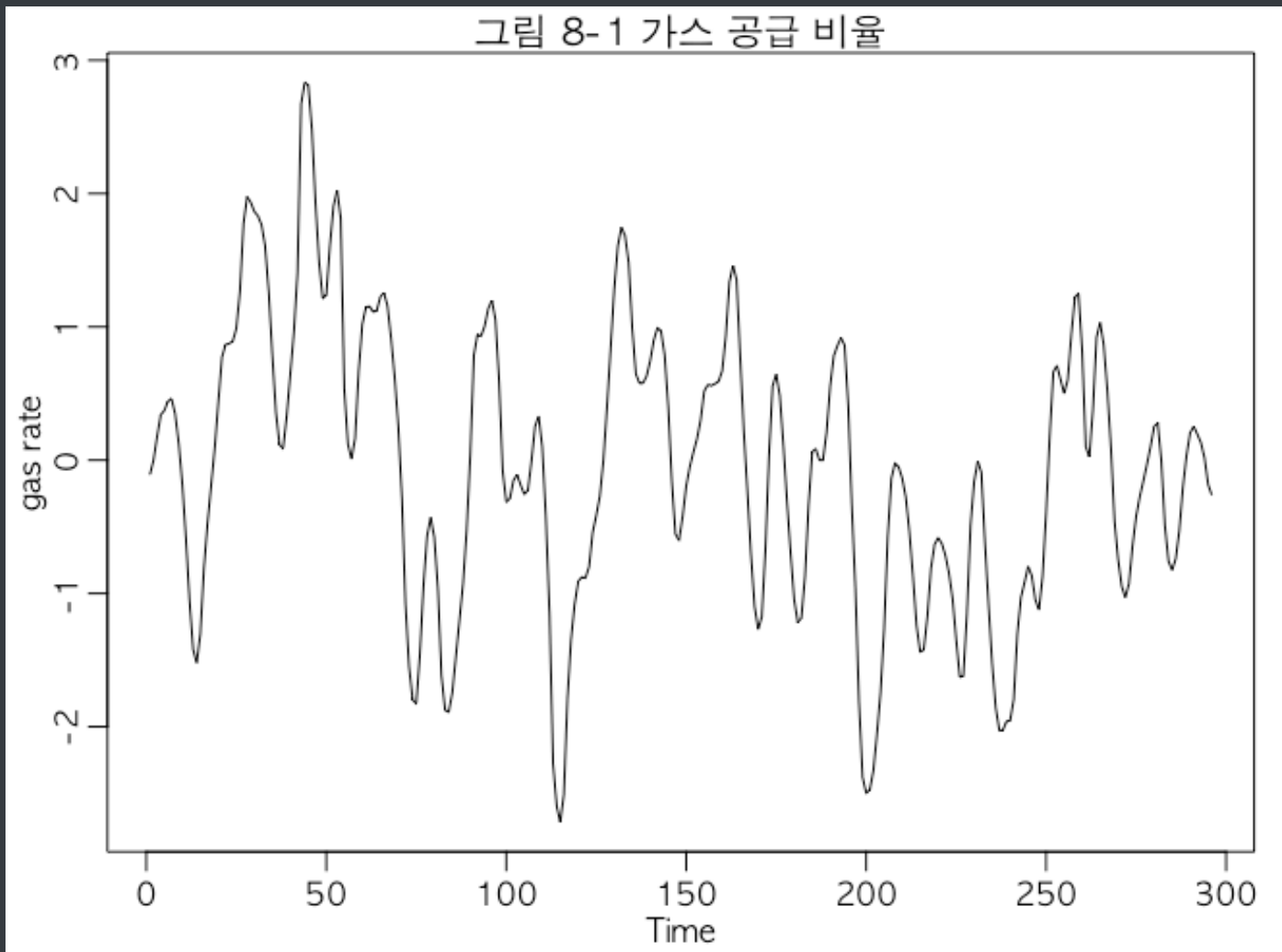


HW 08

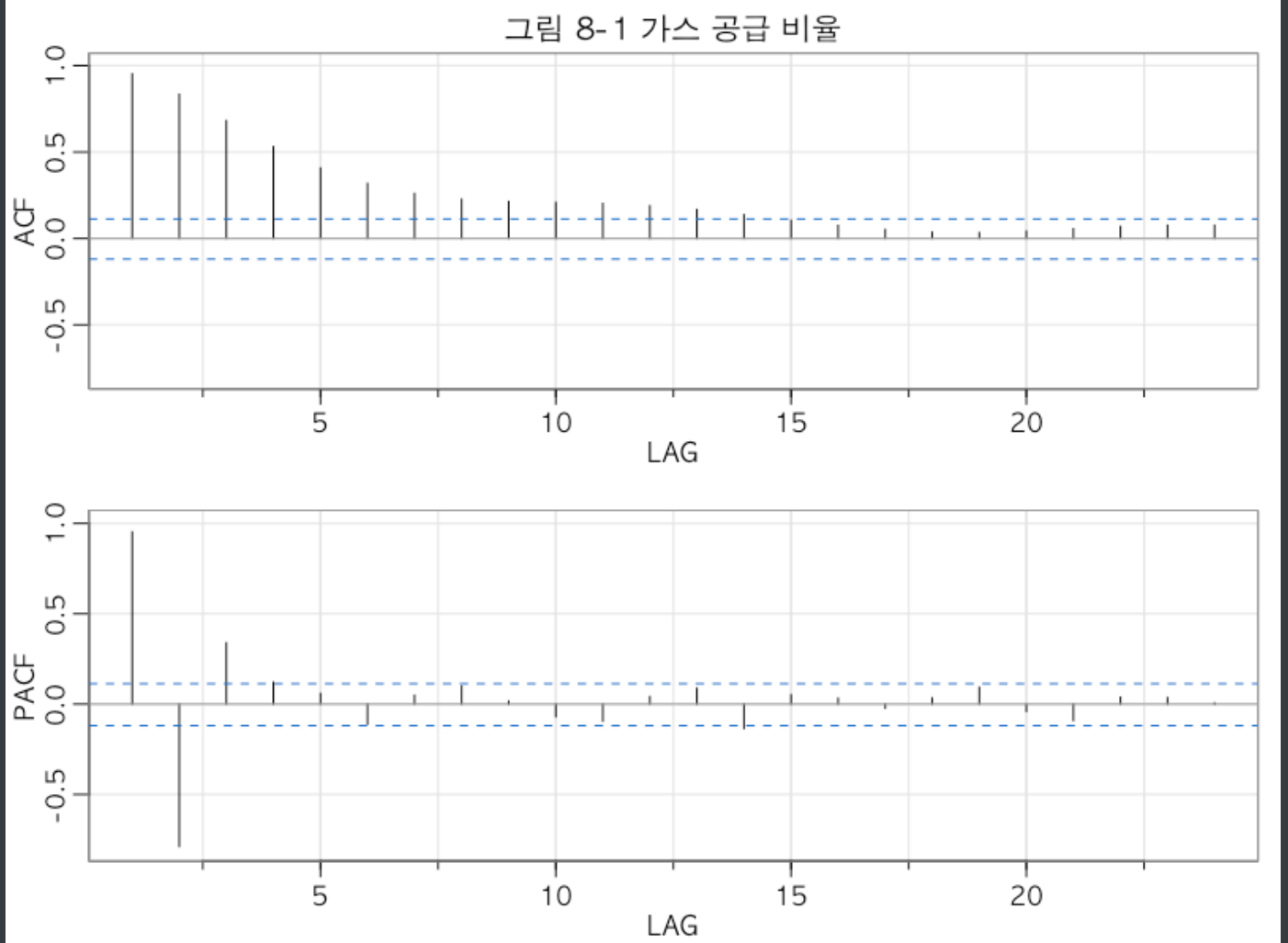
2021234640 이종현

Figure 8-1

아래 그림은 가스 공급 비율에 대한 시계열 그래프이다.

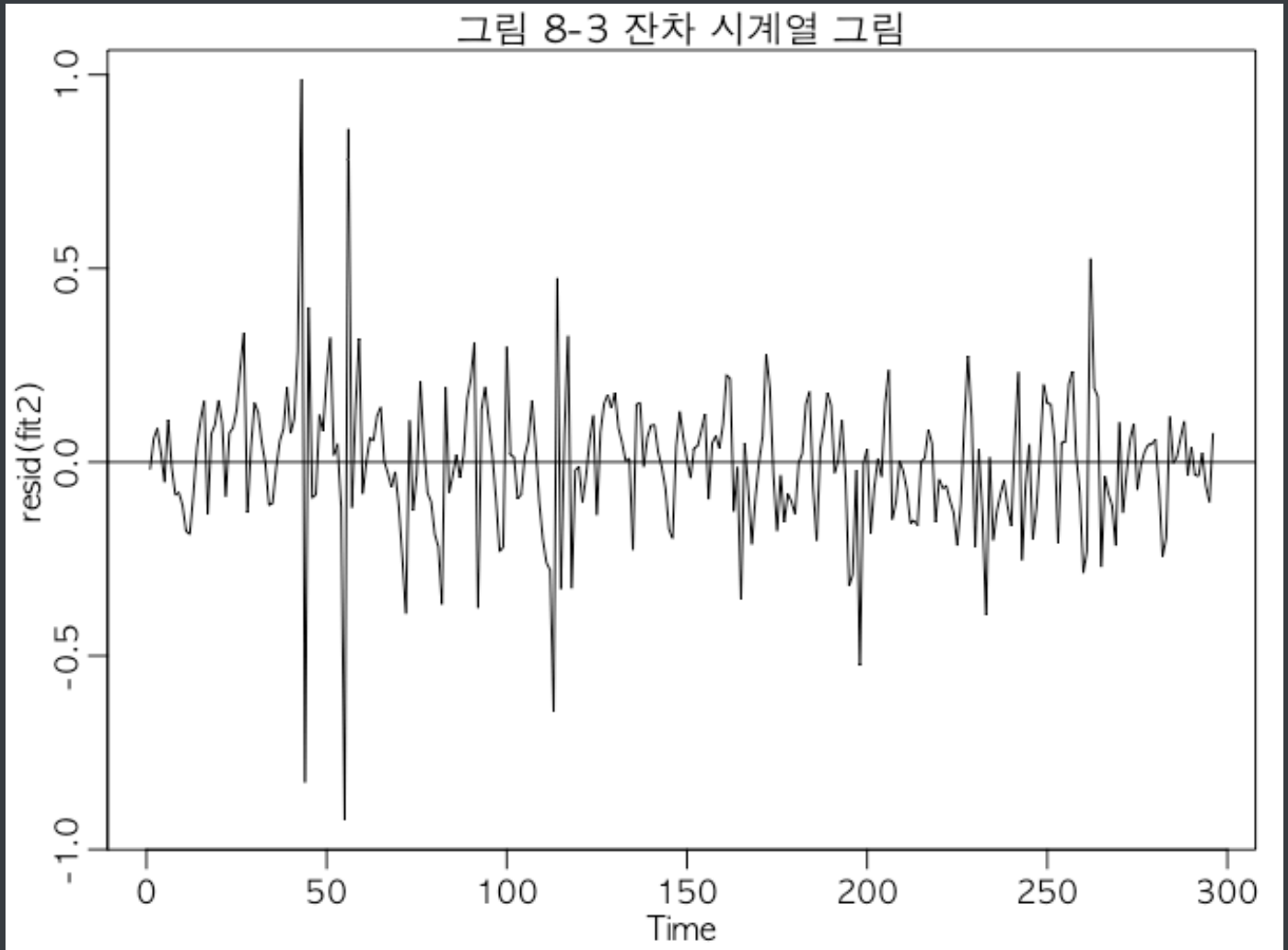


해당 시계열은 정상성을 보이지는 않는 듯 하다. ACF 와 PACF 를 확인해보자.



PACF에서 3 이상에서 절단되는 형태를 보인다. 후보 모델은 AR(3)이 적절할 것으로 보인다.

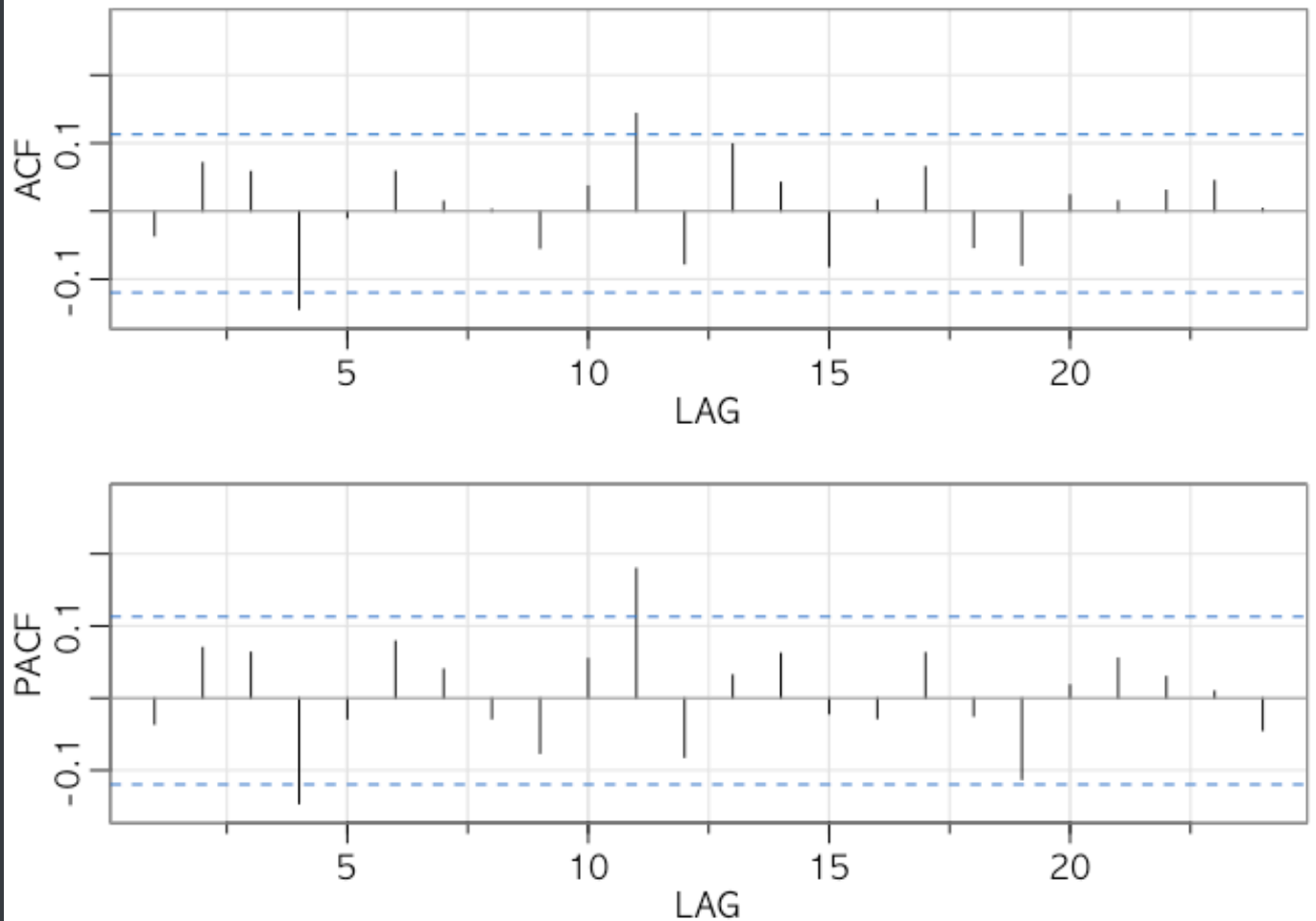
Figure 8–3



AR(3) 모델을 적합한 이후의 잔차이다. 다소 튀는 부분이 있지만 전반적인 경향성은 안정적으로 보인다.

Figure 8-4

그림 8-4 잔차 SACF와 SPACF



앞서 AR(3) 모델을 적합한 이후, 잔차에 대해서 SACF, SPACF를 확인하였다. 앞서 보였던 PACF 에서의 패턴이 사라졌음을 확인할 수 있다. 다만 12 lag 에서 PACF 가 튀는 모습이 보이긴 하지만 추후 분석에 지대한 영향을 주지는 않을 것으로 보인다.

Figure 8-5

추가로 잔차의 정규성을 QQ plot 을 이용하여 확인하였다. 극단 부분을 제외하면 정규성을 따르고 있다고 볼 수 있을 듯하다.

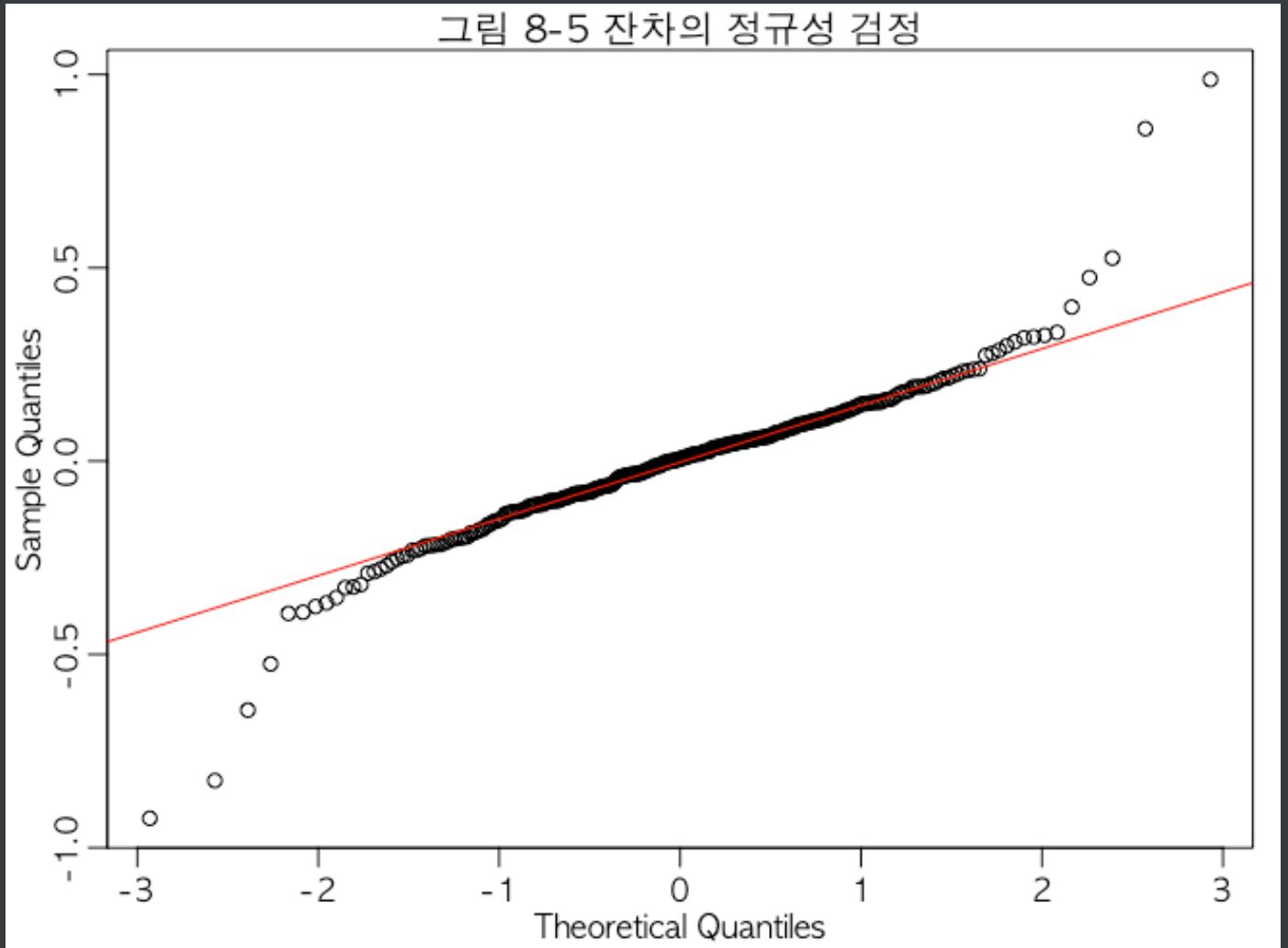
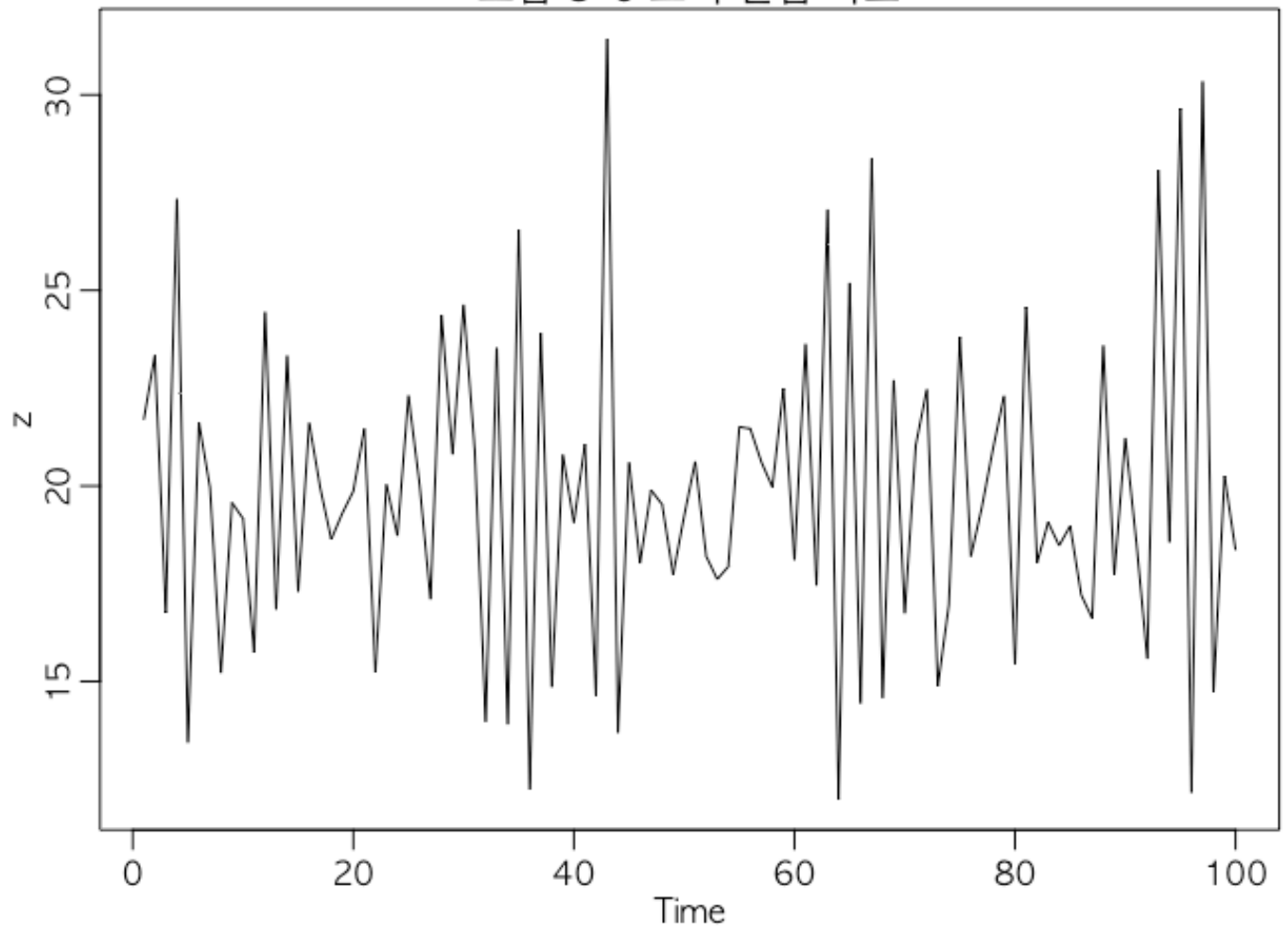


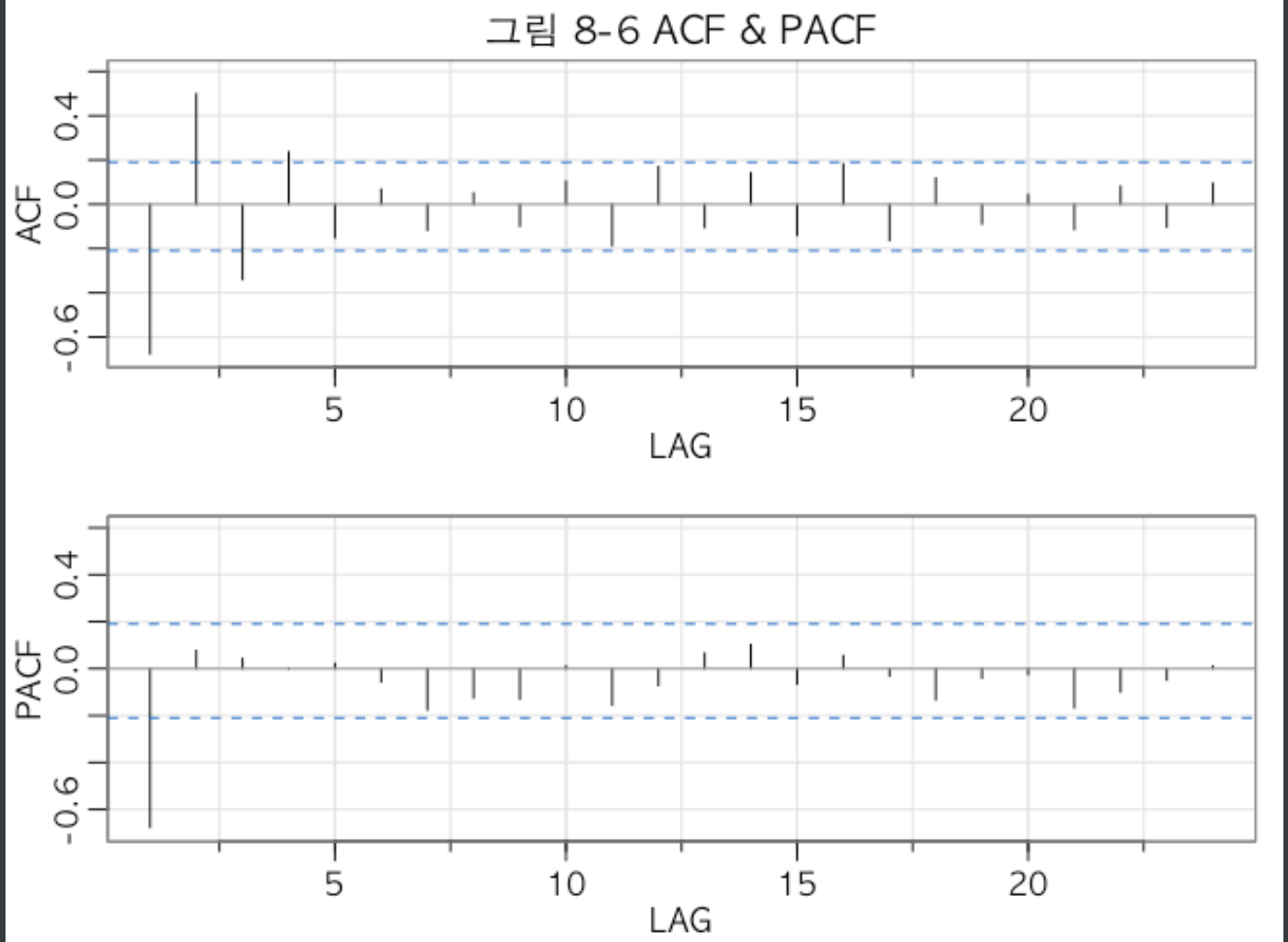
Figure 8-6

다음은 모의 실험 자료에 대한 시계열 그래프이다.

그림 8-6 모의 실험 자료

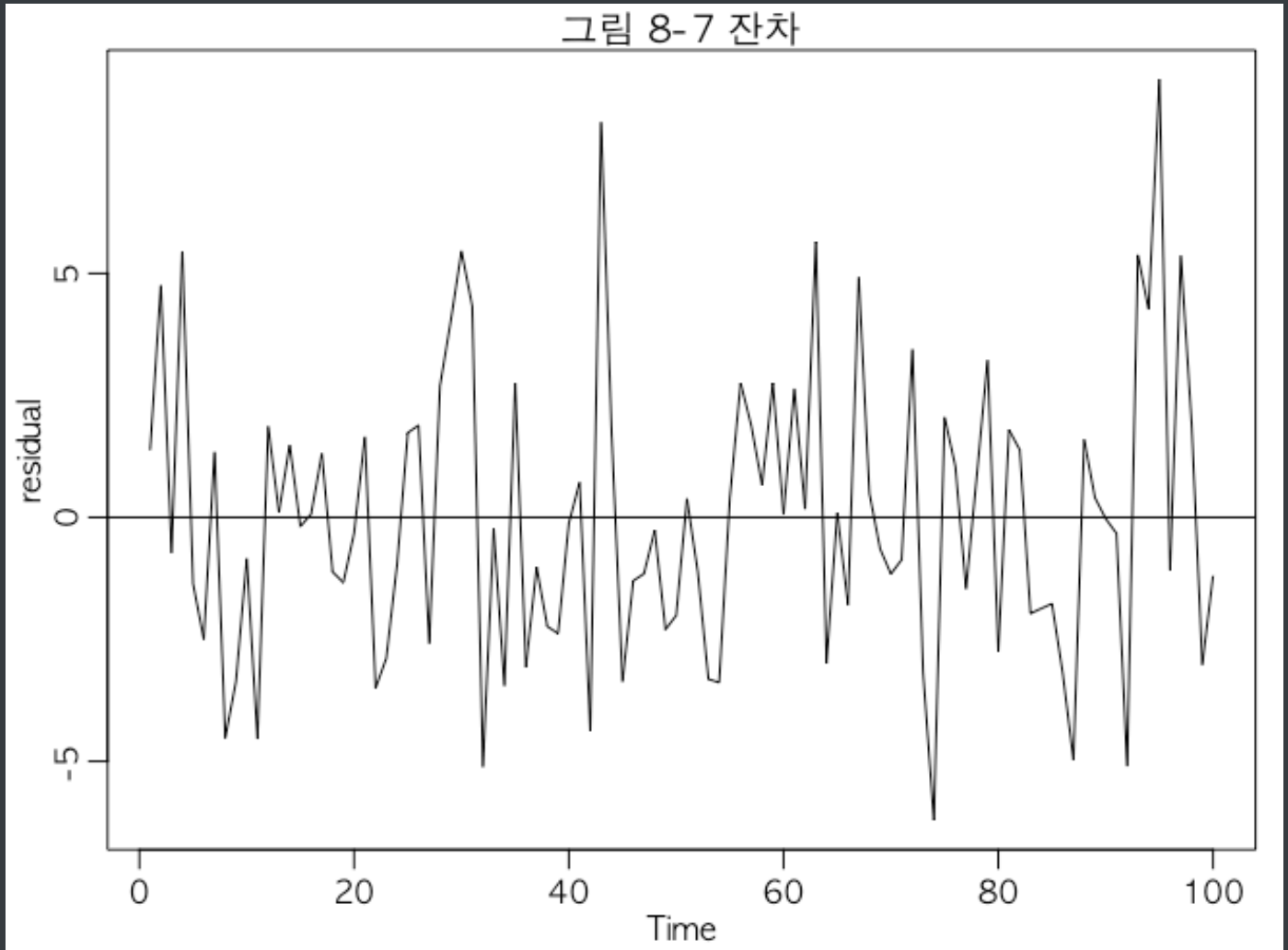


정상 시계열 여부를 확인하기 위해 ACF, PACF 를 확인해보자.



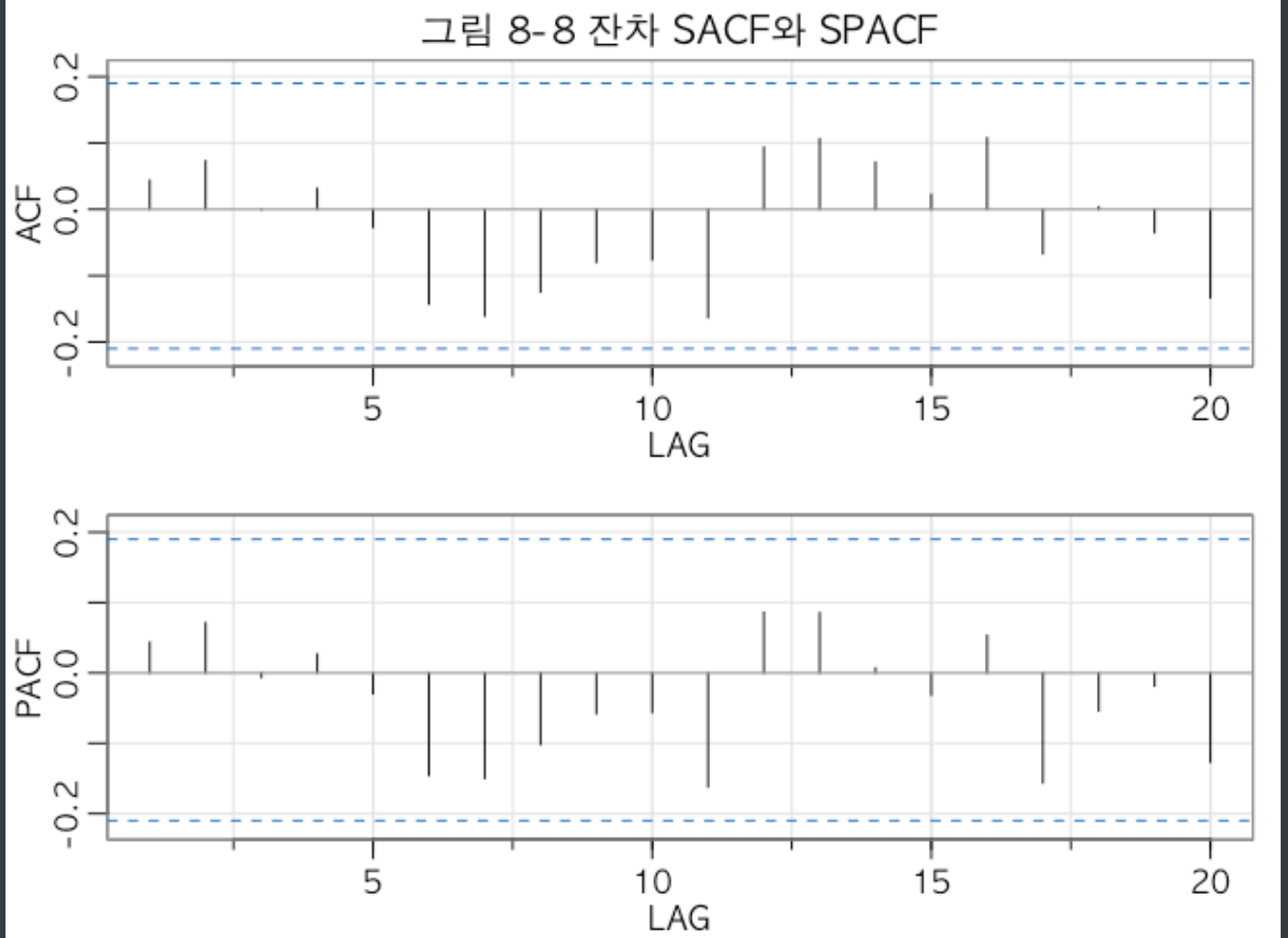
PACF 가 1에서 절단되어 AR(1) 이 적절해 보인다.

Figure 8-7



AR(1) 모델을 적합한 후의 잔차이다.

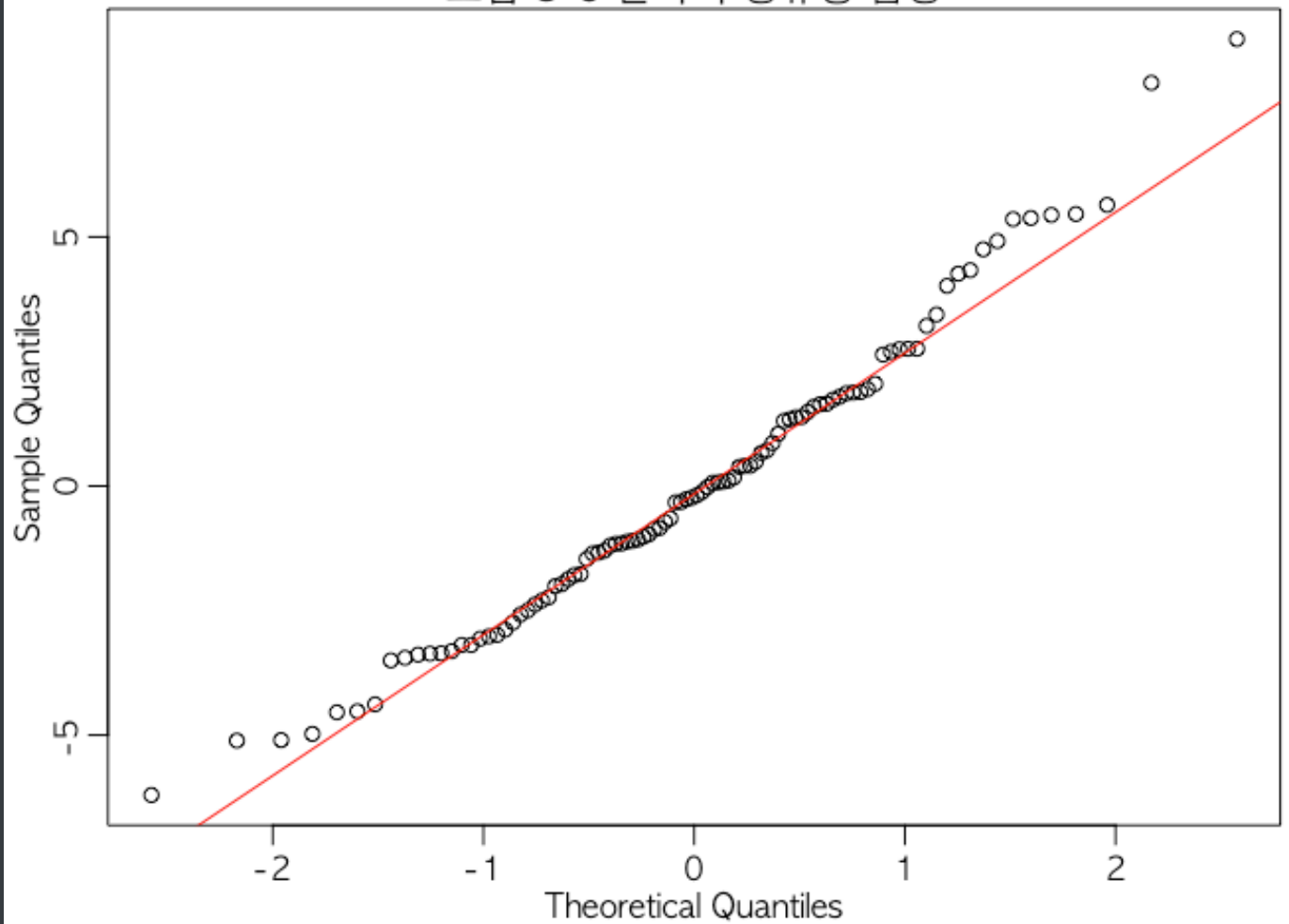
Figure 8-8



앞서 보였던 패턴이 많이 사라져 정상성을 확보했음을 확인할 수 있다.

Figure 8-9

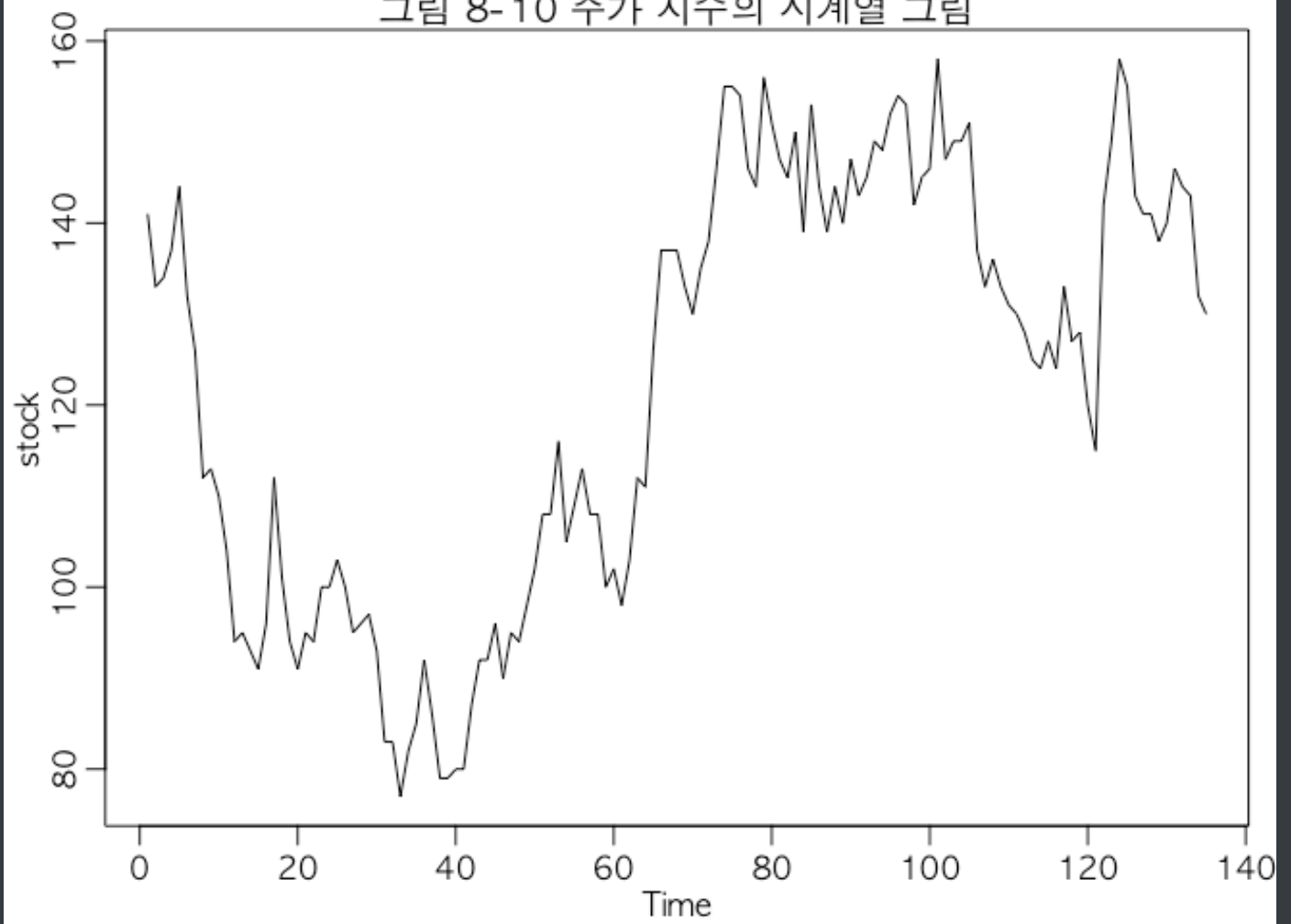
그림 8-9 잔차의 정규성 검정



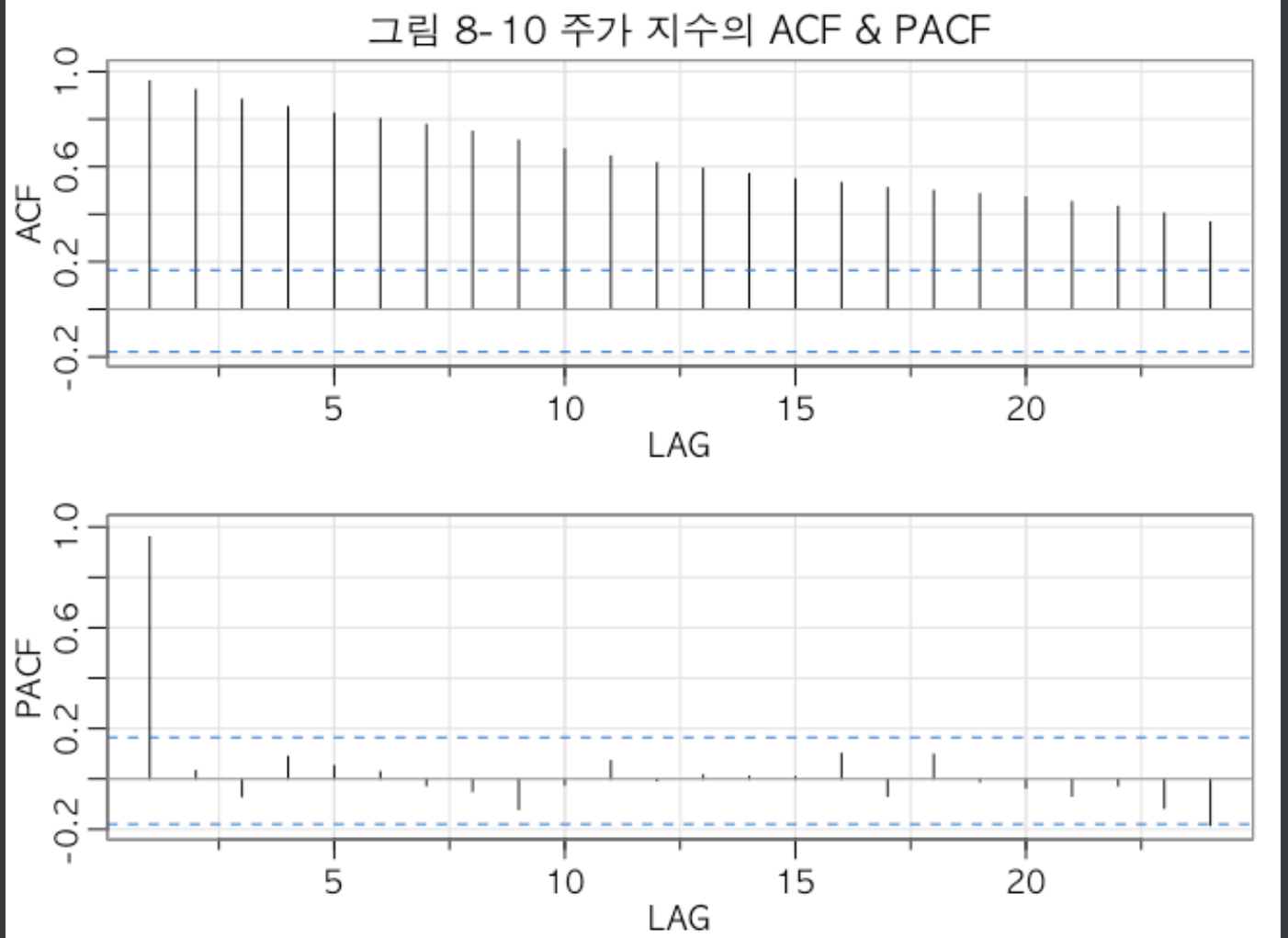
QQ plot 을 이용하여 정규성을 확인하였을 때도 안정적인 추세를 보이고 있다.

Figure 8-10

그림 8-10 주가 지수의 시계열 그림



다음은 주가 시계열 데이터이다.

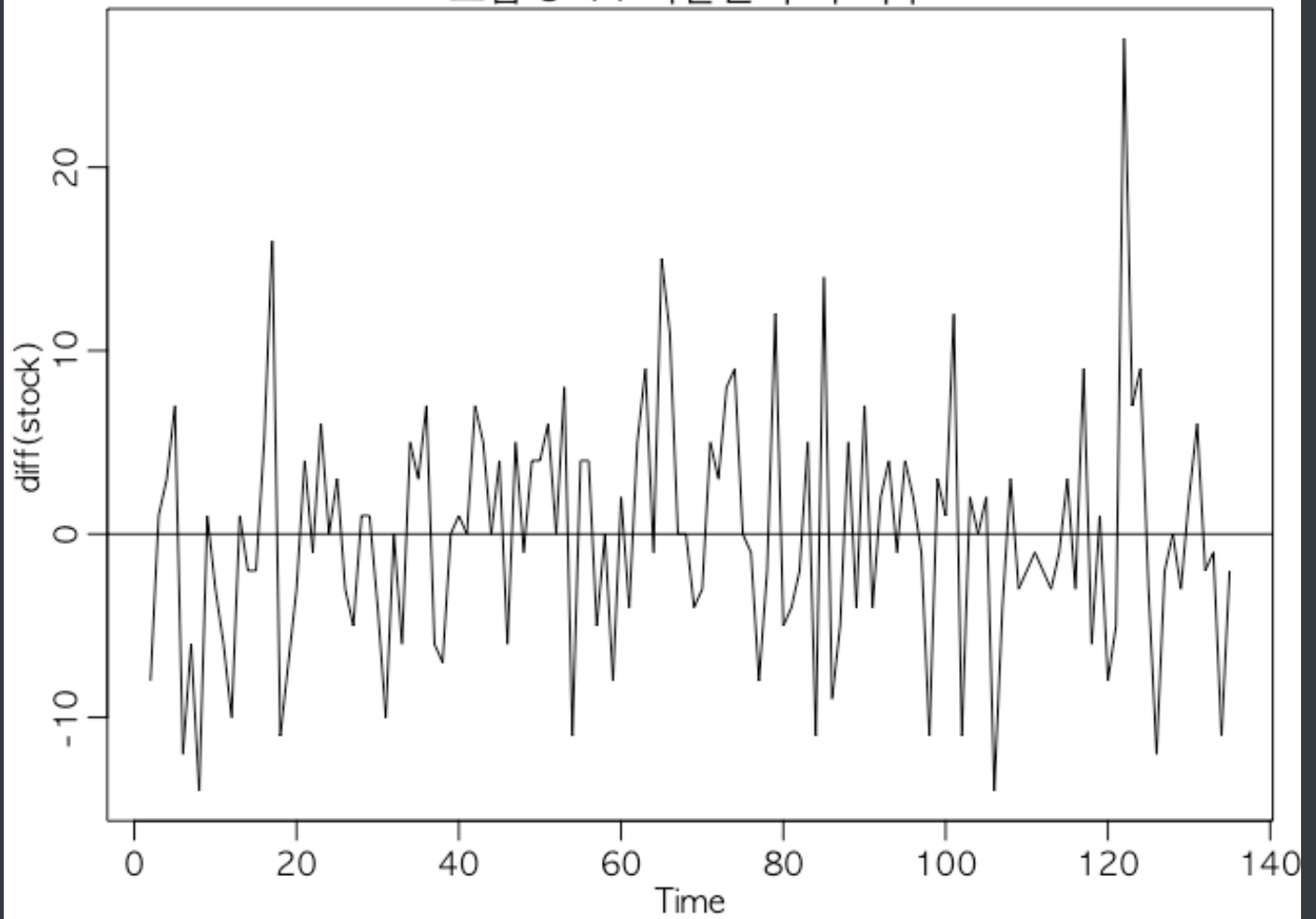


ACF 가 천천히 감소하는 모습으로 미루어 보아 강한 자기상관성이 있음을 알 수 있다. 따라서 1차 차분을 적용하는 것이 적절할 듯하다. (ARIMA(0,1,0))

Figure 8-11

아래는 1차 차분을 적용한 이후의 결과값이다.

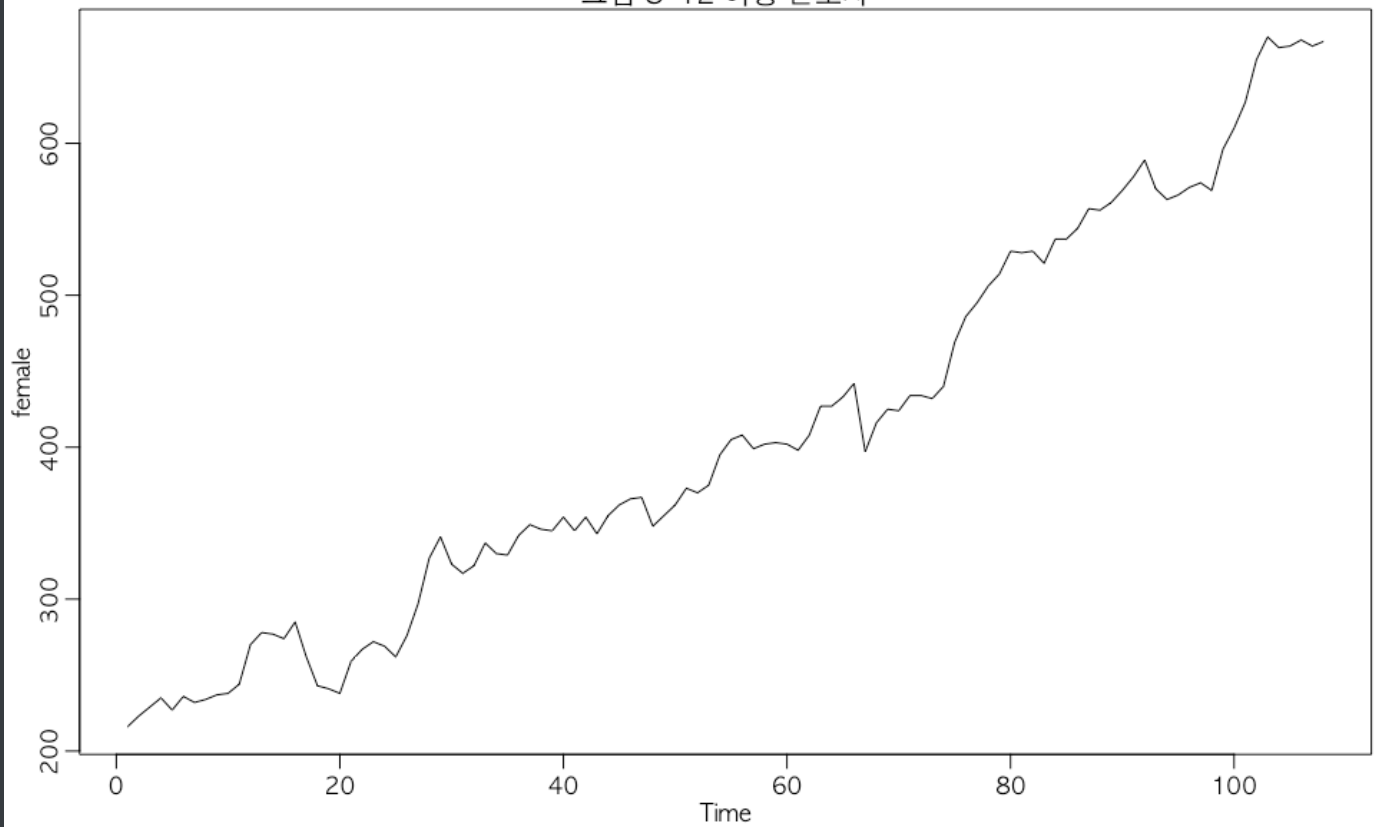
그림 8-11 차분된 주가 지수



앞서 보였던 패턴이 상당히 제거되었음을 확인할 수 있다.

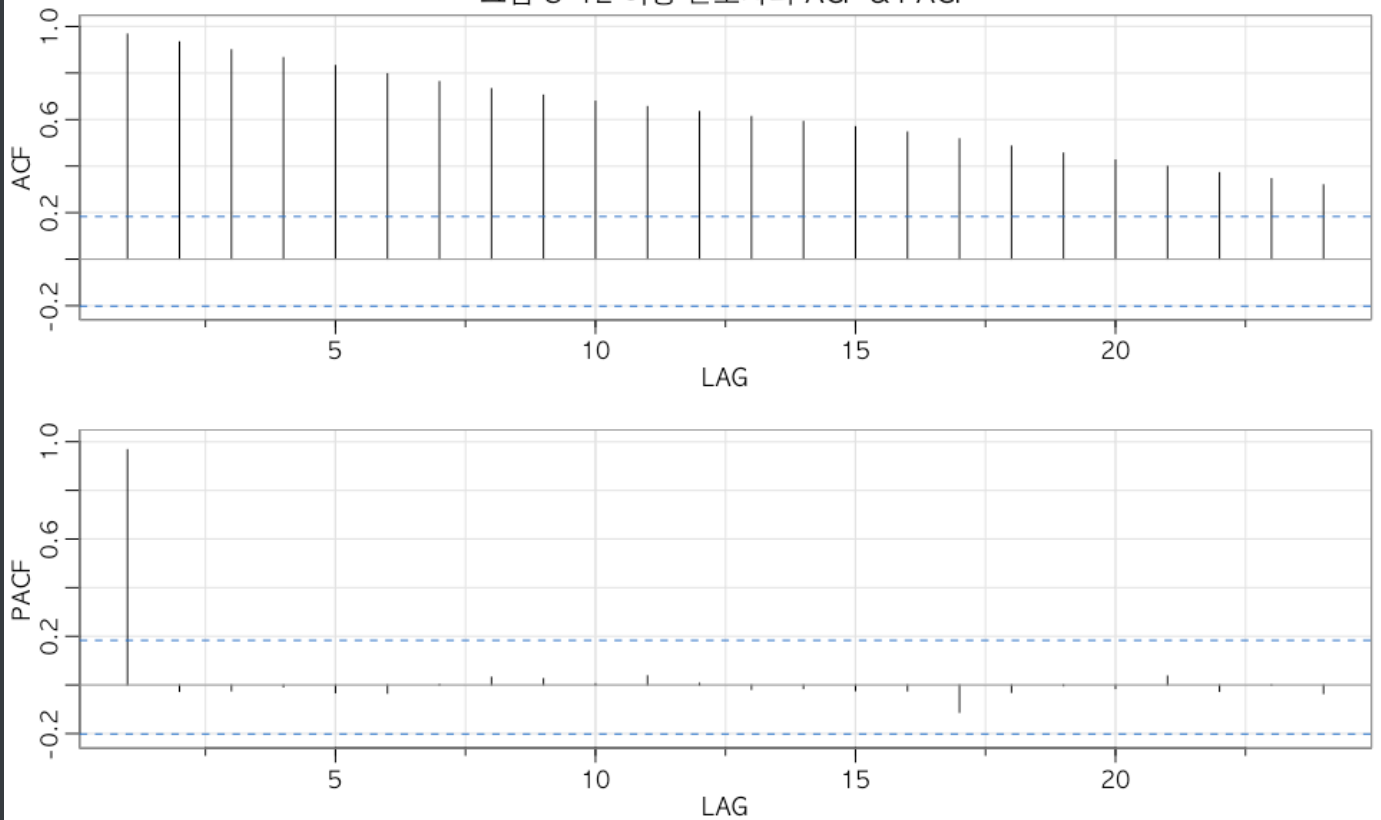
Figure 8-12

그림 8-12 여성 근로자



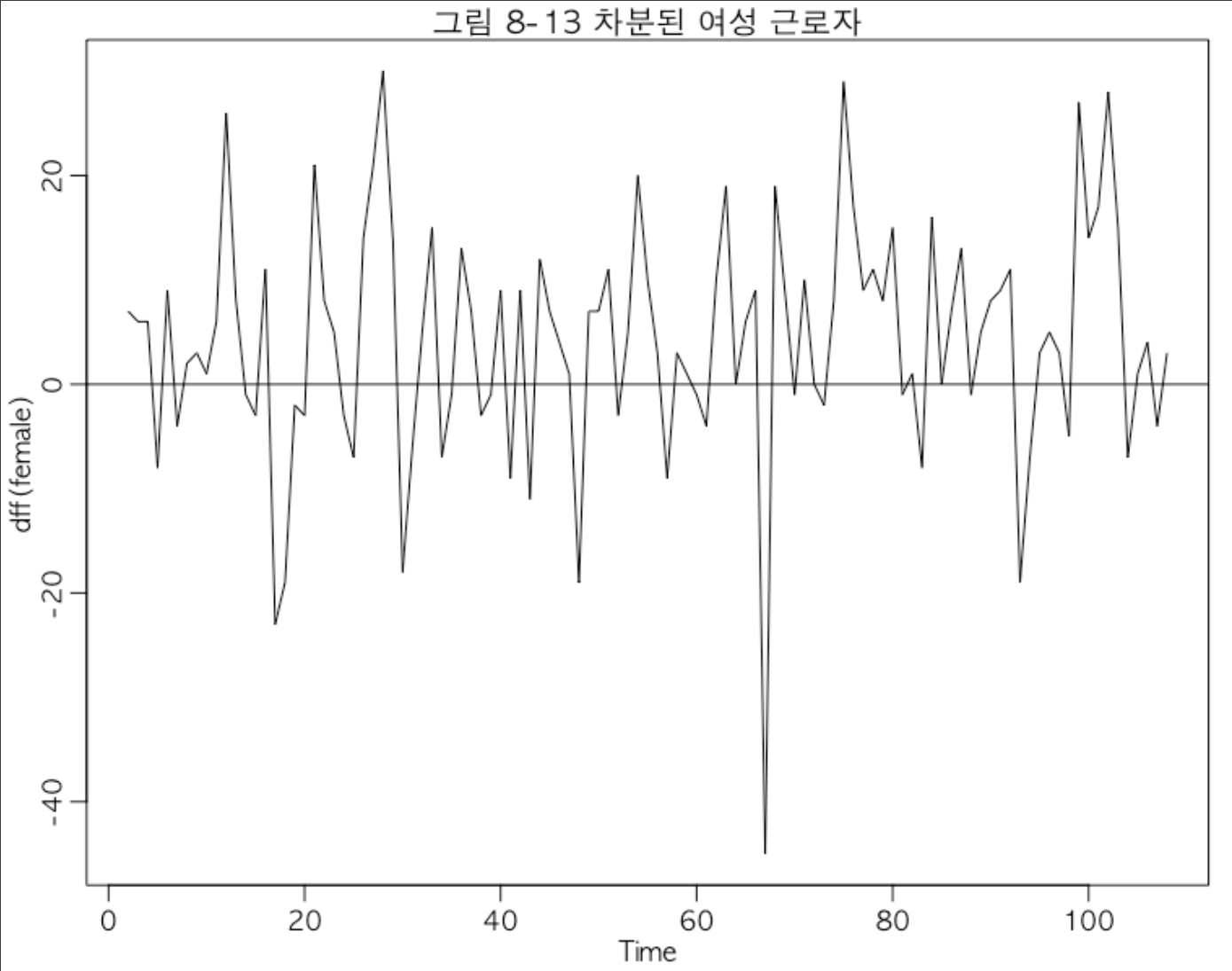
전문직종 종사자 중 여성 종사자의 수이다. 선형 트렌드가 눈에 띈다.

그림 8-12 여성 근로자의 ACF & PACF



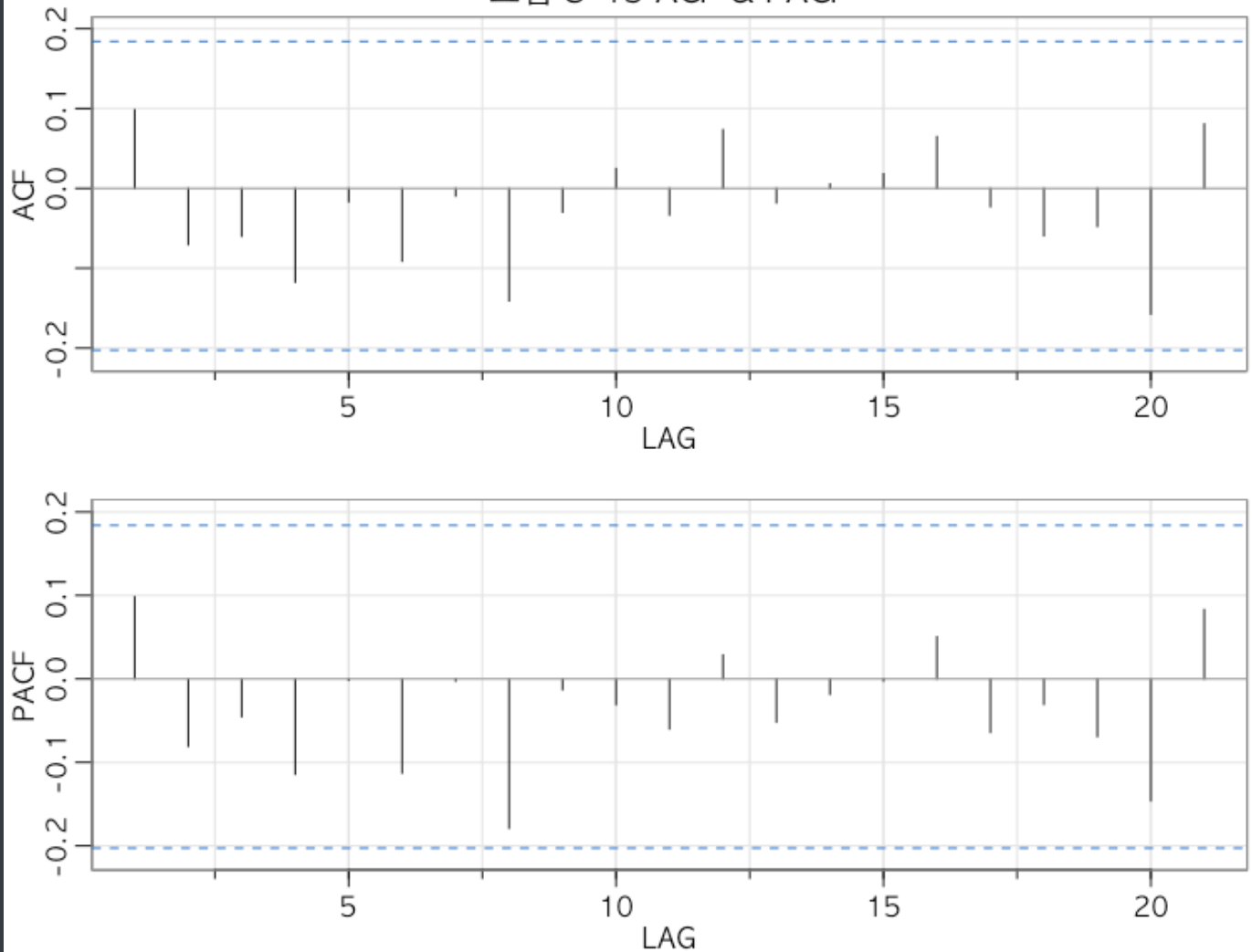
앞서 주가 그래프와 마찬가지로 자기 상관성이 높음을 볼 수 있다. 1차 차분을 한 모델이 적절할 것 같다.
(ARIMA(0,1,0))

Figure 8-13



1차 차분을 한 후, 정상성을 확보한 것으로 보인다.

그림 8-13 ACF & PACF



ACF, PACF 지표 역시 정상 시계열로 판단할 수 있다 .

Appendix: R code

```
rm(list=ls())  
setwd("~/Workspace/2022-Fall_TimeSeriesAnalysis/data")  
  
library(astsa)  
library(portes)  
library(lmtest)
```



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

```
# 8.6
```

```
gas = scan("gas.txt", what=list(0, 0))
```

```
gas = data.frame(gas)
```

```
names(gas) = c("rate", "co2")
```

```
time = 1:nrow(gas)
```

```
rate = ts(gas$rate)
```

```
co2 = ts(gas$co2)
```

```
par(mfrow=c(1,1))
```

```
ts.plot(rate, ylab="gas rate", main="그림 8-1 가스 공급 비율")
```

```
acf2(rate, max.lag=24, main="그림 8-1 가스 공급 비율")
```

```
fit1 = arima(rate, order=c(3, 0, 0))
```

```
summary(fit1)
```

```
coeftest(fit1)
```

```
fit2 = arima(rate, order=c(3, 0, 0), include.mean=FALSE)
```

```
summary(fit2)
```

```
par(mfrow=c(1,1))
```

```
ts.plot(resid(fit2), main="그림 8-3 잔차 시계열 그림")
```

```
abline(h=0)
```

```
acf2(resid(fit2), max.lag=24, main="그림 8-4 잔차 SACF와 SPACF")
```

```
par(mfrow=c(1,1))
```

```
qqnorm(resid(fit2), main="그림 8-5 잔차의 정규성 검정")
```

```
qqline(resid(fit2), col="red")
```

```
# LjungBox(fit2, lags=seq(6, 24, 6))
```

```
sarima(rate, 3, 0, 0)
```

```
sarima.for(rate, 12, 3, 0, 0)
```

```
# 8.7
```

```

z = scan("eg8_7.txt")
z.ts = ts(z)
par(mfrow=c(1,1))
ts.plot(z.ts, ylab="z", main="그림 8-6 모의 실험 자료")

acf2(z.ts, max.lag=24, main="그림 8-6 ACF & PACF")
# LjungBox(z.ts, order=c(1, 0, 0))
fit = arima(z.ts, order=c(1, 0, 0))
summary(fit)
coeftest(fit)

par(mfrow=c(1,1))
ts.plot(resid(fit), main="그림 8-7 잔차", ylab="residual")
abline(h=0)

acf2(resid(fit), maxlen=24, main="그림 8-8 잔차 SACF와 SPACF")

par(mfrow=c(1,1))
qqnorm(resid(fit), main="그림 8-9 잔차의 정규성 검정")
qqline(resid(fit), col="red")
# LjungBox(z.ts, lag=seq(6, 24, 6))
sarima.for(z.ts, 25, 1, 0, 0)
sarima(z.ts, 2, 0, 0)
sarima(z.ts, 1, 0, 1)

# 8.8
library(fUnitRoots)
z = scan("elecstock.txt")
stock = ts(z)
par(mfrow=c(1,1))
ts.plot(stock, ylab="stock", main="그림 8-10 주가 지수의 시계열 그림")

acf2(stock, max.lag=24, main="그림 8-10 주가 지수의 ACF & PACF")
# LjungBox(stock, lags=seq(6, 24, 6))
adfTest(stock, lags=0, type="c")
adfTest(stock, lags=1, type="c")

```

```

adfTest(stock, lags=2, type="c")

dstock = diff(stock, lag=1)
par(mfrow=c(1,1))
ts.plot(dstock, ylab="diff(stock)", main="그림 8-11 차분된 주가 지수")
abline(h=0)

acf2(dstock, maxlag=24)
# LjungBox(dstock, lags=seq(6, 24, 6))

fit = arima(stock, order=c(1, 0, 0), method="CSS")
fit

acf2(resid(fit))
fit1 = arima(stock, order=c(0, 1, 0))
summary(fit1)
acf2(resid(fit1))

# 8.9
library(ggplot2)
library(lubridate)

z = scan("female.txt")
female = ts(z)
par(mfrow=c(1,1))
ts.plot(female, ylab="female", main="그림 8-12 여성 근로자")

date = ymd("821201") + months(1:length(female) - 1)
female_df = data.frame(female, date)
acf2(female, max.lag=24, main="그림 8-12 여성 근로자의 ACF & PACF")

# LjungBox(female, lags=seq(6, 24, 6))
adfTest(female, lags=0, type="ct")
adfTest(female, lags=1, type="ct")
adfTest(female, lags=2, type="ct")

```

```

fit1 = lm(female ~ date, data=df)
coefficients(fit1)
residual = residuals(fit1)
resdf = data.frame(date, residual)
par(mfrow=c(1,1))
ggplot(aes(x=date, y=residual), data=resdf) + geom_line()
dfemale = diff(female, lag=1)

par(mfrow=c(1,1))
ts.plot(dfemale, ylab="dff(female)", main="그림 8-13 차분된 여성 근로자")
abline(h=0)

acf2(dfemale, maxlag=24, main="그림 8-13 ACF & PACF")
fit2 = arima(female, order=c(0, 1, 0), method="ML")
fit2
# LjungBox(fit2, lags=seq(6, 24, 6))

```

Appendix: pyhton code

```
In [7]: 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
```

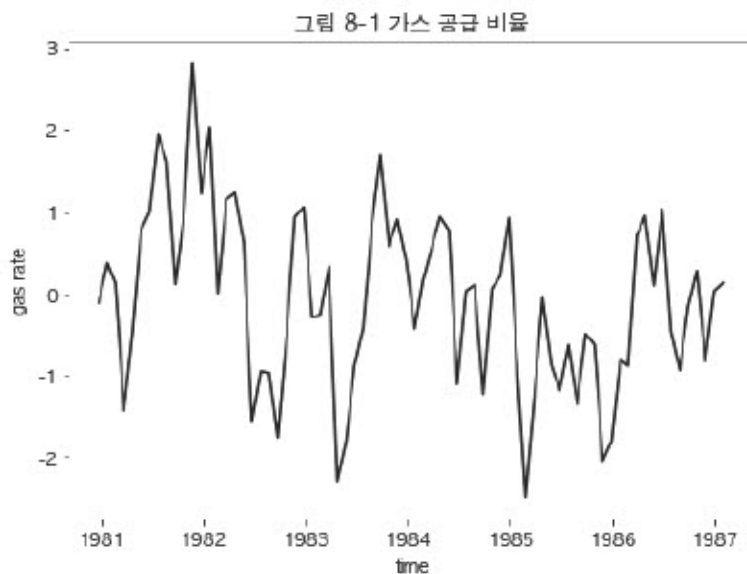
```
In [6]: # Example 4.1
z = []

temp = []
with open('../data/gas.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem) and len(temp) < 2:
                temp.append(float(elem))

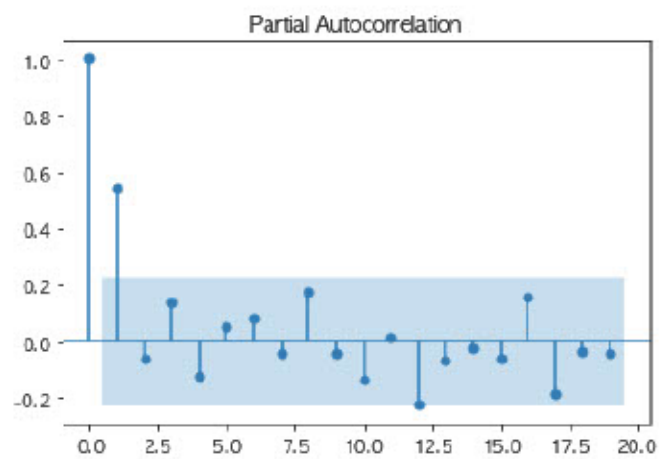
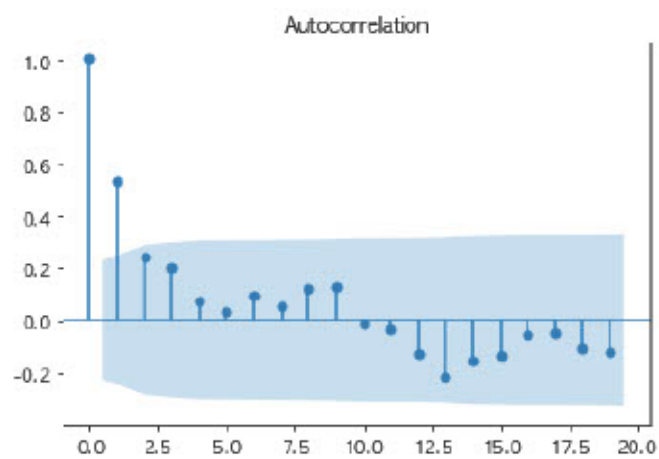
        if len(temp) >= 2:
            z.append(temp)
            temp = []

index = pd.date_range(start="1981", periods=len(z), freq="MS")
data = pd.DataFrame(z, index=index, columns=["rate", "co2"])

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data['rate'], 'black')
ax.set_xlabel("time")
ax.set_ylabel("gas rate")
ax.set_title("그림 8-1 가스 공급 비율")
plt.show()
```



```
In [10]: plot_acf(data['rate'])  
plot_pacf(data['rate'])  
plt.show()
```



```
In [24]: model = ARIMA(data['rate'], order=(3, 0, 0)).fit()
resid = model.resid

plt.plot(resid)
plt.title("그림 8-3 잔차 시계열 그림")
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 = 4 M = 12

At X0 0 variables are exactly at the bounds

At iterate 0 f= 1.30721D+00 |proj g|= 3.85114D-03

At iterate 5 f= 1.30715D+00 |proj g|= 1.77636D-07

* * *

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

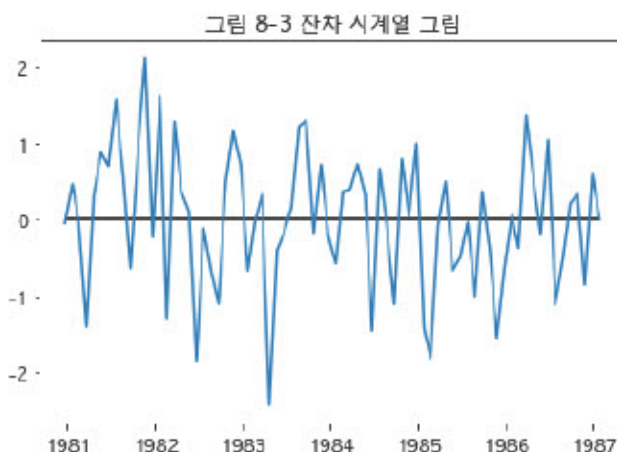
* * *

N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
4	7	9	1	0	0	2.220D-08	1.307D+00

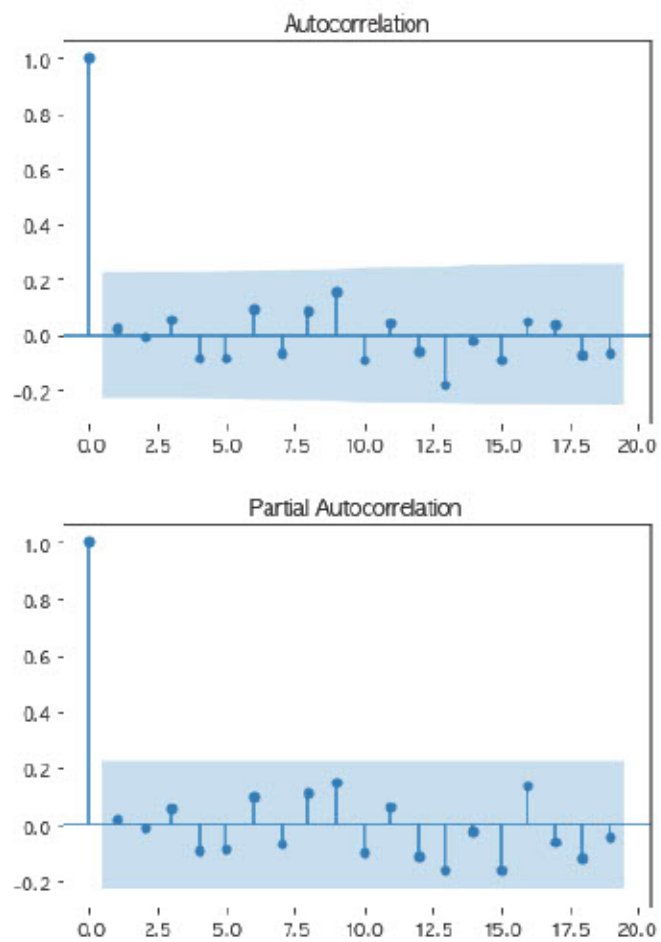
F = 1.3071460988809354

CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

This problem is unconstrained.

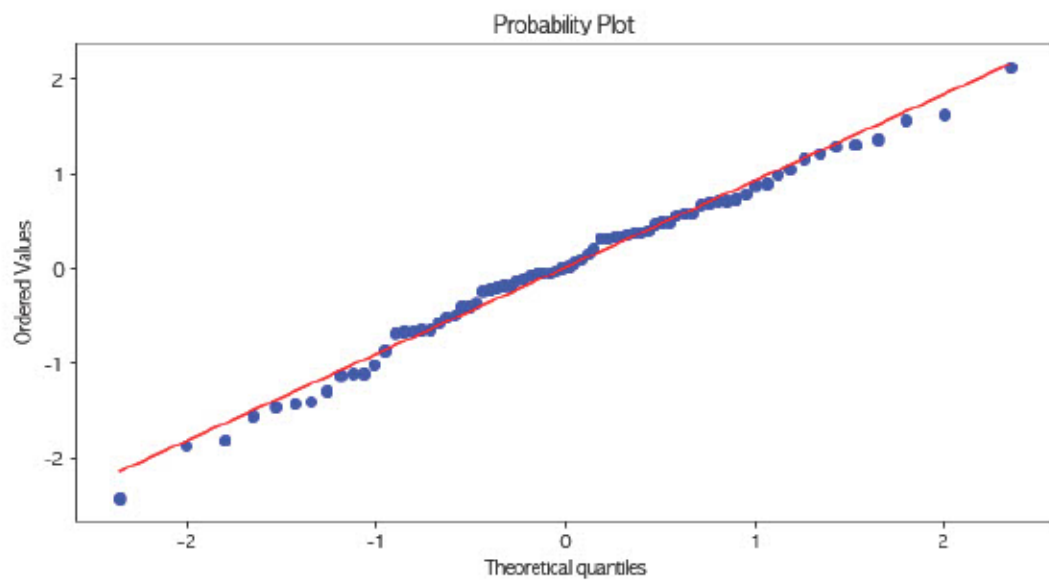


```
In [25]: plot_acf(resid)
plot_pacf(resid)
plt.show()
```




```
In [27]: import scipy.stats as stats

plt.figure(figsize=(10, 5))
plt.title("그림 8-5 잔차의 정규성 검정")
stats.probplot(resid, dist=stats.norm, plot=plt)
plt.show()
```



```

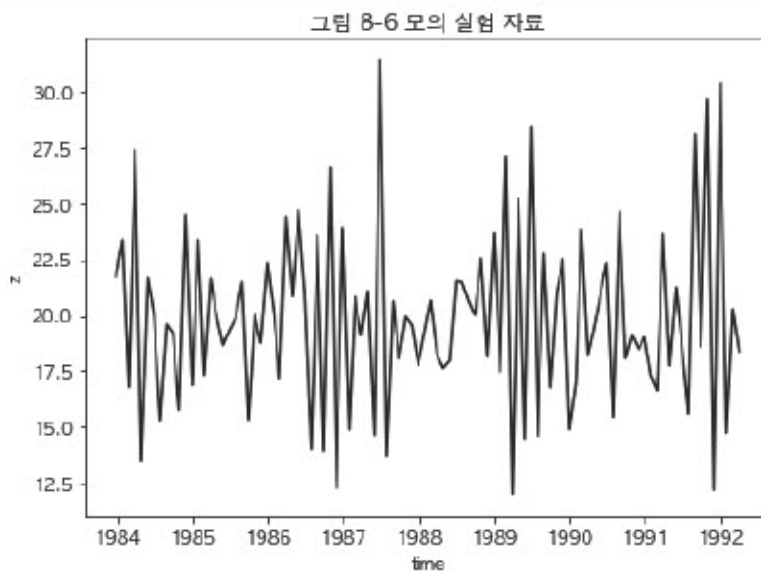
In [28]: # Example 8-7
z = []

with open('../data/eg8_7.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

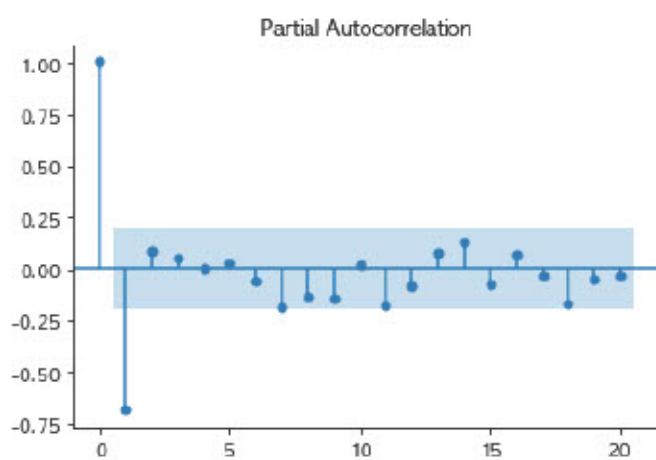
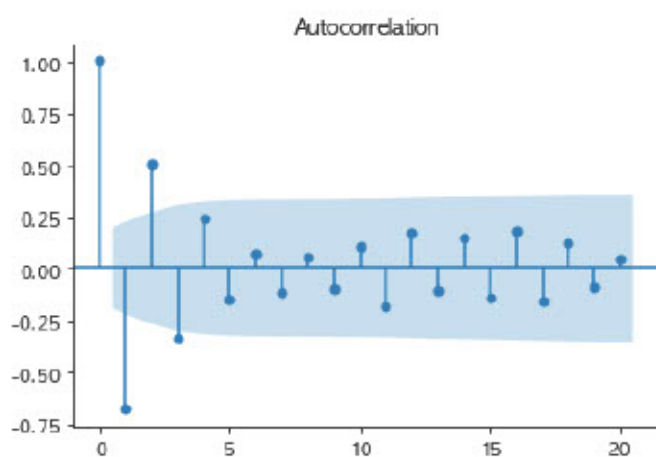
index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("z")
ax.set_title("그림 8-6 모의 실험 자료")
plt.show()

```



```
In [30]: plot_acf(data)
plot_pacf(data)
plt.show()
```



```
In [31]: model = ARIMA(data, order=(1, 0, 0)).fit()
        resid = model.resid

        plt.plot(resid)
        plt.title("그림 8-7 잔차 시계열 그림")
        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= 2.50614D+00 |proj g|= 2.65561D-03

At iterate 5 f= 2.50611D+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	2.506D+00

F = 2.5061142051458765

CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/arma_model.py:472: FutureWarning:

statsmodels.tsa.arma_model.ARMA and statsmodels.tsa.arma_model.ARIMA have been deprecated in favor of statsmodels.tsa.arma.model.ARIMA (note the . between arma and model) and statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

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

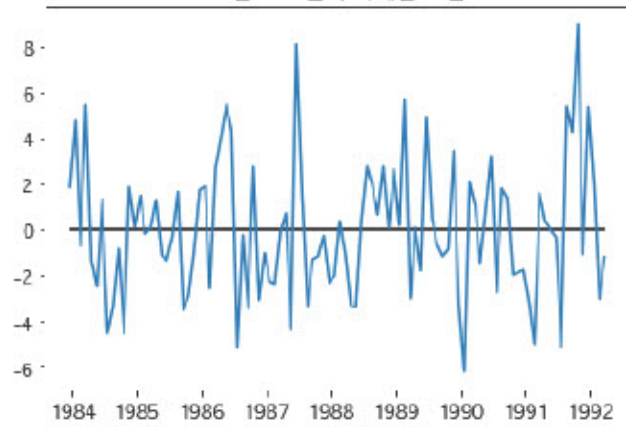
To silence this warning and continue using ARMA and ARIMA until they are removed, use:

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

```
warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
```

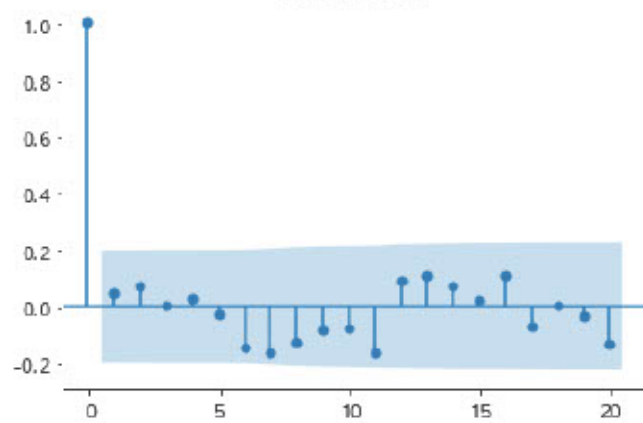
This problem is unconstrained.

그림 8-7 잔차 시계열 그림

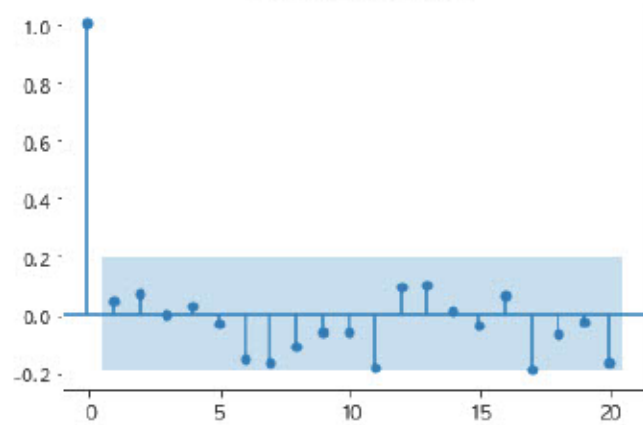


```
In [32]: plot_acf(resid)
plot_pacf(resid)
plt.show()
```

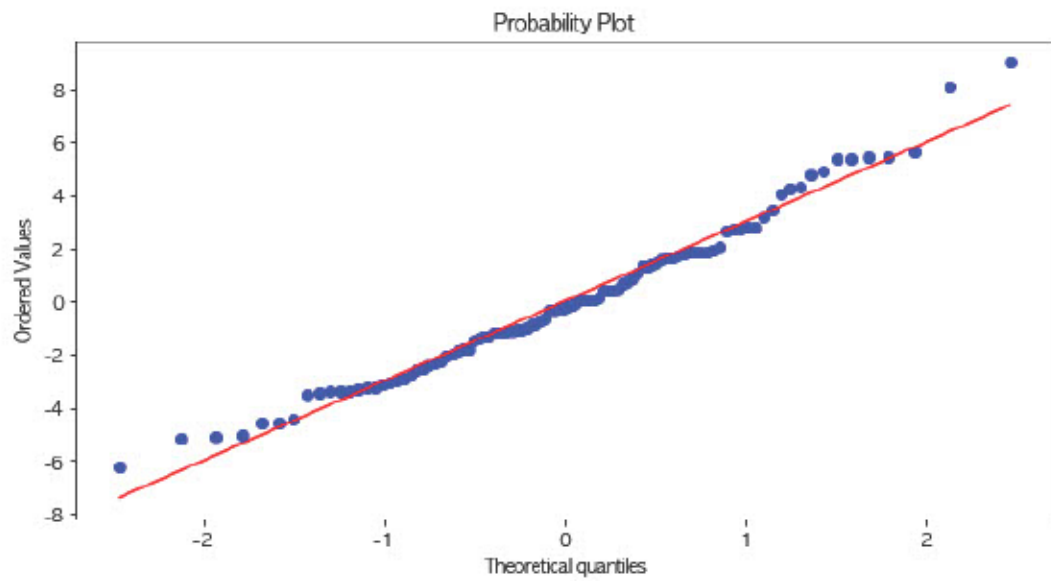
Autocorrelation



Partial Autocorrelation



```
In [34]: plt.figure(figsize=(10, 5))
plt.title("그림 8-9 잔차의 정규성 검정")
stats.probplot(resid, dist=stats.norm, plot=plt)
plt.show()
```



```

In [36]: # Example 8-8
z = []

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

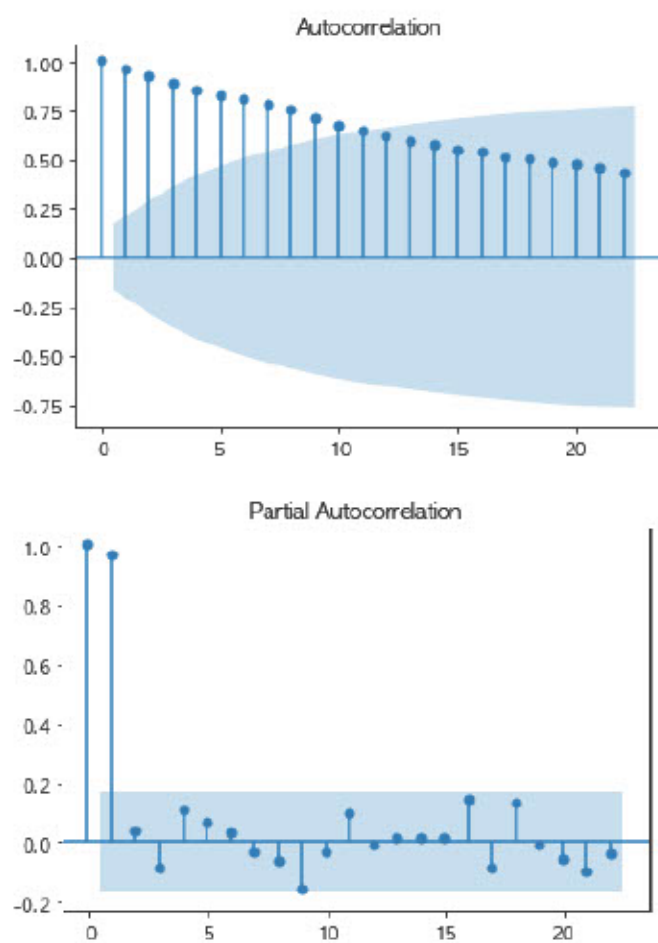
index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.set_title("그림 8-10 주가 지수의 시계열 그림")
plt.show()

```



```
In [37]: plot_acf(data)
plot_pacf(data)
plt.show()
```

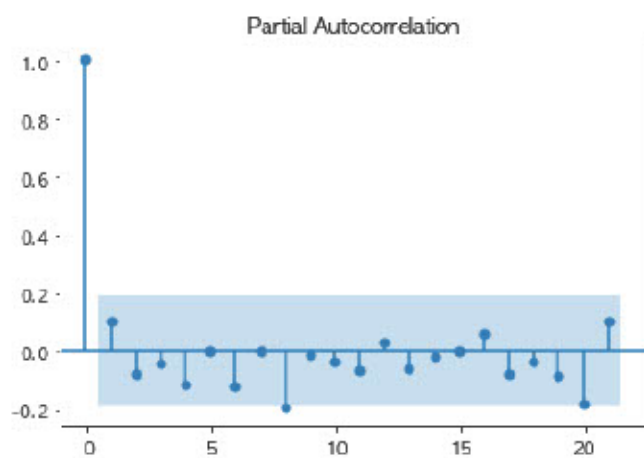
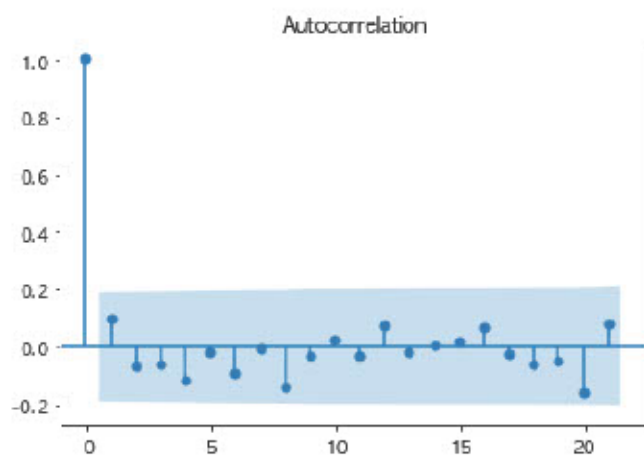



```
In [46]: diff_data = data.diff(1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(diff_data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.hlines(0, diff_data.index.min(), diff_data.index.max())
ax.set_title("그림 8-11 차분된 주가 지수")
plt.show()
```



```
In [47]: plot_acf(diff_data[1:])  
plot_pacf(diff_data[1:])  
plt.show()
```



```

In [49]: # Example 8-9
z = []

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

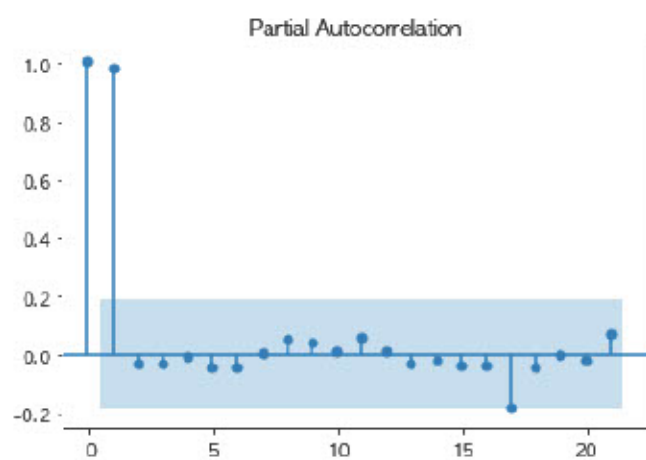
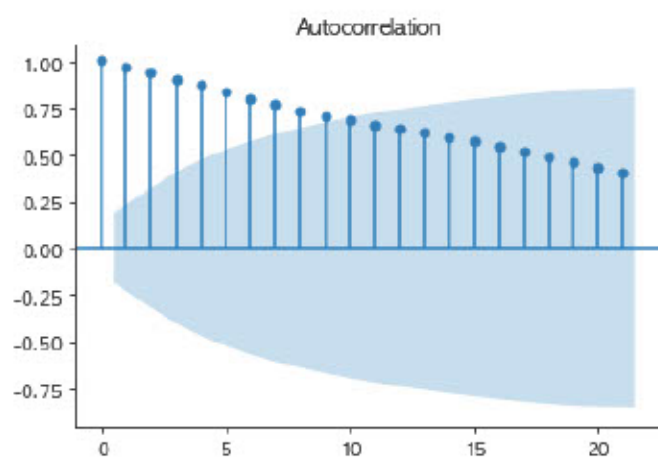
index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.set_title("그림 8-12 여성 근로자")
plt.show()

```

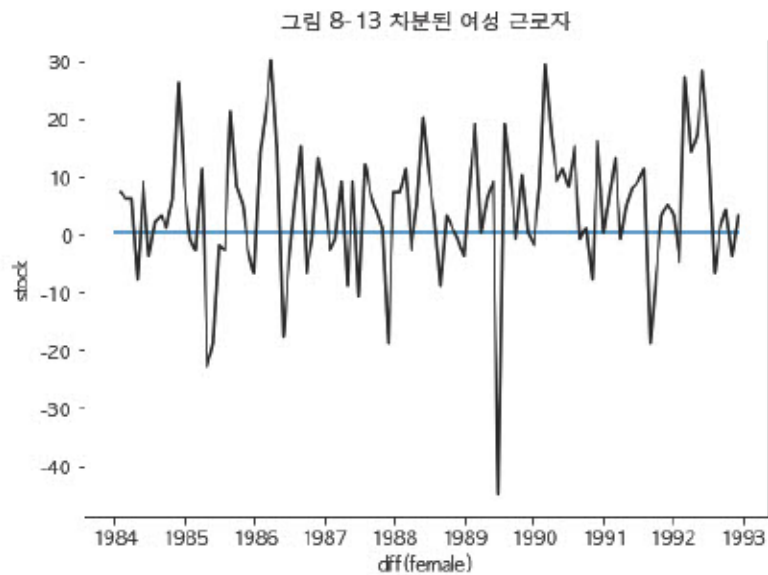


```
In [50]: plot_acf(data)
plot_pacf(data)
plt.show()
```

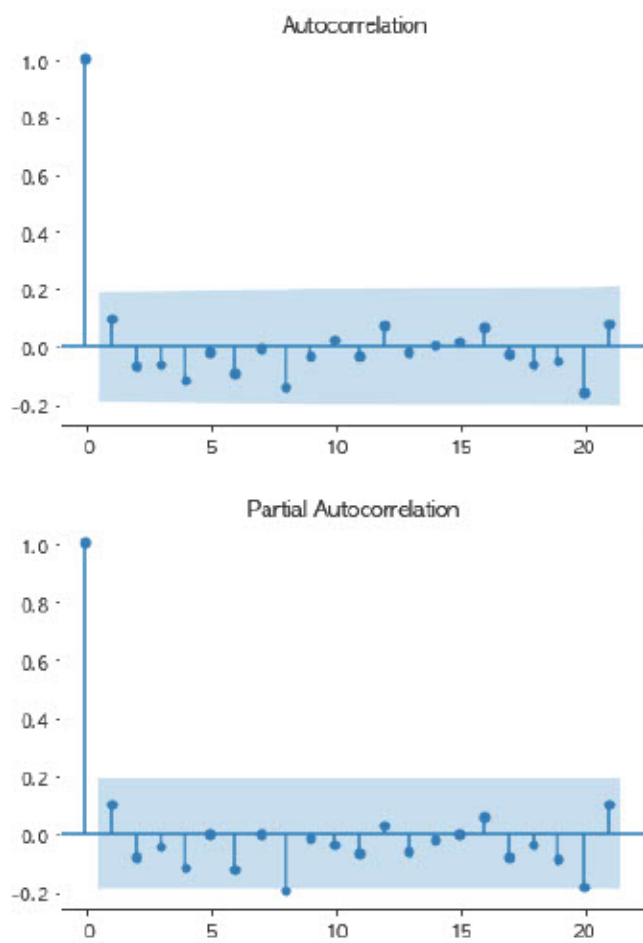


```
In [53]: diff_data = data.diff(1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(diff_data, 'black')
ax.set_xlabel("diff(female)")
ax.set_ylabel("stock")
ax.hlines(0, diff_data.index.min(), diff_data.index.max())
ax.set_title("그림 8-13 차분된 여성 근로자")
plt.show()
```



```
In [55]: plot_acf(diff_data[1:])  
plot_pacf(diff_data[1:])  
plt.show()
```



In []: