== 사례연구 #4 ==

(통계기반 데이터 분석)

이상언

- 순 서 -

Ⅰ. 다음의 방법을 이용하여 **2020**년과 **2021**년 우리나라 총 인구를 예측하시오.

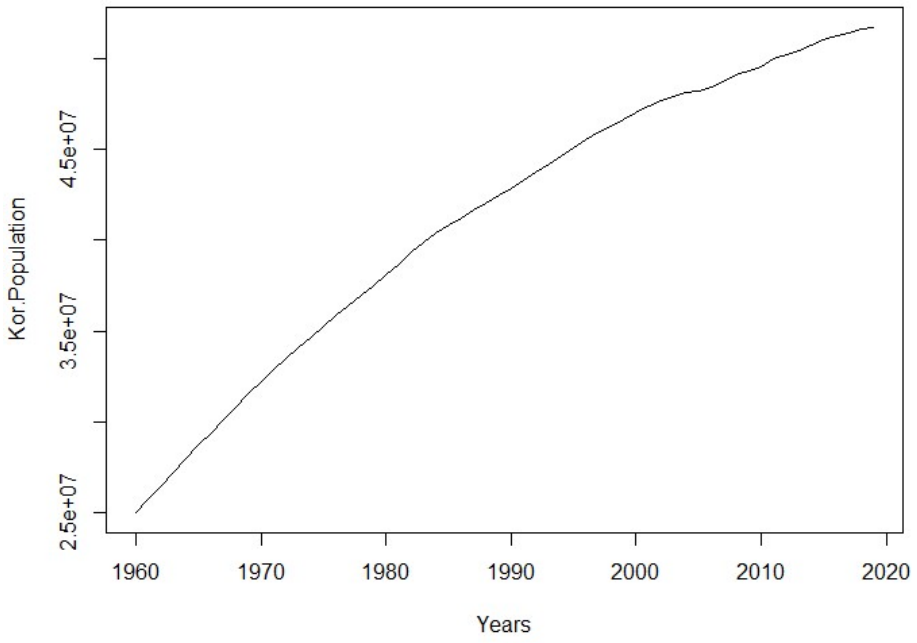
1. **[세계은행]에 있는 연도별 우리나라 인구 데이터를 기반으로 회귀분석 또는 시계열분석을 이용하여 예측하시오.**

   1) **데이터 준비 및 전처리 (시계열 자료로 변환)**

```
# 데이터 준비


kor.pop <- read_excel("/rwork/worldbank_korea.xlsx")

# 데이터 전처리
kor.pop <- t(kor.pop)  # 전치행렬
df.kor.pop <- as.data.frame(kor.pop)  # 구조 변경
colnames(df.kor.pop) <- "인구수"  # 열이름 정리

kor.pop <- ts(df.kor.pop, start = 1960, end = 2019, frequency = 1)
kor.pop
```

   2) **준비된 시계열 자료 시각화**

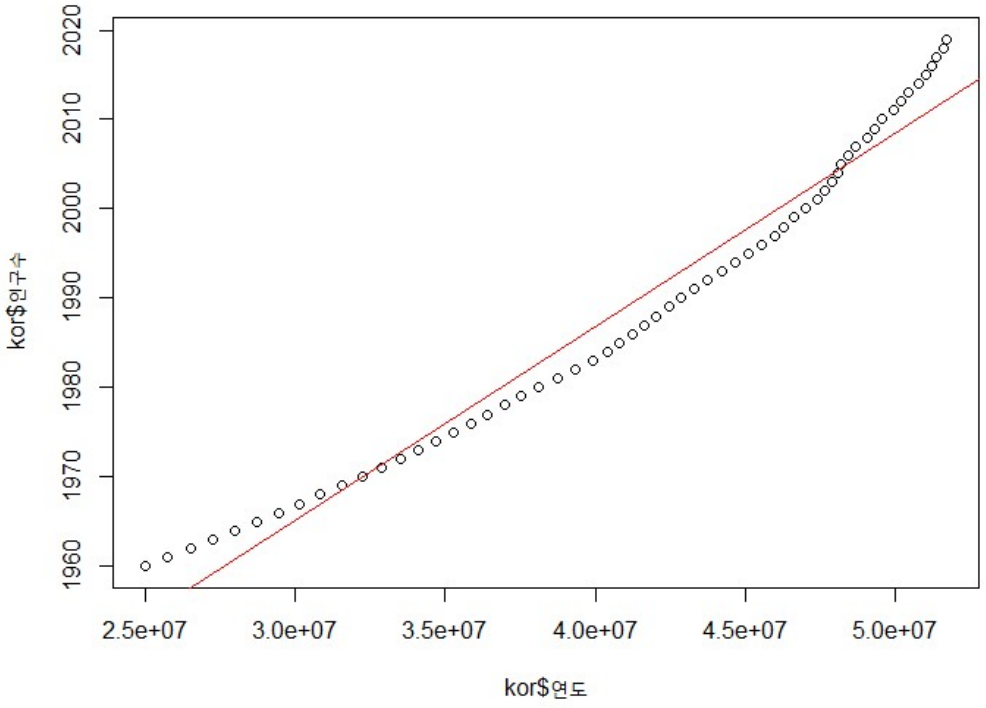
```
# 단일 시계열 자료 시각화
ts.plot(kor.pop, xlab = "Years", ylab = "Kor.Population")
```



시각화 결과 추세 요인은 뚜력하게 보이나 계절요인과 순환 요인은 볼 수 없다. 그러므로 단일 추세 분석이 가장 적합하다. 또한 정상 시계열(Stationary)이 아니므로 회귀 분석을 하는 데 문제가 있다.

3) 회귀 분석법 적용 검정

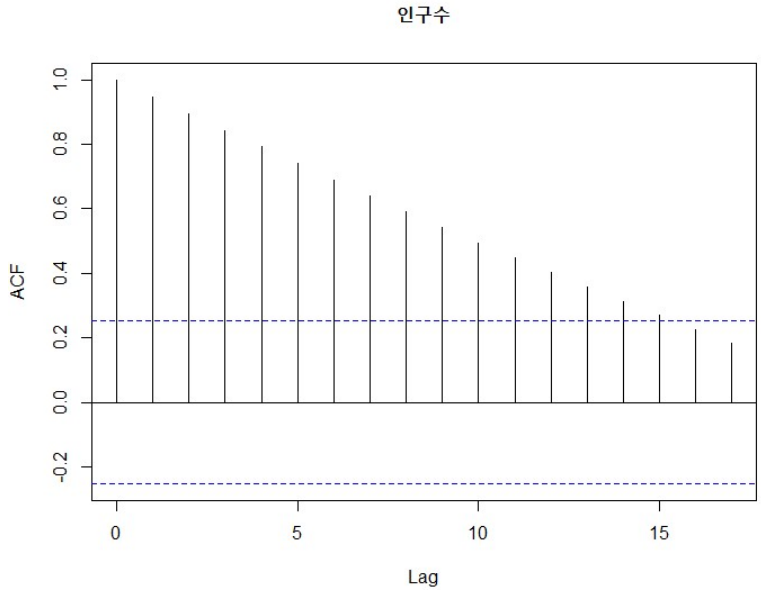
```
# 회귀 분석법 적용 검증
df.kor <- df.kor.pop %>% mutate(year = row.names(df.kor.pop))
colnames(df.kor) <- c("인구수", "연도")  # 이중 데이터를 만들기 위해 시계열
                                         # 자료를 데이터프레임으로 변환
df.kor <- df.kor[, c(2, 1)]

qqplot(df.kor$인구수, df.kor$연도)

formula <- kor$인구수 ~ kor$연도
lm.kor <- lm(formula = formula, data = df.kor)
summary(lm.kor)  # 분석 결과 p-value가 0.05보다 작으므로 회귀식으로서 성립한다.

plot(formula = formula, data = df.kor)
abline(lm.kor, col = "red")  # 회귀 분석 (잔차)에 의한 추세선
```



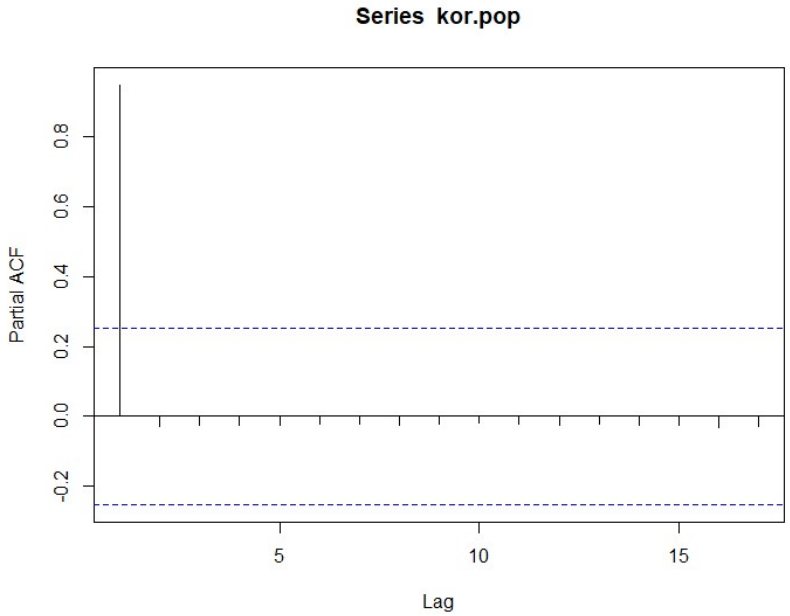
qqplot은 선형적 모양의 그래프에서는 기울기가 45도인 직선에 점들이 밀집 되어 있어야 정규분포를 가정할 수 있다. 따라서 윗 그래프와 같이 나오는 형태는 정규분포를 가정 할 수 없다. 이는 회귀분석의 가정중 정규성, 등분산성, 독립성 중 정규서에 위배되는 상황. 또한 분석 결과 p-value가 0.05보다 작으므로 회귀식으로서 성립하지만 정규성에 위배 된다.

4) ACF(자기 상관 함수)와 PACF(부분 자기 상관 함수) 확인

```
# ACF와 PACF 확인
acf(kor.pop)
pacf(kor.pop)
```



ACF 1 ~ 15까지(시차 0은 제외) 모든 시차(Lag)에서 임계값을 초과 하고 있으므로 서로 이웃한 시점간의 자기 상관성이 매우 뚜렷하다.



PACF 첫 번째 시차에서 임계값을 넘어가므로 자기 상관성이 충분히 있다

5) 모형 식별(ARIMA)

```
# 모형 식별과 모델링
auto.arima(kor.pop)
model <- arima(kor.pop, order = c(0, 2, 1))
plot(model$residuals)

# 모형의 타당성 검정
tsdiag(model)

Box.test(model$residuals, lag = 1, type = "Ljung")

# 차분을 통한 모형 확인
diff <- diff(kor.pop, differences = 2)
plot(diff)
```

※ Auto.arima 분석 결과

Series: kor.pop

ARIMA(0,2,1)

Coefficients:

ma1

-0.2626

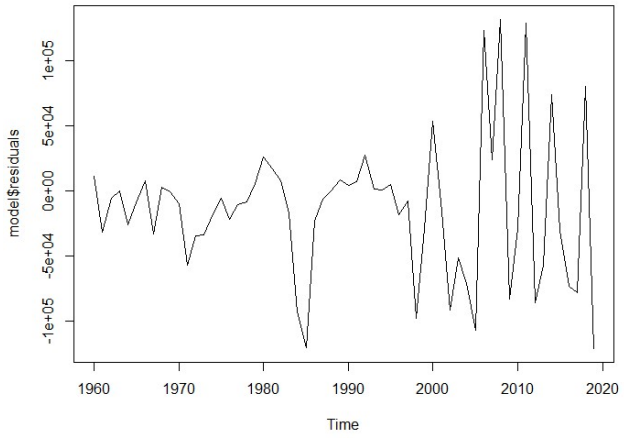
s.e. 0.1280

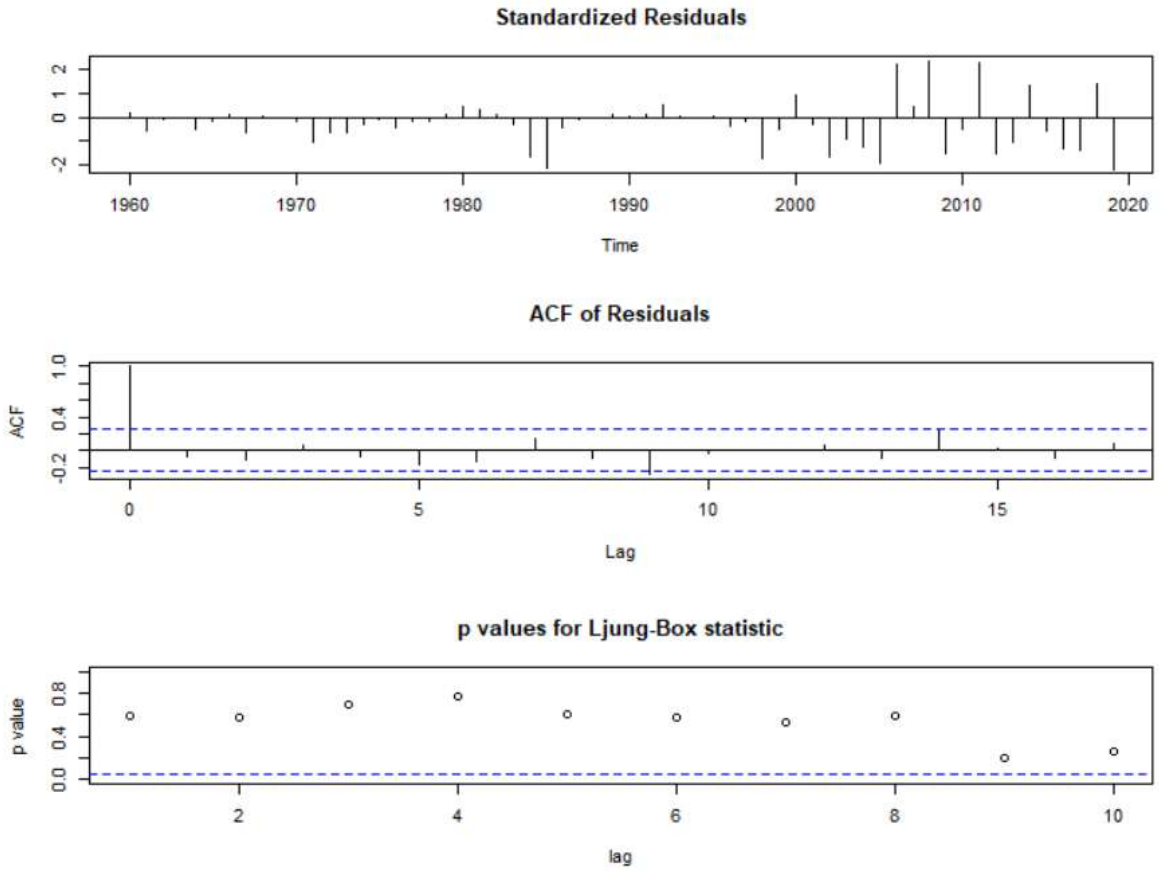
sigma^2 estimated as 3.231e+09: log likelihood=-716.63

AIC=1437.27 AICc=1437.49 BIC=1441.39

ARIMA 모형과 파라미터는 ARIMA(0, 2, 1)로 확인된다. 그러므로 IMA(2, 1)모형이며 이는 2번 차분하면 MA(1) 모형을 따른 다는 것을 알 수 있다.

※ **잔차를 사용한 그래프**





Tsdiag ACF에서 자기 상관이 발견되지 않고, p-value값이 0 이상으로 분포되어 있으므로 현재 모델은 타당하다고 추론할 수 있다.

Box test 잔차항 모형진단 역시 p-value 값 0.5963 으로 0.05보다 크므로 통계적으로 의미 있다.

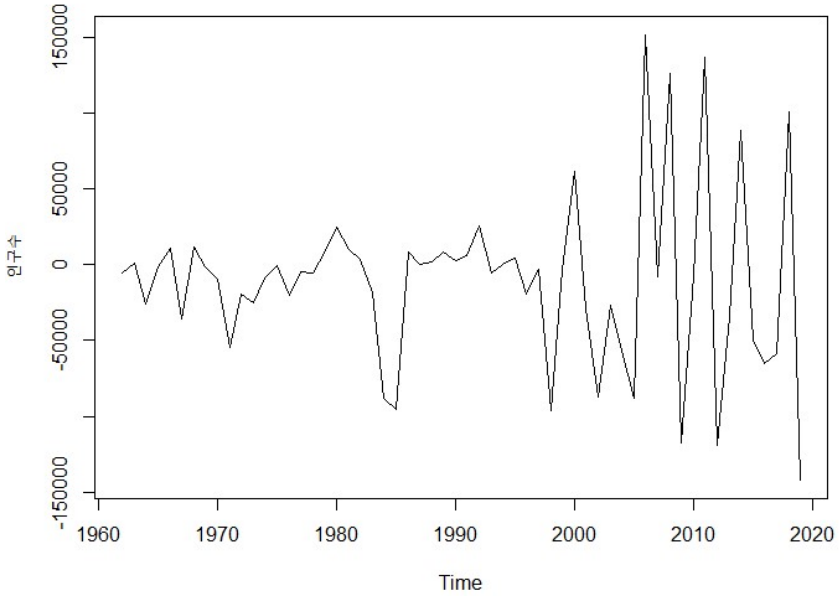
※ Boxtest 진단 결과

Box-Ljung test

data: model$residuals

X-squared = 0.28056, df = 1, p-value = 0.5963

※ **2번 차분된 그래프**



두 번 차분을 하게 되면 arima 분석에 파라미터와 동일해진다는 것을 알 수 있다.

6) 2020년, 2021년 예측

```
# 예측
fore <- forecast(kor.pop, h = 2)
plot(fore, type = "l")
fore
```

※ **예측 결과**

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2020	51851806	51759550	51944062	51710713	51992900
2021	51991503	51818119	52164887	51726335	52256671

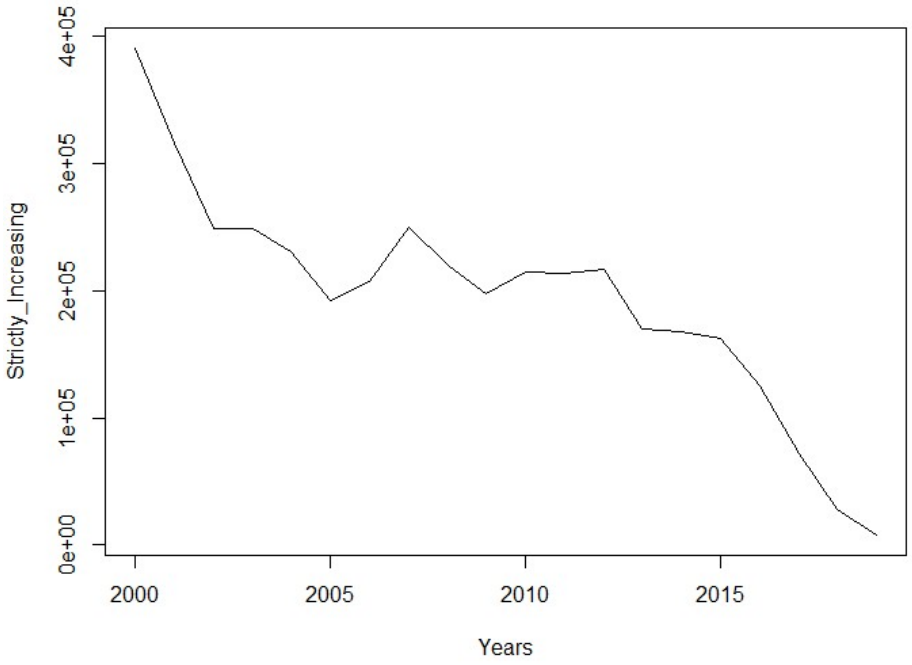
2. 과거 연도별 인구의 순증감을 기반으로 회귀분석 또는 시계열분석을 이용하여 예측하시오.

1) 분석 과정 코드

```
# 데이터 준비
raw.birth <- read_excel("/rwork/birth.xlsx")  # 출생아수
raw.death <- read_excel("/rwork/death.xlsx")  # 사망자수
birth <- t(as.data.frame(raw.birth))  # 데이터 전치 및 데이터프레임 변환
death <- t(as.data.frame(raw.death))  # 데이터 전치 및 데이터프레임 변환
strictly.increasing <- birth - death  # 순증감 데이터 만들기
colnames(strictly.increasing) <- "순증감"  # 컬럼명 변경
ts.si <- ts(strictly.increasing, start = 2000,
            end = 2019, frequency = 1)  # 시계열 자료로 변환
ts.si
```

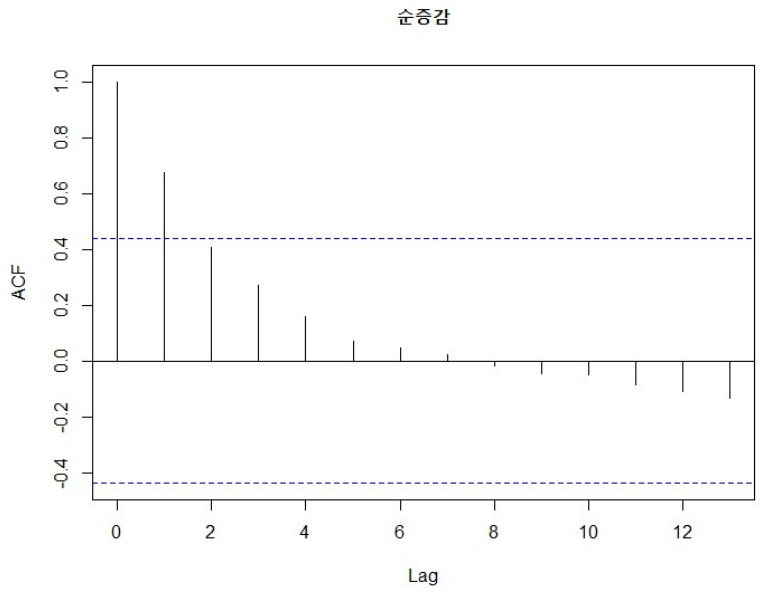
```
# 시각화
plot(ts.si, xlab = "Years", ylab = "Strictly_Increasing")

# ACF와 PACF 확인
acf(ts.si)
pacf(ts.si)

# 모형 식별과 모델링
auto.arima(ts.si)
model2 <- arima(ts.si, order = c(0, 1, 1))
plot(model2$residuals)

# 모형의 타당성 검정
tsdiag(model2)

Box.test(model2$residuals, lag = 1, type = "Ljung")

# 예측
fore.si <- forecast(ts.si, h = 2)
plot(fore.si, type = "l")
fore.si
```

2) 인구 순증감 시계열 자료 시각화

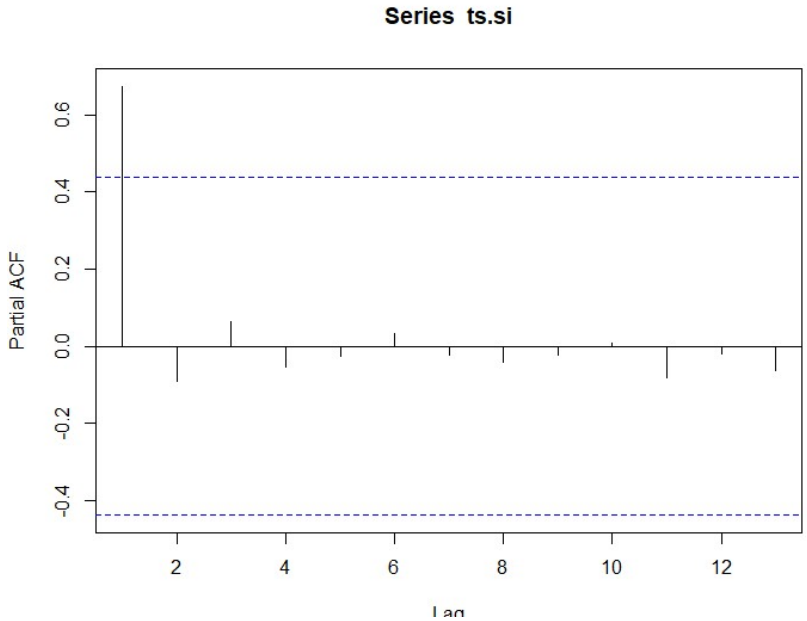


시각화 결과 추세 요인은 뚜렷하게 보이나 계절요인과 순환 요인은 볼 수 없다. 그러므로 단일 추세 분석이 가장 적합하다. 또한 정상 시계열(Stationary)이 아니므로 회귀 분석을 하는데 문제가 있다.

3) ACF(자기 상관 함수)와 PACF(부분 자기 상관 함수) 확인



ACF 분석 결과 두 번째 시차가(시차 0은 제외)임계값을 초과 하고 있으므로 서로 이웃한 시점간의 자기 상관성이 있다.



PACF 분석 결과첫 번째 시차에서 임계값을 넘어가므로 자기 상관성이 충분히 있다.

4) 모형 식별과 모델링 (ARIMA)

※ AutoArima 결과

ARIMA(0,1,1) with drift

Coefficients:

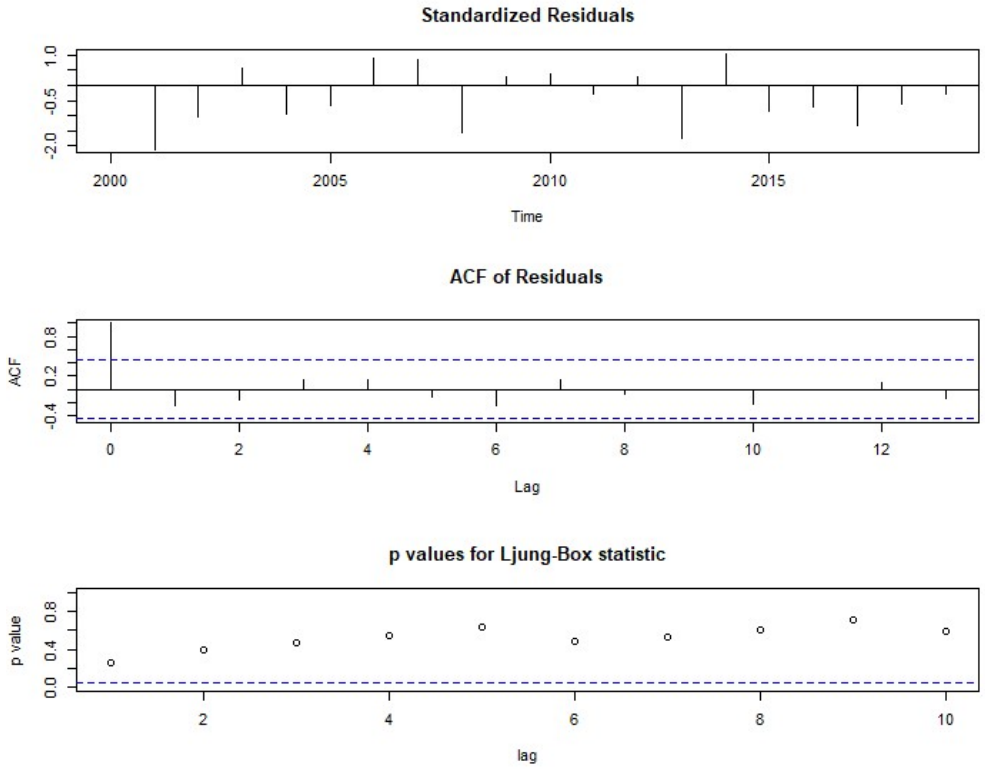
	ma1	drift
	0.5341	-20774.017
s.e.	0.2344	9397.924

sigma^2 estimated as 826064139: log likelihood=-221.13

AIC=448.25 AICc=449.85 BIC=451.09

모형과 파라미터는 ARIMA(0, 1, 1)로 확인된다. 그러므로 IMA(1, 1)모형이며 이는 1번 차분하면 MA(1) 모형을 따른 다는 것을 알 수 있다.

5) 모델링 검정



ACF에서 자기 상관이 발견되지 않고, p-value값이 0 이상으로 분포되어 있으므로 현재 모델은 타당하다고 추론할 수 있다.

※ Box.test 결과

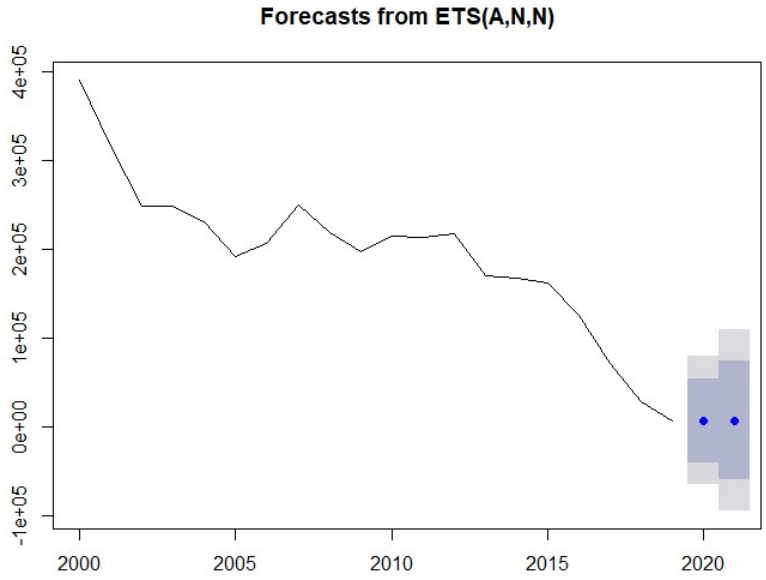
Box-Ljung test

data: model2$residuals

X-squared = 1.2831, df = 1, p-value = 0.2573

잔차항 모형진단 역시 p-value 값 0.5963 으로 0.05보다 크므로 통계적으로 의미 있다.

6) 2020년, 2021년 예측



※ 예측 결과

Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2020	7568.048	-39672.57	54808.67	-64680.25	79816.34
2021	7568.048	-59236.94	74373.03	-94601.36	109737.45

3. 연도별 우리나라 인구 데이터를 기반으로 **AAGR(Average Annual Growth Rate)**을 계산하시오.

1) AAGR 공식

The Formula for the Average Annual Growth Rate (AAGR) Is

$$AAGR = \frac{GR_A + GR_B + \ldots + GR_n}{N}$$

where:

GR_A = Growth rate in period A

GR_B = Growth rate in period B

GR_n = Growth rate in period n

N = Number of payments

2) AAGR을 계산한 코드

```
# 1-3 AAGR 계산
install.packages("magicfor")  # for문 처리용 패키지
library(magicfor)

magic_for(print, silent = T)  # for문을 저장하기 위한 함수
i = 1
for(i in 1:59){
  aagr <- (kor.pop[i + 1] / kor.pop[i] - 1) * 100
  print(aagr)
}
aagr <- magic_result_as_dataframe(aagr)
aagr <- aagr[, -1]

result.aagr <- round(sum(aagr) / length(aagr), 2)
result.aagr  # AAGR = 1.24%
```

※ **AAGR 값은 1.24%**

4. 연도별 우리나라 인구 데이터를 기반으로 **CAGR(Compound Annula Growth Rate)** 을 계산하시오.

1) CAGR 공식

The Formula for CAGR Is:

$$CAGR = \frac{\text{Ending Balance}}{\text{Beginning Balance}}^{\frac{1}{\#\text{Years}}} - 1$$

2) CAGR을 계산한 코드

```
cagr <- round((((kor.pop[60] / kor.pop[1]) ^ (1 / length(kor.pop))) - 1) * 100,
             2)
cagr  # CAGR = 1.22%
```

※ **CAGR 값은 1.22%**

5. 다른 기법을 알고 있다면 다른 기법을 사용하여 예측하시오.

1) 이동평균법을 통한 예측

```
rm.kor <- df.kor %>% mutate(ra = runMean(인구수, 2))  # 이동평균법을 통한 예측
forecast(kor$ra, h = 2)
```

※ 이동평균법을 통한 예측 결과

Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
61	51827080	51775159	51879002	51747673	51906488
62	51992020	51877100	52106940	51816265	52167775

2) 단순지수평활법을 통한 예측

```
ses.kor <- ses(kor.pop, h = 2)  # 단순지수평활법을 통한 예측
ses.kor
```

※ 단순지수평활법을 통한 예측 결과

Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2020	51709088	50868343	52549832	50423280	52994896
2021	51709088	50520155	52898021	49890771	53527404

3) 홀트선형지수 평활법을 통한 예측

```
holt.kor <- holt(kor.pop, damped = T, h = 2)  # 홀트 선형지수평활법을 통한 예측
kor.holt
plot(kor.holt)
```

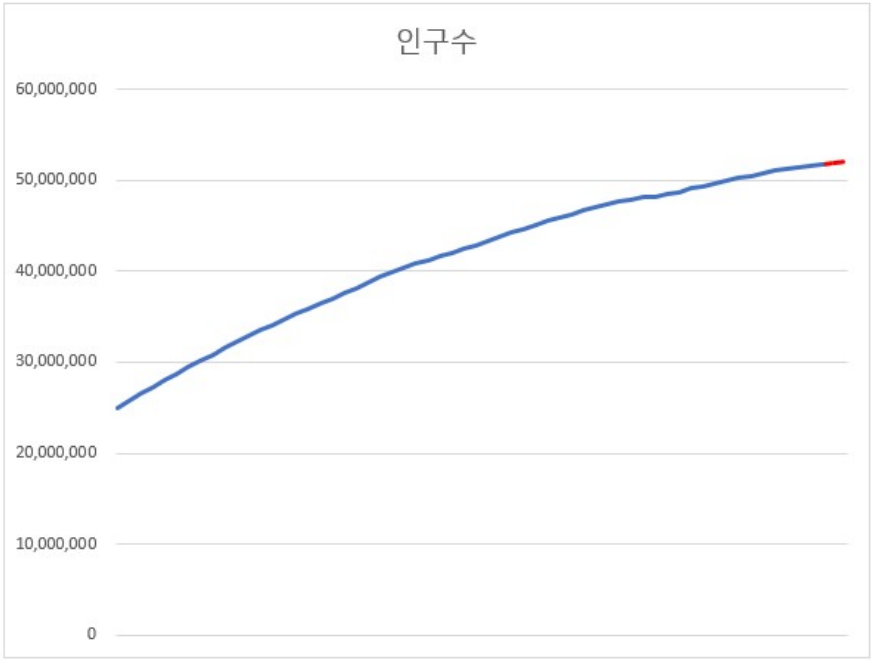
※ 홀트선형지수 평활법을 통한 예측 결과

Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2020	51882256	51758939	52005574	51693659	52070854
2021	52055403	51842533	52268272	51729847	52380958

4) 지수평활법을 통한 예측 (엑셀 함수 사용)

이전 데이터	인구수
1960년 01월	25,012,374
1961년 01월	25,765,673
1962년 01월	26,513,030
1963년 01월	27,261,747
1964년 01월	27,984,155
1965년 01월	28,704,674
1966년 01월	29,435,571
1967년 01월	30,130,983
1968년 01월	30,838,302
1969년 01월	31,544,266
1970년 01월	32,240,827
1971년 01월	32,882,704
1972년 01월	33,505,406
1973년 01월	34,103,149
1974년 01월	34,692,266
1975년 01월	35,280,725
1976년 01월	35,848,523
1977년 01월	36,411,795
1978년 01월	36,969,185
1979년 01월	37,534,236
1980년 01월	38,123,775
1981년 01월	38,723,248
1982년 01월	39,326,352
1983년 01월	39,910,403
1984년 01월	40,405,956
1985년 01월	40,805,744
1986년 01월	41,213,674
1987년 01월	41,621,690
1988년 01월	42,031,247

예측 데이터	예상 인구수
2020년 01월	51,844,313.26
2021년 01월	51,967,913.51



6. 사용된 기법의 결과 비교

기법 이름	예측 결과
ARIMA 모델링을 통한 예측	**Point Forecast Lo 80 Hi 80 Lo 95 Hi 95** **2020 51851806 51759550 51944062 51710713 51992900** **2021 51991503 51818119 52164887 51726335 52256671**
이동평균법을 통한 예측	**Point Forecast Lo 80 Hi 80 Lo 95 Hi 95** **61 51827080 51775159 51879002 51747673 51906488** **62 51992020 51877100 52106940 51816265 52167775**
단순지수평활법을 통한 예측	**Point Forecast Lo 80 Hi 80 Lo 95 Hi 95** **2020 51709088 50868343 52549832 50423280 52994896** **2021 51709088 50520155 52898021 49890771 53527404**
홀트 선형지수평활법을 통한 예측	**Point Forecast Lo 80 Hi 80 Lo 95 Hi 95** **2020 51882256 51758939 52005574 51693659 52070854** **2021 52055403 51842533 52268272 51729847 52380958**
지수평활법을 통한 예측**(EXCEL)**	**2020 : 51844313** **2021 : 51967913**

Ⅱ. 제공된 **SPSS**파일인 “**drinking_water_example.sav**”파일의 데이터 셋을 가지고 각 단계별로 요인 분석을 수행하시오.

1. **1**단계 : 데이터 가져오기

```
# 데이터 준비
install.packages("memisc")
library(memisc)
data.spss <- as.data.set(spss.system.file("/rwork/drinking_water_example.sav"))
data.spss

# 데이터프레임으로 변환
drinking.water <- data.spss[1:7]  # 11개 변수선택
drinking.water.df <- as.data.frame(data.spss[1:7])
str(drinking.water.df)
drinking.water.df
```

2. **2**단계 : 베리맥스 회전법, 요인점수 회귀분석 방법을 적용하여 요인 분석

```
result <- factanal(drinking.water.df, factor = 2, rotation = "varimax",
                   scores = "regression")
result
```

```
Call:
factanal(x = drinking.water.df, factors = 2, scores = "regression",     rotation = "varimax")

Uniquenesses:
   Q1    Q2    Q3    Q4    Q5    Q6    Q7
0.333 0.222 0.298 0.388 0.200 0.231 0.410

Loadings:
   Factor1 Factor2
Q1 0.212   0.789
Q2 0.182   0.863
Q3 0.170   0.820
Q4 0.724   0.296
Q5 0.882   0.149
Q6 0.860   0.172
Q7 0.742   0.198

               Factor1 Factor2
SS loadings      2.700   2.219
Proportion Var   0.386   0.317
Cumulative Var   0.386   0.703

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 12.93 on 8 degrees of freedom.
The p-value is 0.114
```

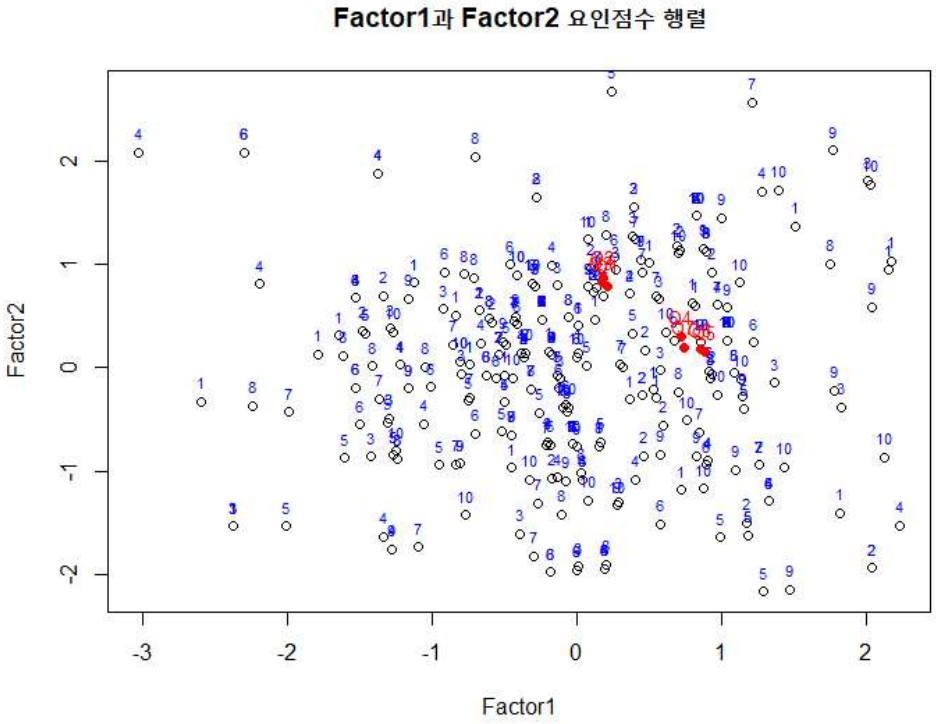
해석 : Uniquenesses(유효성 판단 항목)의 값이 통상 0.5 이하이면 유효한 것으로 판단 할 수 있다. 따라서 7개의 변수 모두 유효하다고 볼 수 있음. // Loadings 항목은 요인 적재값을 보여주는 항목으로 요인부하량이 통상 +0.4 이상이면 유의하다고 볼 수 있다. Factor1에서는 Q4, Q5, Q6, Q7 변수의 적재값이 이에 해당하며, Factor2 에서는 Q1, Q2, Q3 변수가 해당된다. 따라서 첫 번째 요인으로 Q4, Q5, Q6, Q7이 묶이고, 두 번째 요인으로 Q1, Q2, Q3이 묶인다. // Cumulative Var 항목은 누적분산비율로 Factor2 까지 누적 합이 0.703 으로 정보 손실은 약 0.3 이다. // p값은 0.05보다 큰 0.114 이므로 요인선택은 적절하다고 볼 수 있다.

3. 3단계 : 요인적재량 행렬의 칼럼명 변경

```
loadings <- result$loadings
colnames(loadings) <- c("제품만족도", "제품친밀도")
loadings
```

4. 4단계 : 요인점수를 이용한 요인 적재량 시각화

```
# 4단계
plot(result$scores[, c(1, 2)],
     main = "Factor1과 Factor2 요인점수 행렬")  # 시각화
text(result$scores[, 1], result$scores[, 2],
     labels = name, cex = 0.7, pos = 3, col = "blue")

# 요인적재량 추가
points(result$loadings[, c(1, 2)], pch = 19, col = "red")  # 요인적재량 표시
text(result$loadings[, 1], result$loadings[, 2],
     labels = rownames(result$loadings),
     cex = 0.8, pos = 3, col = "red")
```



4 단계 해석 : x축을 구성하는 Factor1의 요인점수를 기준으로 1에 가까운 요인은 Q4, Q5, Q6, Q7로 나타나고 y축을 구성하는 Factor2의 요인점수를 기준으로 1에 가까운 요인은 Q1, Q2, Q3으로 나타난다.

5. 5단계 : 요인별 변수 묶기

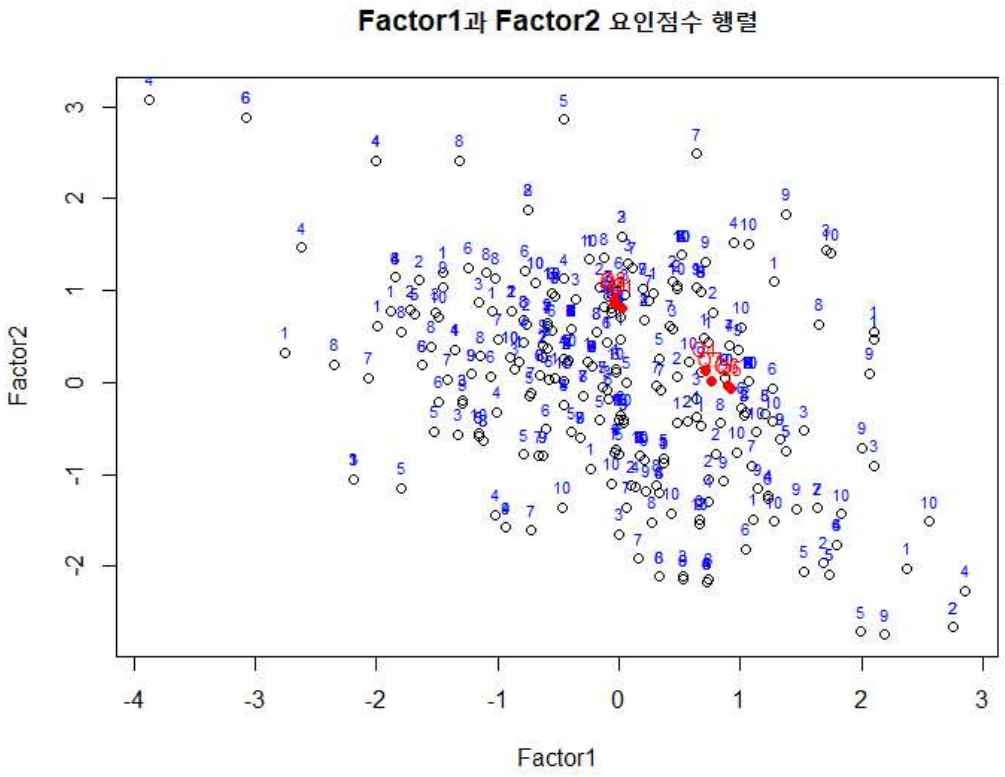
```
# 제품친밀도 데이터프레임 - Q1, Q2, Q3
c <- data.frame(drinking.water.df$Q1, drinking.water.df$Q2, drinking.water.df$Q3)
# 제품만족도 데이터프레임 - Q4, Q5, Q6
s <- data.frame(drinking.water.df$Q4, drinking.water.df$Q5,
                drinking.water.df$Q6, drinking.water.df$Q7)
# 요인별 산술평균 계산
closeness <- round((c$drinking.water.df.Q1 + c$drinking.water.df.Q2 +
                      c$drinking.water.df.Q3) / ncol(c), 2)
satisfaction <- round((s$drinking.water.df.Q4 + s$drinking.water.df.Q5 +
                         s$drinking.water.df.Q6 + s$drinking.water.df.Q7) /
                        ncol(s), 2)
```

6. 6단계 : 프로맥스 회전법을 적용하여 요인 분석

```
result2 <- factanal(drinking.water.df, factor = 2, rotation = "promax",
                    scores = "regression")
result2
```

7. 베리맥스와 프로맥스 회전법 비교

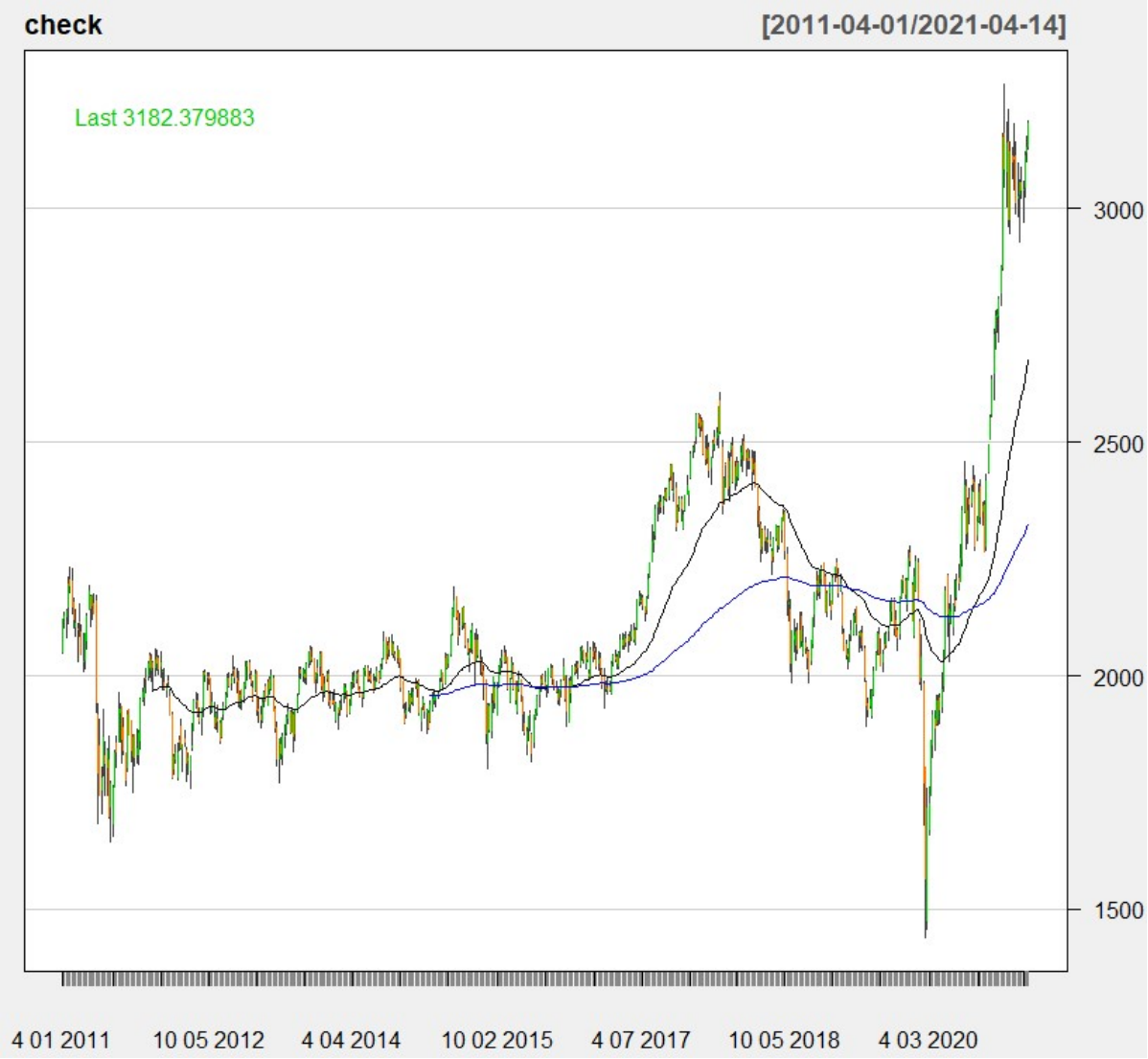
```
plot(result2$scores[, c(1, 2)],
     main = "Factor1과 Factor2 요인점수 행렬")  # 시각화
text(result2$scores[, 1], result2$scores[, 2],
     labels = name, cex = 0.7, pos = 3, col = "blue")

# 요인적재량 추가
points(result2$loadings[, c(1, 2)], pch = 19, col = "red")  # 요인적재량 표시
text(result2$loadings[, 1], result2$loadings[, 2],
     labels = rownames(result2$loadings),
     cex = 0.8, pos = 3, col = "red")
```



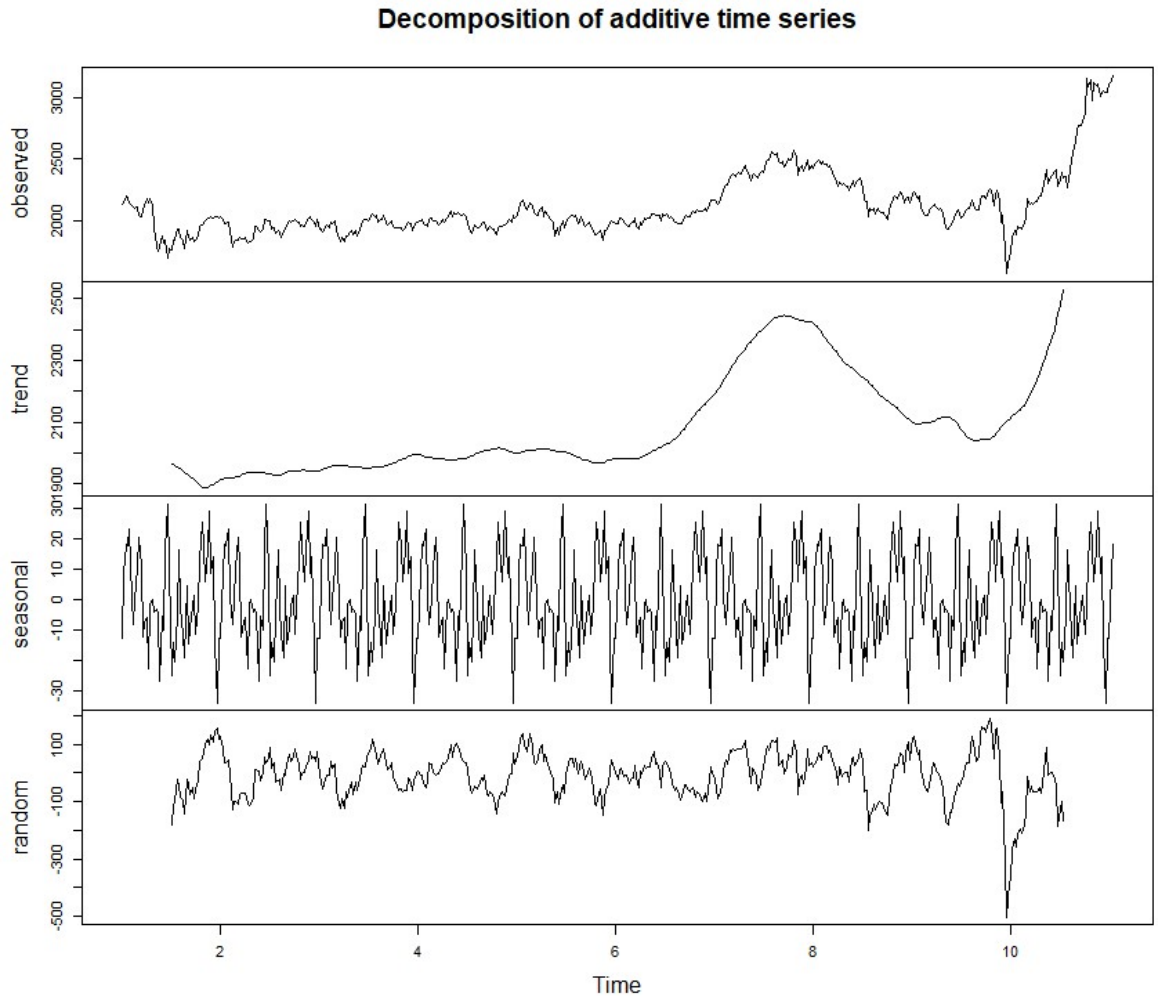
해석 : 결과 값은 요인적재량(Loadings)를 제외하면 큰 차이가 없었다. 요인적재량에서는 promax 분석시 각 Factor에서 다소 높게 측정되었다. 이는 plot결과로 확인해 보았을때 promax 분석을 했을 경우 varimax 분석 보다 변수들이 가깝게 군집하는 결과를 가져다는 것을 알 수 있다. 뿐만 아니라 promax 분석은 요인 간에 얼마나 관계가 있는지 보여줌으로써 요인간의 상관관계를 파악하는데 편리하다.

Ⅲ. 과거 10년간 일별 KOSPI 지수(종가) 데이터를 기준으로 시계열분석을 실시하시오.

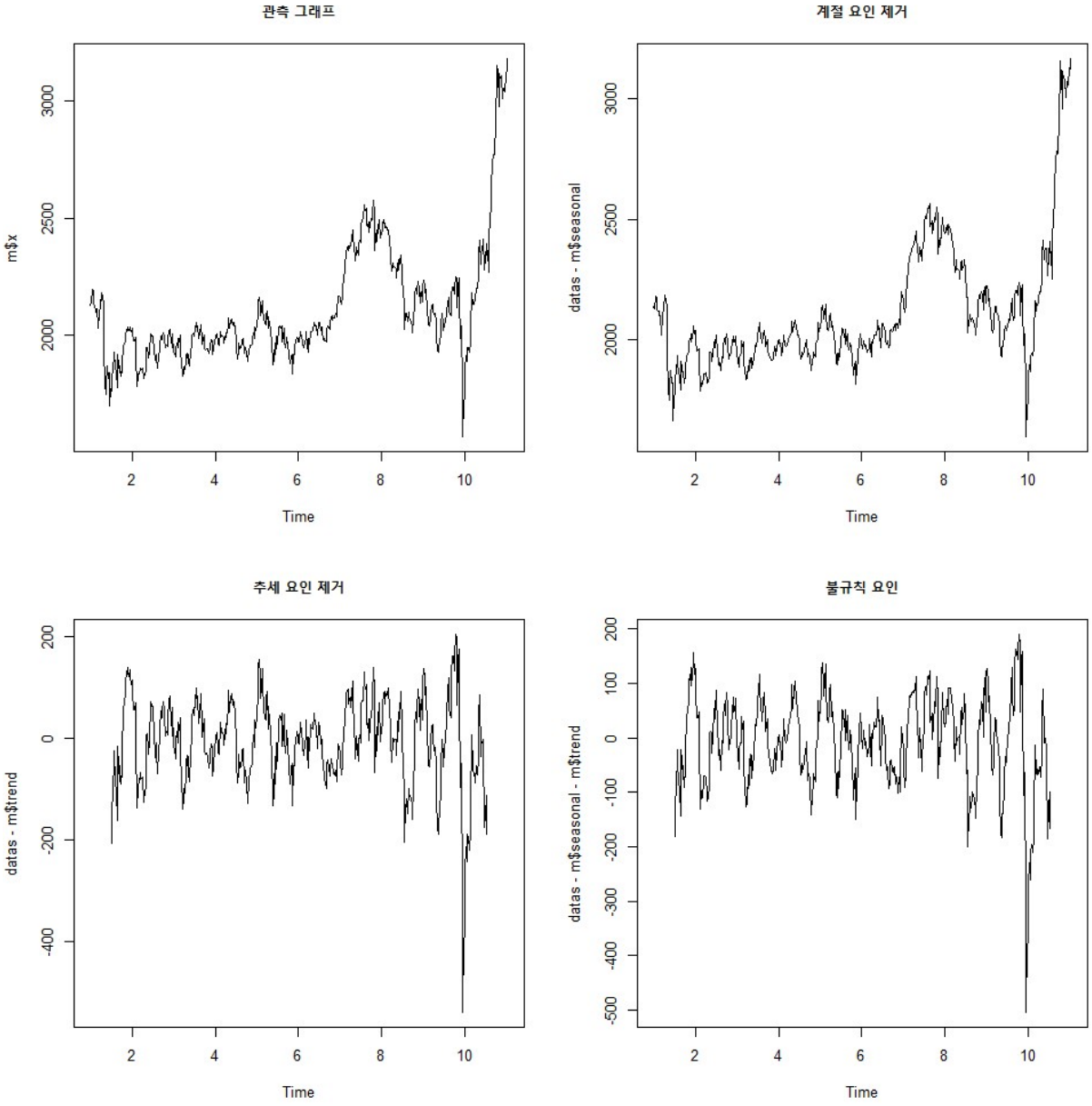
1. 추세선 확인



2. 4가지 시계열 자료의 변동 요인을 분해



3. 분해 상세 시각화


관측 그래프
m$x
Time
계절 요인 제거
datas - m$seasonal
Time
추세 요인 제거
datas - m$trend
Time
불규칙 요인
datas - m$seasonal - m$trend
Time

4. R코드

```r
library(quantmod)
library(dplyr)
library(TTR)

# 데이터 준비
start <- as.Date(Sys.Date() - 3670)
end <-   as.Date(Sys.Date())

Symbol <- c("^KS11")
Stocks = lapply(Symbol[1], function (sym) {
  to.weekly(na.omit(getSymbols(sym, from= start, to = end, auto.assign=FALSE)))
})

# 데이터 전처리
data <- do.call(merge, Stocks)
check <- data
date = index(data)

colnames(data) <- c("Date", "open", "high", "low",
                    "close","volume", "adj", "roc")
data <- na.omit(data)

# 추세선 확인
chartSeries(check, theme=chartTheme('white'),
            type = c("auto", "matchsticks"),
            subset = '2011-04::',
            show.grid = TRUE,
            major.ticks='auto', minor.ticks=TRUE,
            multi.col = FALSE,
            TA="addEMA(50, col='black');addEMA(200, col='blue')")

# 4가지 시계열 자료의 변동요인을 분해
datas = ts((data$close), frequency=52)  # 분해를 위해 일주일 단위로 시즌 형성
m <- decompose(datas)

plot(m)  # 시계열 분해 시각화

par(mfrow = c(2, 2))

plot(m$x, main = "관측 그래프")  # 관측 그래프
plot(datas - m$seasonal, main = "계절 요인 제거")  # 계절 요인 제거 그래프
plot(datas - m$trend, main = "추세 요인 제거")  # 추세 요인 제거
plot(datas - m$seasonal - m$trend, main = "불규칙 요인")  # 불규칙 요인만 출력
```

※ 사용된 라이브러리

```r
library(dplyr)  # 데이터 편집
library(lmtest)  # 선형회귀 모형 진단
library(TTR)  # 이동평균 계산
library(forecast)  # 시계열 자료 예측
library(readxl)  # 엑셀 파일 불러오기
library(memisc)  # spss 파일 불러오기
library(quantmod)  # 주가 정보 불러오기
library(magicfor)  # for문 편집기
```