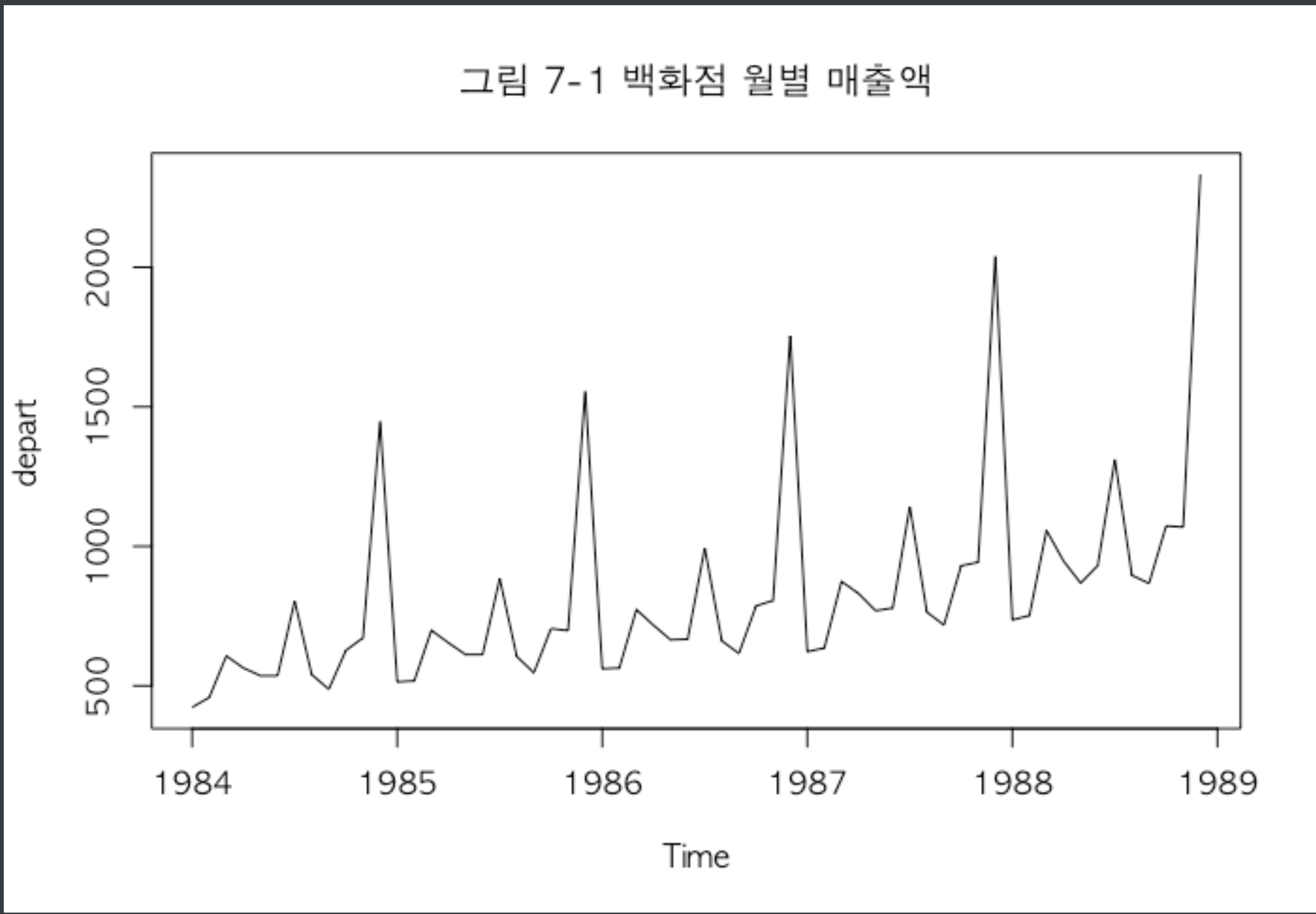
HW_07

2021234640 이종현

Figure 7-1

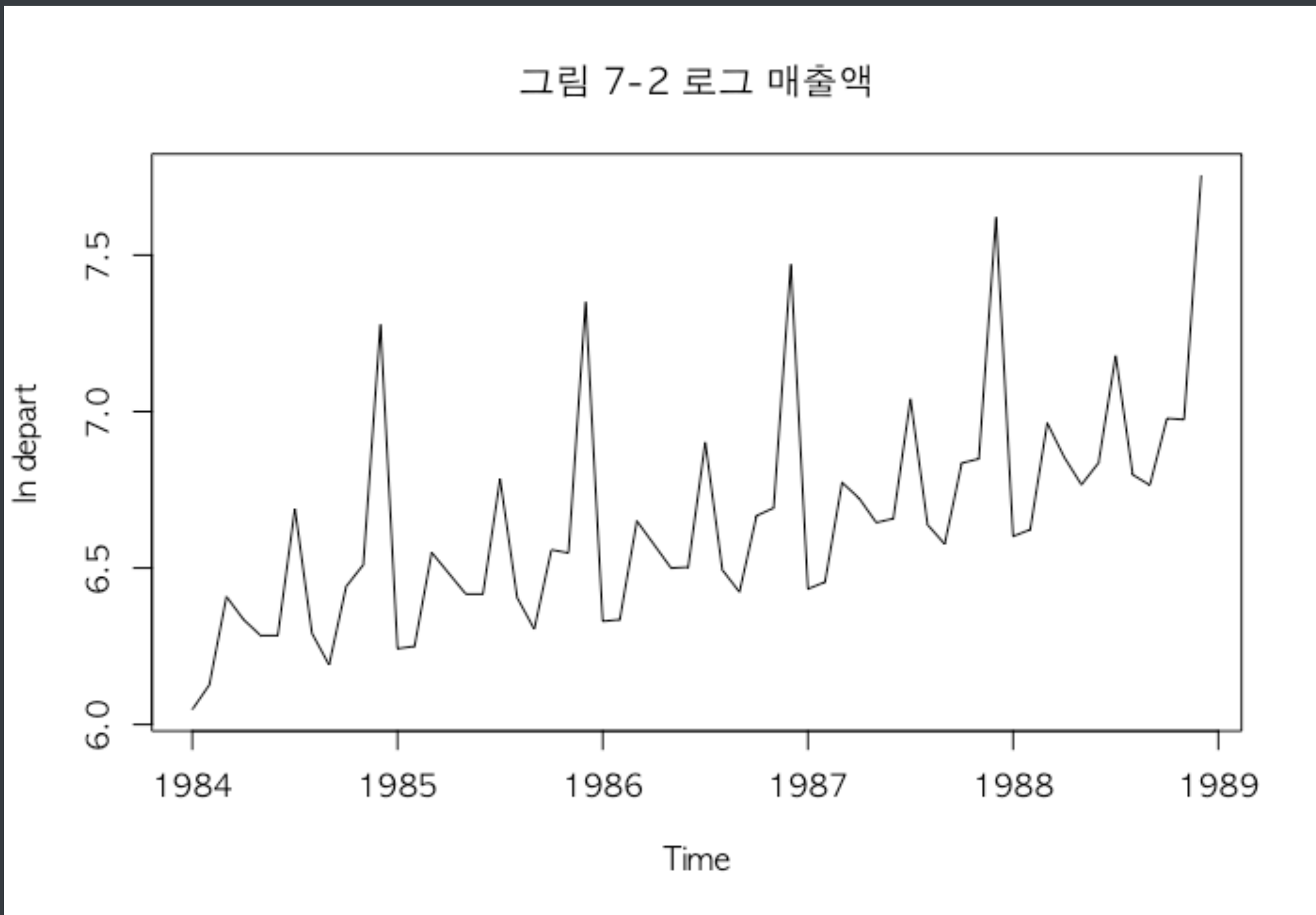
우리가 다루는 대부분의 시계열은 비정상 시계열인 경우가 많다. 때문에 정상 시계열에서 사용할 수 있는 좋은 통계량을 적용하기가 어렵다. 이 때문에 차분 등의 방법이 도입된다. 본 장에서는 시계열의 정상성을 판단하는 방법과 비정상인 경우 차분을 이용하여 정상으로 만드는 방법에 대해 논의한다.



위와 같은 시계열은 선형 추세와 계절 성분이 있음을 확인할 수 있다.

Figure 7-2

시간에 따라 변동폭이 커지는, 즉 분산이 시간에 관련이 있는 시계열은 분산 안정화를 해준다. Log 변환을 취해주면 분산이 안정화된다. 앞서 얻은 백화점 매출액 데이터에 로그를 씌우면 다음과 같아진다.



아까보다 분산이 안정적인 모습으로 잡힌 것을 확인할 수 있다.

Figure 7-3

백화점 매출액 같은 경우에 분산이 시간에 영향을 받아 분산 안정화를 위해 로그 변환을 취해주었다. 이는 추세가 결정적이기 때문에 가능하였다. 추세가 결정적이지 못하고 확률적인 경우, 추세 모형으로 해결할 수 없는 문제가 있다. 예를 들어 아래의 이자율 시계열은 1994년을 기준으로 전혀 다른 추세를 보인다. 즉, 국지적으로는 결정적 추세를 보이지만 전체적으로는 서로 다른 양상을 보이기 때문에 확률적 추세로 결론내릴 수 있으며, 이러한 경우를 동질적 비정상성이라 한다. 이런 시계열은 차분을 통해 정상성을 확보할 수 있

다.

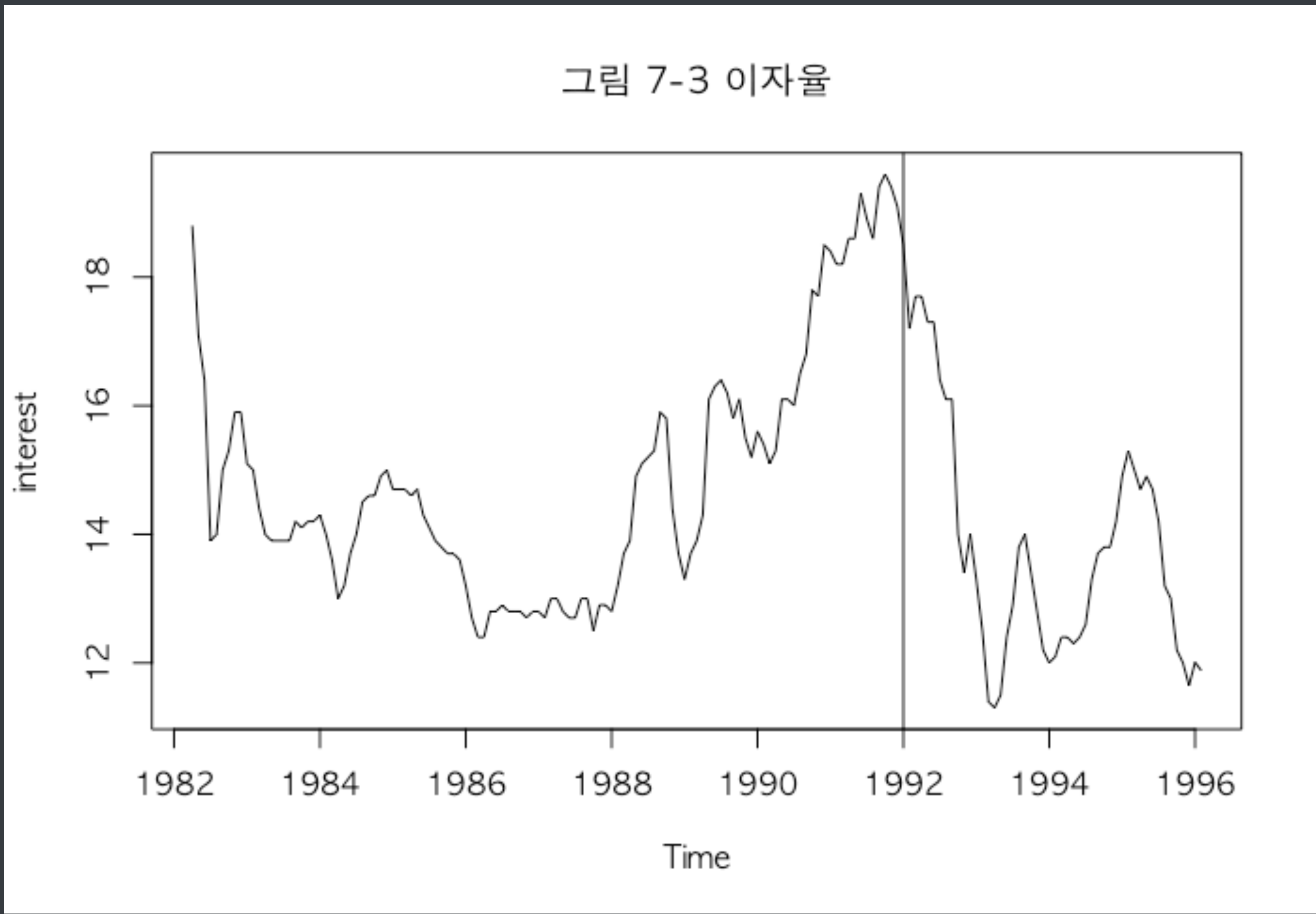


Figure 7-4

random walk process 는 대표적인 비정상 시계열의 예이다. random walk process의 특성상, Z_t 에서 Z_t-1 을 빼주면 error 만 남기 때문에, 차분을 실시하면 정확하게 error 만 남아 정상 시계열이 된다.

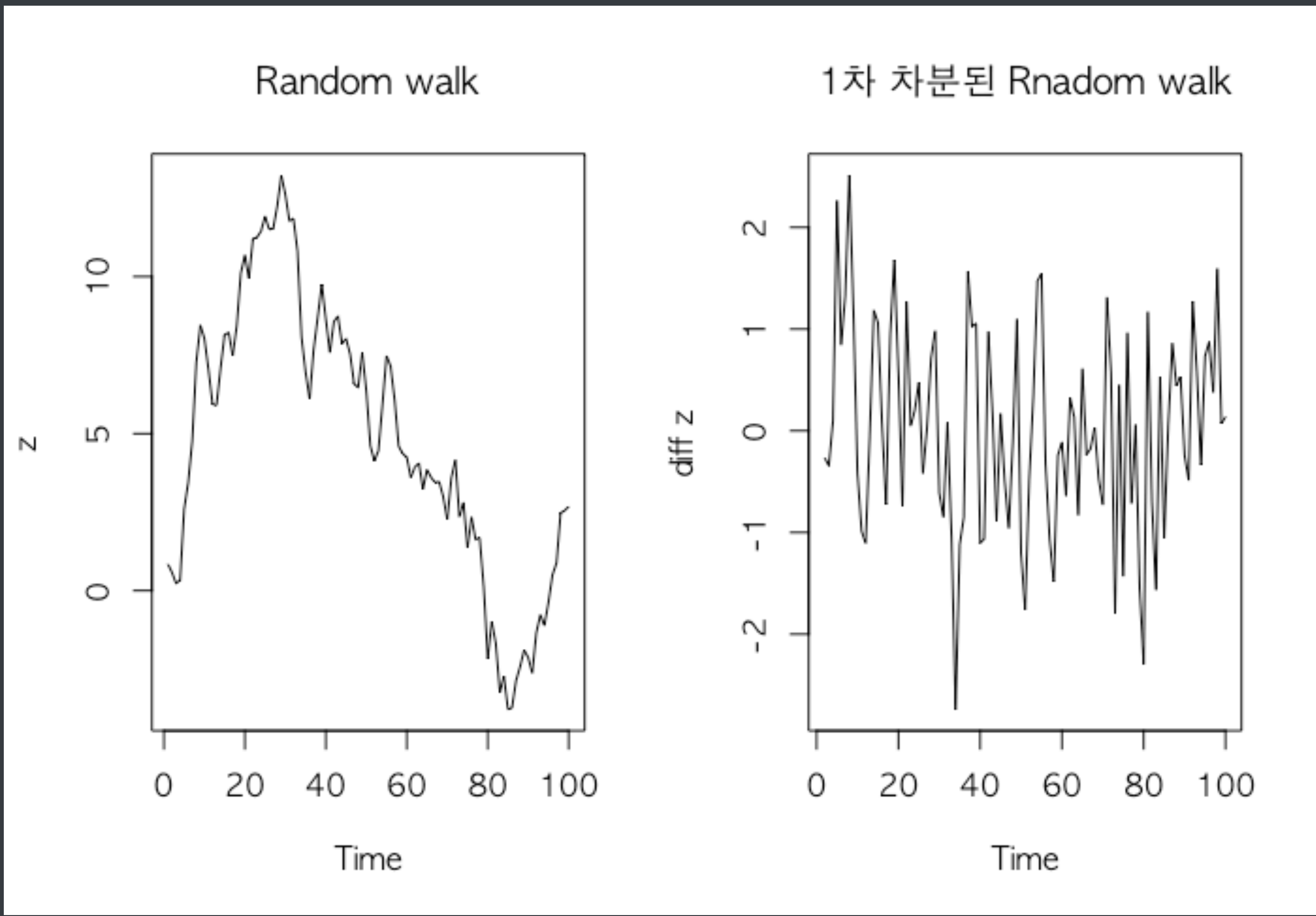
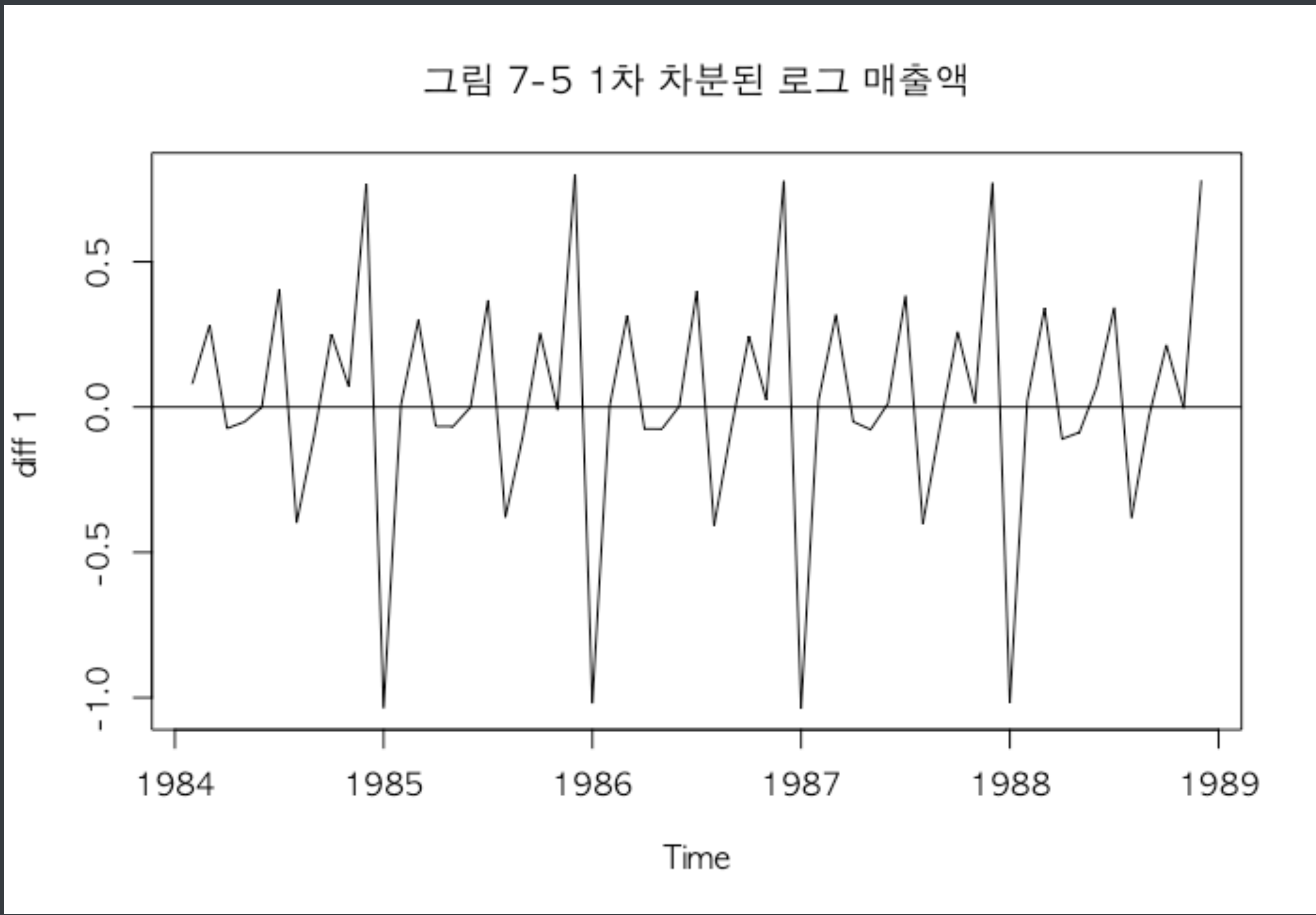


Figure 7-5

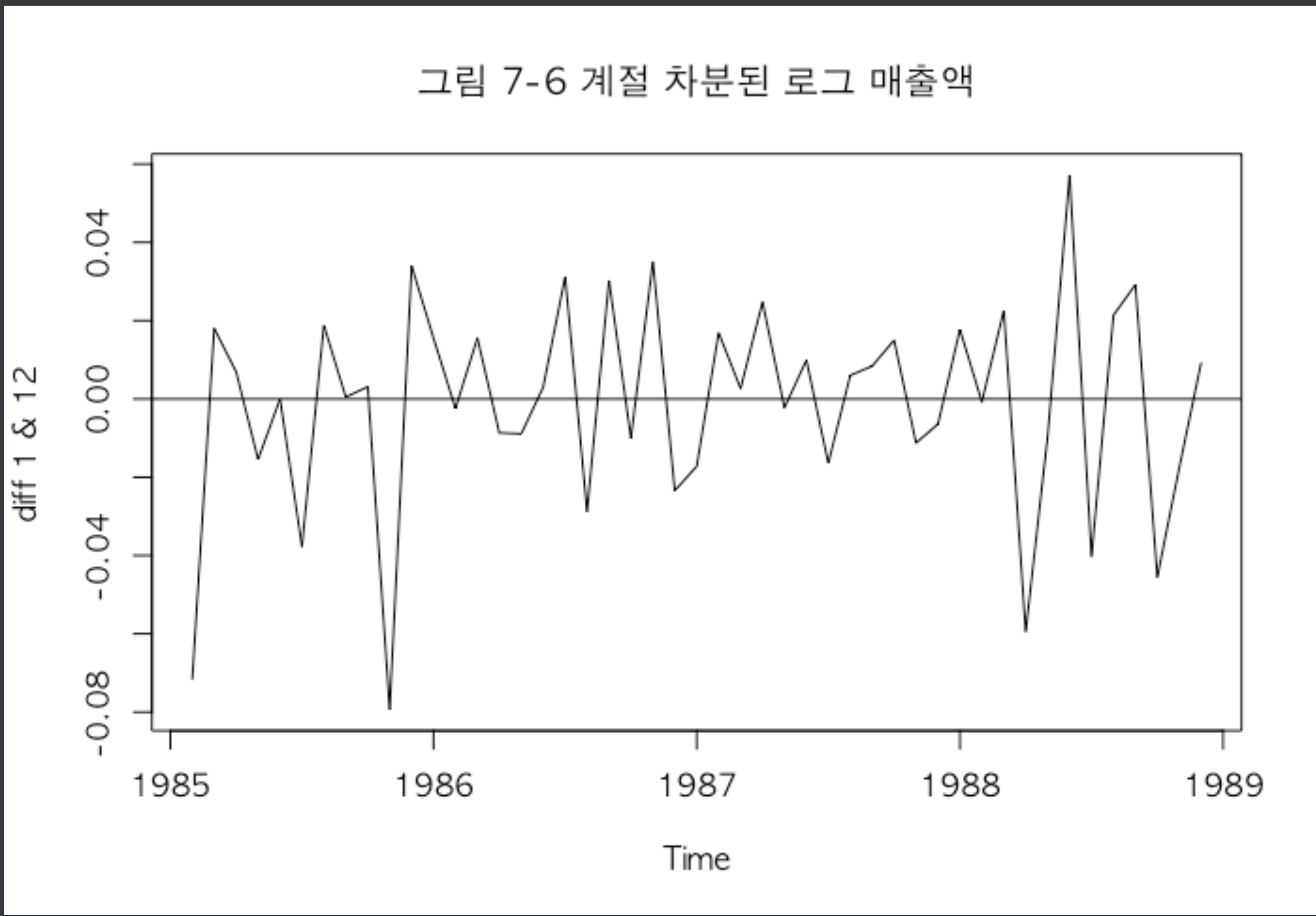
다시 백화점 매출액 데이터로 돌아와 차분을 실시하고자 한다. 일단 선형 추세가 확실히 드러나기 때문에 1차 차분을 실시한다.



선형 추세가 확실히 잡힌 모습을 볼 수 있다. 그러나 여전히 계절 성분이 남아있다.

Figure 7-6

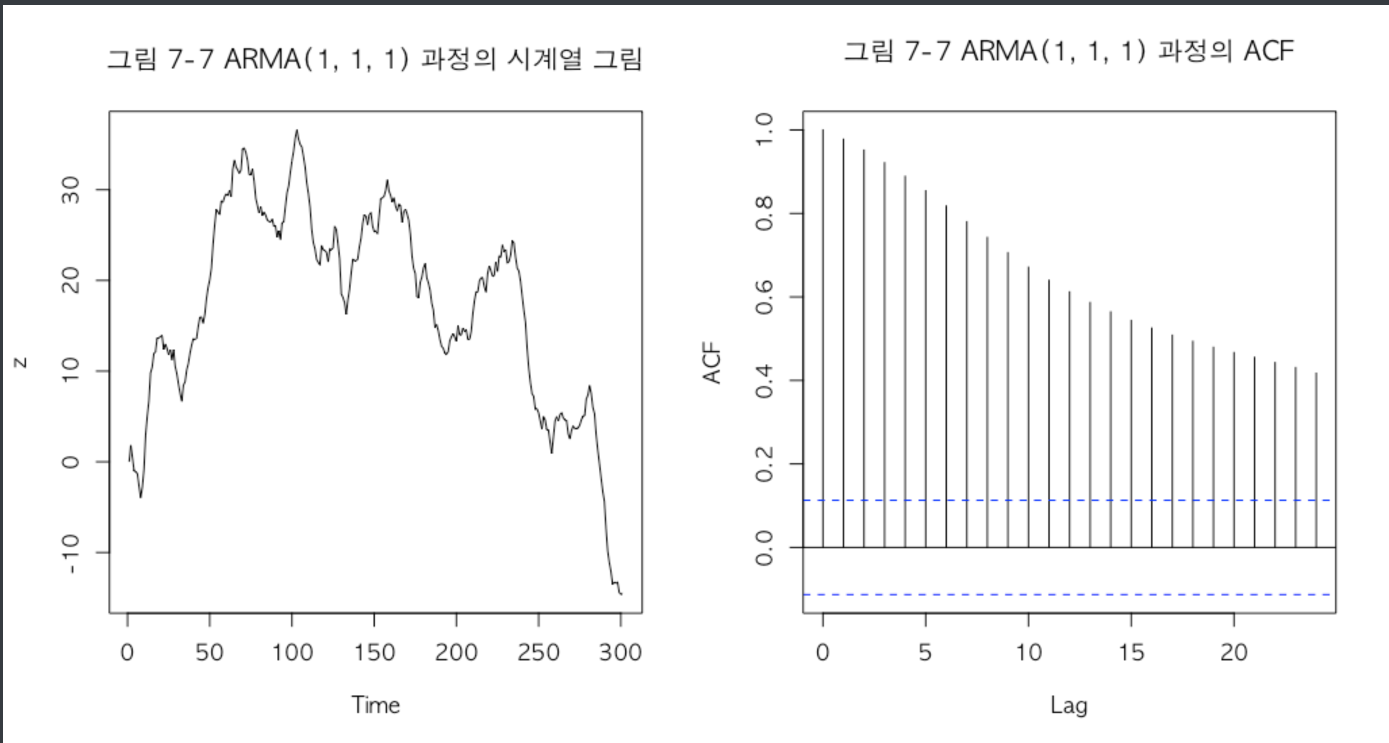
이번에는 lag 를 12 로 두어 계절 성분을 잡아보자. 단, 앞서 선형 추세를 잡은 시계열에서 실시한다.



이제 정상 시계열의 모습을 하고 있으므로 여기에서 시계열 분석 방법을 적용할 수 있다.

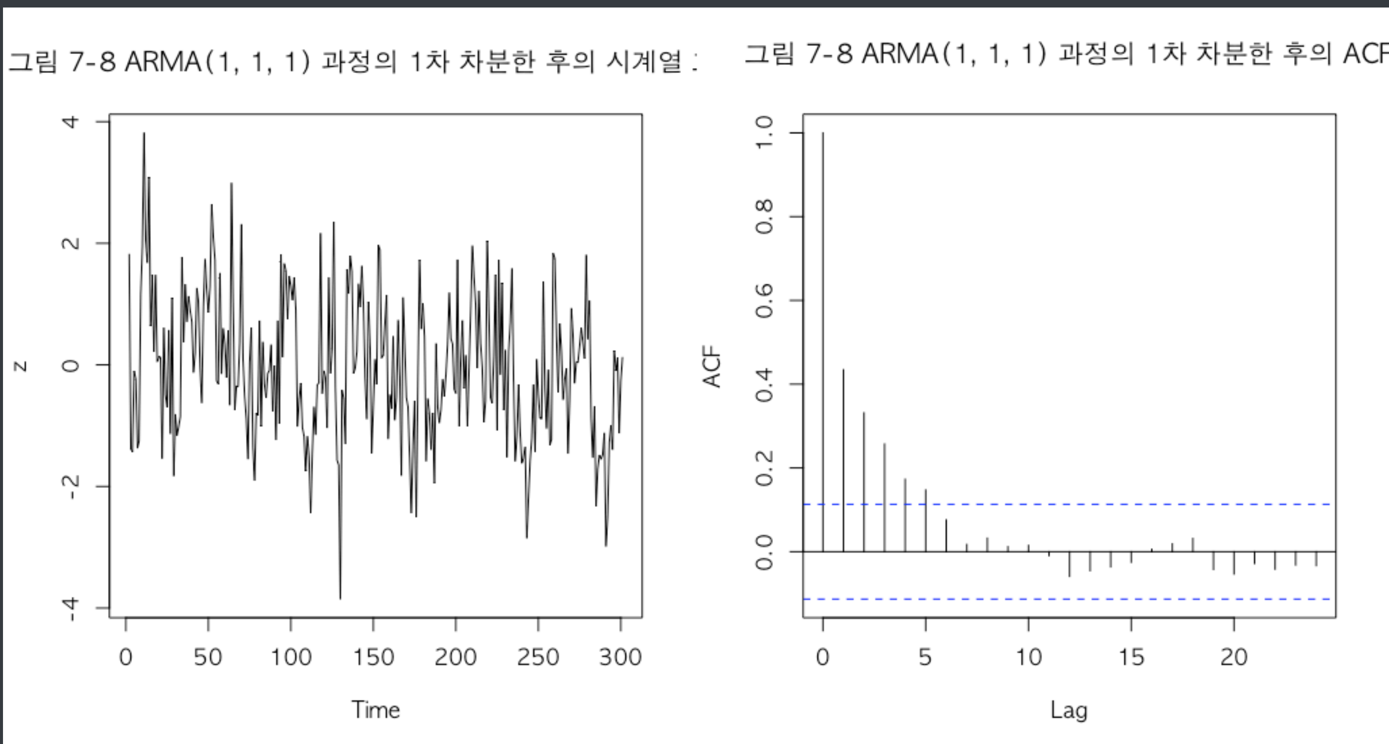
Figure 7-7

ARIMA(1, 1, 1) 프로세스를 시뮬레이션해보자. 이는 이후 차분의 효과를 보기 위함이다.



원 시계열은 위와 같으며, 아래의 그림에서 차분의 결과를 확인할 것이다. ACF 가 아주 천천히 감소하는 패턴을 볼 수 있다.

Figure 7-8



1차 차분을 실시한 결과이다. 시계열의 패턴이 상당히 줄어들었으며, ACF가 빠르게 감소하는 모습을 확인할 수 있다.

Exercise

단위근 검정 요약

단위근 검정 summary

∘ 단위근 검정은 주로 AR process 또는 AR process 가 포함된 확률 과정에 이용된다. 이는 단위근 검정이 해당 확률 과정의 정상성(stationarity)을 확인하는 것이기 때문이다. 이전 장에서 다루었듯, MA process는 무조건 stationarity의 성질을 만족하기 때문에 단위근 검정의 목적에 맞지 않는다.

· AR(1)을 따르는 $z_t - z_{t-1} = \varepsilon_t$ 의 예에서 $z_t - \phi z_{t-1} = \varepsilon_t$ 로 나타낼수 있는 예에서 ϕ는 1로 주어진다.

이 경우 $z_t = \sum_{i=1}^{t} \varepsilon_t$ 로 나타낼수 있으며

$$Cov(z_t, z_{t+k}) = E\left[\left(\sum_{i=1}^{t} \varepsilon_i\right)\left(\sum_{j=1}^{t+k} \varepsilon_j\right)\right]$$

$= t \cdot \sigma_\varepsilon^2$ 으로 나타내서 정상 시계열이 아님을 알 수 있다.

· 비정상 시계열은 차분을 이용해서 정상 시계열로 만들 수 있다. 이 과정을 도식화하면 다음과 같다.

단위근 검정
- 정상성 O → 분석
- 정상성 X → 차분 → 분석

∘ 단위근 검정의 통계량

- LSE를 이용한 ϕ의 추정치, $\hat{\phi}$이 이용된다.
- $\hat{\phi} = \sum_{t=1}^{n} z_{t-1} z_t \Big/ \sum_{t=1}^{n} z_{t-1}^2$
- $|\phi| < 1$ 인 경우, 최소제곱법 성질에 의해

$$\sqrt{n}(\phi - \hat{\phi}) \rightsquigarrow N(0, 1-\phi^2)$$

을 만족한다는 것을 알수 있다.

- $\phi = 1$ 이면 $\sqrt{n}(\hat{\phi} - 1) \to 0$

이므로 점근적 분포가 유용하지 못하다.

$\therefore n(\hat{\phi} - 1)$ 의 점근적 분포를 이용

$$n(\hat{\phi} - 1) \to \frac{\frac{1}{2}\{[W(1)]^2 - 1\}}{\int_0^1 [W(r)]^2 dr},$$

이때 $W(r)$은 brownian motion을 따르는 확률변수이다.

이제 얻은 점근적 분포를 이용하여

$H_0 : \phi = 1$ vs $H_1 : |\phi| < 1$ 을 검정한다.

이것이 Dickey-Fuller 단위근 검정이다.

이때 $n(\hat{\phi} - 1)$의 점근적 분포가 자료 형태에 따라 다르다

i) 절편이 없는 AR process

ii) 절편이 있는 AR process

iii) 절편과 추세항이 있는 AR process

이때 실제 모형과 적합 모형은 다음과 같다.

i) 실제모형: $Z_t = Z_{t-1} + \varepsilon_t$. $\varepsilon_t \sim WN(0, \sigma_\varepsilon^2)$

적합모형: $Z_t = \phi Z_{t-1} + \varepsilon_t$.

ii) 실제 모형: $Z_t = Z_{t-1} + \varepsilon_t$, $\varepsilon_t \sim WN(0, \sigma_\varepsilon^2)$

적합모형: $Z_t = \phi Z_{t-1} + \delta + \varepsilon_t$.

iii) 실제 $Z_t = \delta + Z_{t-1} + \varepsilon_t$. $\varepsilon_t \sim WN(0, \sigma_\varepsilon^2)$

적합: $Z_t = \delta + \phi Z_{t-1} + \beta t + \varepsilon_t$

위 세 가지는 데이터를 보고 결정하면 된다.

∘Augmented Ducky-Fuller 검정
단위근

- 기존 DF 단위근 검정은 AR(1)에 대해 적용하는 반면 ADF는 AR(p)에 대해 적용한다.

$AR(p) \rightarrow (1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3 - \cdots - \phi_p B^p) Z_t = \varepsilon_t$

ϕ 와 ξ_j 는 $\phi = \phi_1 + \phi_2 + \phi_3 + \cdots + \phi_p$

$\xi_j = [\phi_{j+1} + \phi_{j+2} + \cdots + \phi_p]$

$j = 1, 2 \cdots, p-1$

이라면 AR(p)는

$[(1-\phi B) - (\xi_1 B + \xi_2 B^2 + \cdots + \xi_{p-1} B^{p-1})(1-B)] Z_t = \varepsilon_t$

$\Leftrightarrow Z_t = \phi Z_{t-1} + \xi_1 \nabla Z_{t-1} + \xi_2 \nabla Z_{t-2} + \cdots + \xi_{p-1} \nabla Z_{t-p+1} + \varepsilon_t$

으로 정리할 수 있다.

이때 $\phi = 1$ 이면 Z_t가 단위근을 가짐을 의미한다

ADF는 $\phi = 1$ 여부를 검정하며

$$Z_t = \frac{(\hat{\phi} - 1)}{\hat{\sigma}_{\hat{\phi}}}$$, 단 $\hat{\phi}$는 LSE 추정량.

$Z(\phi) = n(\lambda / \sigma)(\hat{\phi} - 1)$, $\lambda / \sigma = (1 - \xi_1 - \xi_2 - \cdots - \xi_{p-1})^{-1}$

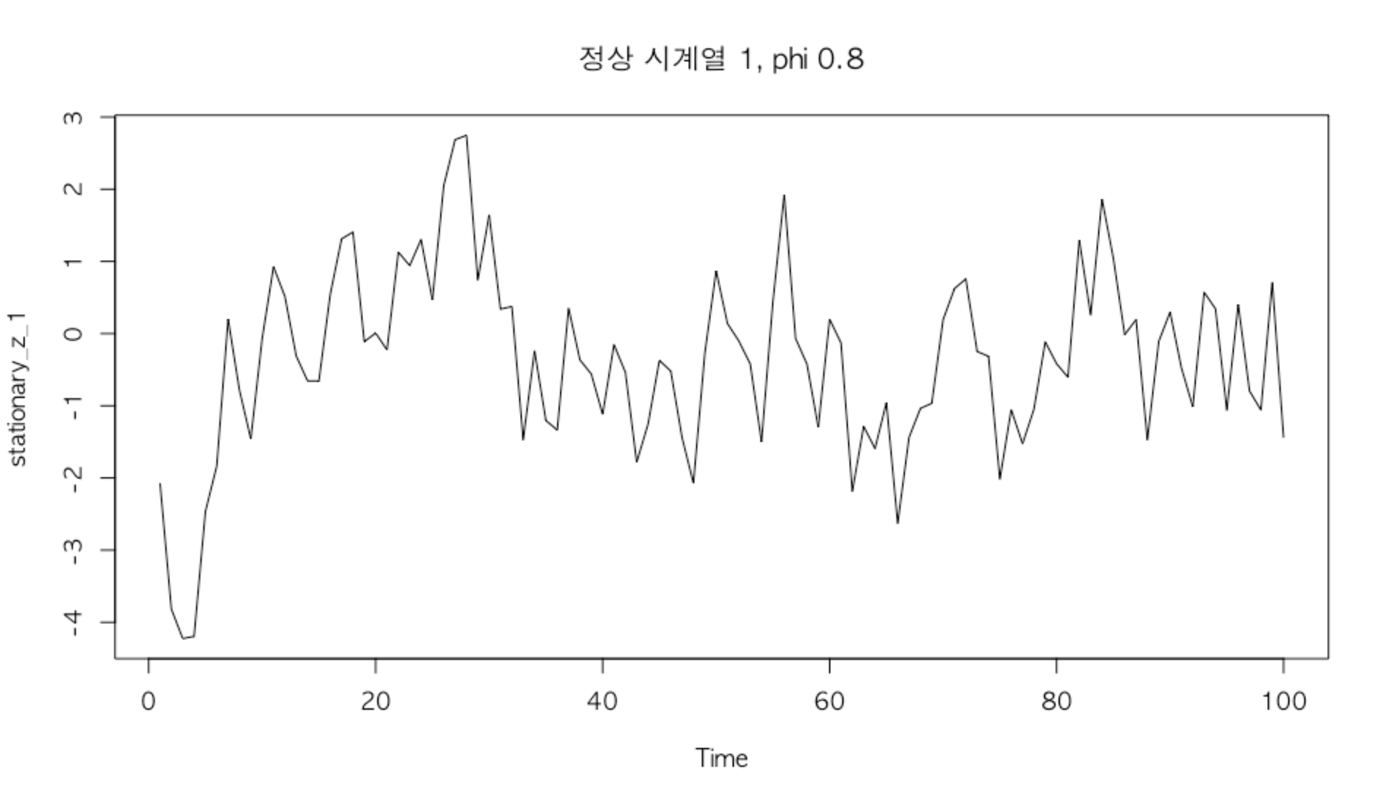
세가지 Case로 나누는 것은 DF와 유사하다.

∘Phillips - perron 비모수

PP는 AR 확률과정을 구한 후 오차를 MA로 가정하여 검정한다.

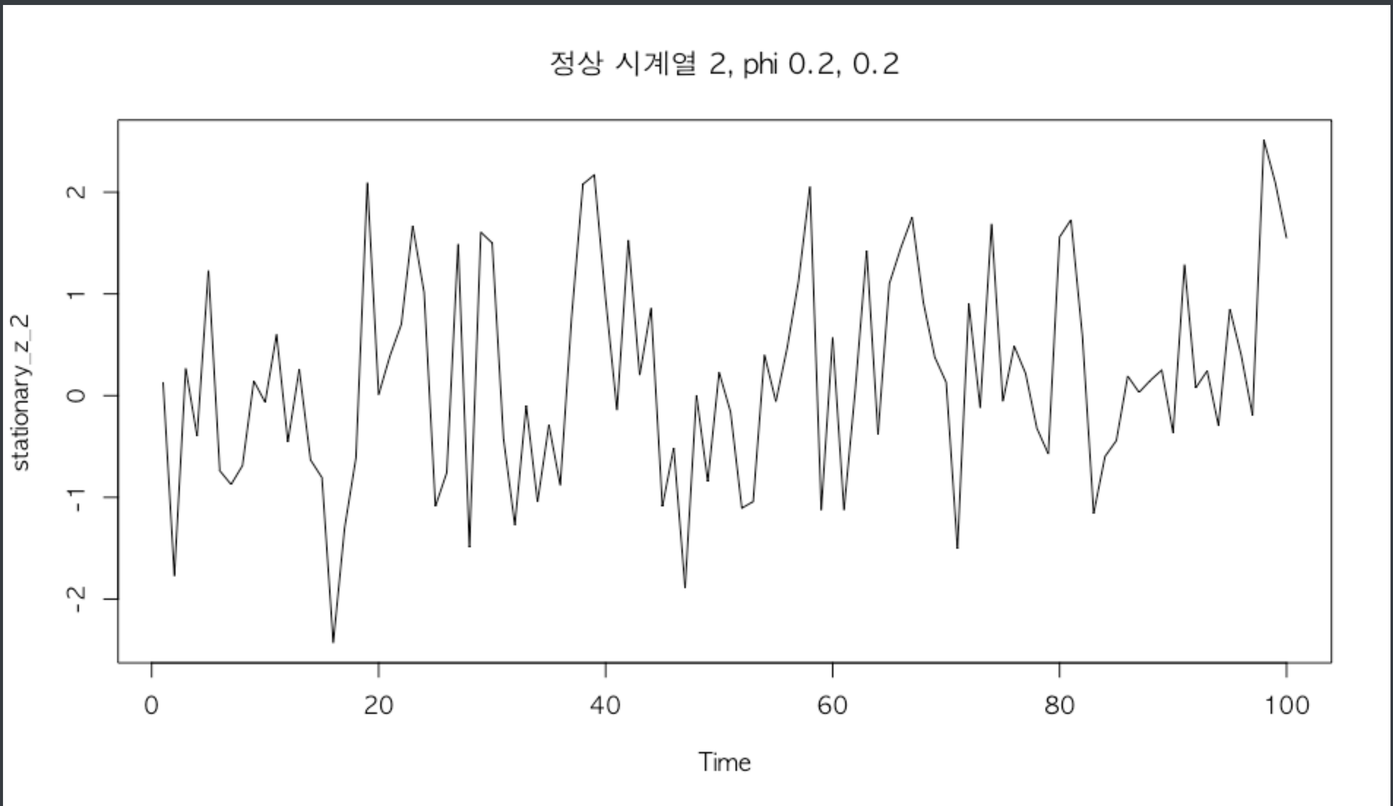
이 역시 3가지의 다른 경우를 가지는 단위근 검정을 실시한다.

정상 시계열




```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -3.7405
  P VALUE:
    0.01
```

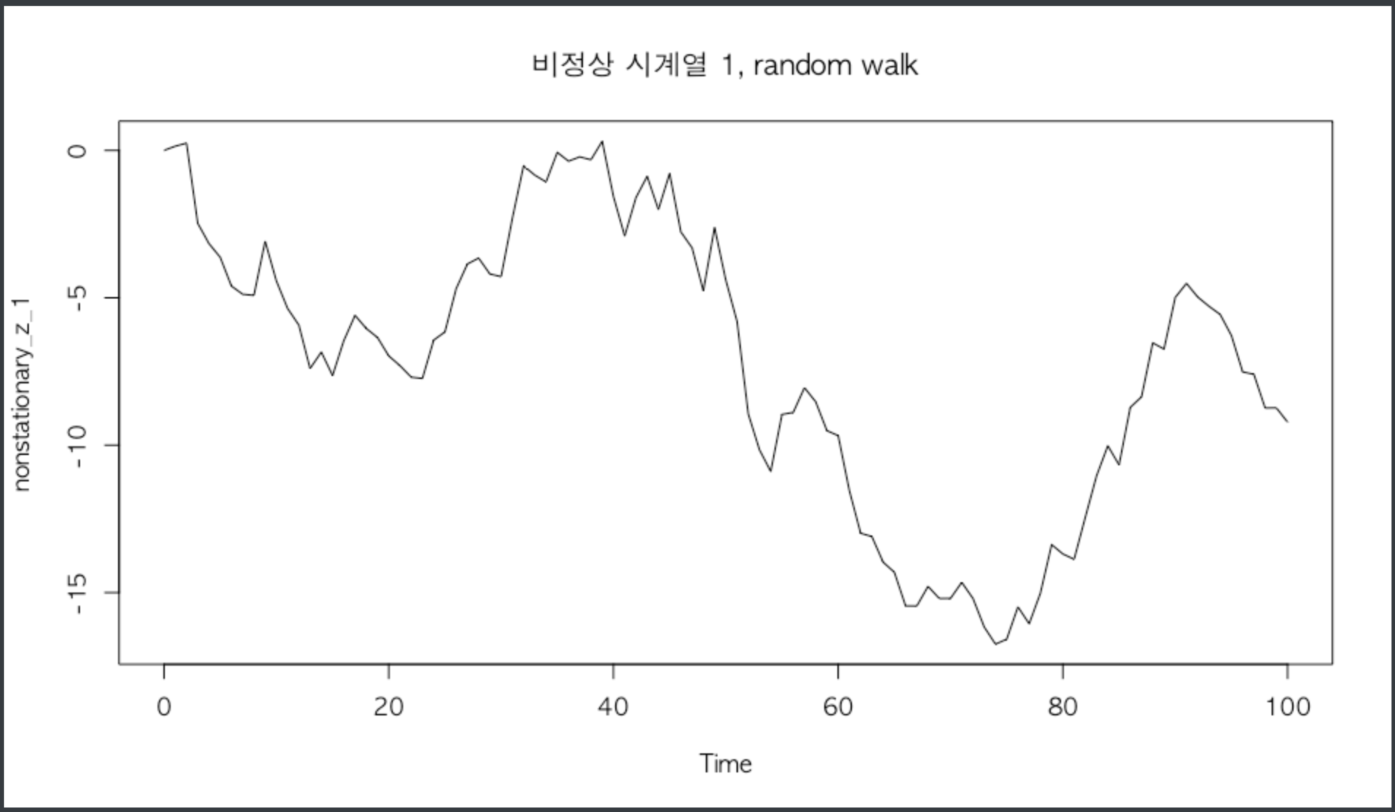
귀무 가설을 기각하고 정상성 가설 채택




```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -5.8487
  P VALUE:
    0.01
```

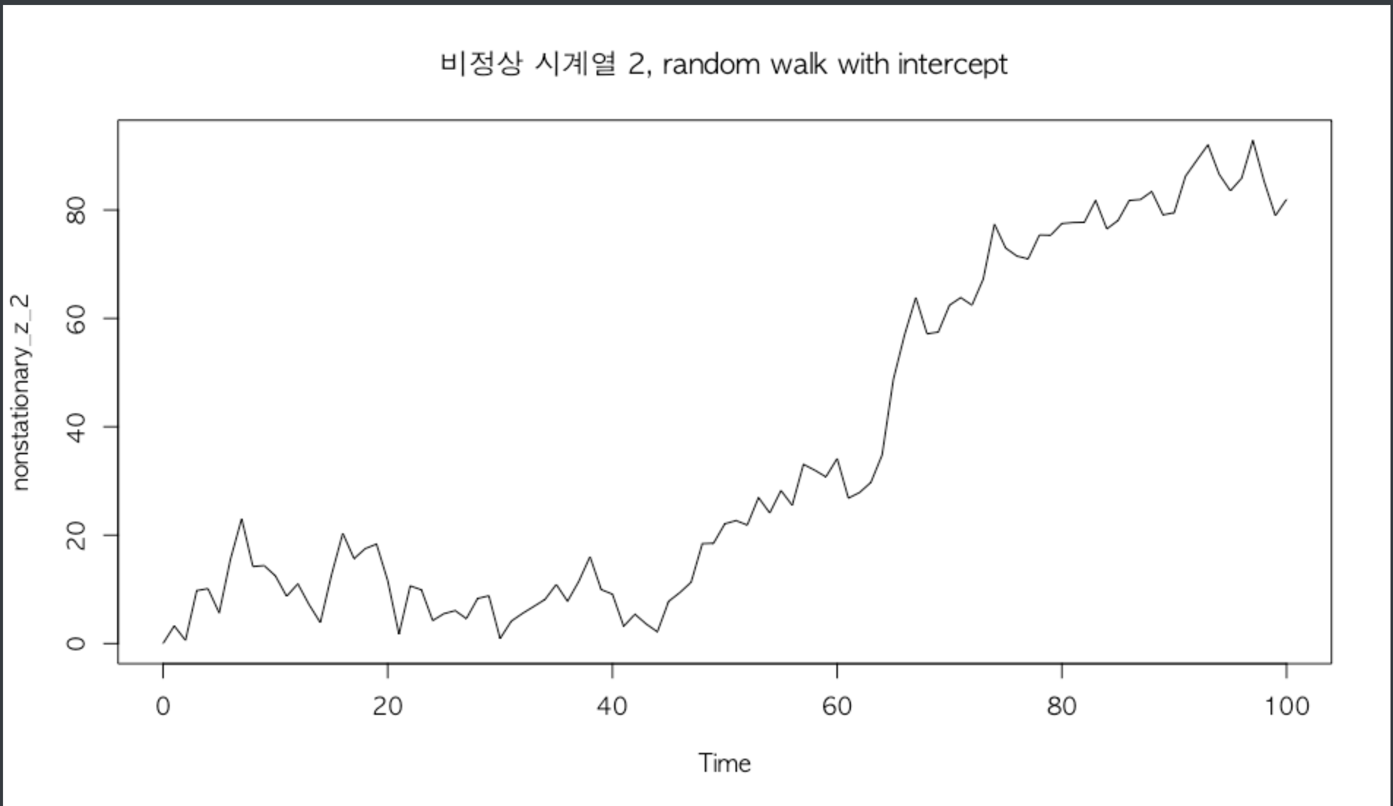
마찬가지로 귀무 가설 기각

비정상 시계열




```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -1.0688
  P VALUE:
    0.2725
```

귀무 가설을 기각하지 못함. 정상 시계열이 아님




```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: 1.283
  P VALUE:
    0.9491
```

귀무 가설 기각 X. 비정상 시계열임.

Appedix, R code

```
rm(list=ls())

setwd("Workspace/2022-Fall_TimeSeriesAnalysis/")
```

```r
par(family="AppleGothic")

# Figure 7-1
z = scan("data/depart.txt")
dept = ts(z, start=c(1984, 1), frequency=12)
ldept=log(dept)
dif_1 = diff(ldept, lag=1)
dif_12 = diff(ldept, lag=12)
dif_112 = diff(dif_1, lag=12)

ts.plot(dept, ylab="depart", main="그림 7-1 백화점 월별 매출액")

ts.plot(ldept, ylab="ln depart", main="그림 7-2 로그 매출액")

ts.plot(dif_1, ylab="diff 1", main="그림 7-5 1차 차분된 로그 매출액")
abline(h=0)

ts.plot(dif_12, ylab="diff 12", main="계절 차분된 로그 매출액")

ts.plot(dif_112, ylab="diff 1 & 12", main="그림 7-6 계절 차분된 로그 매출액")
abline(h=0)

# Figure 7-3
z = scan("data/interest.txt")
interest = ts(z, start=c(1982, 4), frequency=12)
ts.plot(interest, ylab="interest", main="그림 7-3 이자율")
abline(v=1992)

# Figure 7-4
set.seed(123456)
z = ts(cumsum(rnorm(100,.01, 1)))
difz = diff(z, lag=1)
par(mfrow=c(1, 2))

ts.plot(z, ylab="z", main="Random walk")
ts.plot(difz, ylab="diff z", main="1차 차분된 Rnadom walk")
```

```r
# Figure 7.7~7.10
set.seed(16732)
par(mfrow=c(1, 2))
z = arima.sim(n=300, list(order=c(1,1,1), ar=0.8, ma=-0.5),
rand.gen=rnorm)

# Figure 7.7
ts.plot(z, ylab="z", main="그림 7-7 ARMA(1, 1, 1) 과정의 시계열 그림")
acf(z, maxlag=24, main="그림 7-7 ARMA(1, 1, 1) 과정의 ACF")

# Figure 7.7
diff_z = diff(z, lag=1)
ts.plot(diff_z, ylab="z", main="그림 7-8 ARMA(1, 1, 1) 과정의 1차 차분한 후의
시계열 그림")
acf(diff_z, maxlag=24, main="그림 7-8 ARMA(1, 1, 1) 과정의 1차 차분한 후의
ACF")


# or
t = 1:300
z = rep(0, 302)
a1 = rnorm(1)
for (i in 1:300){
  a = rnorm(1)
  z[i+2] = 1.8 * z[i+1] - 0.8 * z[i] + a - 0.5 * a1
  a1 = a
}
z = z[3:302]
ts.plot(z, ylab="z",
main=expression(ARIMA(1,1,1)~~~~~phi==0.8~~theta==0.5))


# Exercise
library(fUnitRoots)
par(mfrow=c(1, 1))
```

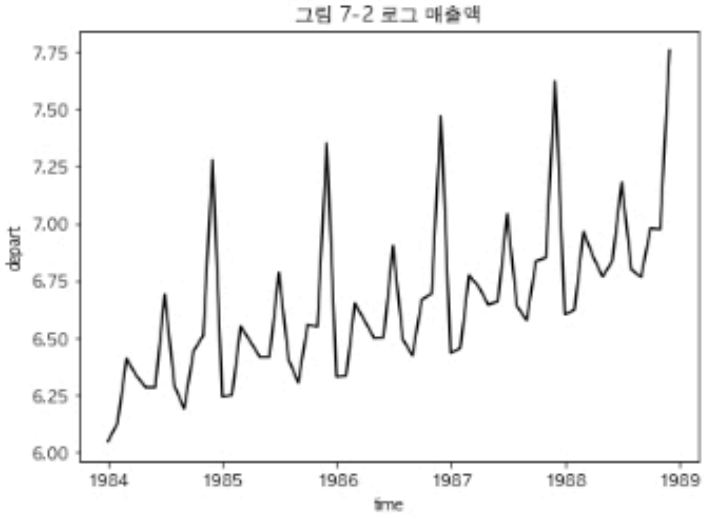
```r
stationary_z_1 = arima.sim(n=100, list(order=c(1,0,0), ar=0.8))
ts.plot(stationary_z_1, main="정상 시계열 1, phi 0.8")
adfTest(stationary_z_1)

stationary_z_2 = arima.sim(n=100, list(order=c(2,0,0), ar=c(0.2, 0.2)))
ts.plot(stationary_z_2, main="정상 시계열 2, phi 0.2, 0.2")
adfTest(stationary_z_2)

nonstationary_z_1 = arima.sim(model=list(order=c(0, 1, 0)), n=100)
ts.plot(nonstationary_z_1, main="비정상 시계열 1, random walk")
adfTest(nonstationary_z_1)

nonstationary_z_2 = arima.sim(model=list(order=c(0, 1, 0)), n=100,
mean=1,sd=5)
ts.plot(nonstationary_z_2, main="비정상 시계열 2, random walk with
intercept")
adfTest(nonstationary_z_2)
```

Appendix, Python code

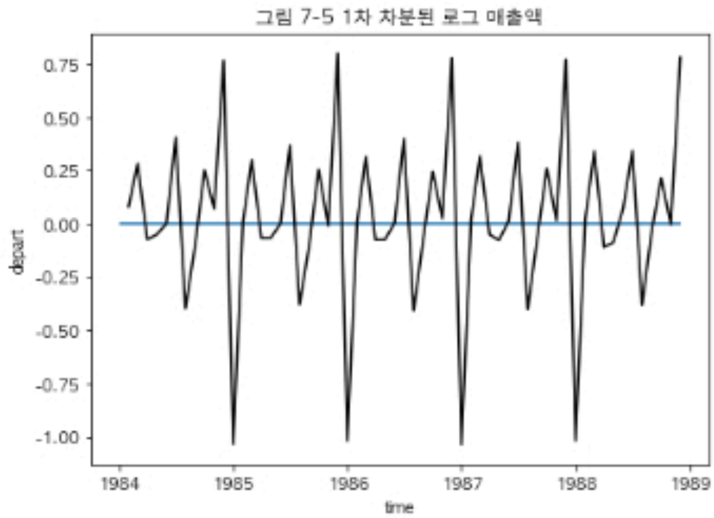
```
In [1]: import math
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        plt.rc('font', family='AppleGothic')
        plt.rcParams['axes.unicode_minus'] = False
```

```
In [4]: # Example 7-1
        z = []

        with open('../data/depart.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="1984", periods=len(z), freq="MS")
        data = pd.Series(z, index)

        fig, ax = plt.subplots(figsize=(7, 5))
        ax.plot(data, 'black')
        ax.set_xlabel("time")
        ax.set_ylabel("depart")
        ax.set_title("그림 7-1 백화점 월별 매출액")
        plt.show()
```

In [5]:
```
ldep = np.log(data)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(ldep, 'black')
ax.set_xlabel("time")
ax.set_ylabel("depart")
ax.set_title("그림 7-2 로그 매출액")
plt.show()
```



In [9]:
```
dif_1 = ldep.diff(periods=1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(dif_1, 'black')
ax.hlines(0, dif_1.index.min(), dif_1.index.max())
ax.set_xlabel("time")
ax.set_ylabel("depart")
ax.set_title("그림 7-5 1차 차분된 로그 매출액")
plt.show()
```



In [11]:
```python
dif_12 = ldep.diff(periods=12)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(dif_12, 'black')
ax.set_xlabel("time")
ax.set_ylabel("depart")
ax.set_title("그림 7-5 계절 차분된 로그 매출액")
plt.show()
```

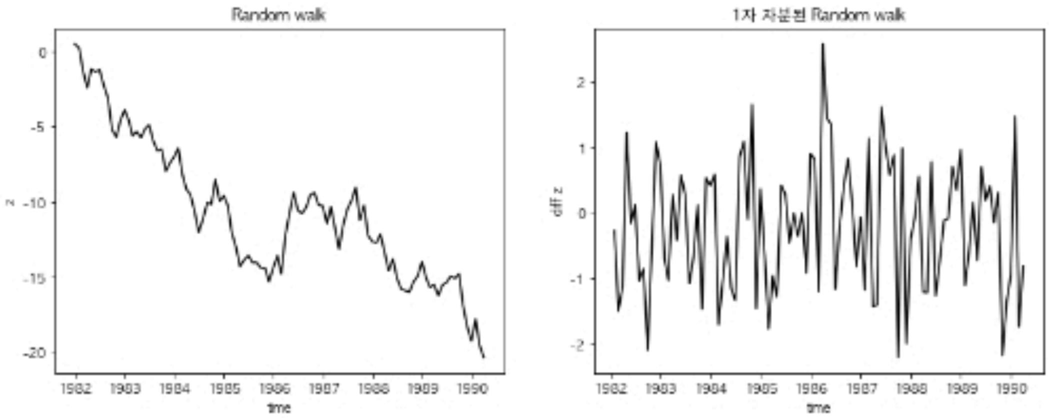
In [15]:
```
dif_112 = dif_1.diff(periods=12)
dif_112 = dif_112.dropna()

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(dif_112, 'black')
ax.hlines(0, dif_112.index.min(), dif_112.index.max())
ax.set_xlabel("time")
ax.set_ylabel("depart")
ax.set_title("그림 7-6 계절 차분된 로그 매출액")
plt.show()
```

In [19]:

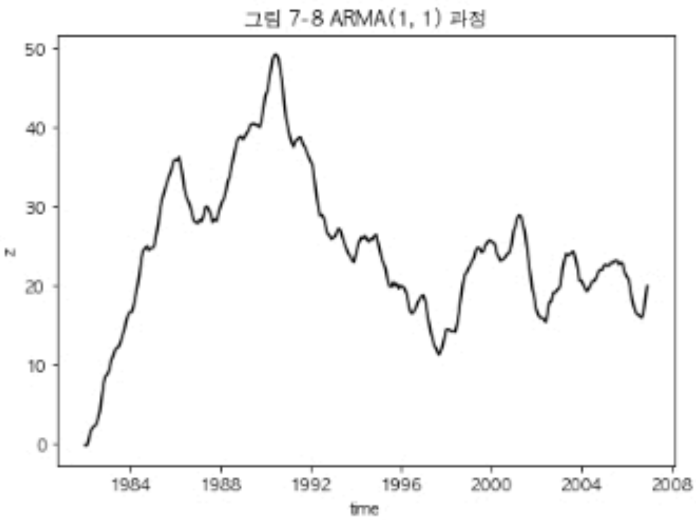
```python
# Example 7-3
z = []

with open('../data/interest.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="1982", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.vlines(pd.to_datetime("1992-01-01"), 10, 20)
ax.set_xlabel("time")
ax.set_ylabel("interest")
ax.set_title("그림 7-3 이자율")
plt.show()
```

In [30]:

```python
# Figure 7-4

np.random.seed(123456)

z = np.cumsum(np.random.normal(0.01, 1, (100)))

index = pd.date_range(start="1982", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))
ax1.plot(data, 'black')
ax1.set_title("Random walk")
ax1.set_xlabel("time")
ax1.set_ylabel("z")

ax2.plot(data.diff(1), 'black')
ax2.set_title("1차 차분된 Random walk")
ax2.set_xlabel("time")
ax2.set_ylabel("diff z")

plt.show()
```



In [36]:
```python
# Figure 7-7 ~ 7-10
np.random.seed(123456)

z = np.zeros(302)
a1 = np.random.normal()

for i in range(300):
    a = np.random.normal()
    z[i+2] = 1.8 * z[i+1] - 0.8 * z[i] - 0.5 * a1
    a1 = a

z = z[2:]

index = pd.date_range(start="1982", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("z")
ax.set_title("그림 7-8 ARMA(1, 1) 과정")
plt.show()
```



In [ ]:

In [ ]: