

Timeseries_Analysis_HW4

Na SeungChan

2025-05-16

Q.07

난수 생성

```
set.seed(42)
phi <- runif(1, -1, 1)
theta <- runif(1, -1, 1)

set.seed(1)
ARt <- arima.sim(n = 100, model = list(ar = phi))
set.seed(1)
MAt <- arima.sim(n = 100, model = list(ma = theta))
```

AIC 계산 : AR 데이터

```
AR1 <- arima(ARt, order = c(1,0,0))
AR1$aic #AR(1) 적합시 AIC
```

```
## [1] 255.6853
```

```
AR2 <- arima(ARt, order = c(1,0,1))
AR2$aic #ARMA(1) 적합시 AIC
```

```
## [1] 257.6424
```

AIC 계산 : MA 데이터

```
MA1 <- arima(MAt, order = c(0,0,1))
MA1$aic #AR(1) 적합시 AIC
```

```
## [1] 269.1867
```

```
MA2 <- arima(MAt, order = c(1,0,1))  
MA2$aic #ARMA(1) 적합시 AIC
```

```
## [1] 271.1844
```

여담으로, 이 결과는 일관성이 없다. 이론상 AIC는 불필요한 변수를 포함하지 않은 AR1과 MA1 모델에서 작아야 하지만, 실제로 자료 생성 시 계수에 따라 ARMA(1,1)을 적합했을 때 AIC가 더 작아지기도 한다.

Q.10

data load

```
df10 <- read_csv('ex_ch4_10.txt')
```

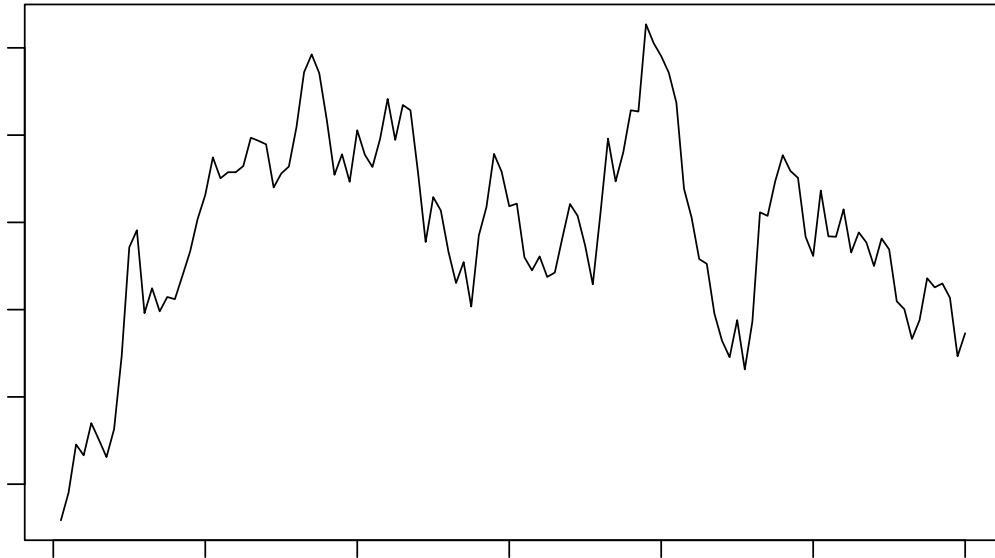
```
## Rows: 120 Columns: 1  
## -- Column specification -----  
## Delimiter: ","  
## dbf (1): data  
##  
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df10 <- ts(df10$data)  
head(df10)
```

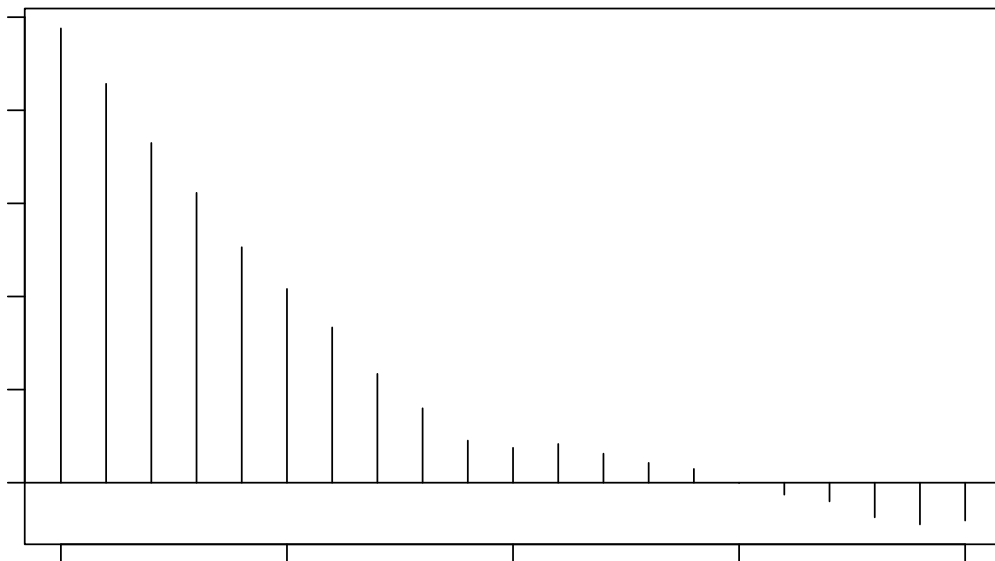
```
## Time Series:  
## Start = 1  
## End = 6  
## Frequency = 1  
## [1] -30.83 -30.20 -29.09 -29.34 -28.60 -28.98
```

(1)

```
plot(df10)
```



```
acf.df10 = acf(x = df10, type = "covariance")
```



시계열도와 SACF를 통해 비정상성을 의심하는 정도는 가능하지만, 정상시계열일 가능성을 배제할 수 없다. 추가적인 단위근검정이 필요해 보인다.

(2)

```
adf.test(df10, alternative = 'stationary')
```

```
##
## Augmented Dickey-Fuller Test
##
## data: df10
```

```
## Dickey-Fuller = -3.0271, Lag order = 4, p-value = 0.15
## alternative hypothesis: stationary
```

해당 시계열이 정상시계열이라고 하기 어렵다.

(3)

```
adf.test(diff(df10))
```

```
## Warning in adf.test(diff(df10)): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: diff(df10)
## Dickey-Fuller = -5.4477, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

1회 차분함으로써 정상시계열이 될 수 있다.

```
auto.arima(df10, d = 1, trace = TRUE)
```

```
##
## ARIMA(2,1,2) with drift      : Inf
## ARIMA(0,1,0) with drift      : 312.1437
## ARIMA(1,1,0) with drift      : 313.4298
## ARIMA(0,1,1) with drift      : 313.3855
## ARIMA(0,1,0)                 : 310.2727
## ARIMA(1,1,1) with drift      : Inf
##
## Best model: ARIMA(0,1,0)

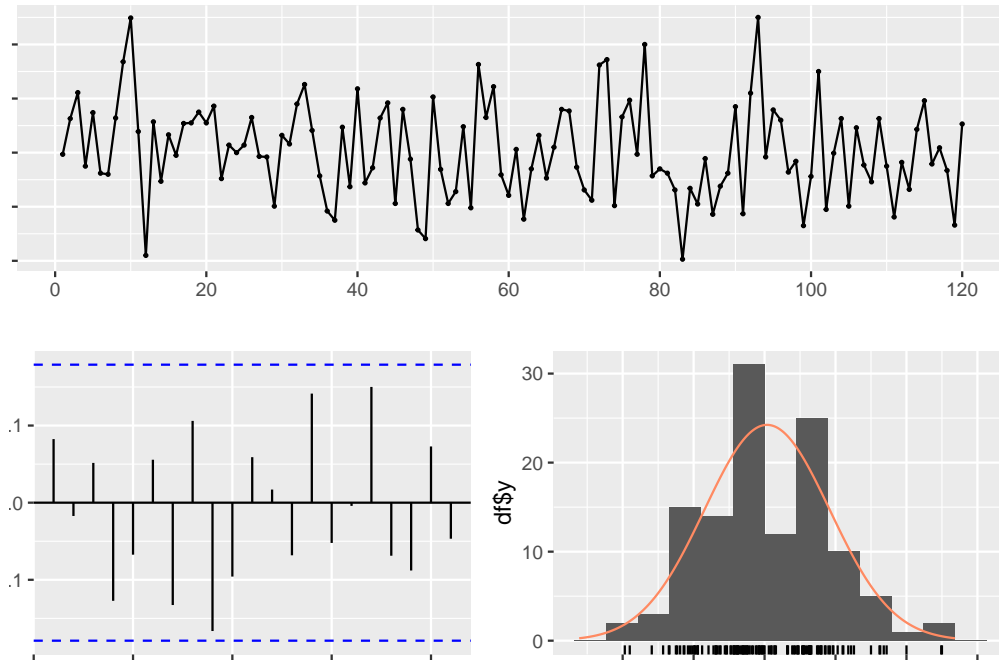
## Series: df10
## ARIMA(0,1,0)
##
## sigma^2 = 0.7807: log likelihood = -154.12
## AIC=310.24 AICc=310.27 BIC=313.02
```

```
model <- arima(df10, order = c(0,1,0))
```

ARIMA(0, 1, 0)을 채택하였다. 차분을 한 번 하는 랜덤워크이다.

(4)

```
checkresiduals(model)
```



```
##
## Ljung-Box test
##
## data: Residuals from ARIMA(0,1,0)
## Q* = 12.852, df = 10, p-value = 0.2321
##
## Model df: 0. Total lags used: 10
```

```
arima(df10, order = c(1,1,0))
```

```
##
## Call:
## arima(x = df10, order = c(1, 1, 0))
##
## Coefficients:
##      ar1
##  0.0839
## s.e. 0.0913
##
## sigma^2 estimated as 0.7751: log likelihood = -153.7, aic = 311.4
```

```
arima(df10, order = c(0,1,1))
```

```
##
## Call:
## arima(x = df10, order = c(0, 1, 1))
##
## Coefficients:
##      ma1
##  0.0889
```

```
## s.e. 0.0947
##
## sigma^2 estimated as 0.7748: log likelihood = -153.68, aic = 311.35
```

잔차분석 및 과적합 진단 결과 ARIMA(1, 1, 0)과 ARIMA(0, 1, 1) 모두 계수가 낮아 유의성을 말하기 어렵다. 잔차의 형태 역시 적절하다.

Q.13

data load

```
gld <- read_csv('ex_ch4_13_gold.csv')
```

```
## Rows: 474 Columns: 2
## -- Column specification -----
## Delimiter: ","
## num (1): price
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gld$date <- as_date(gld$date)
gld
```

```
## # A tibble: 474 x 2
##   date      price
##   <date>   <dbl>
## 1 2015-01-04 1216.
## 2 2015-01-11 1277.
## 3 2015-01-18 1293.
## 4 2015-01-25 1279.
## 5 2015-02-01 1235.
## 6 2015-02-08 1227.
## 7 2015-02-15 1205.
## 8 2015-02-22 1213.
## 9 2015-03-01 1164.
## 10 2015-03-08 1152.
## # i 464 more rows
```

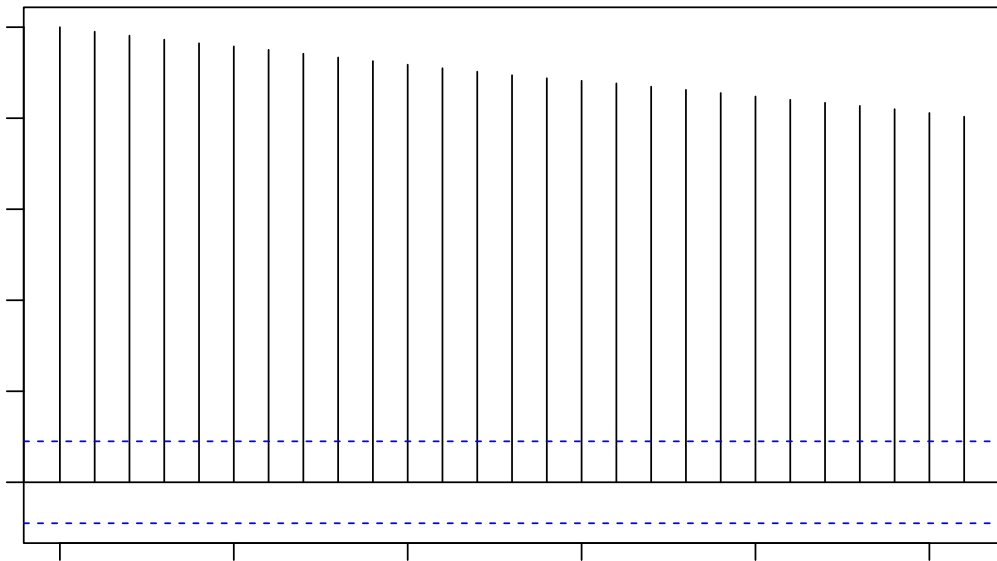
```
gold <- xts(gld$price, order.by = gld$date, col.names = 'price')
```

(1)

```
plot(gold)
```



```
acf(gold)
```



```
adf.test(x = gold, alternative = "stationary")
```

```
##
## Augmented Dickey-Fuller Test
##
## data: gold
## Dickey-Fuller = -2.8001, Lag order = 7, p-value = 0.2396
## alternative hypothesis: stationary
```

plot상 증가 추세가 있고, ACF가 빠르게 감소하지 않으며, ADF test를 통해서도 대립가설인 정상시계열을 기각할 수 없다. 정상시계열이라고 보기 어렵고, 차분과 로그변환을 통해 정상시계열로 만드는 것을 고려할 수 있다.

(2)~(3)

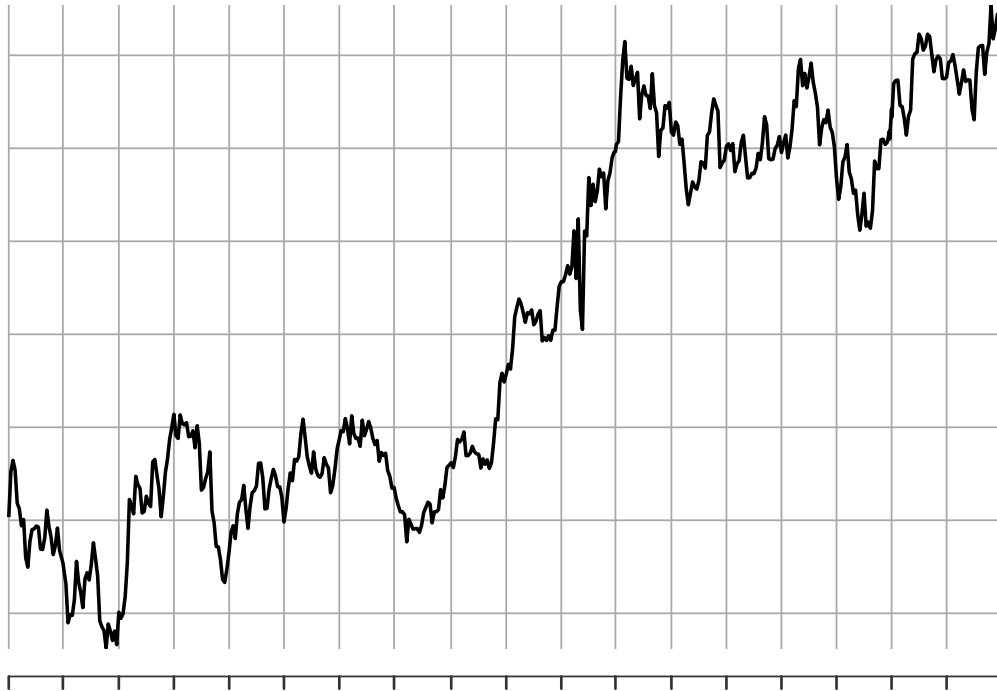
차분 필요. unit root test를 통해 차분의 차수를 구한다.

여러 변환을 고려해 볼 때, 로그를 씌운 가격 데이터를 차분함으로써 분산이 일정하지는 않지만 정상시계열과 유사한 모습을 볼 수 있다. 이와 같은 데이터를 통해 auto.arima를 수행한다.

```
plot(BoxCox(x = gold, lambda = 0.2), type = "l")
```



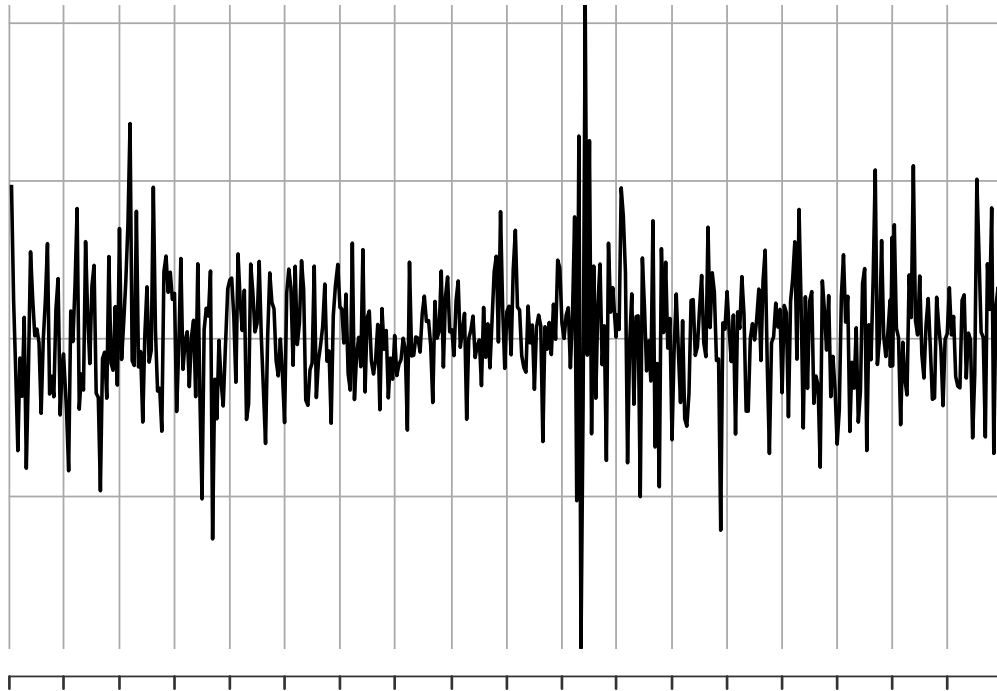
```
plot(log(gold), type = "l")
```

```
adf.test(x = log(gold),  
        alternative = "stationary")
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: log(gold)  
## Dickey-Fuller = -2.8532, Lag order = 7, p-value = 0.2171  
## alternative hypothesis: stationary
```

```
plot(diff(log(gold)), type = "l")
```



```
adf.test(x = na.omit(diff(log(gold))),
        alternative = "stationary")
```

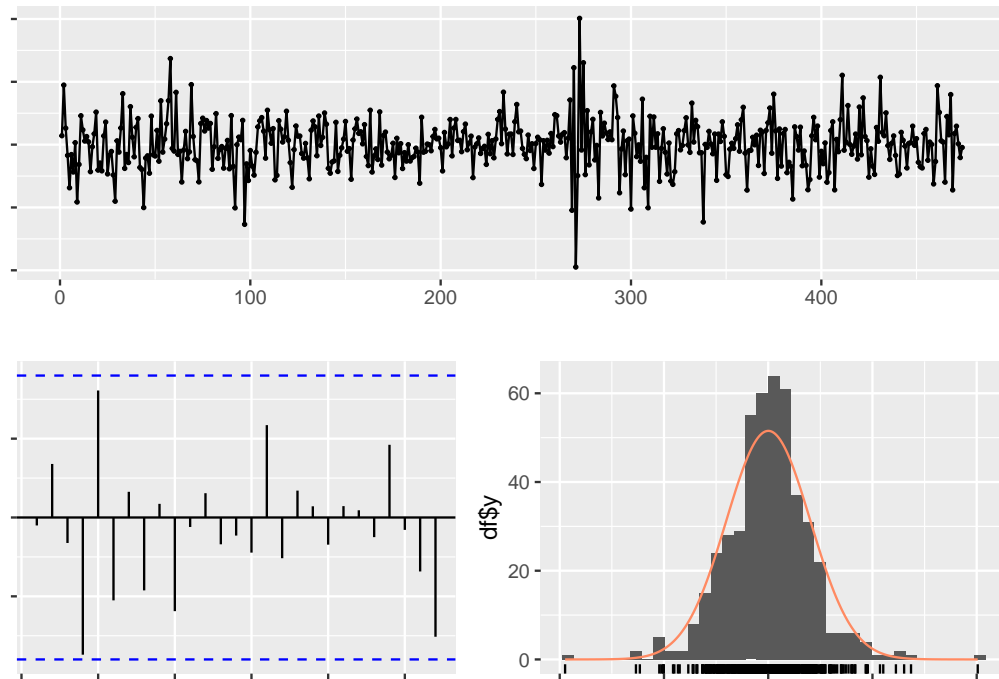
```
## Warning in adf.test(x = na.omit(diff(log(gold))), alternative = "stationary"):
## p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: na.omit(diff(log(gold)))
## Dickey-Fuller = -7.8164, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
model <- auto.arima(log(gold))
model
```

```
## Series: log(gold)
## ARIMA(2,1,2) with drift
##
## Coefficients:
##      ar1   ar2   ma1   ma2  drift
##      1.5455 -0.8529 -1.5996 0.8825 0.0011
## s.e. 0.2198 0.1350 0.2175 0.1566 0.0009
##
## sigma^2 = 0.0004113: log likelihood = 1175.16
## AIC=-2338.32 AICc=-2338.14 BIC=-2313.37
```

```
checkresiduals(model)
```



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,2) with drift  
## Q* = 11.692, df = 6, p-value = 0.06921  
##  
## Model df: 4. Total lags used: 10
```

```
arima(log(gold), order = c(3,1,2))
```

```
##  
## Call:  
## arima(x = log(gold), order = c(3, 1, 2))  
##  
## Coefficients:  
##      ar1   ar2   ar3   ma1   ma2  
##  0.6261 -0.4344 -0.0992 -0.6791 0.4598  
## s.e. 0.4168 0.2741 0.0529 0.4192 0.2799  
##  
## sigma^2 estimated as 0.0004082: log likelihood = 1174.42, aic = -2336.84
```

```
arima(log(gold), order = c(2,1,3))
```

```
##  
## Call:  
## arima(x = log(gold), order = c(2, 1, 3))
```

```
##
## Coefficients:
##      ar1  ar2  ma1  ma2  ma3
##    0.6310 -0.5129 -0.6846 0.5404 -0.1050
## s.e. 0.5547 0.3728 0.5526 0.3771 0.0556
##
## sigma^2 estimated as 0.0004081: log likelihood = 1174.49, aic = -2336.98
```

해당 자료에 대해 로그변환 후 차분 후 시계열을 만들어냄으로써 ARIMA 모델을 적합한다. $\ln(\text{price}) \sim \text{ARIMA}(2, 1, 2)$ 모델을 적합하였으며, 그 적절성을 잔차를 통해 검증하고 한 단계 큰 모델을 통해 확인하였다. 한 단계 큰 모델들을 검토함으로써 해당 모델의 적합성을 확인하였으나, 분산이 안정되지 않은 점 아쉽다.