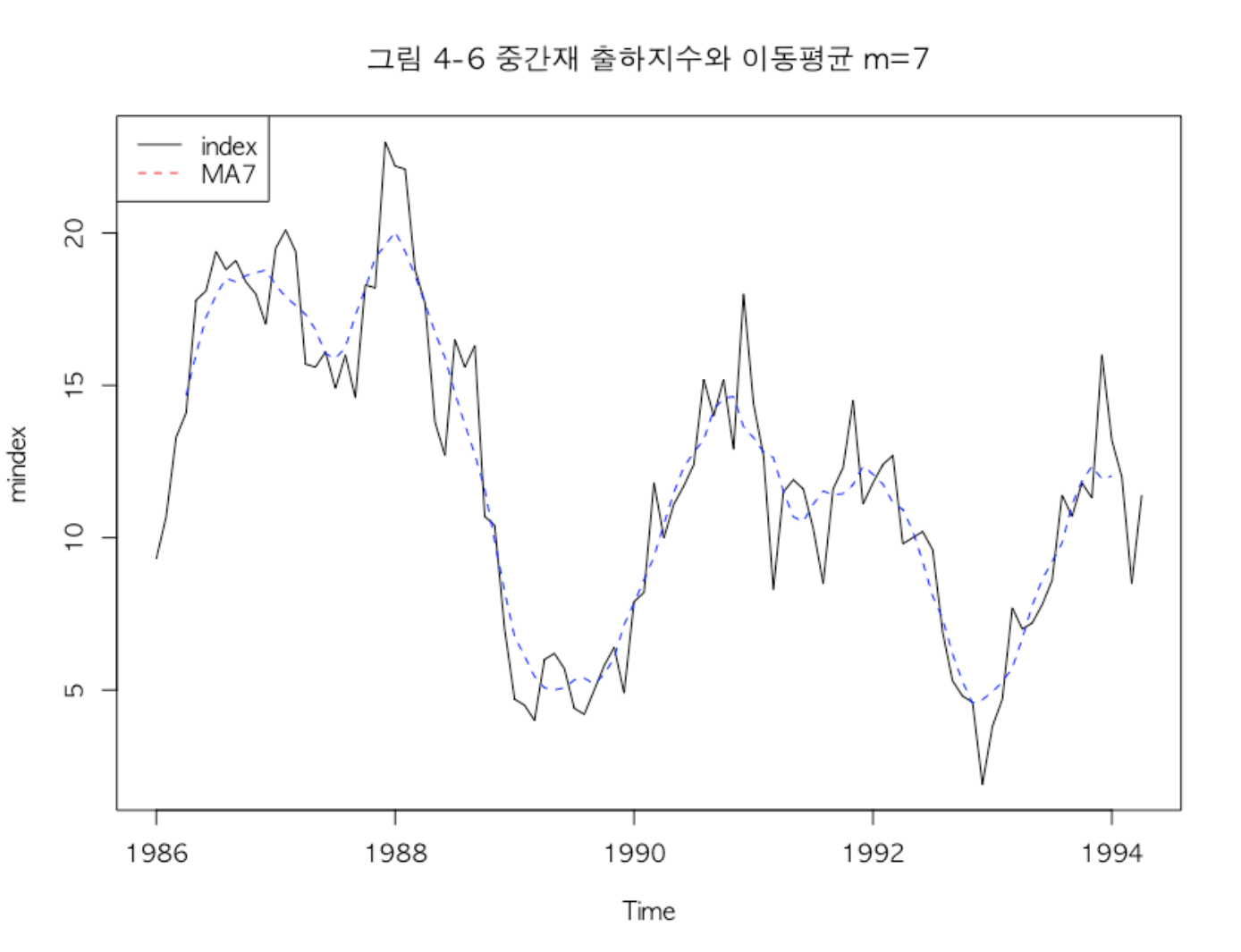


Figure 4-7

다음은 원시계열에서 추세, 순환 성분을 먼저 분리하고, 잔차에 대해 계절 성분을 분리하는 순으로 진행되는 성분 분해의 예시이다. 먼저 원시계열에서 추세, 순환성분을 분리한다. 일반적으로 순환 성분은 알기 어렵다는 문제가 있어 따로 고려하지 않거나 추세 성분과 묶어서 처리한다.

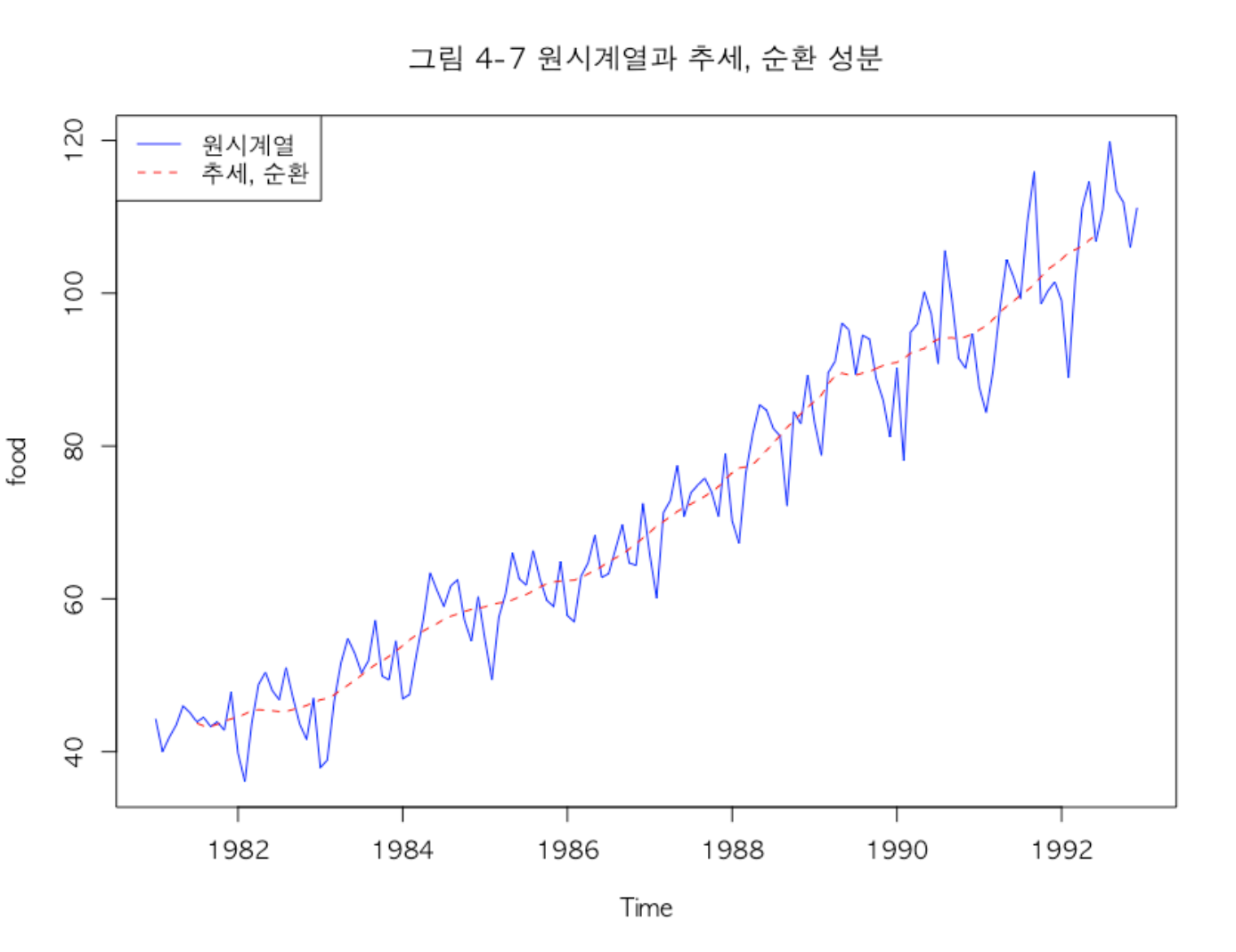


Figure 4-8

원시계열에서 추세, 순환 성분을 분리하고 남은 잔차에 대해서 계절 성분을 분해한다. 이때는 가법 모형을 고려하기 때문에 각 성분을 원시계열에서 빼주면 된다.

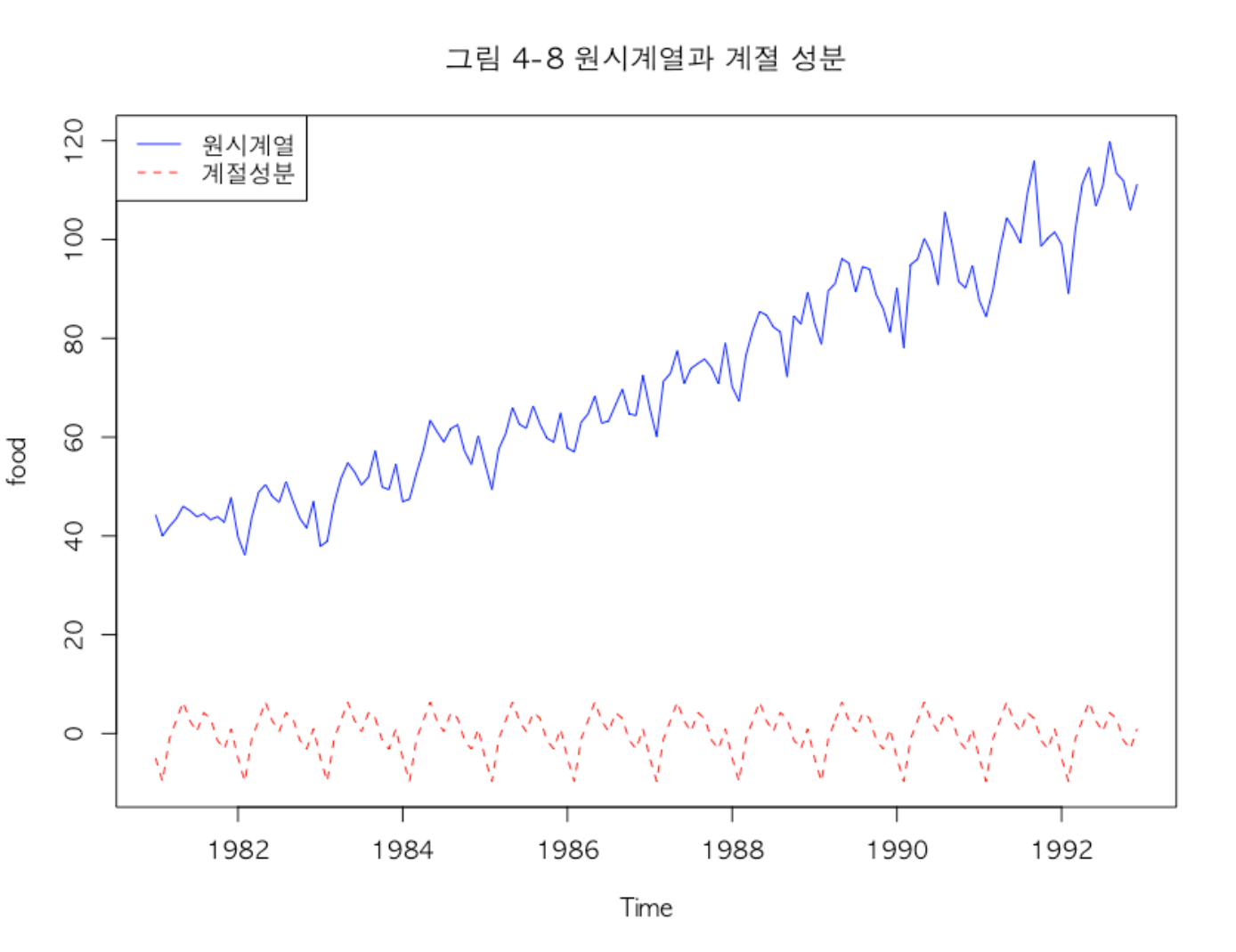


Figure 4-9

추세, 순환 성분을 먼저 제거한 후, 계절 성분도 제거가된 불규칙 성분이다. 분산이 조금씩 커지는 모습이 확인되는데, 이 경우에는 가법 모형보다 승법 모형이 적절했을 것 같다.

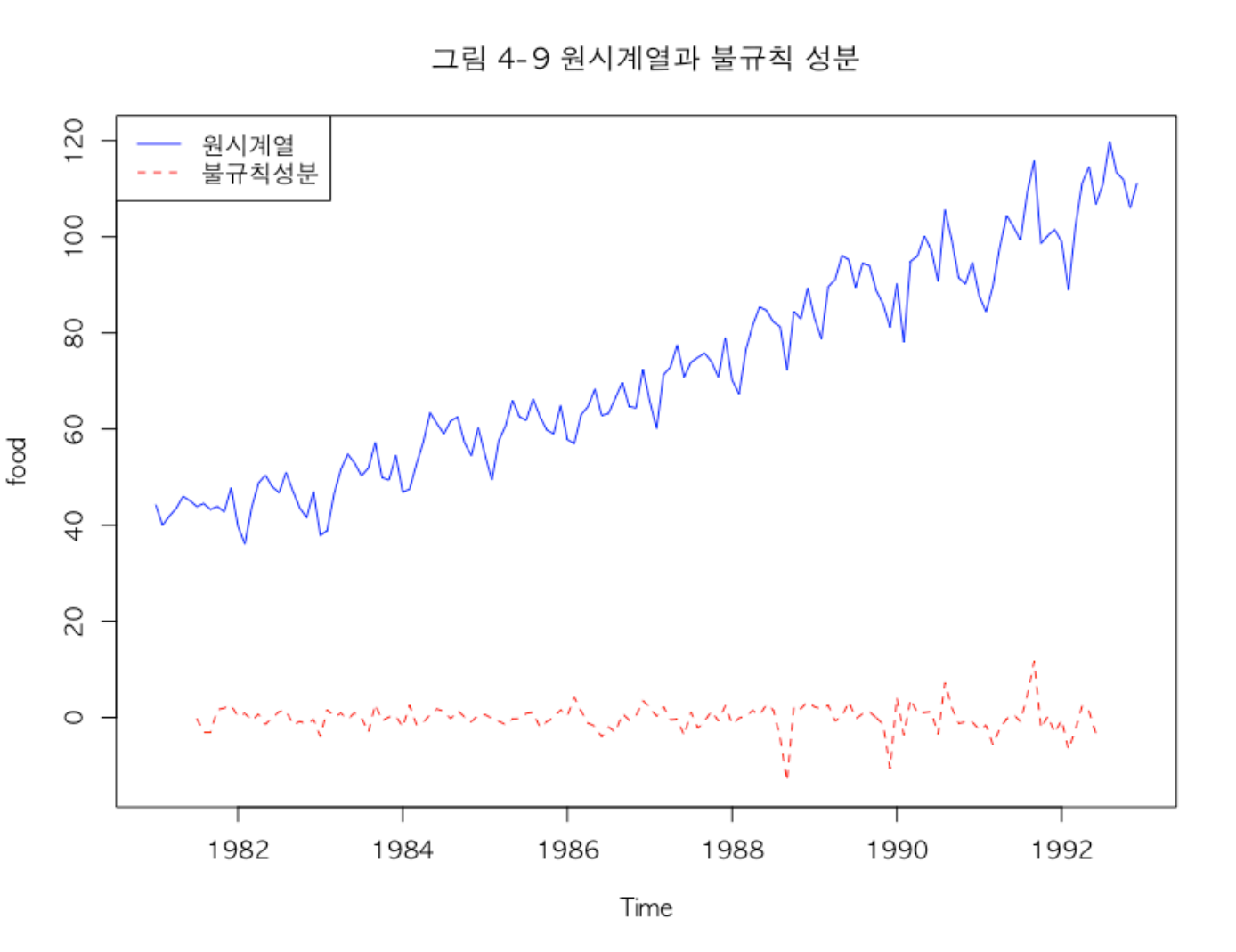


Figure 4-10

최종 예측 결과이다. 예측 결과는 추정된 추세, 순환 성분 + 계절 성분으로 구할 수 있다.

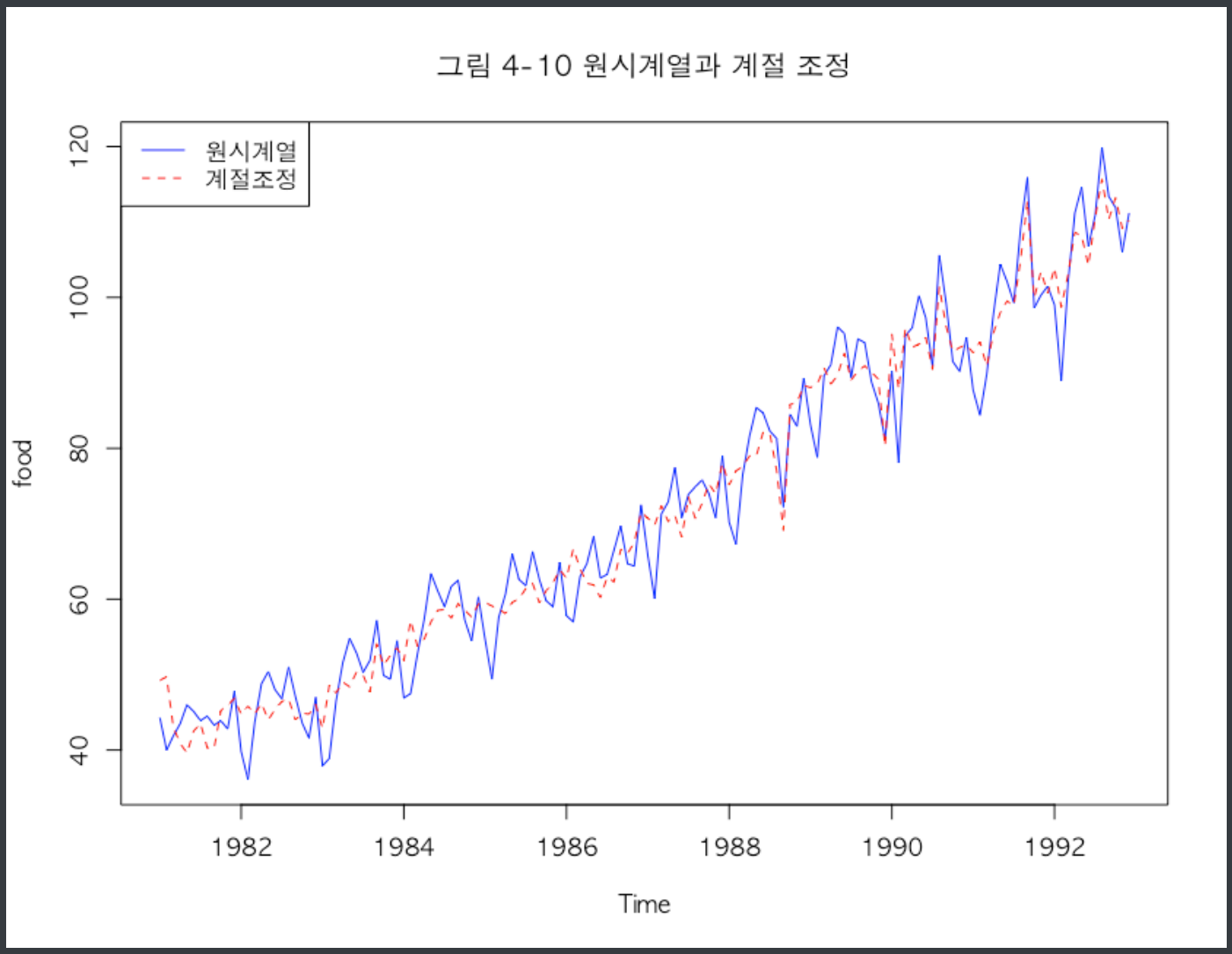


그림 4-10 원시계열과 계절 조정
원시계열
계절조정
food
120
100
80
60
40
1982
1984
1986
1988
1990
1992
Time

Exercise 4-2

4-2) $Z_t = \beta_0 + \beta_1 t + \varepsilon_t$ 이때,

a) $E[M_n^{(1)}] = ?$

$$M_n^{(1)} = \frac{1}{m}\sum_{t}^{n} Z_t = \frac{1}{m}\left[Z_{n-m+1} + Z_{n-m+2} + \cdots + Z_n\right]$$

$$= \frac{1}{m}\left[\underbrace{\beta_0 + \beta_1(n-m+1)+\varepsilon_t}_{Z_{n-m+1}} + \underbrace{\beta_0 + \beta_1(n-m+2)+\varepsilon_t}_{Z_{n-m+2}} + \cdots + \underbrace{\beta_0+\beta_1(n)+\varepsilon_t}_{Z_n}\right]$$

$$= \frac{1}{m}\left[\underbrace{m\beta_0 + m\cdot n\beta_1 - m\cdot m\beta_1 + \frac{m(m+1)}{2}\beta_1}_{} + m\varepsilon_t\right]$$

$\longrightarrow$ $-m+1 \quad -m+2 + \cdots \quad -m+m$

m ① ② $\quad m$ ① ② $\quad m$ ① ②

$-m\cdot m + \frac{m(m+1)}{2} \longrightarrow \sum_{k=1}^{n} k$

① ②

$$= \beta_0 + n\beta_0 - m\beta_1 + \frac{m+1}{2}\beta_1 + \varepsilon_t$$

$$E[M_n^{(1)}] = \beta_0 + n\beta_1 - m\beta_1 + \frac{m+1}{2}\beta_1$$

$$= \beta_0 + n\beta_1 - \frac{m}{2}\beta_1 + \frac{1}{2}\beta_1 + \beta_0 - \beta_1$$

$$= \beta_0 + n\beta_1 + \beta_1 - \frac{m+1}{2}\beta_1$$

$$= \beta_0 + (n+1)\beta_1 - \frac{m+1}{2}\beta_1$$

bias

①

b)

$$M_n^{(2)} = \frac{1}{m}\sum_{t}^{n} M_t^{(1)} = \frac{1}{m}\left[M_{n-m+1}^{(1)} + M_{n-m+2}^{(1)} + \cdots + M_n^{(1)}\right]$$

$M_n^{(1)} = \beta_0 + (n+1)\beta_1$, $-\frac{m+1}{2}\beta_1 + \varepsilon_t$

$$= \frac{1}{m}\left[\beta_0 + (n-m+2)\beta_1 - \frac{m+1}{2}\beta_1 + \varepsilon_t + \cdots + \beta_0 + (n+1)\beta_1 - \frac{m+1}{2}\beta_1 + \varepsilon_t\right]$$

$$= \frac{1}{m}\left[m\beta_0 + m\cdot n\beta_1 - (m+1)m\beta_1 + \frac{m(m+1)}{2}\beta_1 - \frac{m(m+1)}{2}\beta_1 + m\cdot\varepsilon_t\right]$$

$$= \frac{1}{m}\left[m\beta_0 + mn\beta_1 - (m-1)m\beta_1 + m\cdot\varepsilon_t\right]$$

$$= \beta_0 + n\beta_1 - (m-1)\beta_1 + \varepsilon_t$$

$$E\left[M_n^{(2)}\right] = \beta_0 + n\beta_1 - (m-1)\beta_1$$

$-m-1$

$$\begin{cases} M_n^{(1)} = \beta_0 + (n+1)\beta_1 - \frac{m+1}{2}\beta_1 \\ M_n^{(2)} = \beta_0 + n\beta_1 - (m-1)\beta_1 \end{cases}$$

$+m+1$

c) $M_n^{(1)} - M_n^{(2)} = \beta_0 + (n+1)\beta_1 - \frac{m+1}{2}\beta_1 - \beta_0 - n\beta_1 + (m-1)\beta_1$

$$= \beta_1 + \frac{-m-1+2m-2}{2}\beta_1 = \beta_1 + \frac{m-3}{2}\beta_1 = \frac{m-1}{2}\beta_1$$

$$\therefore \hat{\beta}_1 = \frac{2}{m-1}\left(M_n^{(1)} - M_n^{(2)}\right)$$

$$2m_n^{(1)} - m_n^{(2)} = 2\beta_0 + 2(n+1)\beta_1 - (n+1)\beta_1 - (\beta_0 + n\beta_1 - (n-1)\beta_1)$$

$$= \beta_0 + (2n+2 - n - 1 - n + n - 1)\beta_1$$

$$2m_n^{(1)} - m_n^{(2)} = \beta_0 + n\beta_1$$

$$\hat{\beta}_0 = 2m_n^{(1)} - m_n^{(2)} - n\beta_1$$

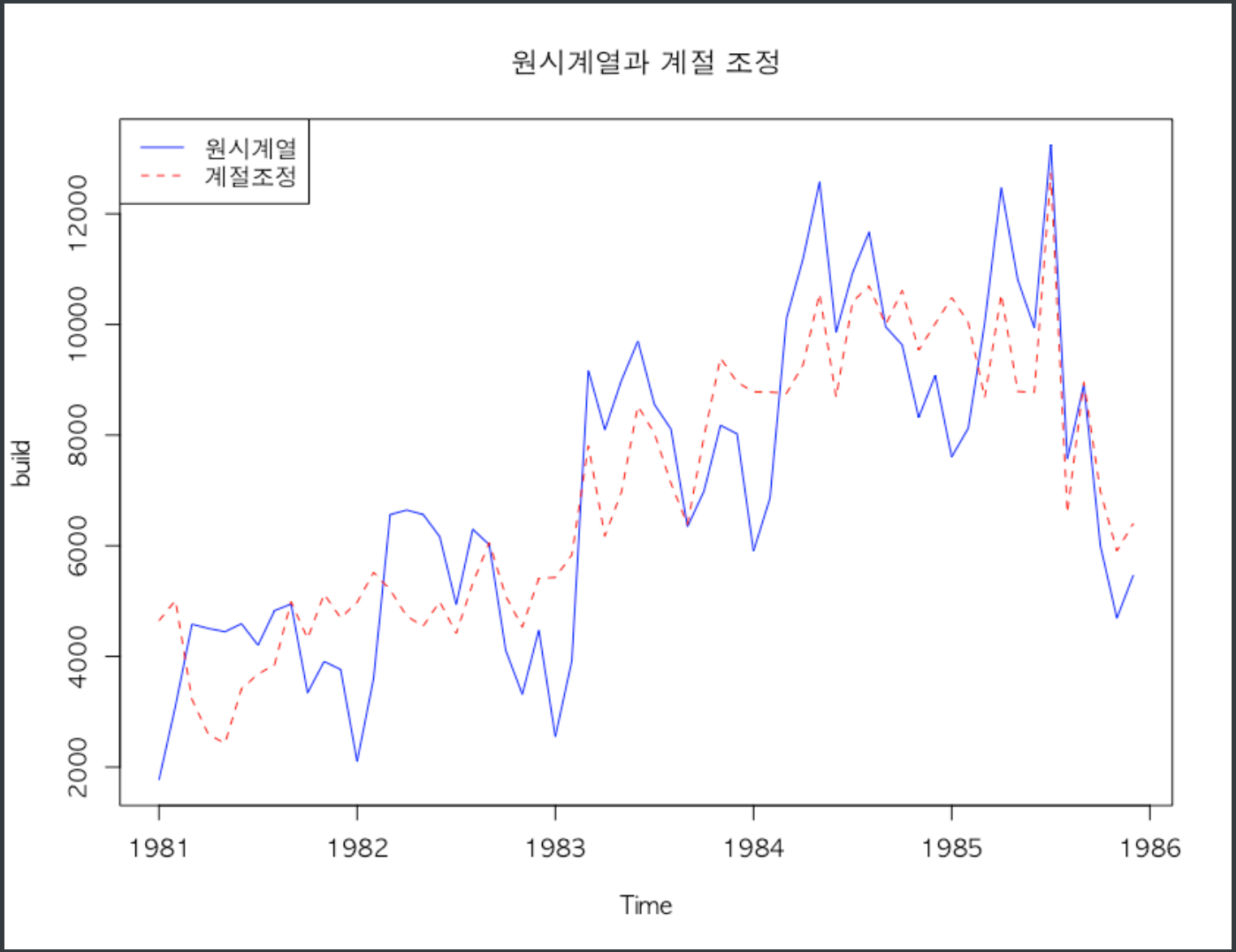
=

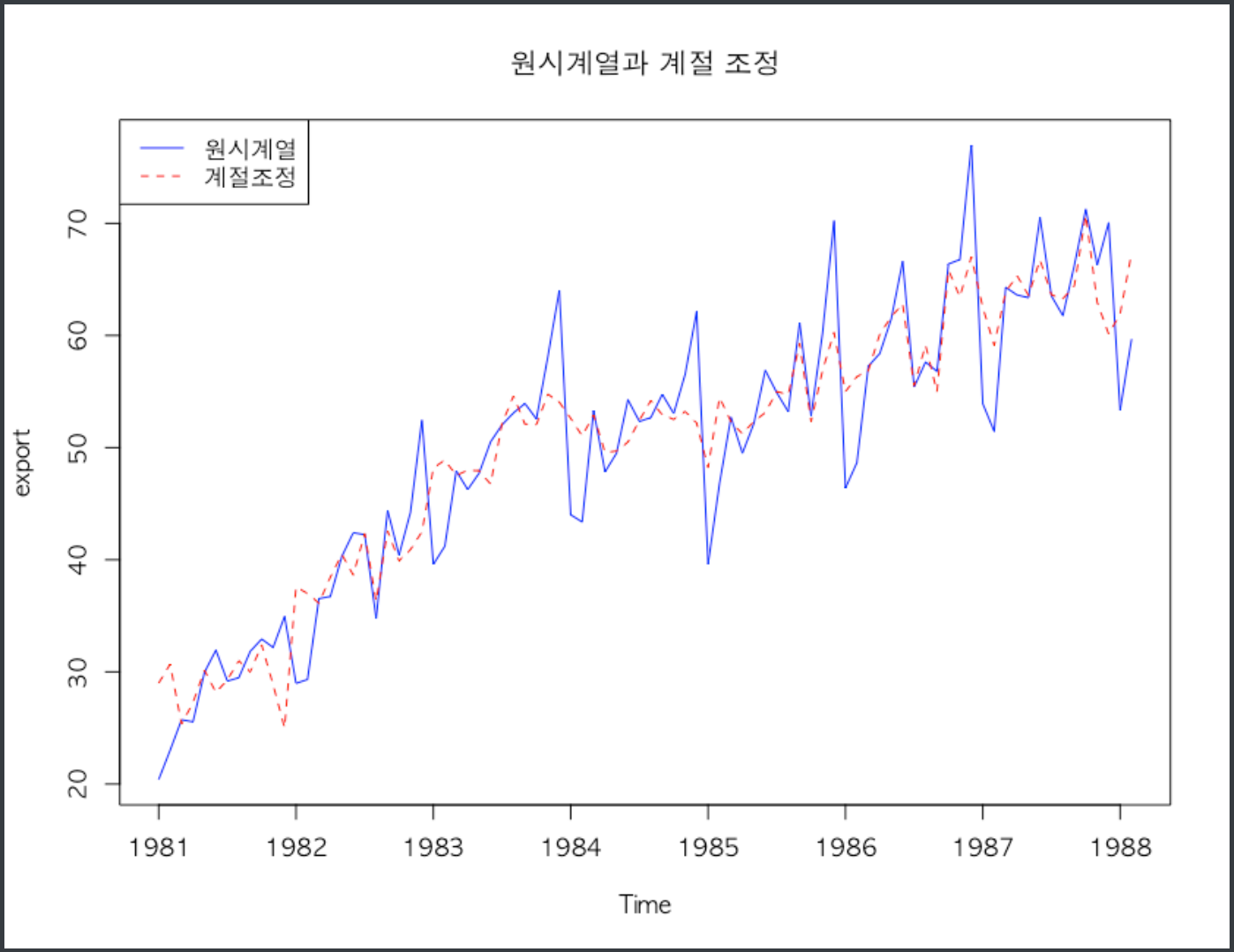
$$\hat{Z}_n(\ell) = \hat{\beta}_0 + \hat{\beta}_1(n+\ell)$$

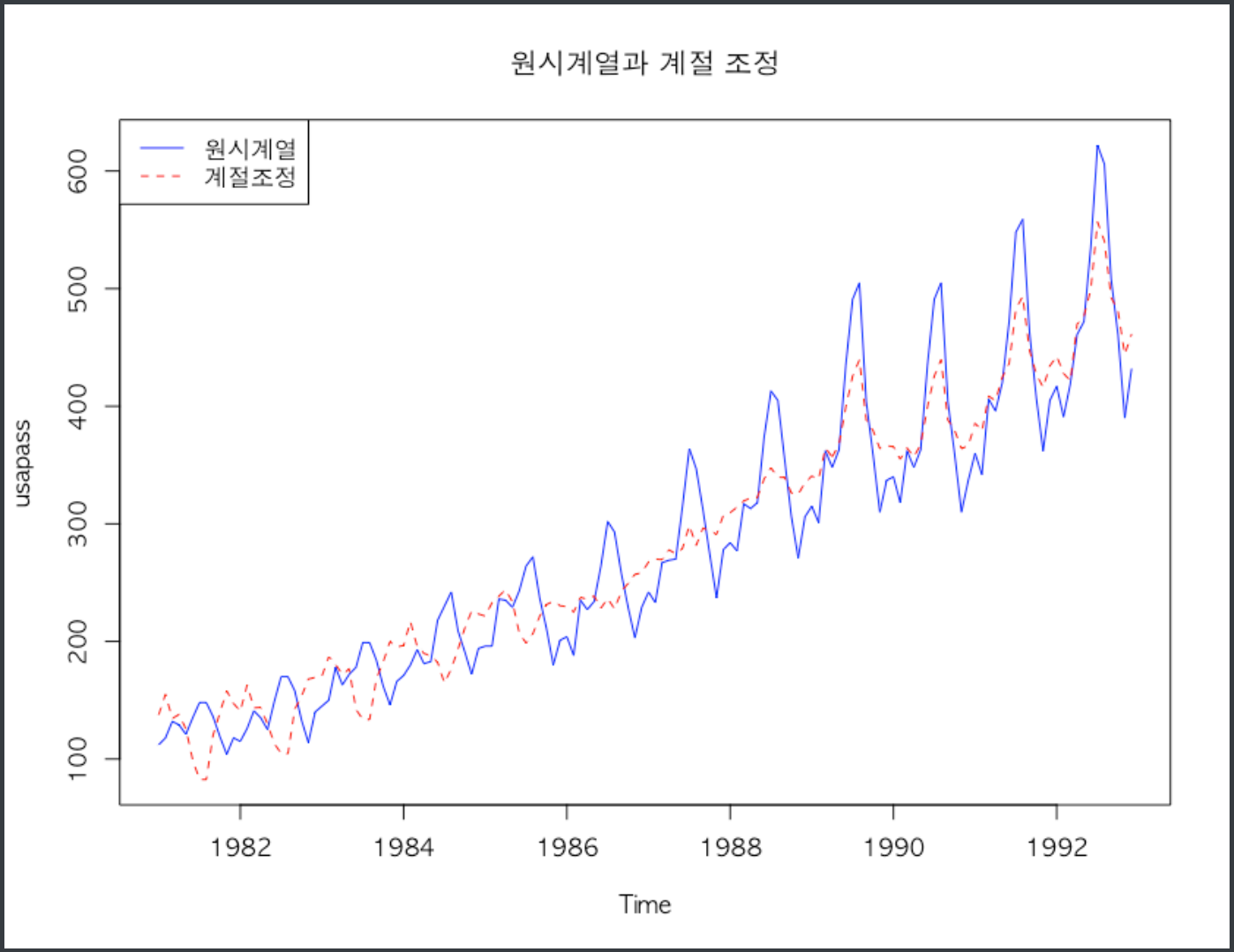
$$= 2m_n^{(1)} - m_n^{(2)} + \hat{\beta}_1 \ell$$

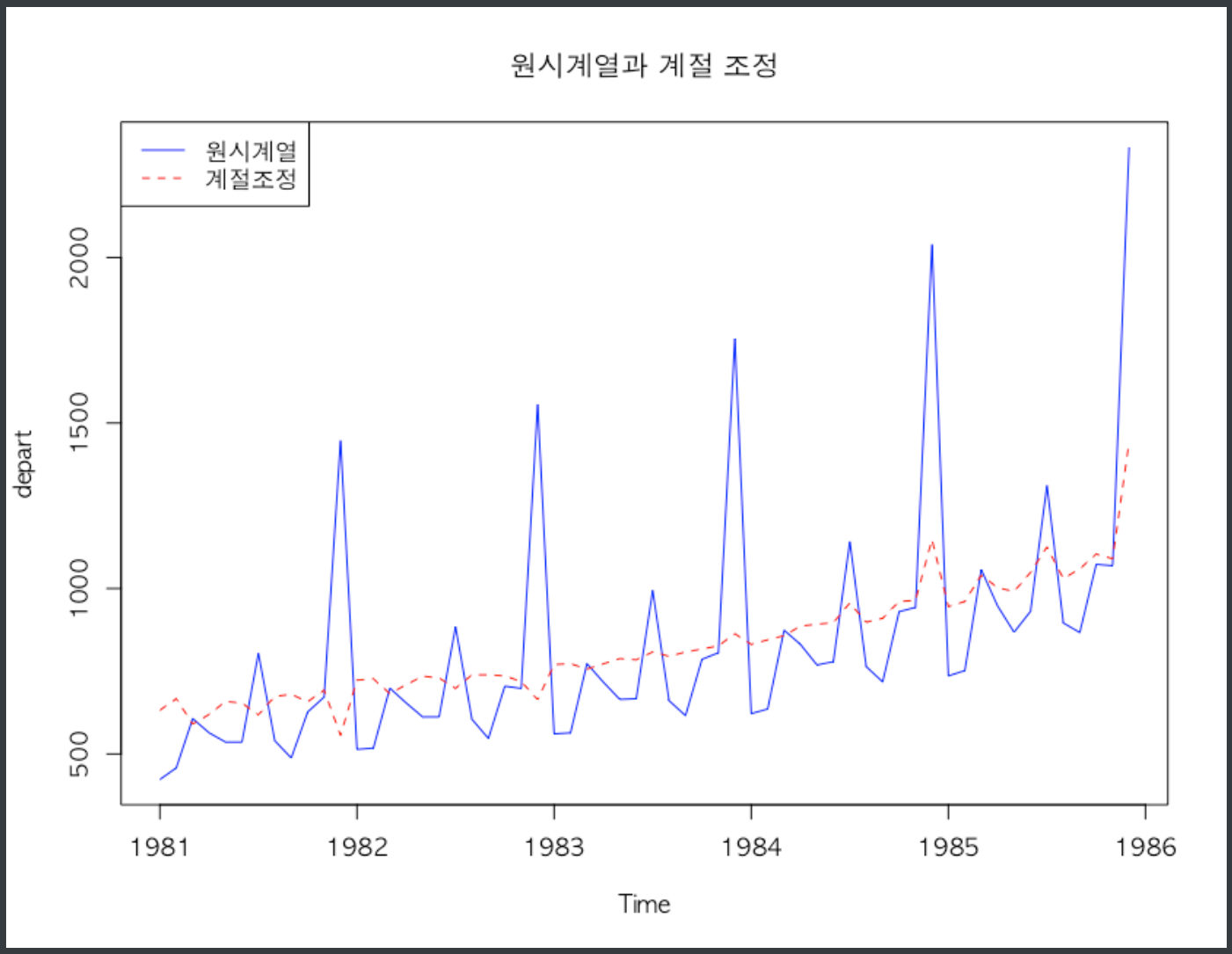
Exercise 4-4

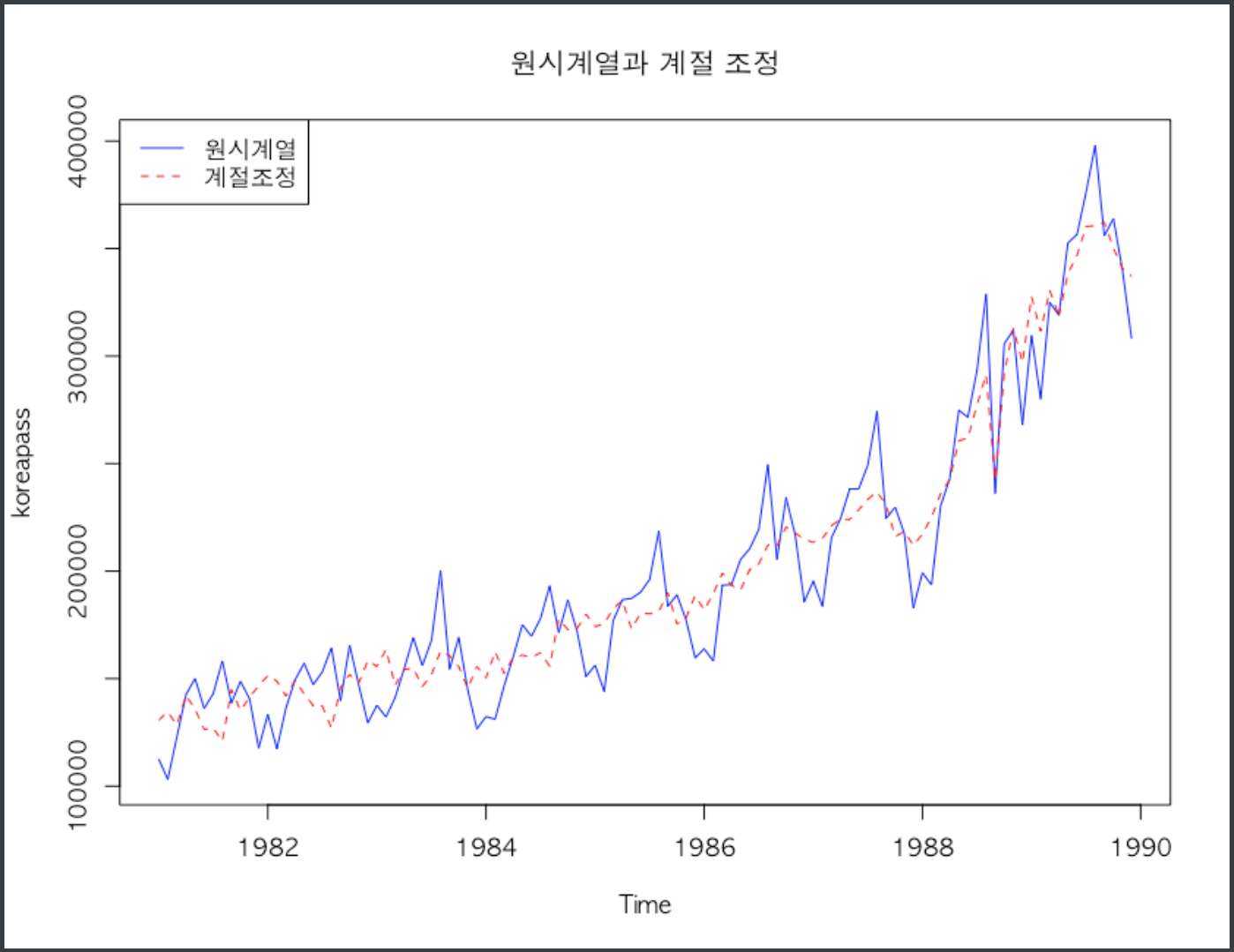
아래 그림은 순서대로 4-4-1, 4-4-2, 4-4-3, 4-4-4, 4-4-5 에 해당한다.




원시계열과 계절 조정
원시계열
계절조정
export
20
30
40
50
60
70
1981
1982
1983
1984
1985
1986
1987
1988
Time


원시계열과 계절 조정
원시계열
계절조정
usapass
100
200
300
400
500
600
1982
1984
1986
1988
1990
1992
Time


원시계열과 계절 조정
원시계열
계절조정
depart
500
1000
1500
2000
1981
1982
1983
1984
1985
1986
Time



Appendix, R code

```r
rm(list=ls())

setwd("~/Workspace/2022-Fall_TimeSeriesAnalysis/")
par(family="AppleGothic")

library(lmtest)
library(forecast)
library(lubridate)

# Example 4.1
z = scan("data/food.txt")
```

```r
t = 1:length(z)
food = ts(z, start=c(1981, 1),frequency=12)
fit = lm(food ~ t)

anova(fit)
trend = fitted(fit)
ts.plot(food, trend, col=1:2, lty=1:2, ylab="food", xlab="time",
        main="그림 4-1 원시계열과 분해법에 의한 추세성분")
legend("topleft", lty=1:2, col=1:2, c("원시계열", "추세성분"))

adjtrend = food / trend
y = factor(cycle(adjtrend))

temp_lm = lm(adjtrend ~ y + 0)
dwtest(temp_lm)

fit1 = auto.arima(adjtrend, max.p=2, xreg=model.matrix(~ 0 + y)[, -12],
                  seasonal=F, max.d=0, max.q=0)
fit1

seasonal = fit1$fitted
pred = trend * seasonal
irregular = food / pred
ts.plot(seasonal, main="그림 4-2 분해법에 의한 계절성분")
ts.plot(irregular, main="그림 4-3 분해법에 의한 불규칙성분")

acf(irregular, main="불규칙 성분의 ACF")
ts.plot(food, pred, lty=1:2, ylab="food", col=c("blue", "red"),
        main="그림 4-4 원시계열과 예측값")
legend("topleft", lty=1:2, col=c("blue", "red"),
       c("원시계열", "예측값"))

date = ymd("810101") + months(1:length(food) - 1)
table4 = data.frame(date, food, trend, seasonal, irregular)
table4
```

```r
# Example 4.2
z = scan("data/mindex.txt")
mindex = ts(z, start=c(1986, 1), frequency=12)
m3 = ma(mindex, 3)
m7 = ma(mindex, 7)

plot(mindex, lty=1, main="그림 4-5 중간재 출하지수와 이동평균 m=3")
lines(m3, lwd=1, col="red", lty=2)
legend("topleft", lty=1:2, col=c(1, "red"), c("index", "MA3"))

plot(mindex, lty=1, main="그림 4-6 중간재 출하지수와 이동평균 m=7")
lines(m7, lwd=1, col="blue", lty=2)
legend("topleft", lty=1:2, col=c(1, "red"), c("index", "MA7"))

# Example 4.3
z = scan("data/food.txt")
food = ts(z, start=c(1981, 1), frequency=12)

m = decompose(food, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = food - seasonal

ts.plot(food, trend, ylab="food", lty=1:2, col=c("blue", "red"),
        main="그림 4-7 원시계열과 추세, 순환 성분")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "추세, 순환"))

ts.plot(food, seasonal, ylab="food", lty=1:2, col=c("blue", "red"),
        main="그림 4-8 원시계열과 계절 성분")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절성분"))

ts.plot(food, irregular, ylab="food", lty=1:2, col=c("blue", "red"),
        main="그림 4-9 원시계열과 불규칙 성분")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "불규칙성분"))
```

```r
ts.plot(food, adjseasonal, ylab="food", lty=1:2, col=c("blue", "red"),
        main="그림 4-10 원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))

food.stl = stl(food, "periodic")
food.sa = seasadj(food.stl)
ts.plot(food, food.sa, ylab="food", lty=1:2, col=c("blue", "red"),
        main="Figure 4-10 원시계열과 계절조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))


# Exercise 4.4 - 1
z = scan("data/build.txt")
build = ts(z, start=c(1981, 1), frequency=12)

m = decompose(build, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = build - seasonal

ts.plot(build, adjseasonal, ylab="build", lty=1:2, col=c("blue", "red"),
        main="원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))


# Exercise 4.4 - 2
z = scan("data/export.txt")
export = ts(z, start=c(1981, 1), frequency=12)

m = decompose(export, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = export - seasonal
```

```r
ts.plot(export, adjseasonal, ylab="export", lty=1:2, col=c("blue",
"red"),
        main="원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))


# Exercise 4.4 - 3
z = scan("data/usapass.txt")
usapass = ts(z, start=c(1981, 1), frequency=12)

m = decompose(usapass, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = usapass - seasonal

ts.plot(usapass, adjseasonal, ylab="usapass", lty=1:2, col=c("blue",
"red"),
        main="원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))


# Exercise 4.4 - 4
z = scan("data/depart.txt")
depart = ts(z, start=c(1981, 1), frequency=12)

m = decompose(depart, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = depart - seasonal

ts.plot(depart, adjseasonal, ylab="depart", lty=1:2, col=c("blue",
"red"),
        main="원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))
```

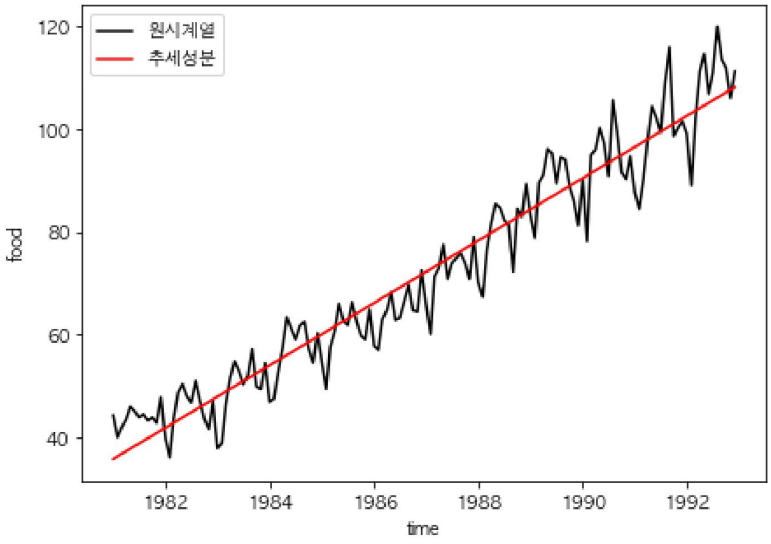
```r
# Exercise 4.4 - 5
z = scan("data/koreapass.txt")
koreapass = ts(z, start=c(1981, 1), frequency=12)

m = decompose(koreapass, type=c("additive"))
trend = trendcycle(m)
seasonal = seasonal(m)
irregular = remainder(m)
adjseasonal = koreapass - seasonal

ts.plot(koreapass, adjseasonal, ylab="koreapass", lty=1:2, col=c("blue",
"red"),
        main="원시계열과 계절 조정")
legend("topleft", lty=1:2, col=c("blue", "red"), c("원시계열", "계절조정"))
```

Appendix, python code

In [1]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('font', family='AppleGothic')
plt.rcParams['axes.unicode_minus'] = False

import statsmodels.api as sm
from statsmodels.tsa.seasonal import STL
from statsmodels.tsa.ar_model import AutoReg
```

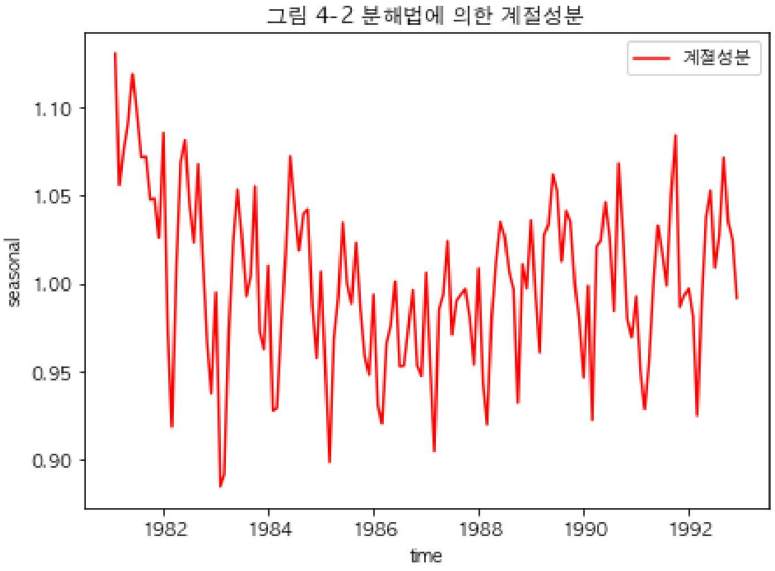
In [2]:
```
# Example 4.1
z = []

with open('../data/food.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1981", periods=len(z), freq="MS")
data = pd.Series(z, index)
t = np.arange(len(z)).reshape(-1, 1)
t_ = sm.add_constant(t)

lm = sm.OLS(data, t_)
res = lm.fit()

trend = res.fittedvalues

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(trend, 'red', label="추세성분")
ax.set_xlabel("time")
ax.set_ylabel("food")
ax.set_title("그림 4-1 원시계열과 분해법에 의한 추세성분")
plt.legend()
plt.show()
```

그림 4-1 원시계열과 분해법에 의한 추세성분

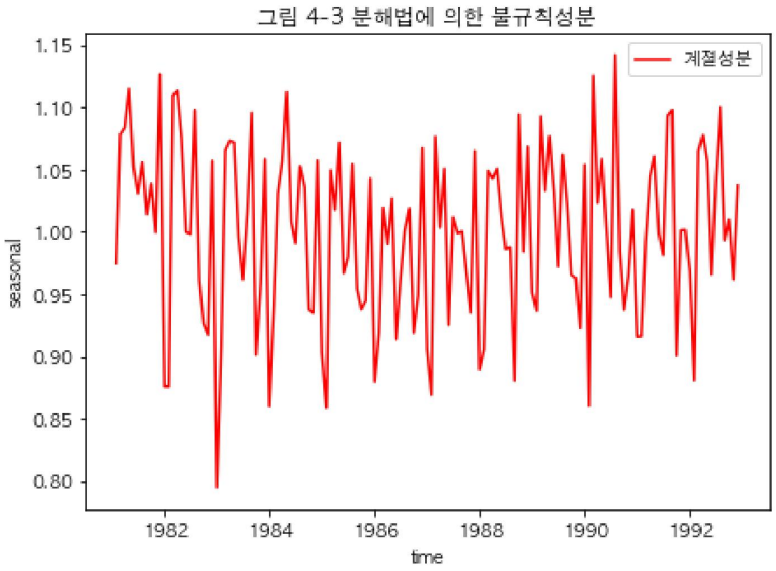


In [3]:
```
adjtrend = data / trend
y = pd.get_dummies(data.index.month).values

auto_reg = AutoReg(adjtrend, 1)
res = auto_reg.fit()

seasonal = res.fittedvalues
pred = trend * seasonal
irregular = data / pred

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(seasonal, 'red', label="계절성분")
ax.set_xlabel("time")
ax.set_ylabel("seasonal")
ax.set_title("그림 4-2 분해법에 의한 계절성분")
plt.legend()
plt.show()
```

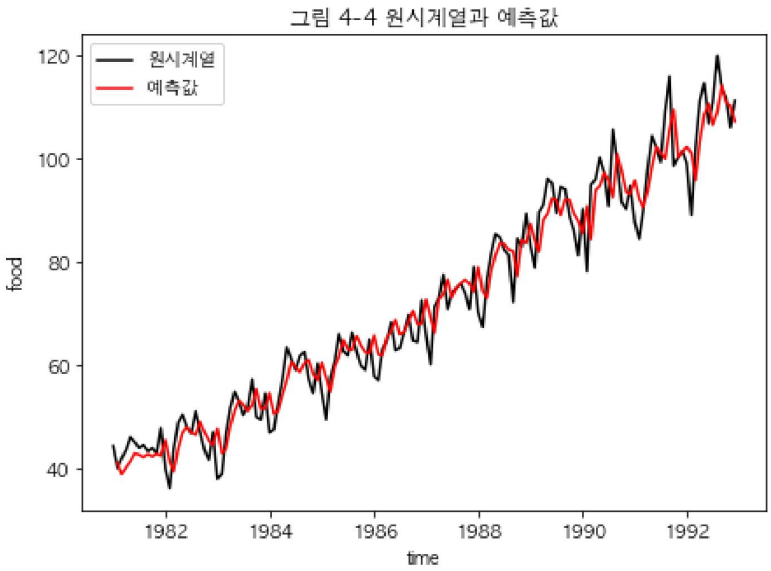
```
/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/ar_mode
l.py:248: FutureWarning: The parameter names will change after 0.12 is release
d. Set old_names to False to use the new names now. Set old_names to True to us
e the old names.
  warnings.warn(
/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/base/tsa
_model.py:132: FutureWarning: The 'freq' argument in Timestamp is deprecated an
d will be removed in a future version.
  date_key = Timestamp(key, freq=base_index.freq)
```



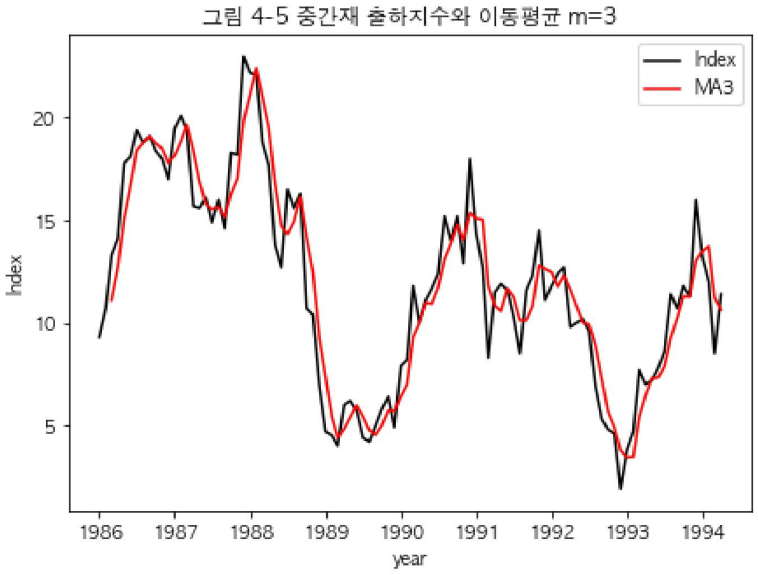
In [4]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(irregular, 'red', label="계절성분")
ax.set_xlabel("time")
ax.set_ylabel("seasonal")
ax.set_title("그림 4-3 분해법에 의한 불규칙성분")
plt.legend()
plt.show()
```



In [5]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(pred, 'red', label="예측값")
ax.set_xlabel("time")
ax.set_ylabel("food")
ax.set_title("그림 4-4 원시계열과 예측값")
plt.legend()
plt.show()
```

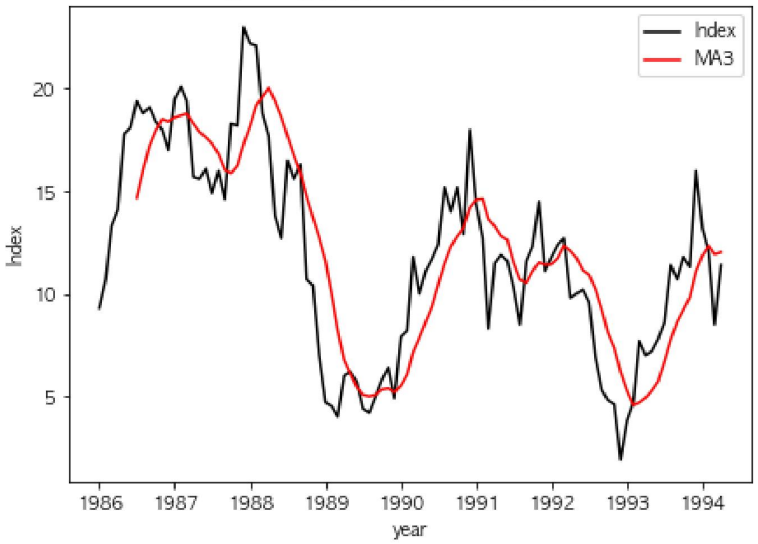


In [6]:
```
# Example 4.2
z = []

with open('../data/mindex.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1986", periods=len(z), freq="MS")
data = pd.Series(z, index)

m3 = data.rolling(3).mean()

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="Index")
ax.plot(m3, 'red', label="MA3")
ax.set_xlabel("year")
ax.set_ylabel("Index")
ax.set_title("그림 4-5 중간재 출하지수와 이동평균 m=3")
plt.legend()
plt.show()
```



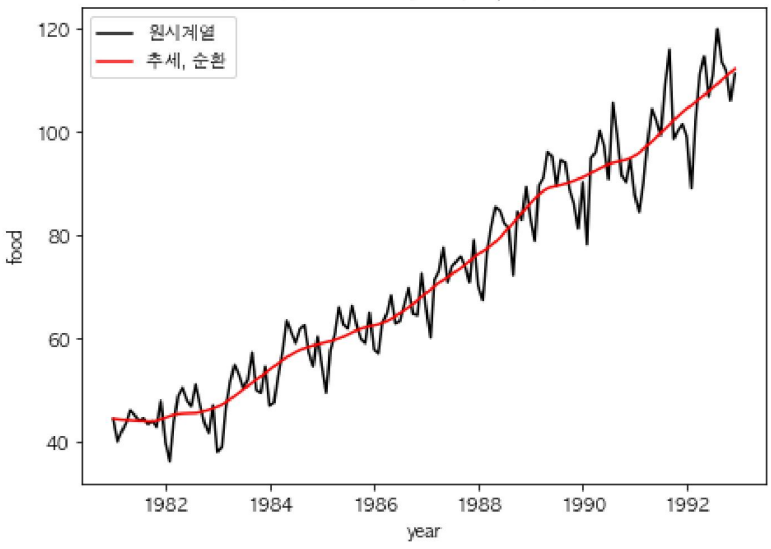
In [7]:
```
m7 = data.rolling(7).mean()

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="Index")
ax.plot(m7, 'red', label="MA3")
ax.set_xlabel("year")
ax.set_ylabel("Index")
ax.set_title("그림 4-6 중간재 출하지수와 이동평균 m=7")
plt.legend()
plt.show()
```

그림 4-6 중간재 출하지수와 이동평균 m=7



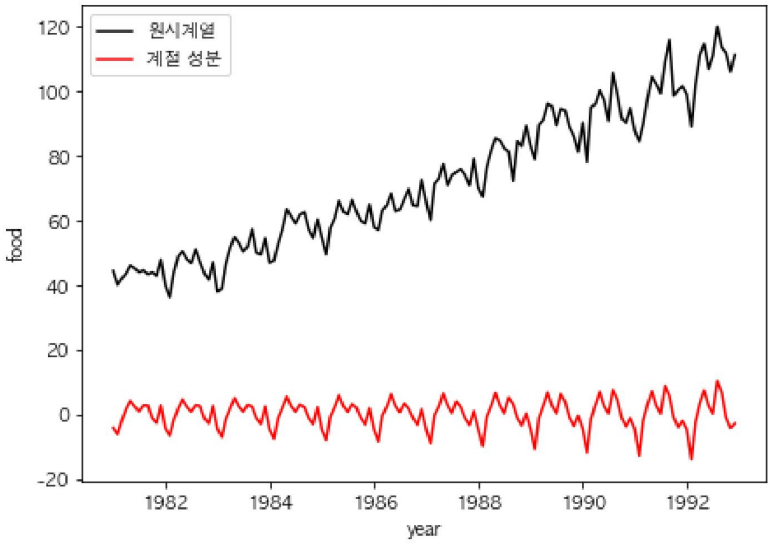
In [8]:
```
# Example 4.3
z = []

with open('../data/food.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1981", periods=len(z), freq="MS")
data = pd.Series(z, index)

stl = STL(data, seasonal=13)
res = stl.fit()

trend = res.trend
seasonal = res.seasonal
irregular = res.resid
adjseasonal = data - seasonal

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(trend, 'red', label="추세, 순환")
ax.set_xlabel("year")
ax.set_ylabel("food")
ax.set_title("그림 4-7 원시계열과 추세, 순환 성분")
plt.legend()
plt.show()
```

그림 4-7 원시계열과 추세, 순환 성분



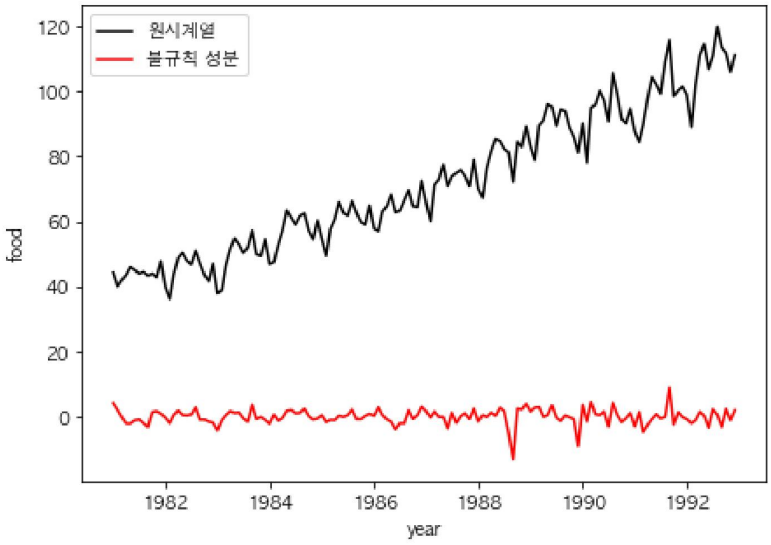
In [9]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(seasonal, 'red', label="계절 성분")
ax.set_xlabel("year")
ax.set_ylabel("food")
ax.set_title("그림 4-8 원시계열과 계절 성분")
plt.legend()
plt.show()
```

그림 4-8 원시계열과 계절 성분



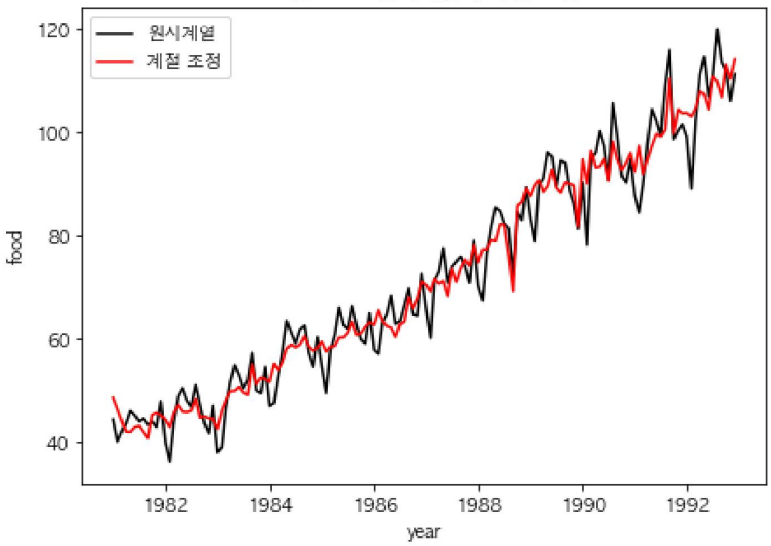
In [10]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(irregular, 'red', label="불규칙 성분")
ax.set_xlabel("year")
ax.set_ylabel("food")
ax.set_title("그림 4-9 원시계열과 불규칙 성분")
plt.legend()
plt.show()
```

그림 4-9 원시계열과 불규칙 성분

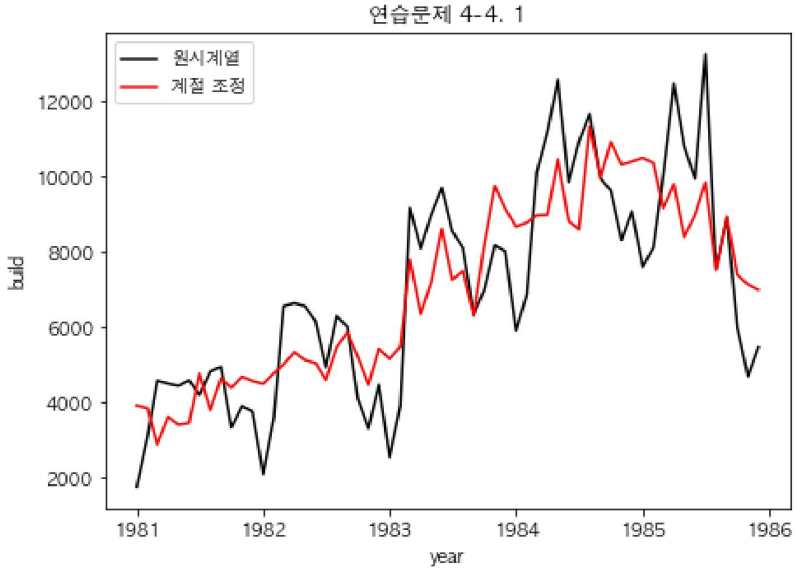


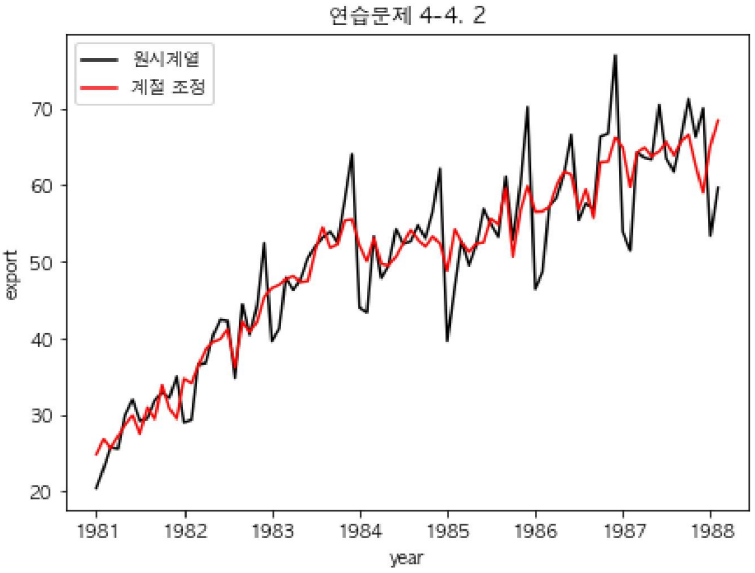
In [11]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black', label="원시계열")
ax.plot(adjseasonal, 'red', label="계절 조정")
ax.set_xlabel("year")
ax.set_ylabel("food")
ax.set_title("그림 4-10 원시계열과 계절 조정")
plt.legend()
plt.show()
```

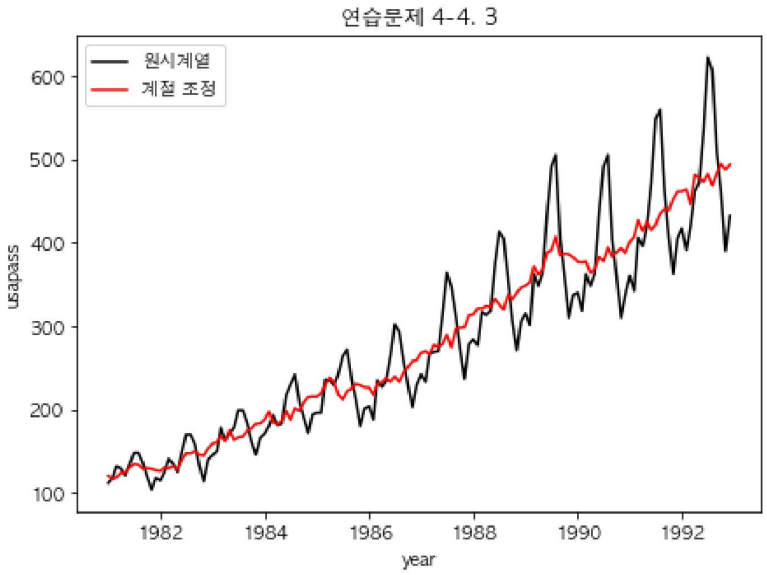
그림 4-10 원시계열과 계절 조정

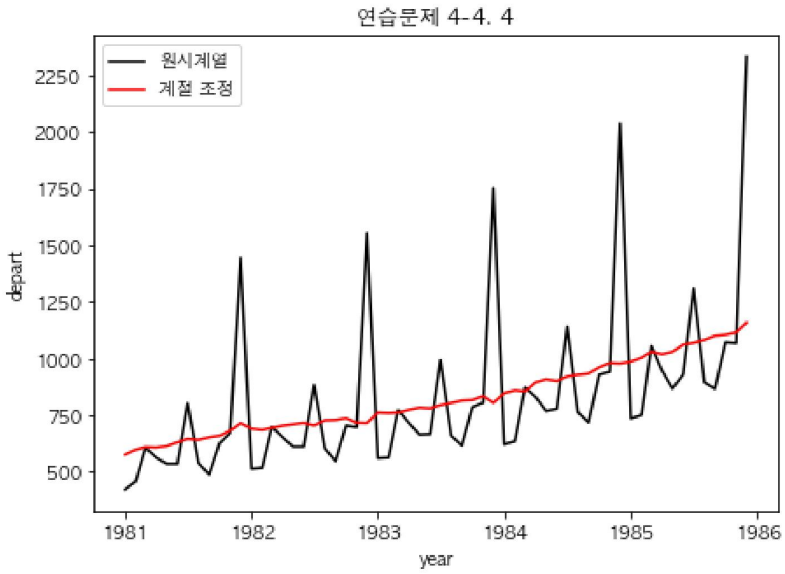


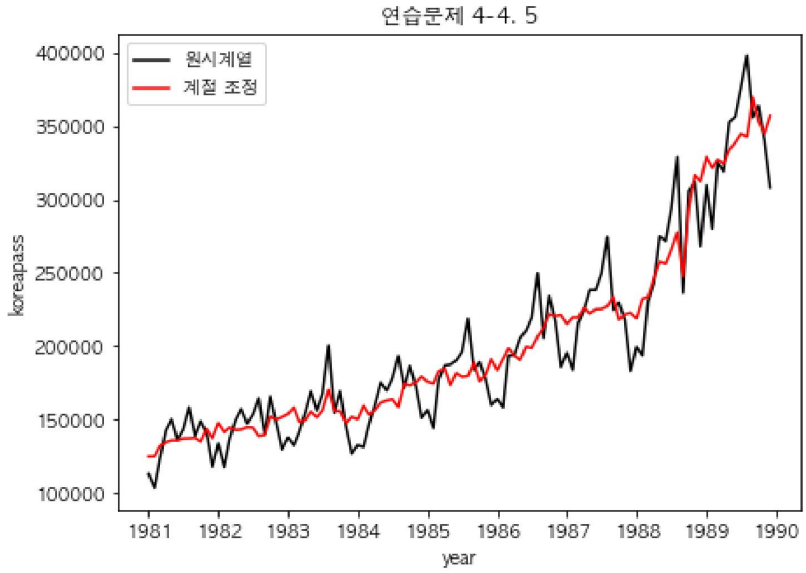
In [12]:
```
# Exercise 4-4
import os

def plot_decomposed_series(fname, number):
    z = []
    fpath = os.path.join("../data", fname)

    with open(fpath) as f:
        for line in f.readlines():
            for elem in line.rstrip().split(" "):
                if len(elem):
                    z.append(float(elem))

    index = pd.date_range(start="1981", periods=len(z), freq="MS")
    data = pd.Series(z, index)

    stl = STL(data, seasonal=13)
    res = stl.fit()

    trend = res.trend
    seasonal = res.seasonal
    irregular = res.resid
    adjseasonal = data - seasonal

    generate_figure(data, adjseasonal, fname.split(".")[0], number)


def generate_figure(data, adj, ylabel, number):
    fig, ax = plt.subplots(figsize=(7, 5))
    ax.plot(data, 'black', label="원시계열")
    ax.plot(adj, 'red', label="계절 조정")
    ax.set_xlabel("year")
    ax.set_ylabel(ylabel)
    ax.set_title(f"연습문제 4-4. {number}")
    plt.legend()
    plt.show()


for i, fname in enumerate(["build.txt", "export.txt", "usapass.txt", "depart.txt"
    plot_decomposed_series(fname, i + 1)
```




연습문제 4-4. 2
원시계열
계절 조정
export
year
20
30
40
50
60
70
1981
1982
1983
1984
1985
1986
1987
1988


연습문제 4-4. 3
원시계열
계절 조정
usapass
year
100
200
300
400
500
600
1982
1984
1986
1988
1990
1992


연습문제 4-4. 4
원시계열
계절 조정
depart
year
2250
2000
1750
1500
1250
1000
750
500
1981
1982
1983
1984
1985
1986


연습문제 4-4. 5
원시계열
계절 조정
koreapass
year
400000
350000
300000
250000
200000
150000
100000
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990

In [ ]: