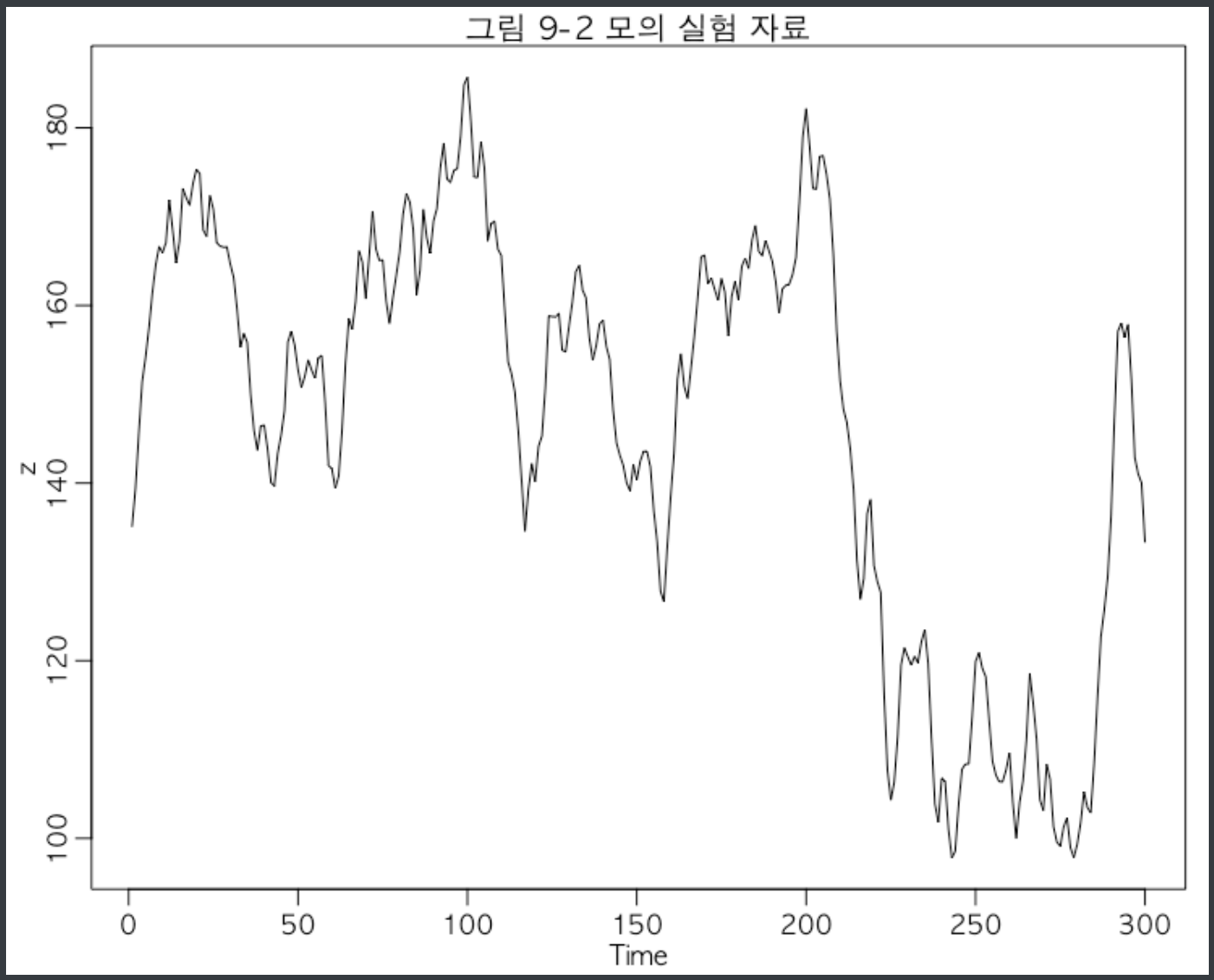
HW 09

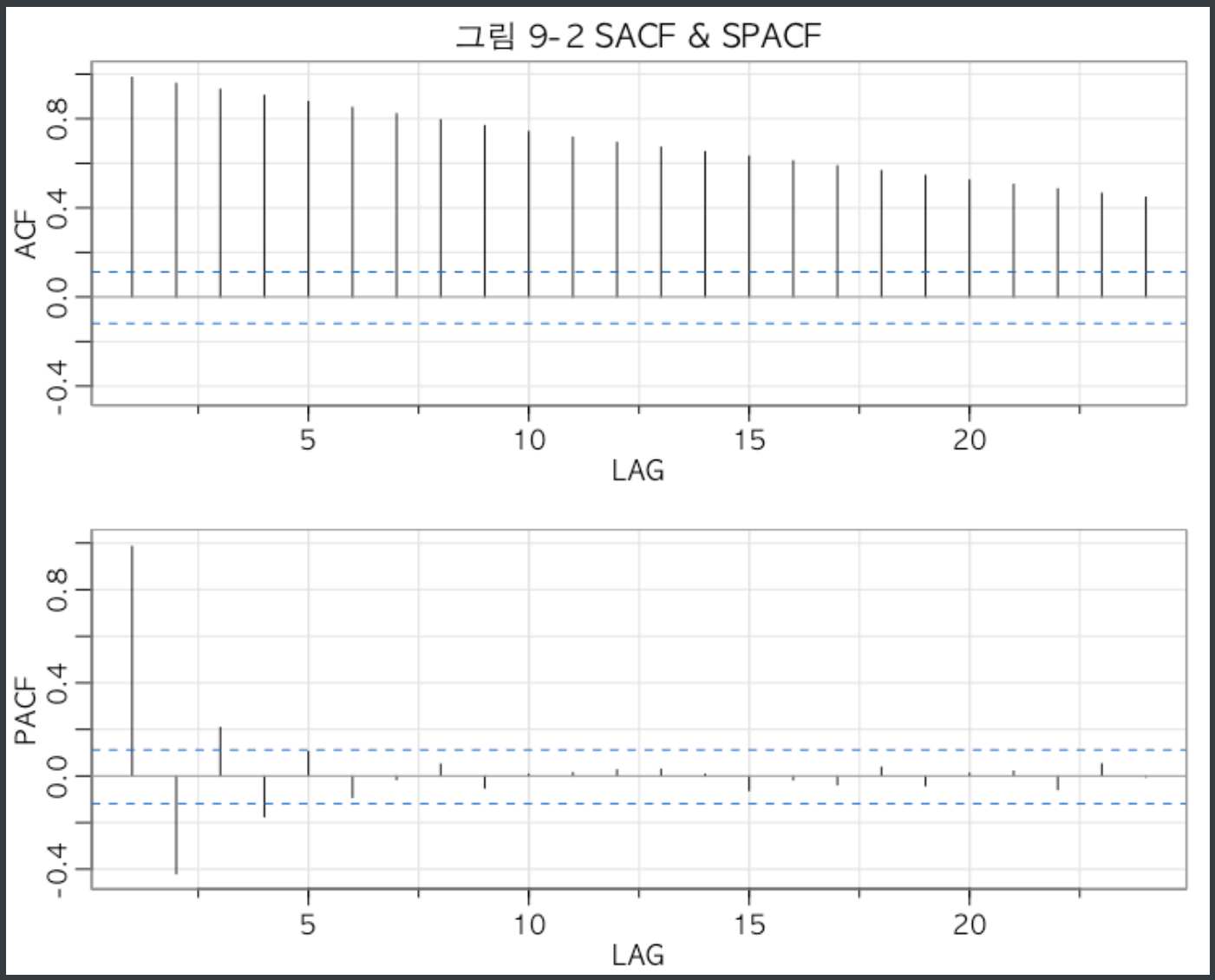
2021234640 이종현

Figure 9-2

다음은 모의 실험 자료의 시계열이다. 한눈에 보아도 정상성을 만족하지 않는 것처럼 보인다.



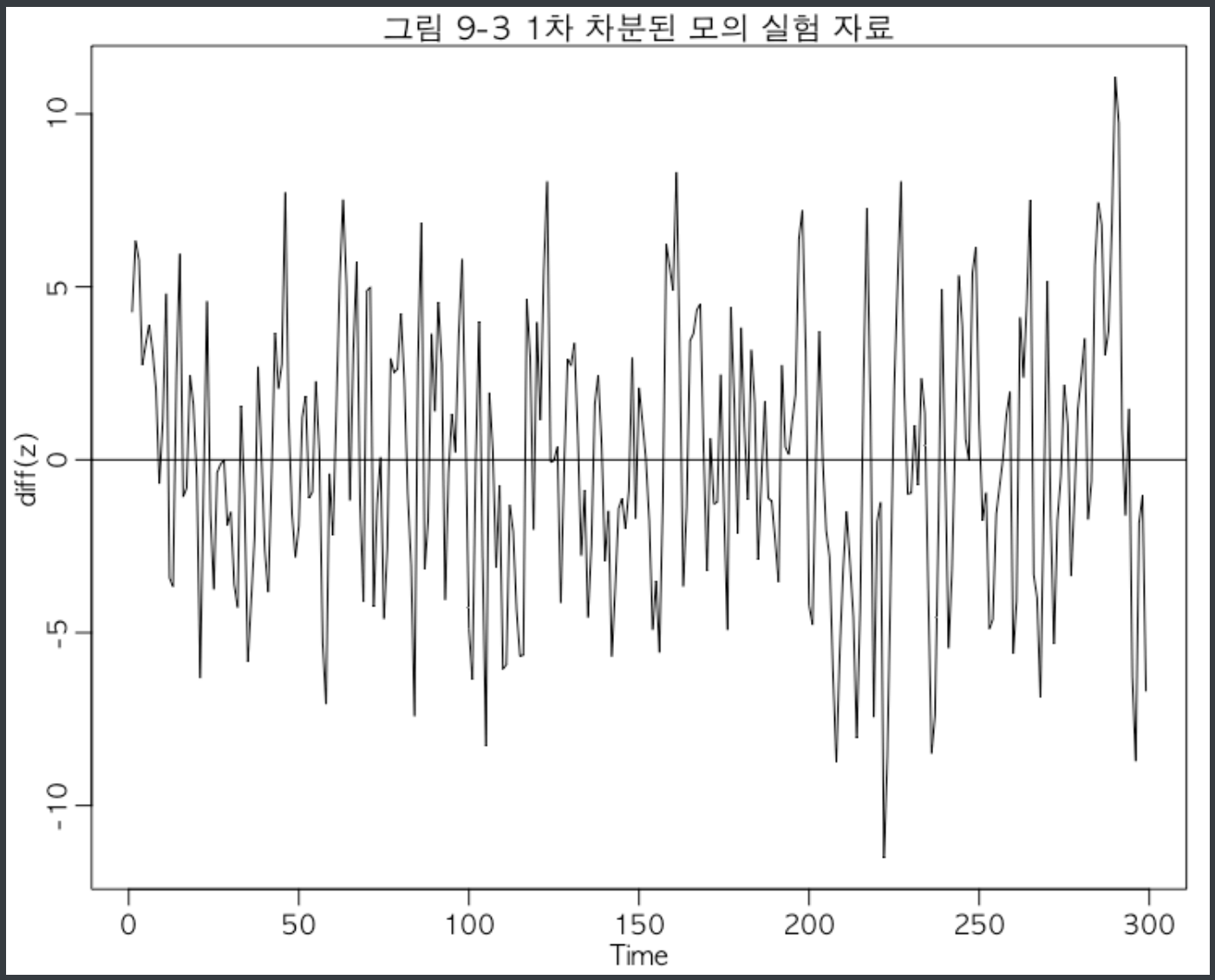
ACF, PACF를 확인해보자.



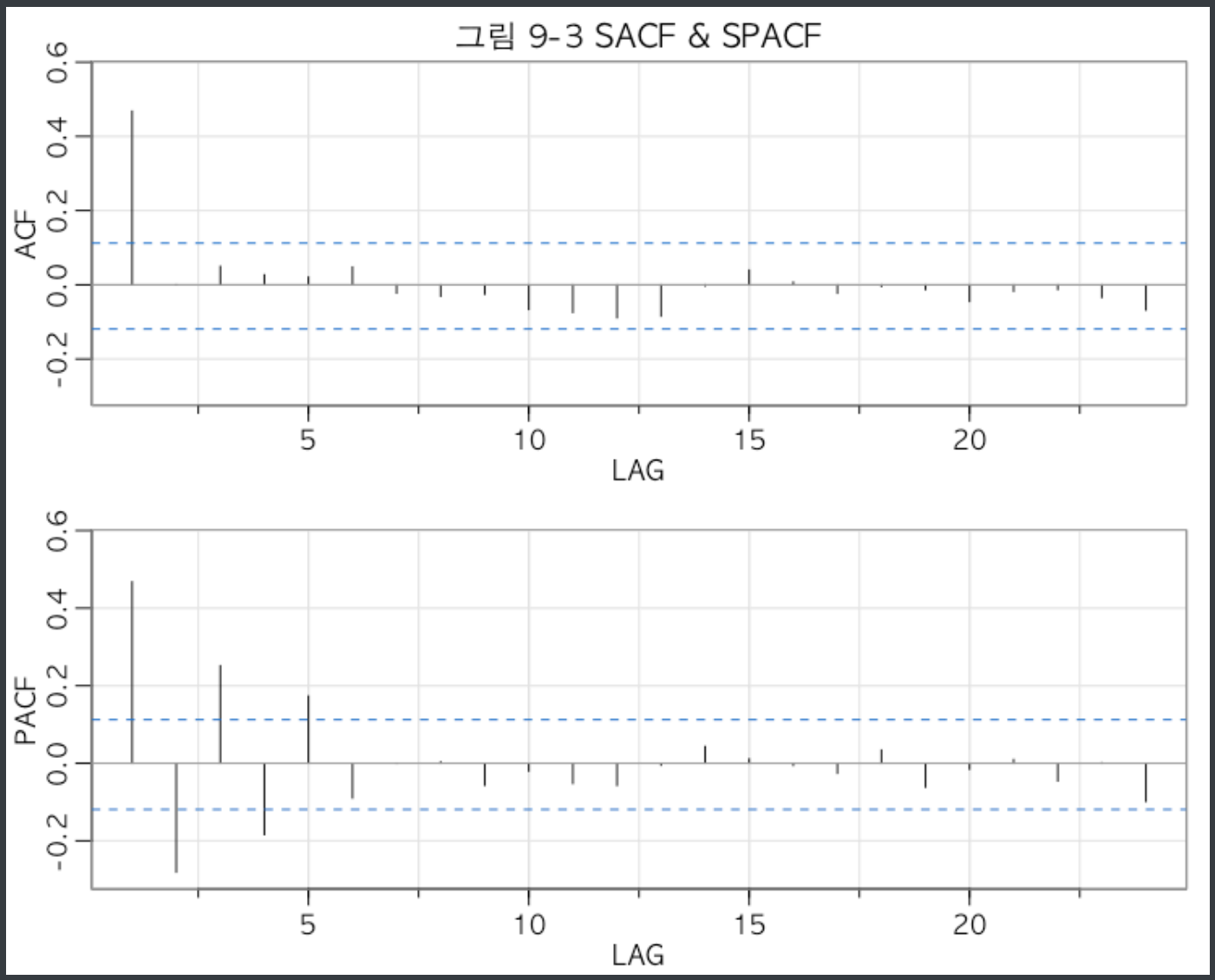
PACF 1, 2 시점에서의 값이 크다. ACF 가 매우 천천히 감소하기 때문에 차분을 고려해볼 수 있다.

Figure 9-3

앞선 시계열을 1차 차분을 진행하였다.

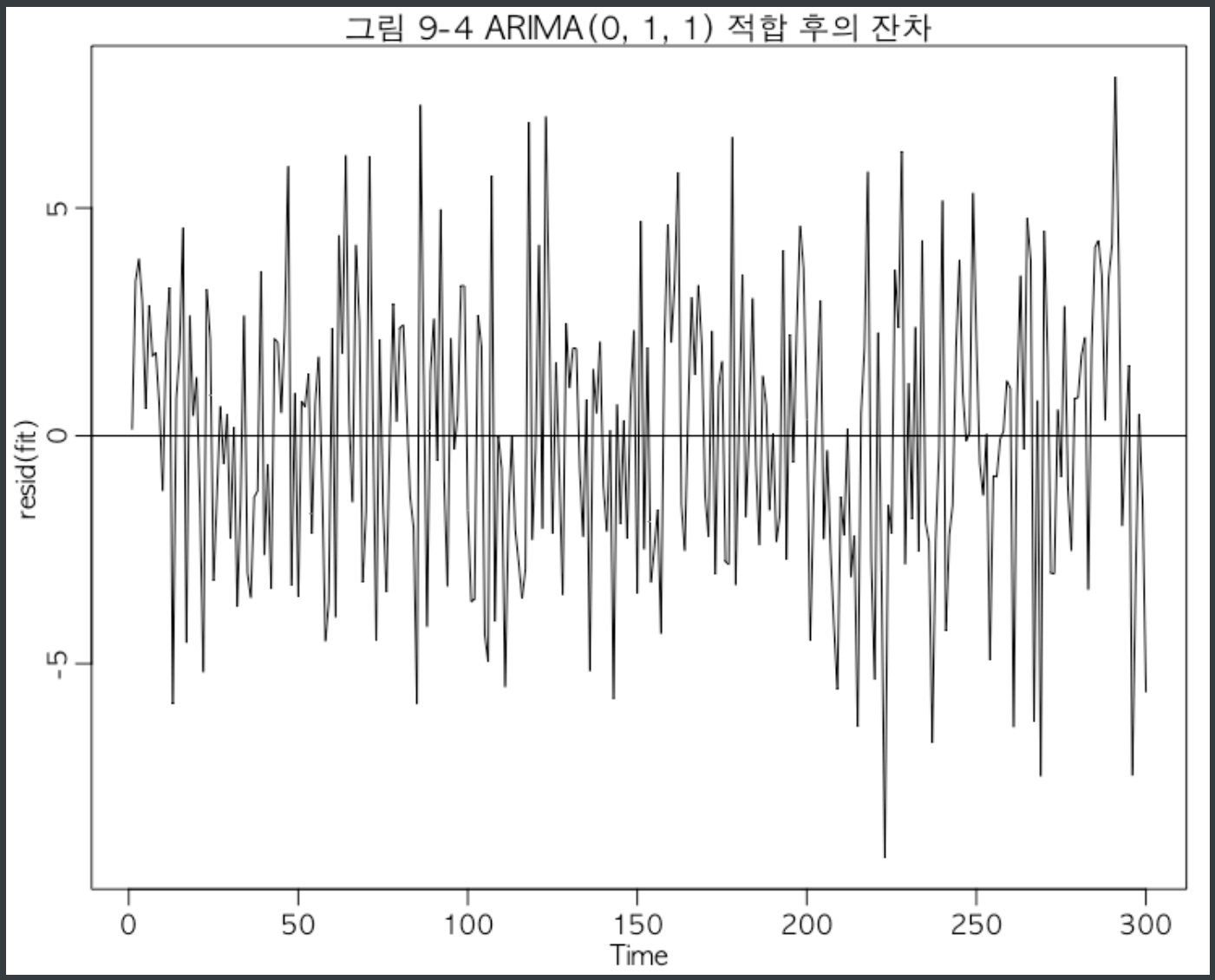


정상성에 가까워졌다.

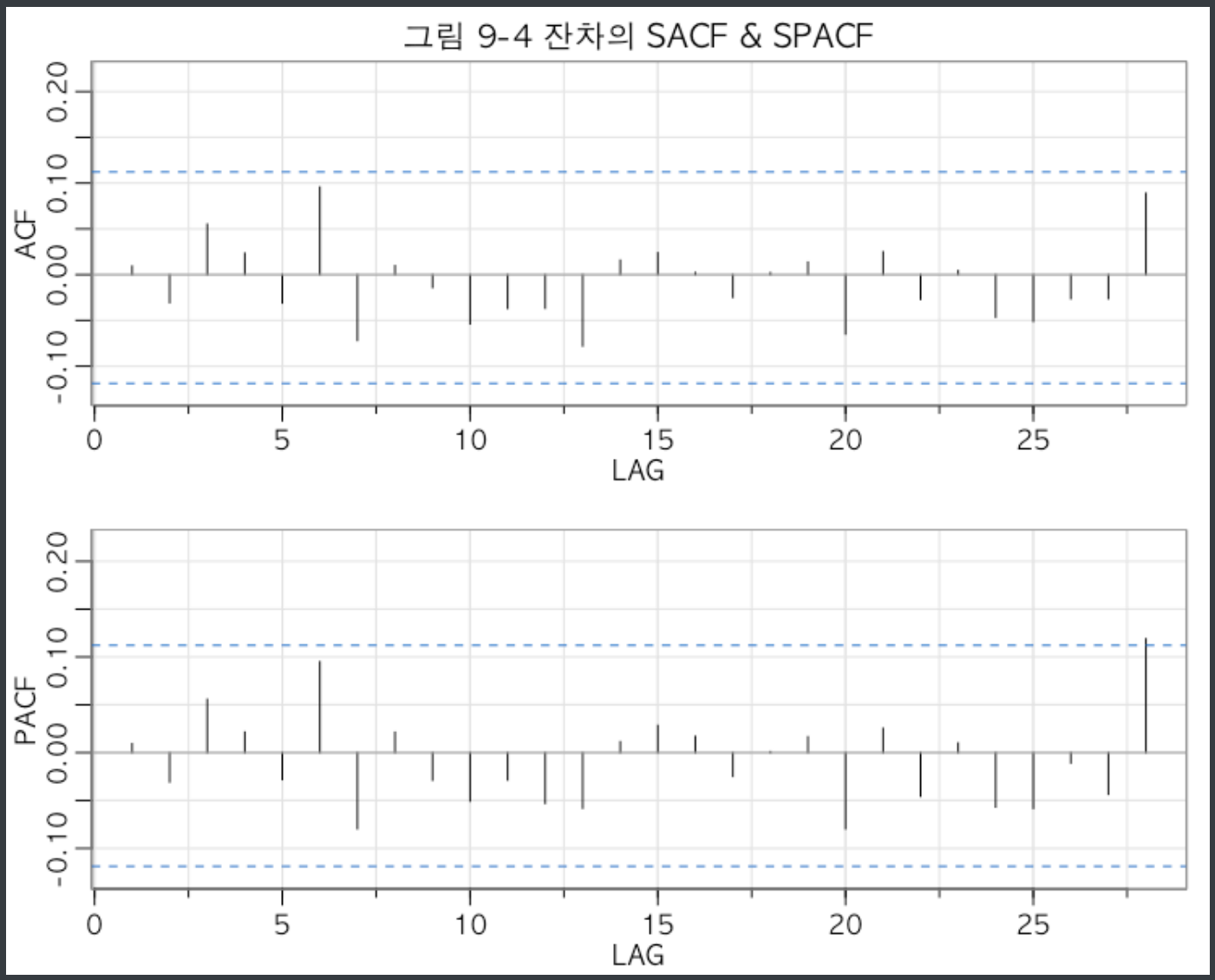


1차 차분을 진행하였으나, ACF, PACF 가 아직 임계점보다 높은 부분들이 식별된다. 추가적인 모델링이 필요하다. ACF가 1에서 절단된다는 점을 고려하여 MA(1) 을 추가로 도입한다.

Figure 9-4



ARIMA(0, 1, 1) 모형 적합 후 정상성을 보인다.



ACF, PACF 역시 안정적인 모습을 보인다.

Appendix: R code

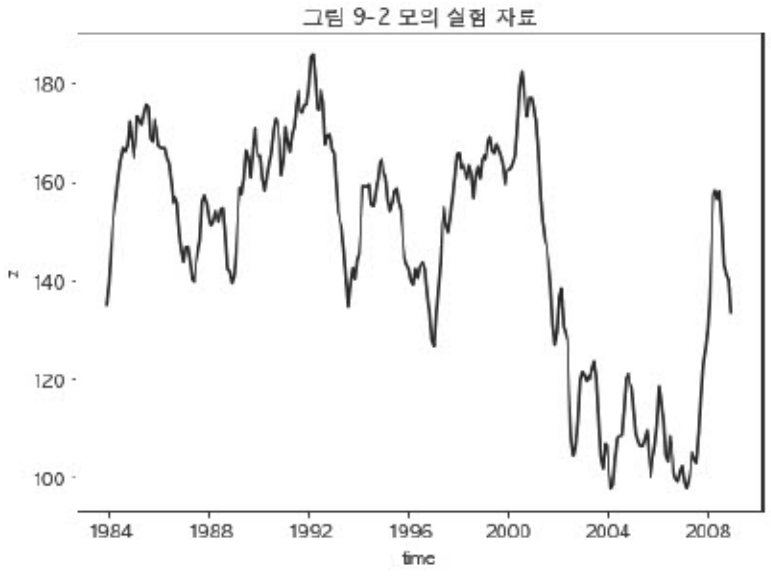
```r
rm(list=ls())

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

# 9.1
```

```r
library(astsa)
z = scan("eg8_7.txt")
par(mfrow=c(1,1))
ts.plot(z, ylab="z", main="Simulated AR(1) process")
acf2(z, max.lag=24, main="AR(1) 과정의 ACF & PACF")
sarima.for(z, 25, 1, 0, 0)

# 9.5
z = scan("eg9_5.txt")
par(mfrow=c(1,1))
ts.plot(z, ylab="z", main="그림 9-2 모의 실험 자료")
acf2(z, max.lag=24, main="그림 9-2 SACF & SPACF")

par(mfrow=c(1,1))
ts.plot(diff(z), main="그림 9-3 1차 차분된 모의 실험 자료")
abline(h=0)
acf2(diff(z), max.lag=24, main="그림 9-3 SACF & SPACF")

fit = arima(z, order=c(0, 1, 1))
par(mfrow=c(1,1))
ts.plot(resid(fit), main="그림 9-4 ARIMA(0, 1, 1) 적합 후의 잔차")
abline(h=0)

acf2(resid(fit), main="그림 9-4 잔차의 SACF & SPACF")
sarima.for(z, 25, 0, 1, 1)
```

Appendix: Pyhton code

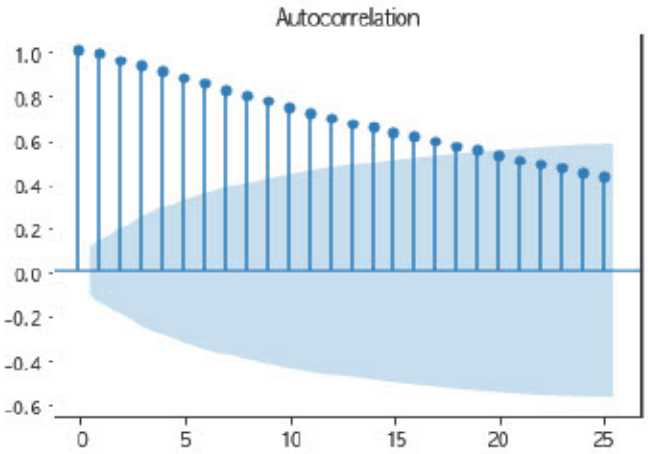
In [2]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('font', family='AppleGothic')
plt.rcParams['axes.unicode_minus'] = False

from statsmodels.tsa.arima_model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

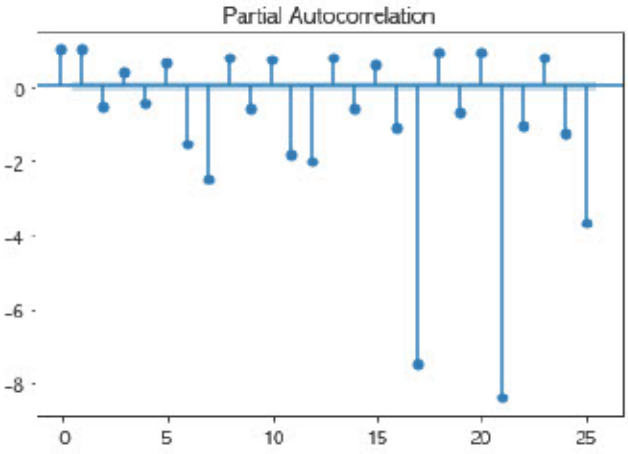
In [3]:
```
# Example 9.1
z = []

with open('../data/eg9_5.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("z")
ax.set_title("그림 9-2 모의 실험 자료")
plt.show()
```



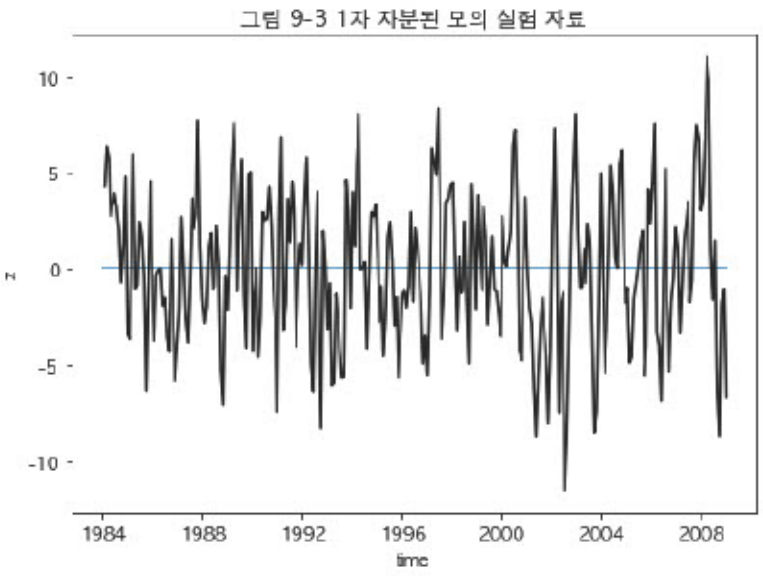
In [4]:
```
plot_acf(data)
plot_pacf(data)
plt.show()
```

```
/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/regression/l
inear_model.py:1434: RuntimeWarning: invalid value encountered in sqrt
  return rho, np.sqrt(sigmasq)
```



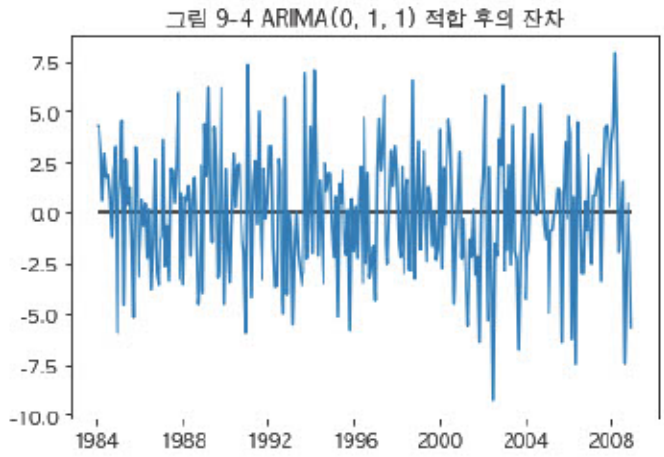


In [6]:
```
diff_data = data.diff(1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(diff_data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("z")
ax.set_title("그림 9-3 1차 차분된 모의 실험 자료")
ax.hlines(0, diff_data.index.min(), diff_data.index.max())
plt.show()
```

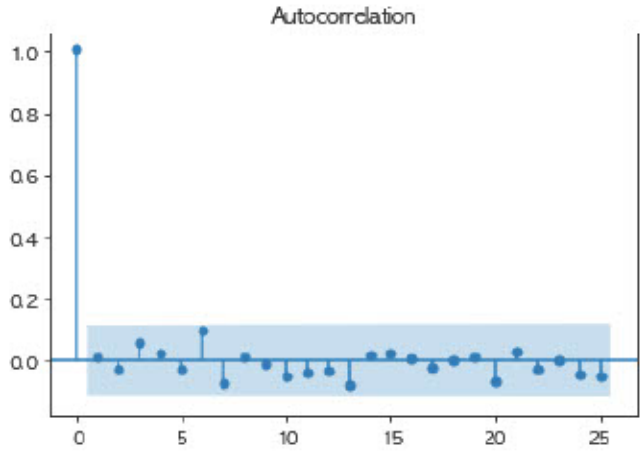


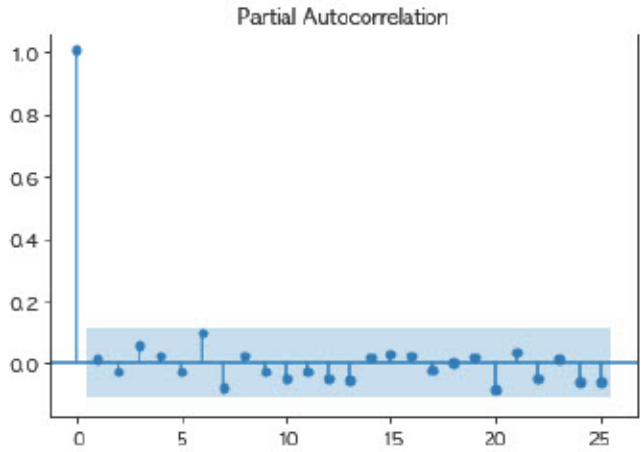
In [9]:
```
model = ARIMA(data, order=(0, 1, 1)).fit()
resid = model.resid

plt.plot(resid)
plt.title("그림 9-4 ARIMA(0, 1, 1) 적합 후의 잔차")
plt.hlines(0, resid.index.min(), resid.index.max(), color="black")
plt.show()
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           12

At X0         0 variables are exactly at the bounds

At iterate    0    f=  2.54492D+00    |proj g|=  7.74447D-04

At iterate    5    f=  2.54491D+00    |proj g|=  4.44089D-08

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2      5      7      1     0     0   4.441D-08   2.545D+00
  F =   2.5449123531058921

CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/arima_mo
del.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)

  warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
 This problem is unconstrained.
```



In [10]:
```
plot_acf(resid)
plot_pacf(resid)
plt.show()
```





In [ ]: