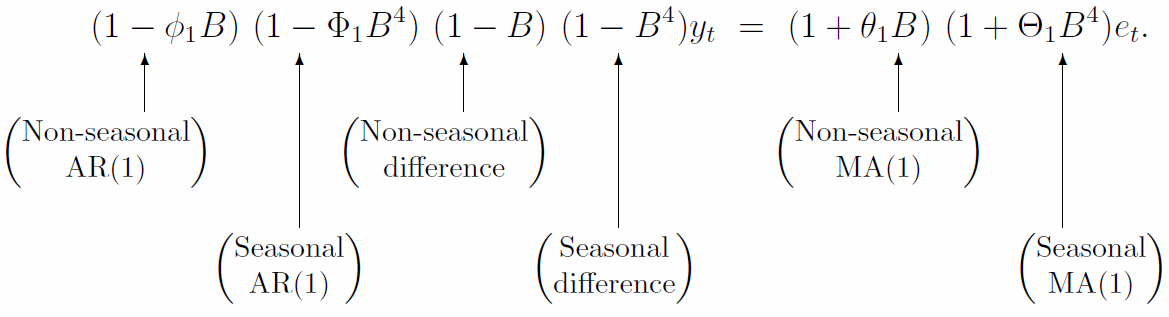
**1 [20 점]. 가 를 따른다고 가정하자. 다음과 같은 확률과정에 대하여 아래의 물음에 답하고 이유를 밝혀라.**



<https://www.datasciencecentral.com/profiles/blogs/tutorial-forecasting-with-seasonal-arima>

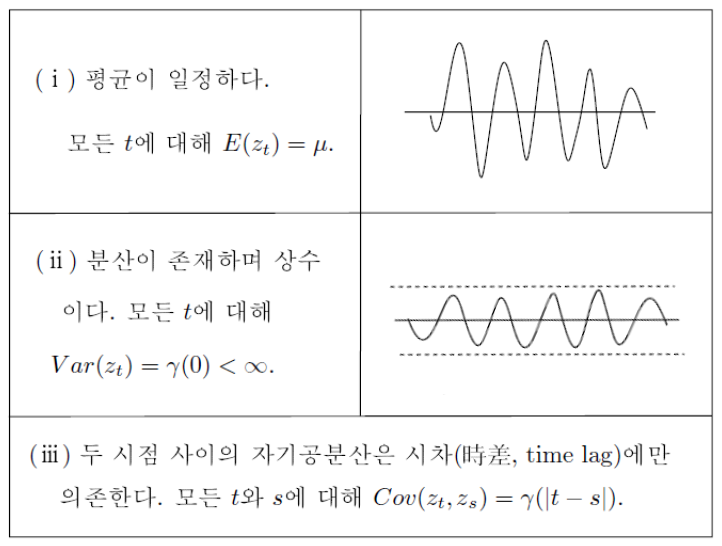
: Non-seasonal difference

: Seasonal difference

: Non-seasonal MA(1)

**(a) 정상시계열의 조건을 기술하고 𝑍𝑡는 정상시계열인지 판단하라.**

* 정상 시계열의 조건 기술



정상성(stationarity)을 나타내는 시계열은 시계열의 특징이 해당 시계열이 관측된 시간에 무관합니다. 추세와 계절성은 서로 다른 시간에 시계열의 값에 영향을 줄 것이기 때문입니다. 반면에, 백색잡음(white noise) 시계열은 정상성을 나타내는 시계열입니다. 일반적으로는, 정상성을 나타내는 시계열은 장기적으로 볼 때 예측할 수 있는 패턴을 나타내지 않을 것입니다. (어떤 주기적인 행동이 있을 수 있더라도) 시간 그래프는 시계열이 일정한 분산을 갖고 대략적으로 평평하게 될 것을 나타낼 것입니다. [ <https://otexts.com/fppkr/stationarity.html> ]

데이터가 정상성을 가진다는 것은 평균과 분산이 안정되어 있어서 분석하기가 쉽다는 의미입니다. 데이터가 정상성을 가지지 않으면, 분석이 어렵기 때문에 정상성을 갖도록 만드는 전처리를 하게 됩니다. 보통 평균이 일정하지 않으면 차분을 취하고, 분산이 일정하지 않으면 변환을 취하면 됩니다. [ <https://freshrimpsushi.tistory.com/907> ]

* 𝑍𝑡는 정상시계열인지 판단하라

: 2차 차분 과 : m = 12 주기 계절성 이 존재하므로, 정상 시계열이 아니다.

**(b) 는 정상시계열인가?**

: 1차 차분과 : m = 12 주기 계절성이 존재하므로, 정상 시계열이 아니다.

**(c) 는 정상시계열인가?**

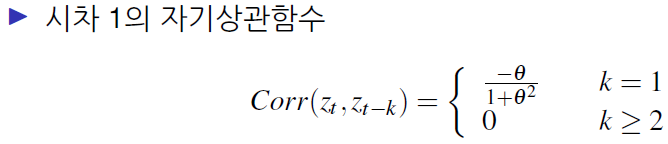
: m = 12 주기 계절성이 존재하므로, 정상 시계열이 아니다.

**(d)** 는 정상시계열인가?

: 2차 차분이 존재하므로 정상 시계열이 아니다.

**(e) 는 정상시계열인가?**

는 Non-seasonal MA(1) 형태 이므로, 정상 시계열이라고 볼 수 있다.

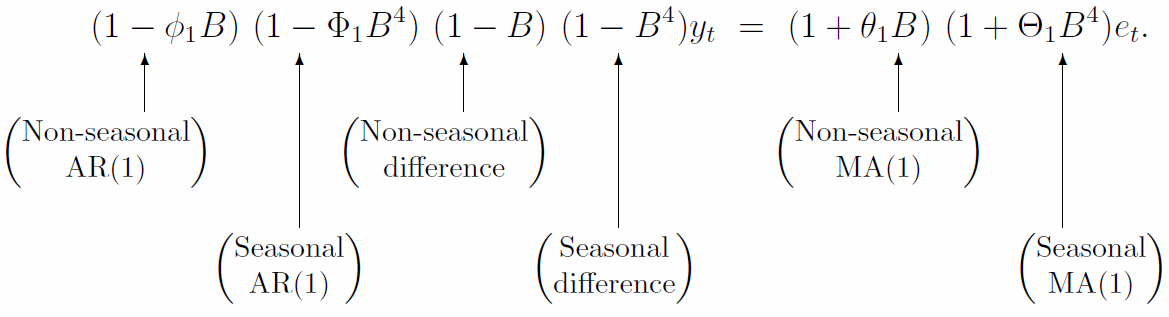


→ MA(1) 과정에서는 시차 1 에서만 공분산을 가진다.

→ MA(1) 과정은 분산이 항상 존재함으로 정상성 조건을 항상 만족한다.

→ 를 설명하는 것은 오직 뿐 이다.

**(f) 시계열 를 의 형식으로 표현하라.**



: Non-seasonal difference

: Seasonal difference

: Non-seasonal MA(1)

**(g) 문제 (e) 의 시계열 를 의 형식으로 표현하라.**

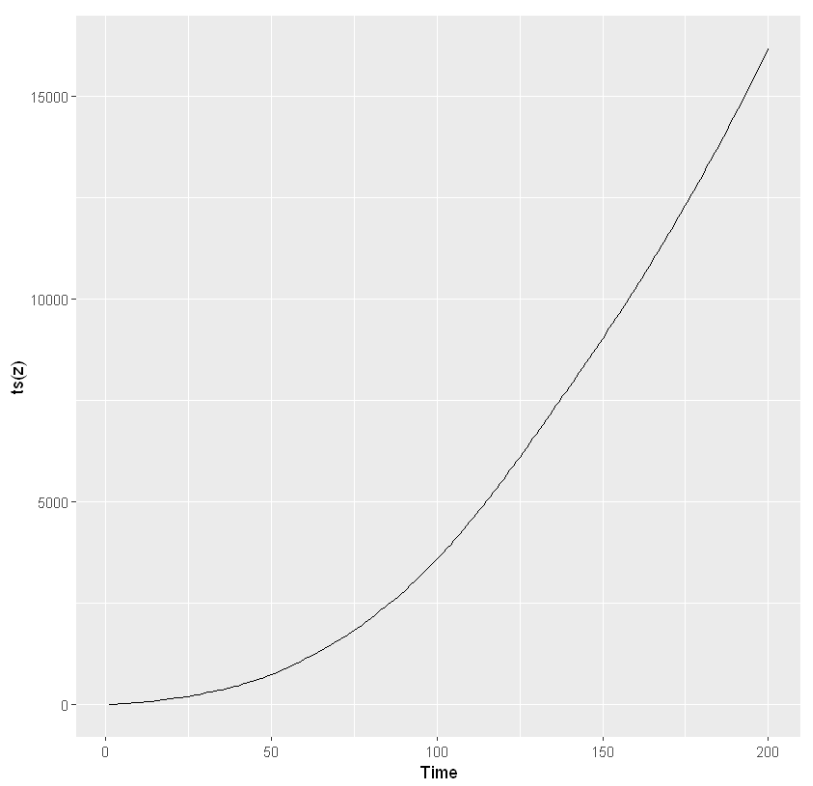
: Non-seasonal MA(1)

**(h) 수식 (1)을 아래와 같이 풀어쓰고자 한다. 계수 (A, B, C, D)를 결정하라.**

**2 [20 점]. R 의 arima.sim 함수는 와 같은 계절형 ARIMA 모형에 대한 모의자료를 생성하지 못한다. 대신 R 을 이용하여 다음과 같이 200 개의 시계열자료를 문제 1 의 모형 (1)로부터 생성하고자 한다. 편의상 초기값을 으로 지정하고 이로부터 200 개의 자료를 생성하여 분석에 사용하고자 한다. 이에 대한 R 샘플코드는 아래와 같다.**

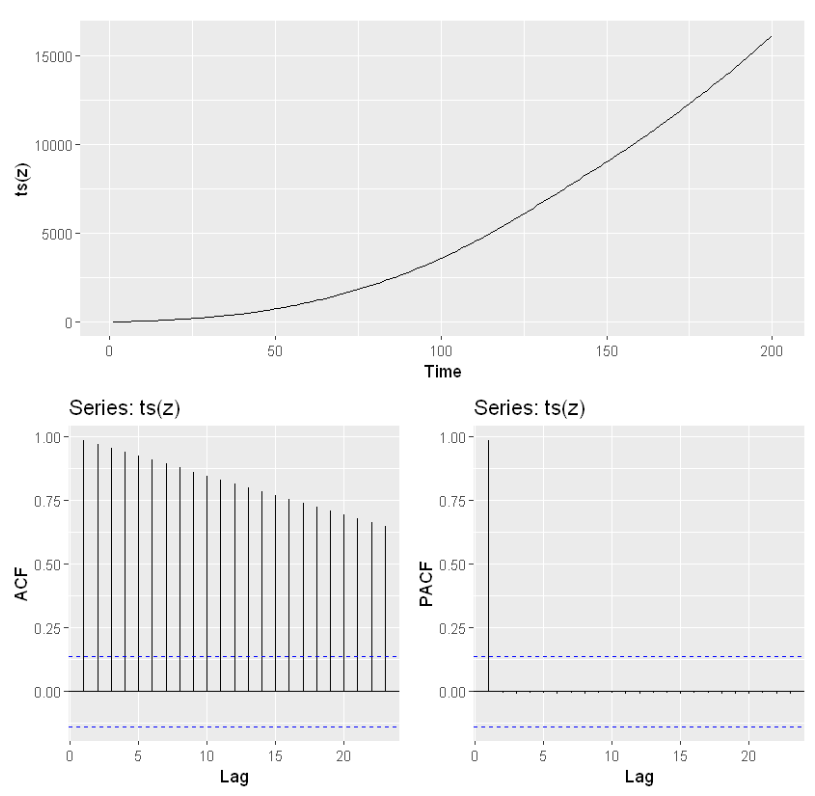
**(a) 문제 1(h)의 결과 (즉, 모형 (2))를 활용하여 샘플코드의 z.temp[t] = (....) 부분을 완성하라. 이를 이용하여 200 개의 모의실험 시계열자료 {𝑍𝑡 }에 대한 시계열그림을 그려라.**

|  |
| --- |
| set.seed(1)  n <- 200 + 14  z.temp <- rep(0,n)  e <- rnorm(n,mean=0,sd=2)  **for** (t **in** 15:n){  z.temp[t] = 2 \* z.temp[t - 1] - z.temp[t - 2] + z.temp[t - 12] -2 \* z.temp[t - 13] + z.temp[t - 14] + e[t] -0.5 \* e[t-1] *## fill out*  }  z <- z.temp[15:n] *## simulated dataset*  autoplot(ts(z)) |



**(b) 모의실험 시계열자료 에 대한 ACF 및 PACF 를 그리고 관측내용을 기술하라.**

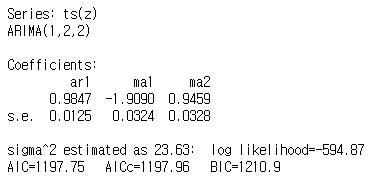
|  |
| --- |
| p0 <- autoplot(ts(z))  p1 <- ggAcf(ts(z))  p2 <- ggPacf(ts(z))  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |



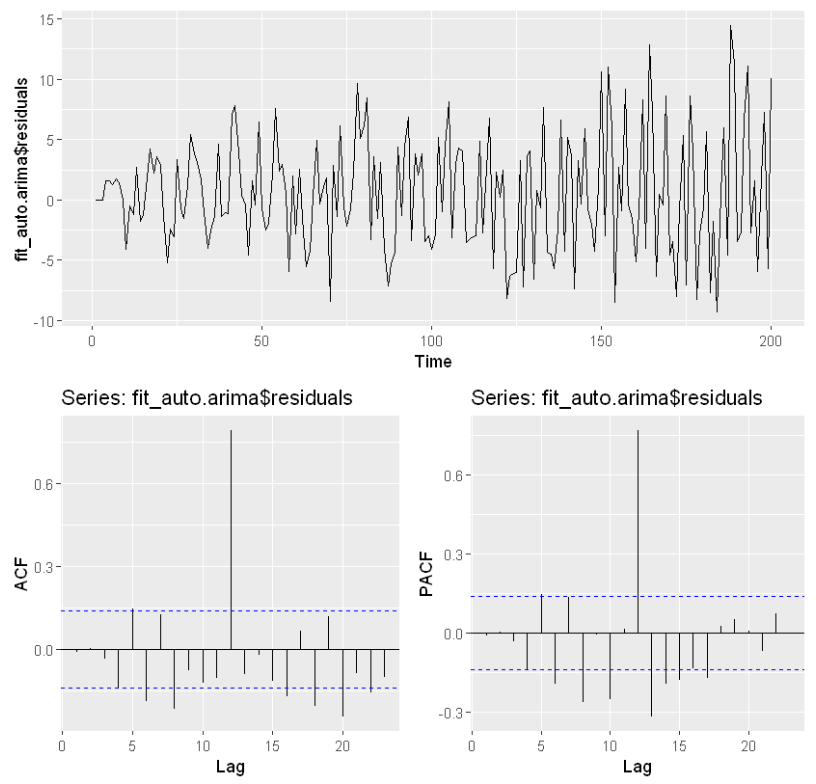
|  |
| --- |
| ACF 는 기준선을 너무 넘어서 천천히 줄고 있고, PACF 는 1만 높고, 기준선을 넘지 않고 있다.  지수적으로 감소하거나, 절단의 형태를 가지면 정상성을 가진 것으로 판단할 수 있는데, 기준선을 너무 크게 넘고 있고, ACF 의 경우 지수적으로 감소한다고 볼 수 없다. |

**(c) 모의실험 자료별 적절한 변화 및 차분 시도한 후 ARIMA 모형을 적합하고 잔차에 대한 ACF 및 PACF 를 그리고 결과를 논하라.**

|  |
| --- |
| fit\_auto.arima <- auto.arima(ts(z))  fit\_auto.arima |

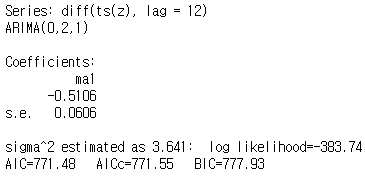


|  |
| --- |
| p0 <- autoplot(fit\_auto.arima$residuals)  p1 <- ggAcf(fit\_auto.arima$residuals)  p2 <- ggPacf(fit\_auto.arima$residuals)  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |

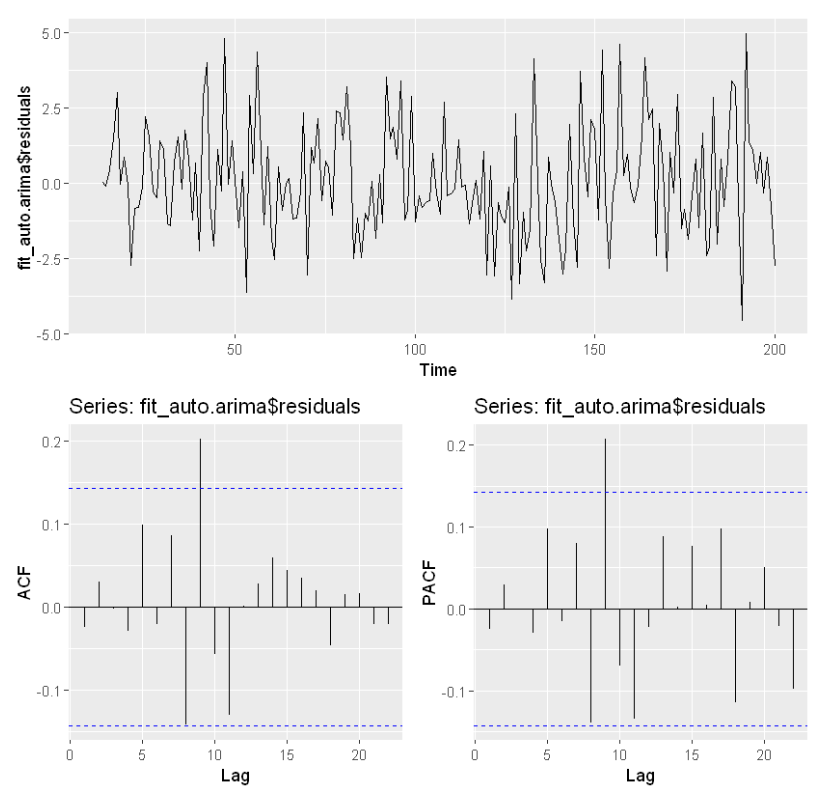


|  |
| --- |
| auto.arima 를 사용 하였을 때, ARIMA(1, 2, 2) 를 추천해 주었고, 실제 ACF 와 PACF 를 보니 Lag 12 에서 기준선을 크게 웃도는 값이 나왔다. 따라서 다음 번에는 12 주기 계절성을 차분해서 auto.arima 를 사용할 것이다. residuals 도 주기와 점점 커지는 추세도 보이는 듯 하다. |

|  |
| --- |
| fit\_auto.arima <- auto.arima(diff(ts(z), lag = 12))  fit\_auto.arima |



|  |
| --- |
| p0 <- autoplot(fit\_auto.arima$residuals)  p1 <- ggAcf(fit\_auto.arima$residuals)  p2 <- ggPacf(fit\_auto.arima$residuals)  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |

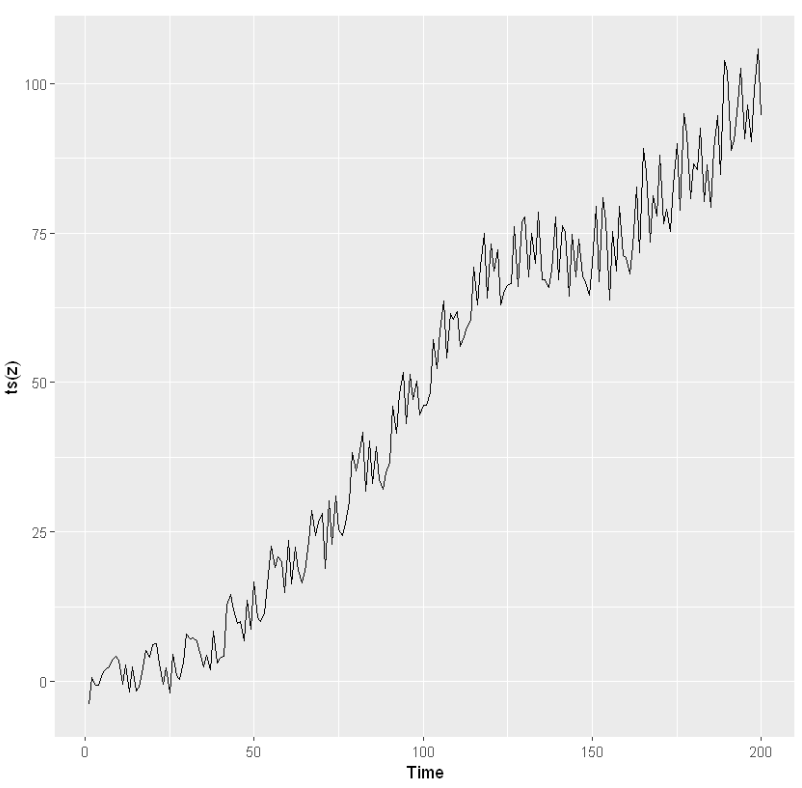


|  |
| --- |
| 계절성 차분 12 로 를 해 주고, auto.arima 를 사용 하였을 때, ARIMA(0, 2, 1) 를 추천해 주었고, 실제 ACF 와 PACF 를 보니 안정적인 값이 나왔다. residuals 도 점점 커지는 추세등 어떤 패턴이 안보이는 듯 하다. 따라서 |

**(d) 다음의 시계열자료 에 대하여 위의 문제 (a)-(c)를 반복하라.**

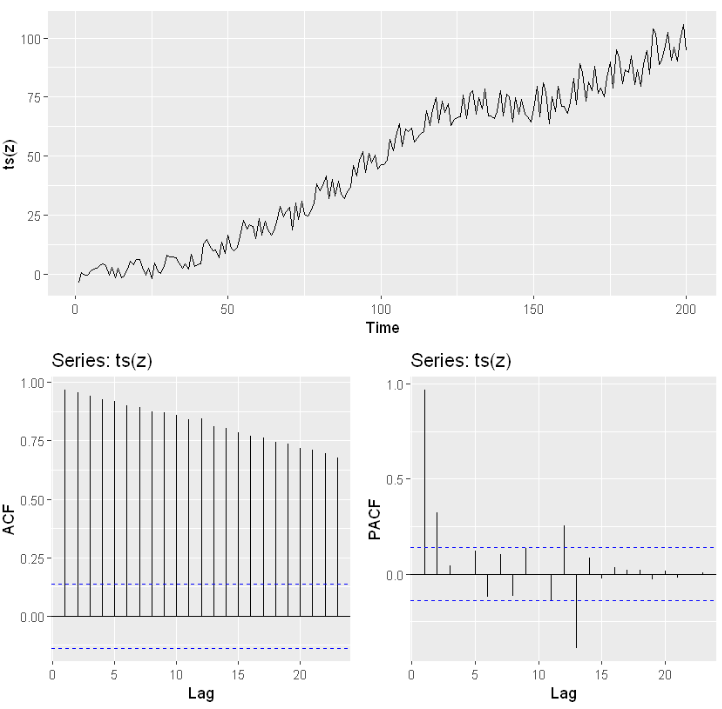
**(d - a) 문제 1(h)의 결과 (즉, 모형 (2))를 활용하여 샘플코드의 z.temp[t] = (....) 부분을 완성하라. 이를 이용하여 200 개의 모의실험 시계열자료 {𝑍𝑡 }에 대한 시계열그림을 그려라.**

|  |
| --- |
| set.seed(1)  n <- 200 + 13  z.temp <- rep(0,n)  e <- rnorm(n,mean=0,sd=2)  **for** (t **in** 14:n){  z.temp[t] = z.temp[t - 1] + z.temp[t - 12] - z.temp[t - 13] + e[t] -0.5 \* e[t-1] *## fill out*  }  z <- z.temp[14:n] *## simulated dataset*  autoplot(ts(z)) |



**(d - b) 모의실험 시계열자료 에 대한 ACF 및 PACF 를 그리고 관측내용을 기술하라.**

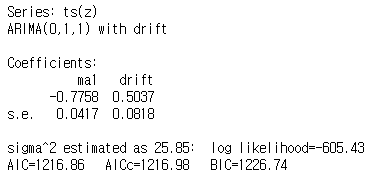
|  |
| --- |
| p0 <- autoplot(ts(z))  p1 <- ggAcf(ts(z))  p2 <- ggPacf(ts(z))  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |



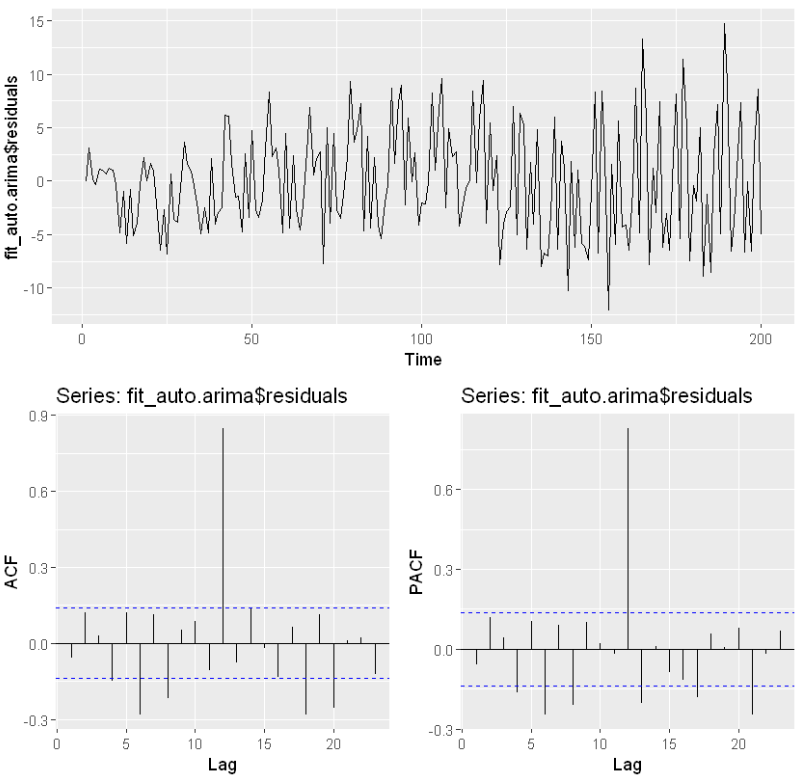
|  |
| --- |
| ACF 는 기준선을 너무 넘어서 천천히 줄고 있고, PACF 기준선을 넘고 있다. 지수적으로 감소하거나, 절단의 형태를 가지면 정상성을 가진 것으로 판단할 수 있는데, 기준선을 너무 크게 넘고 있고, ACF 의 경우 지수적으로 감소한다고 볼 수 없다. |

**(d - c) 모의실험 자료별 적절한 변화 및 차분 시도한 후 ARIMA 모형을 적합하고 잔차에 대한 ACF 및 PACF 를 그리고 결과를 논하라.**

|  |
| --- |
| fit\_auto.arima <- auto.arima(ts(z))  fit\_auto.arima |

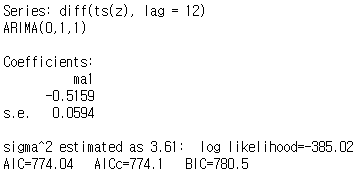


|  |
| --- |
| p0 <- autoplot(fit\_auto.arima$residuals)  p1 <- ggAcf(fit\_auto.arima$residuals)  p2 <- ggPacf(fit\_auto.arima$residuals)  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |

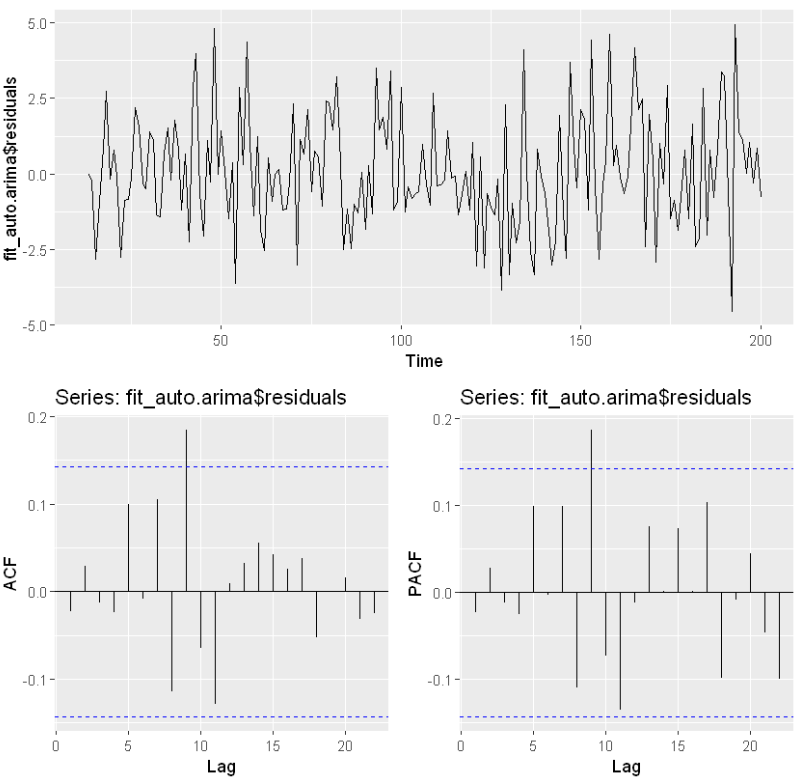


|  |
| --- |
| auto.arima 를 사용 하였을 때, ARIMA(0, 1, 1) 를 추천해 주었고, 실제 ACF 와 PACF 를 보니 Lag 12 에서 기준선을 크게 웃도는 값이 나왔다. 따라서 다음 번에는 12 주기 계절성을 차분해서 auto.arima 를 사용할 것이다. residuals은 추세와 주기가 있어 보인다.. |

|  |
| --- |
| fit\_auto.arima <- auto.arima(diff(ts(z), lag = 12))  fit\_auto.arima |



|  |
| --- |
| p0 <- autoplot(fit\_auto.arima$residuals)  p1 <- ggAcf(fit\_auto.arima$residuals)  p2 <- ggPacf(fit\_auto.arima$residuals)  grid.arrange(p0, p1, p2,  layout\_matrix = rbind(  c(1),  c(2, 3))) |



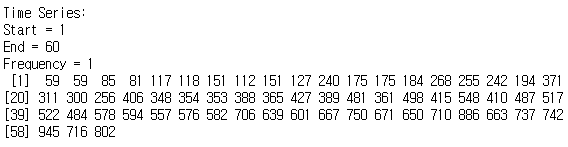
|  |
| --- |
| 계절성 차분 12 로 를 해 주고, auto.arima 를 사용 하였을 때, ARIMA(0, 1, 1) 를 추천해 주었고, 실제 ACF 와 PACF 를 보니 안정적인 값이 나왔다. residuals 도 점점 커지는 추세등 어떤 패턴이 안보이는 듯 하다. 따라서 |

**3 [20 점]. 다음의 시계열자료는 가 를 따를 때 시계열모형**

**으로부터 생성된 모의실험 자료이다.**

|  |
| --- |
| 59 59 85 81 117 118 151 112 151 127 240 175 175 184 268 255 242 194 371 311 300 256 406 348 354 353 388 365 427 389 481 361 498 415 548 410 487 517 522 484 578 594 557 576 582 706 639 601 667 750 671 650 710 886 663 737 742 945 716 802 |

|  |
| --- |
| data <- c(59, 59, 85, 81, 117, 118, 151, 112, 151, 127, 240, 175, 175, 184, 268, 255, 242, 194, 371, 311, 300, 256, 406, 348, 354, 353, 388, 365, 427, 389, 481, 361, 498, 415, 548, 410, 487, 517, 522, 484, 578, 594, 557, 576, 582, 706, 639, 601, 667, 750, 671, 650, 710, 886, 663, 737, 742, 945, 716, 802)  ts\_data <- ts(data)  print(ts\_data) |



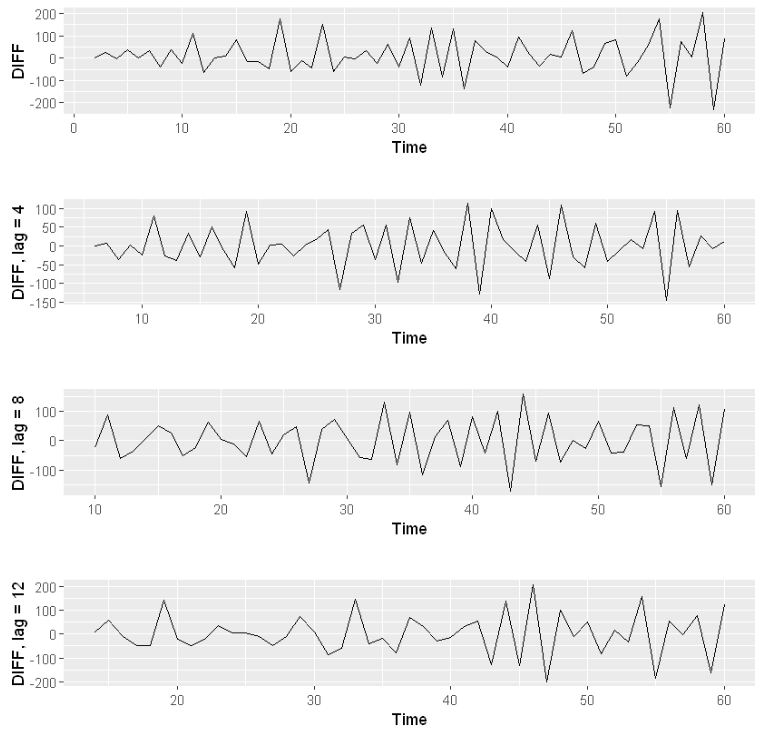
**(a) 의 시계열그림을 그려라. 비정상성이 존재하는가? 존재한다면 비정상성분은 무엇이라 생각하는가?**

|  |
| --- |
| autoplot(ts\_data) + geom\_point() |

|  |  |
| --- | --- |
|  | 1. 우상향을 하는 추세가 있다.  2. 눈으로 볼 때는, 12 의 주기성이 보이지만,  3. 실제 decompose, acf 등으로 분해, 분석 해서 봐야 정확하게 볼 수 있을 것 같다. |

**(b) 1 차 차분된 시계열 의 시계열그림을 그려라. 계절성분이 존재하는가? 존재한다면 계절성분을 제거하기 위하여 계절차분을 하라.**

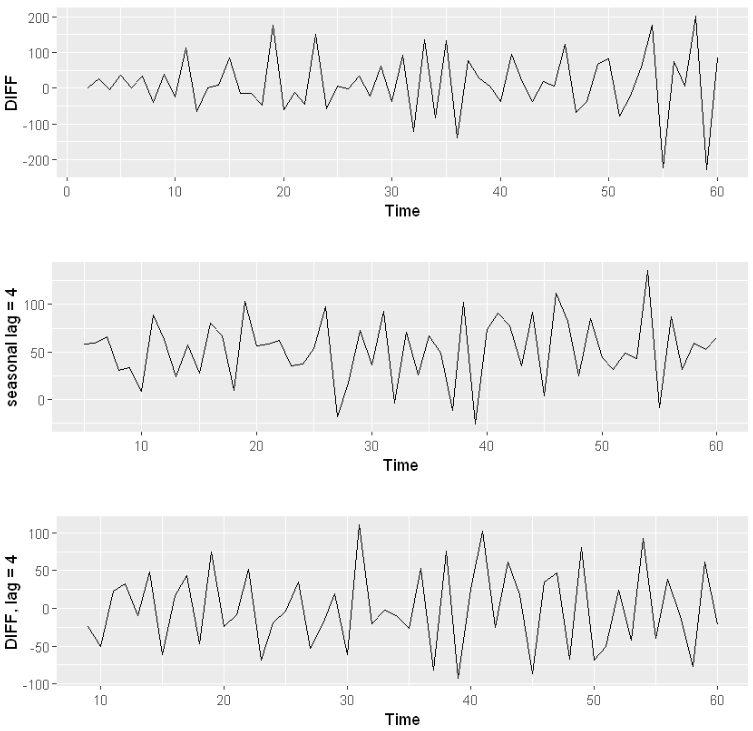
|  |
| --- |
| diff\_ts <- diff(ts\_data)  p1 <- autoplot(diff\_ts) + ylab("DIFF")  diff\_ts4 <- diff(diff\_ts, lag = 4)  p2 <- autoplot(diff\_ts4) + ylab("DIFF, lag = 4")  diff\_ts8 <- diff(diff\_ts, lag = 8)  p3 <- autoplot(diff\_ts8) + ylab("DIFF, lag = 8")  diff\_ts12 <- diff(diff\_ts, lag = 12)  p4 <- autoplot(diff\_ts12) + ylab("DIFF, lag = 12")  grid.arrange(p1, p2, p3, p4,  layout\_matrix = rbind(  c(1),  c(2),  c(3),  c(4))) |



|  |
| --- |
| 1차 차분 했을 때, 우상향의 추세는 없어졌지만, 주기가 12 같은 주기로 변동하고 있어서, 1차차분 데이터를 가지고, 계졀성을 4씩 증가 시켜 차분시켜 보니, 4 / 8 / 12 인 계절 차분을 했을 시에, 확실히 주기의 일정함도 없어져서, 계절성을 찾기가 힘들어 졌다. |

**(c) 계절주기가 4 인 계절차분된 시계열 의 시계열그림을 그려라. 비정상성이 존재하는지 판단하라.**

|  |
| --- |
| diff\_ts <- diff(ts\_data)  p1 <- autoplot(diff\_ts) + ylab("DIFF")  ts4 <- diff(ts\_data, lag = 4)  p2 <- autoplot(ts4) + ylab("seasonal lag = 4")  diff\_ts4 <- diff(diff\_ts, lag = 4)  p3 <- autoplot(diff\_ts4) + ylab("DIFF, lag = 4")  grid.arrange(p1, p2, p3,  layout\_matrix = rbind(  c(1),  c(2),  c(3))) |



|  |
| --- |
| 1차 차분을 먼저 하지 말고, 계절성 주기가 4를 차분부터 하고 그려보았다. 계절 요인만 차분으로 잡아봤는데, 우상향의 추세선도 잡혔다. 평균만 0 에서 50 으로 증가한 것이 보인다. |

**(d) 차분의 순서에 따라 결과가 달라짐을 주목하고 이에 대하여 논하라.**

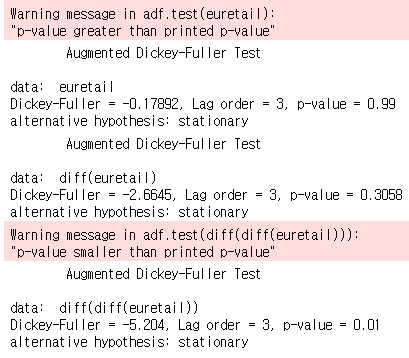
먼저 1차 차분을 했을 때, 우상향 추세선은 잡았지만, 주기성은 보였다. 따라서 그 다음 계절성을 차분하여 주기성을 잡았지만, (c) 에서 보듯이, 궂이 1차 차분을 안해도 계절성 차분만 해서도 우상향 추세선과 주기성을 잡을 수 있었다. 그런데 처음 접하는 데이터를 볼 때, 주기성 보다도 우상향이 더 현상적으로 크게 보여서 차분을 먼저 하기도 하고, ARIMA 의 식을 보더라도 으로 차분을 하고 계절성을 보게 되는 것 같다.

**4 [20 점]. 다음의 방법으로 EURO retail 자료를 R 로 불러들인다**

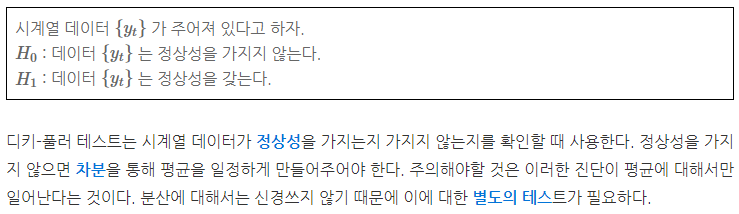
|  |
| --- |
| data(euretail)  str(euretail) |

**(a) euretail 자료에 단위근이 존재하는지 adf.test()를 이용하여 테스트하라.**

|  |
| --- |
| adf.test(euretail)  adf.test(diff(euretail))  adf.test(diff(diff(euretail))) |



Dickey-Fuller Test (디키-풀러 테스트)

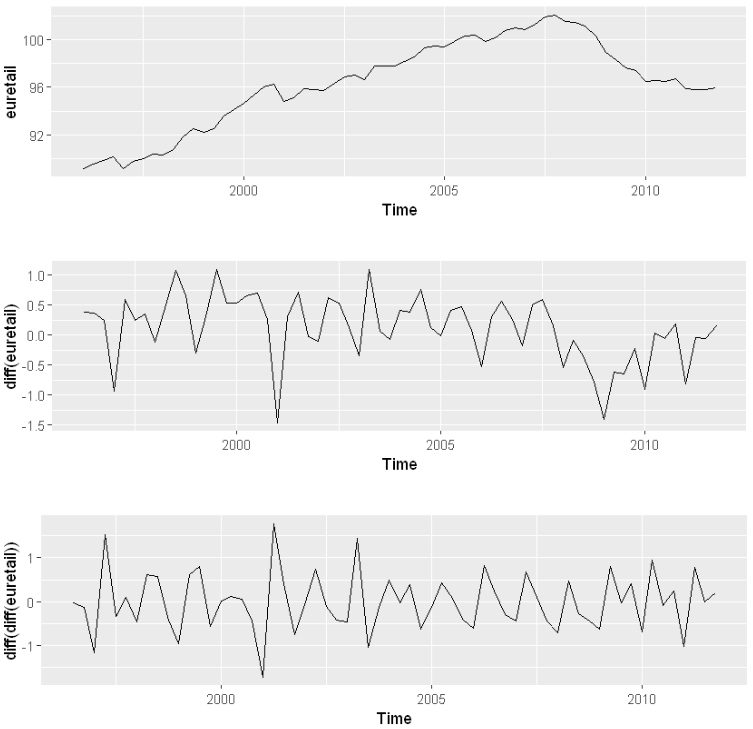


<https://freshrimpsushi.tistory.com/921>

|  |
| --- |
| euretail 의 원 자료는 p-value 가 0.99 로 엄청 높아서 실제 단위근이 존재한다는 귀무가설을 기각하지 못한다. 따라서 단위근이 존재하게 된다. 2차 차분을 했을 때는 p-value 가 0.01 로 떨어져서 를 기각하고, 단위근이 존재하지 않는다는 대립가설을 채택할 수 있다. 하지만, Dickey-Fuller Test (디키-풀러 테스트) 에서 데이터가 정상성을 가진다는 것은 어디까지나 평균이 일정하다는 의미이지, 완전한 정상성 자체를 보장하는 것이 아니다. |

* 시도표를 보자.

|  |
| --- |
| p1 <- autoplot(euretail)  p2 <- autoplot(diff(euretail))  p3 <- autoplot(diff(diff(euretail)))  grid.arrange(p1, p2, p3,  layout\_matrix = rbind(  c(1),  c(2),  c(3))) |

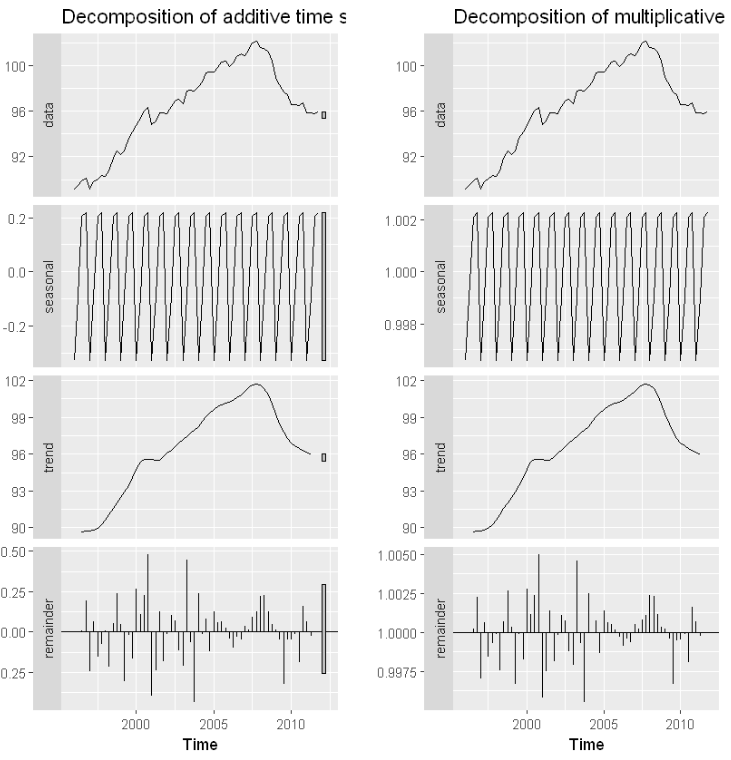


2차 차분한 그림을 보면 주기성도 보이고, 분산도 커짐을 알 수있다. 완전한 정상성 자체를 보장하는 것이 아니니 주의해야 한다.

**(b) euretail 자료를 (i) decompose 방법, (ii) Holt-Winters 방법, (iii) ARIMA 방법으로 적합하라. 각 방법에 따른 적합치 (fitted value) 및 잔차 (random 혹은 residual)에 대한 그림을 그려라. ARIMA 모형의 경우 가장 적절한 모형의 차수를 선택하라.**

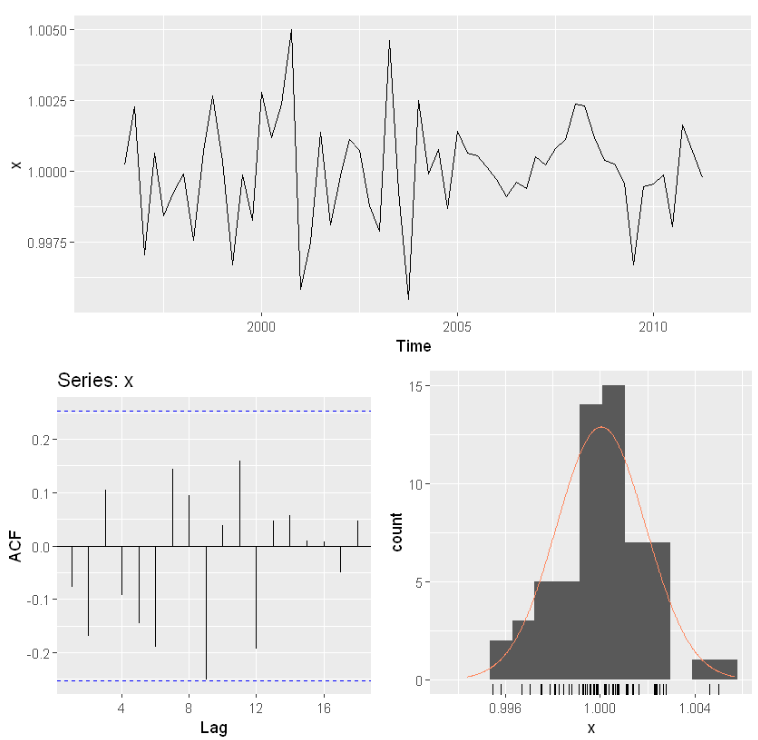
**(i) decompose 방법**

|  |
| --- |
| *# (i) decompose 방법*  fit\_decom\_add <- decompose(euretail, type = 'additive')  fit\_decom\_mul <- decompose(euretail, type = 'multiplicative')  p1 <- autoplot(fit\_decom\_add)  p2 <- autoplot(fit\_decom\_mul)  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1, 2))) |



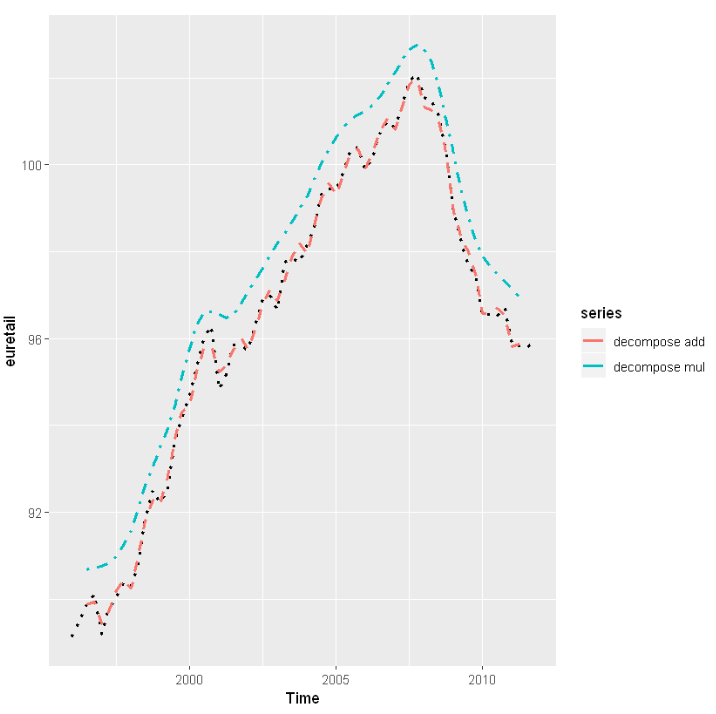
**잔차 (random 혹은 residual)**

|  |
| --- |
| x <- fit\_decom\_mul$random  p1 <- autoplot(x)  p2 <- gghistogram(x, add.normal = **TRUE**)  p3 <- ggAcf(x)  grid.arrange(p1, p2, p3,  layout\_matrix = rbind(  c(1, 1),  c(3, 2))) |



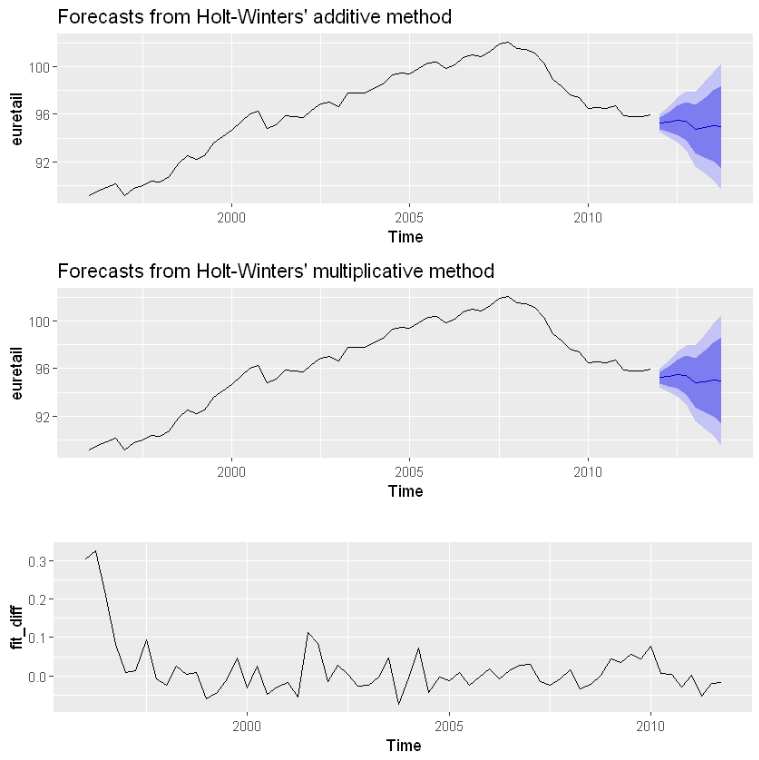
**적합치 (fitted value)**

|  |
| --- |
| autoplot(euretail, linetype = 'dotted', size = 1) +  autolayer(fit\_decom\_add$seasonal + fit\_decom\_add$trend  , series = "decompose add", linetype = 'dashed', size = 1) +  autolayer(fit\_decom\_mul$seasonal + fit\_decom\_mul$trend  , series = "decompose mul", linetype = 'dotdash', size = 1) |



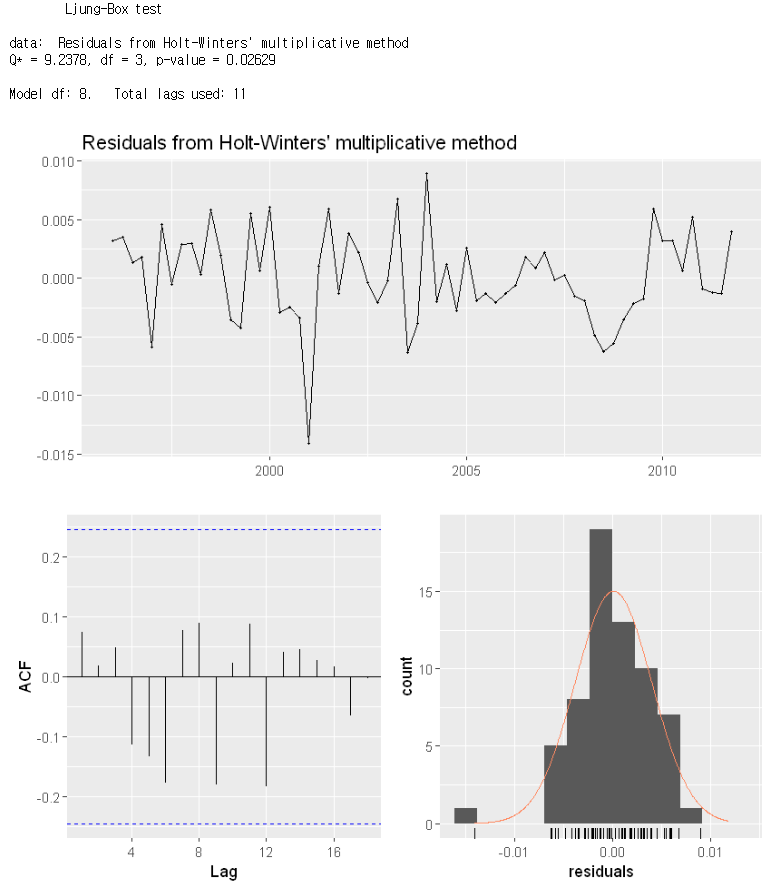
**(ii) Holt-Winters 방법**

|  |
| --- |
| *# (ii) Holt-Winters 방법,*  fit\_hw\_add <- hw(euretail, seasonal = "additive")  p1 <- autoplot(fit\_hw\_add)  fit\_hw\_mul <- hw(euretail, seasonal = "multiplicative")  p2 <- autoplot(fit\_hw\_mul)  fit\_diff <- fit\_hw\_add$fitted - fit\_hw\_mul$fitted  p3 <- autoplot(fit\_diff)  grid.arrange(p1, p2, p3,  layout\_matrix = rbind(  c(1),  c(2),  c(3))) |



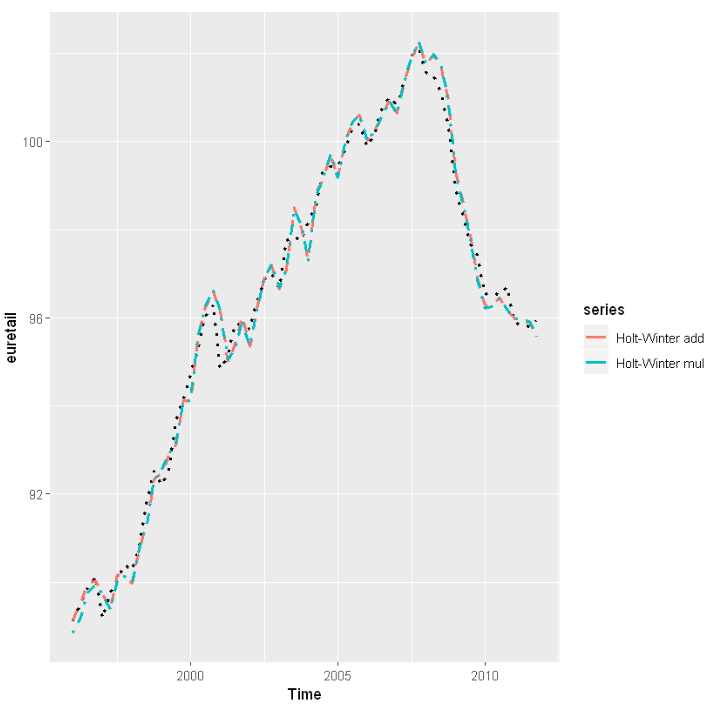
**잔차 (random 혹은 residual)**

|  |
| --- |
| checkresiduals(fit\_hw\_mul) |



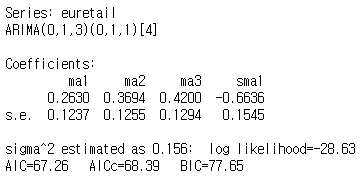
**적합치 (fitted value)**

|  |
| --- |
| autoplot(euretail, linetype = 'dotted', size = 1) +  autolayer(fit\_hw\_add$fitted, series = "Holt-Winter add"  , linetype = 'dashed', size = 1) +  autolayer(fit\_hw\_mul$fitted, series = "Holt-Winter mul"  , linetype = 'dotdash', size = 1) |



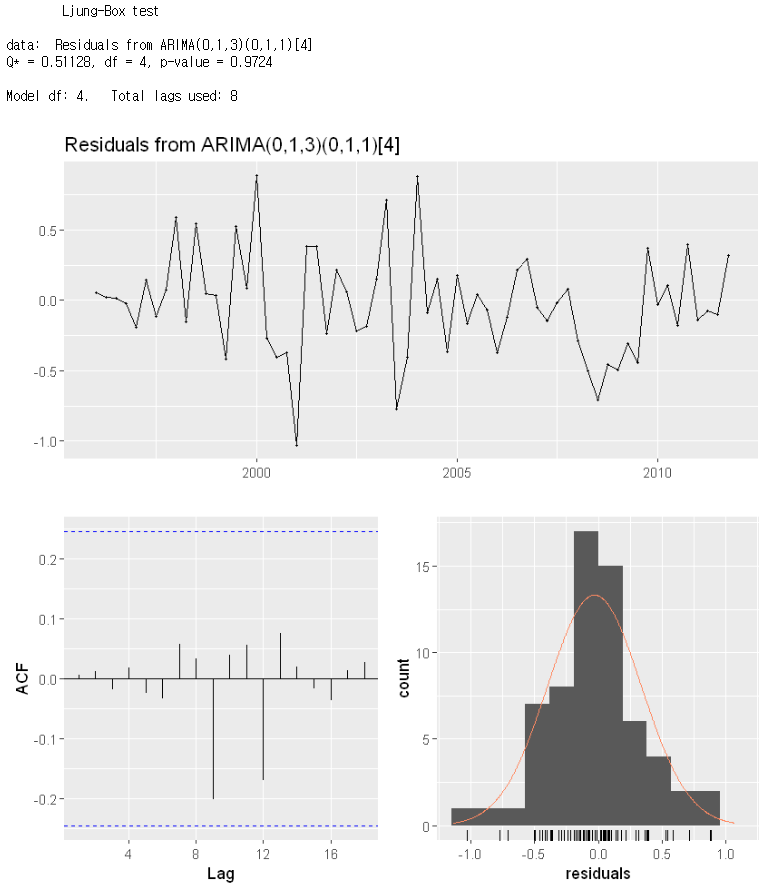
**(iii) ARIMA 방법**

|  |
| --- |
| fit\_auto.arima <- auto.arima(euretail)  fit\_auto.arima |



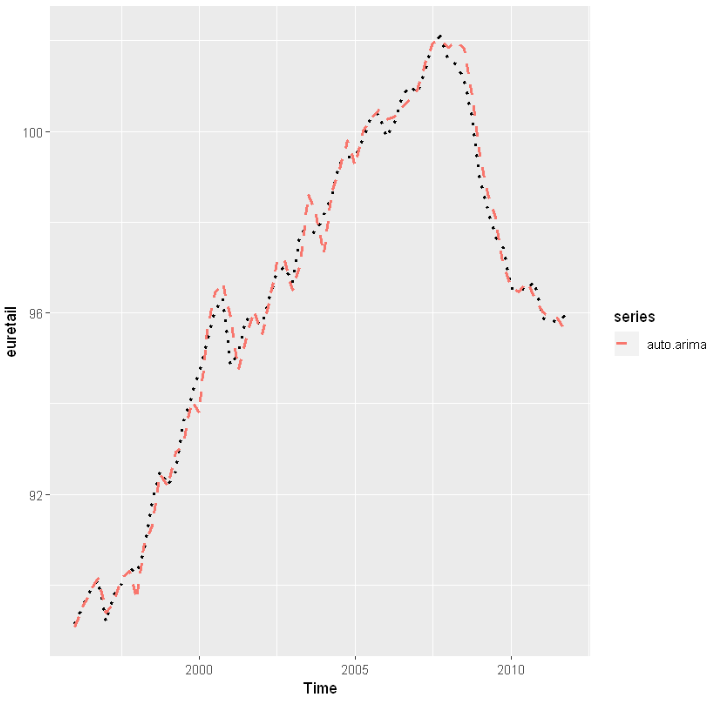
**잔차 (random 혹은 residual)**

|  |
| --- |
| checkresiduals(fit\_auto.arima) |



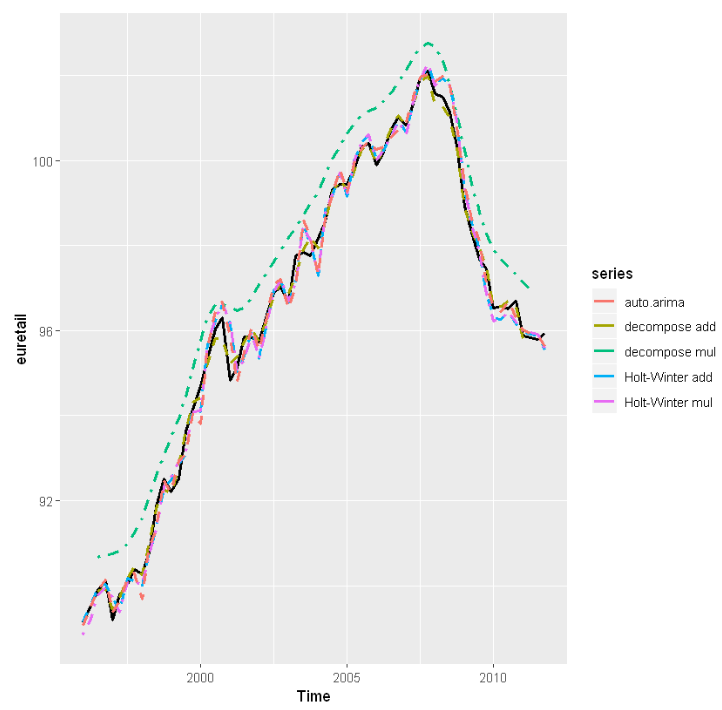
**적합치 (fitted value)**

|  |
| --- |
| autoplot(euretail, linetype = 'dotted', size = 1) +  autolayer(fit\_auto.arima$fitted, series = "auto.arima"  , linetype = 'dashed', size = 1) |



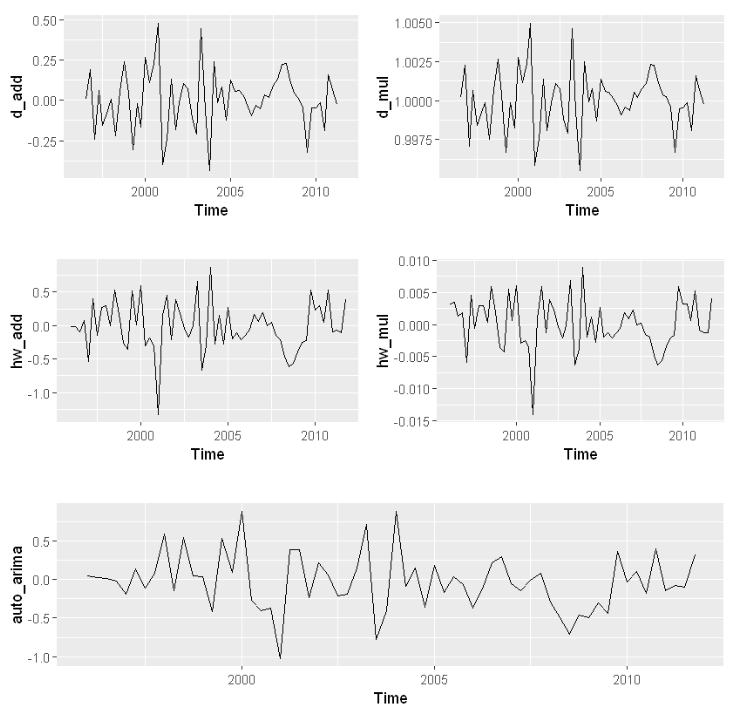
**모든 적합치 (fitted value)**

|  |
| --- |
| autoplot(euretail, size = 1) +  autolayer(fit\_decom\_add$seasonal + fit\_decom\_add$trend  , series = "decompose add", linetype = 'dashed', size = 1) +  autolayer(fit\_decom\_mul$seasonal + fit\_decom\_mul$trend  , series = "decompose mul", linetype = 'dotdash', size = 1) +  autolayer(fit\_hw\_add$fitted, series = "Holt-Winter add"  , linetype = 'dashed', size = 1) +  autolayer(fit\_hw\_mul$fitted, series = "Holt-Winter mul"  , linetype = 'dotdash', size = 1) +  autolayer(fit\_auto.arima$fitted, series = "auto.arima"  , linetype = 'dashed', size = 1) |



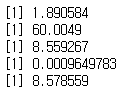
**모든 잔차 (random 혹은 residual)**

|  |
| --- |
| p1 <- autoplot(fit\_decom\_add$random) + ylab('d\_add')  p2 <- autoplot(fit\_decom\_mul$random) + ylab('d\_mul')  p3 <- autoplot(fit\_hw\_add$residuals) + ylab('hw\_add')  p4 <- autoplot(fit\_hw\_mul$residuals) + ylab('hw\_mul')  p5 <- autoplot(fit\_auto.arima$residuals) + ylab('auto\_arima')  grid.arrange(p1, p2, p3, p4, p5,  layout\_matrix = rbind(  c(1, 2),  c(3, 4),  c(5))) |

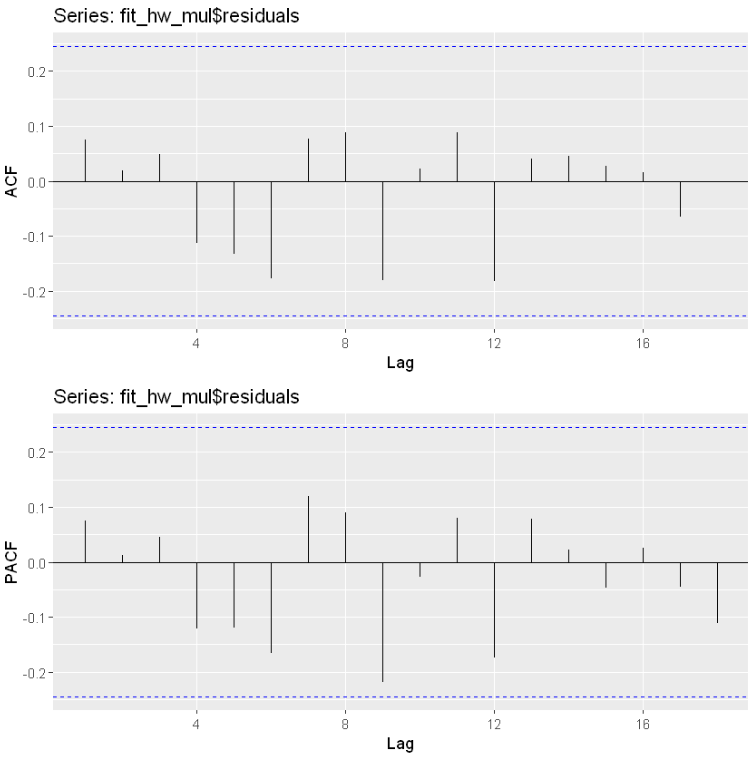


(c) 잔차들의 제곱의 합, 즉 sum(잔차^2)을 계산하고 어떤 모형이 주어진 자료를 가장 잘 설명하는 모형을 선택하고 이 모형의 잔차에 대한 ACF 및 PACF 를 그리고 문제가 있는지 기술하라.

|  |
| --- |
| print(sum(as.numeric((fit\_decom\_add$random)^2), na.rm = **TRUE**)) print(sum(as.numeric((fit\_decom\_mul$random)^2), na.rm = **TRUE**)) print(sum(as.numeric((fit\_hw\_add$residual)^2))) print(sum(as.numeric((fit\_hw\_mul$residual)^2))) print(sum(as.numeric((fit\_auto.arima$residuals)^2))) |



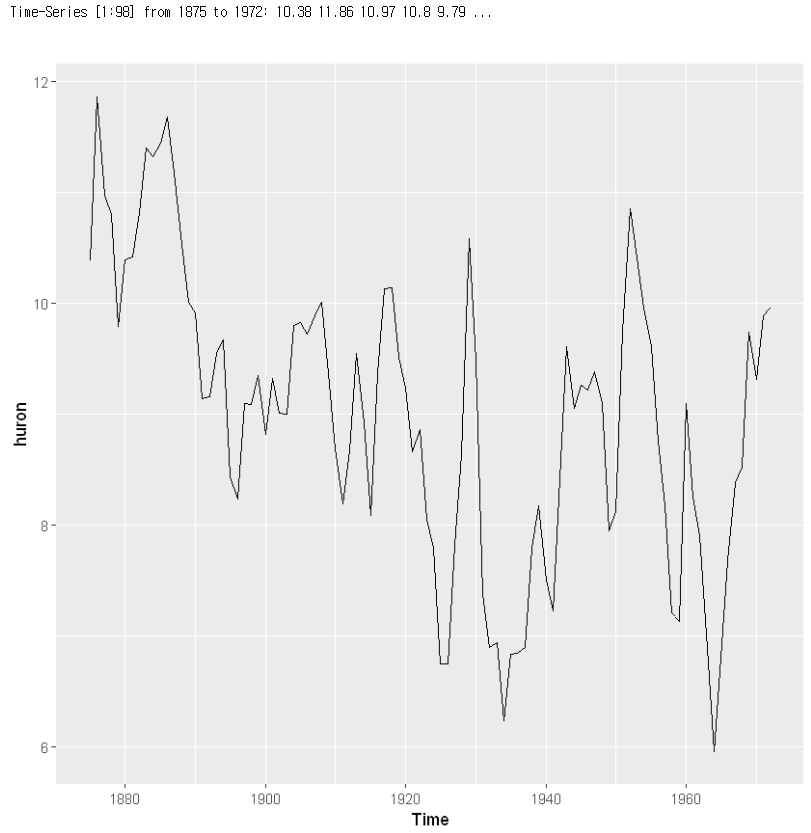
|  |
| --- |
| p1 <- ggAcf(fit\_hw\_mul$residuals, type = c("correlation"))  p2 <- ggPacf(fit\_hw\_mul$residuals)  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1),  c(2))) |



|  |
| --- |
| 각 모형의 잔차 제곱의 합을 했을 때, Holt-Winters 방법의 seasonal 이 multiplicative 로 했을 때 가장 작았다. 따라서 이 모형을 선택하고, 이 모형의 잔차에 대한 ACF, PACF 를 볼 때, 또한 유의 수준 안에서 나와서 문제가 없다. |

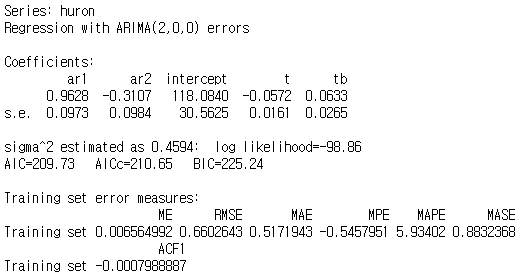
**5 [20 점]. 교재 Forecasting: Principles and Practice, 섹션 9.7 (https://otexts.com/fppkr/dynamicexercises.html)의 2 번 연습문제에 대해 답하라.**

|  |
| --- |
| *# 이 연습문제에서는 1875년부터 1972년까지 휴론(Huron) 호수의 수위를 기록한 데이터 모음 huron을 사용합니다.*  str(huron)  autoplot(huron) |

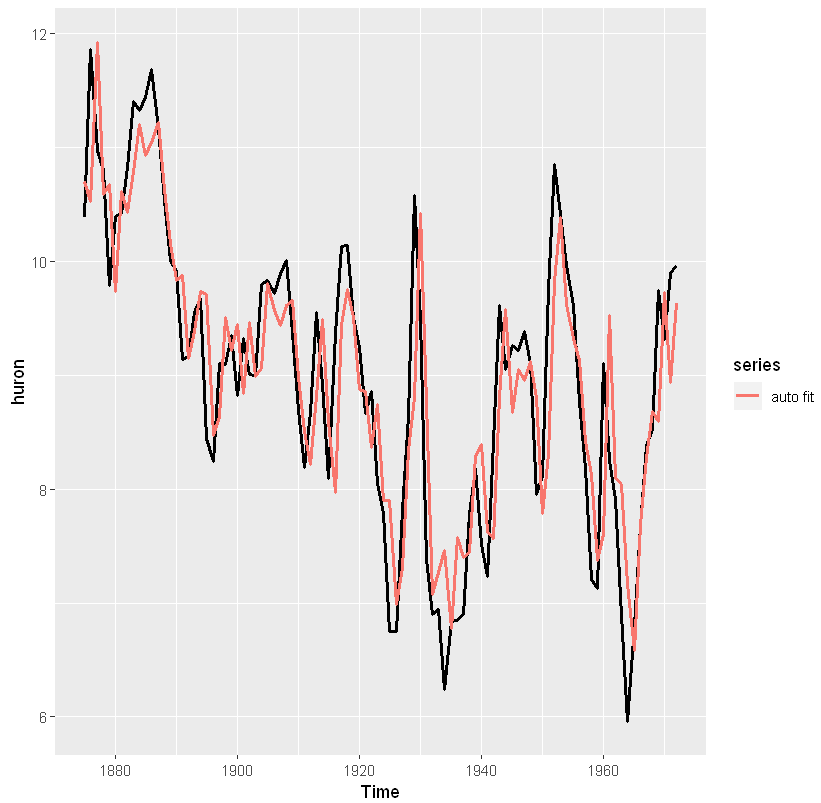


**a. 1920년에 매듭(knot)과 ARMA 오차 구조가 있는 단계적(stepwise) 선형 추세 모델로 휴론 호수 데이터를 맞춰봅시다.**

|  |
| --- |
| *#https://otexts.com/fppkr/nonlinear-regression.html ( knot )*  t <- time(huron)  t.break <- 1920  tb <- ts(pmax(0, t - t.break), start = t[1])  newdata <- cbind(t = t, tb = tb)  *#https://www.rdocumentation.org/packages/forecast/versions/8.10/topics/auto.arima*  *# xreg => Optionally, a numerical vector or matrix of external regressors, which must have the same number of rows as y.*  *# (It should not be a data frame.)*  *# stepwise => If TRUE, will do stepwise selection (faster).*  *# Otherwise, it searches over all models. Non-stepwise selection can be very slow, especially for seasonal models.*  huron\_knot.auto <- auto.arima(huron, xreg = newdata, stepwise = **TRUE**)  summary(huron\_knot.auto) |



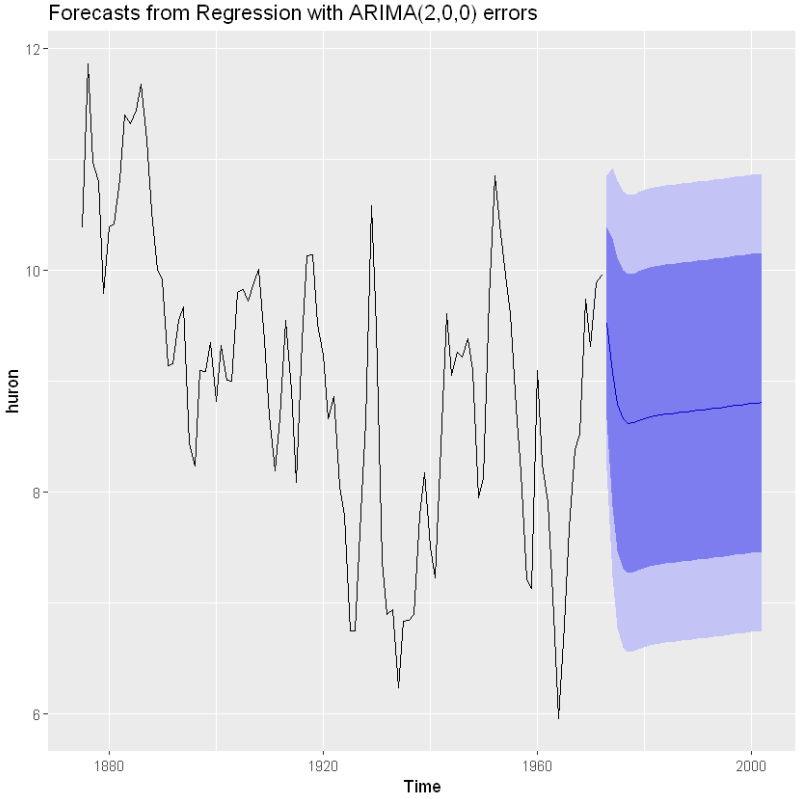
|  |
| --- |
| autoplot(huron, size = 1) +  autolayer(huron\_knot.auto$fitted, series = 'auto fit', size = 1) |



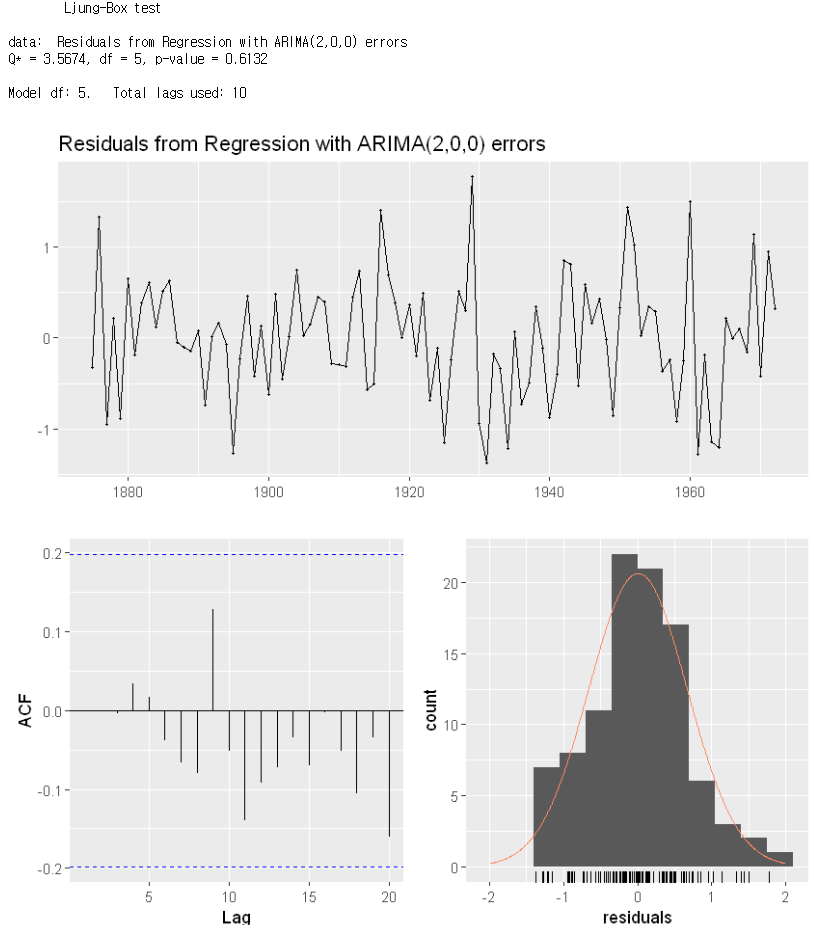
**b. 다음 30년의 수위를 예측해봅시다.**

|  |
| --- |
| *#https://www.rdocumentation.org/packages/forecast/versions/8.10/topics/forecast*  *# https://otexts.com/fppkr/nonlinear-regression.html*  h <- 30  t.new <- t[length(t)] + seq(h)  tb.new <- tb[length(tb)] + seq(h)  newdata <- cbind(t = t.new, tb = tb.new)  *# https://otexts.com/fppkr/dynamic-regression-forecasting.html*  forecast\_huron.auto <- forecast( huron\_knot.auto, xreg = newdata, h = 30 ) |

|  |
| --- |
| autoplot(forecast\_huron.auto) |

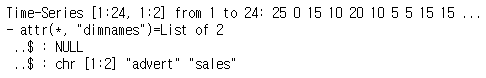


|  |
| --- |
| checkresiduals(forecast\_huron.auto) |



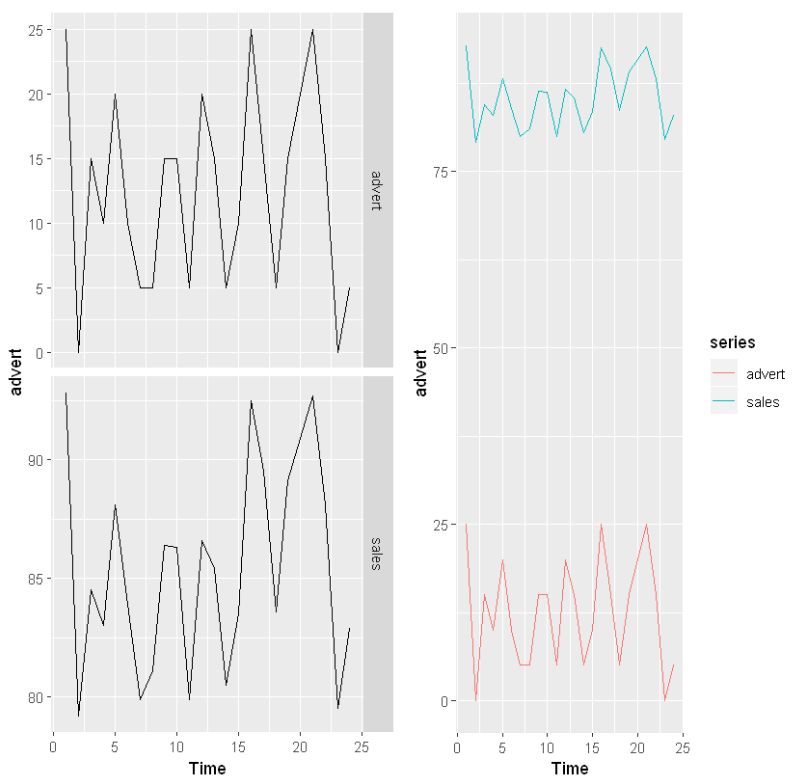
**6 [20 점]. 교재 Forecasting: Principles and Practice, 섹션 9.7 (https://otexts.com/fppkr/dynamicexercises.html)의 1 번 연습문제에 대해 답하라**

|  |
| --- |
| *# 1. 자동차 부품 회사의 월별 판매량과 광고 데이터를 살펴봅시다(데이터 모음 advert)*  str(advert) |



**a. autoplot을 이용하여 데이터를 그래프로 나타내봅시다.facets=TRUE로 두는것이 왜유용합니까?**

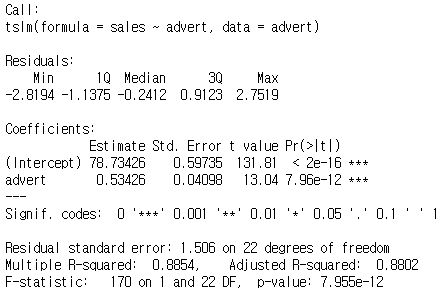
|  |
| --- |
| p1 <- autoplot(advert, facets = **TRUE**)  p2 <- autoplot(advert, facets = **FALSE**)  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1, 2))) |



|  |
| --- |
| advert 단위가 달라서, 둘의 거리감으로 비교 및 시계열의 어떤 추세와 주기를 알기 힘드나, 각각 보게 되면, 자세히 관찰 할 수 있다. |

**b. tslm() 함수를 이용하여 표준 회귀 모델 을 맞춰봅시다. 여기에서 는 판매량이고, 는 광고 데이터입니다.**

|  |
| --- |
| *# https://otexts.com/fppkr/regression-intro.html*  fit\_tslm <- tslm(sales ~ advert, data = advert)  summary(fit\_tslm) |

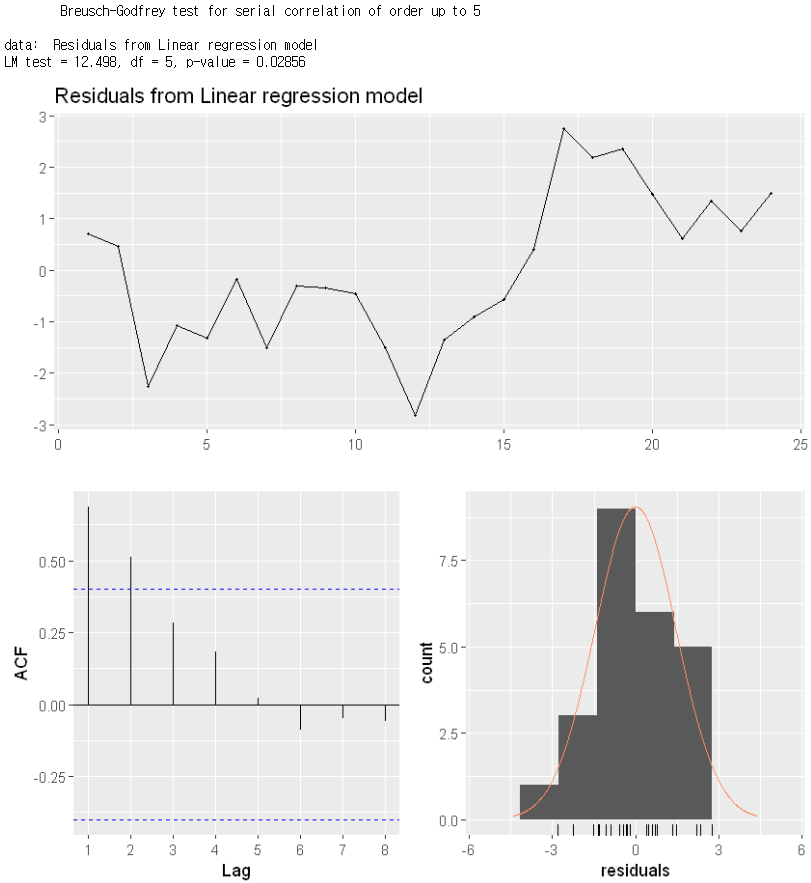


|  |
| --- |
| int(head(advert))  advert %>%  as.data.frame %>%  ggplot(aes(x = advert, y = sales)) +  geom\_point() +  geom\_smooth(method="lm") |

|  |  |
| --- | --- |
|  |  |

**c. 잔차에 유의미한 자기상관이 있다는 것을 증명해봅시다.**

|  |
| --- |
| checkresiduals(fit\_tslm) |



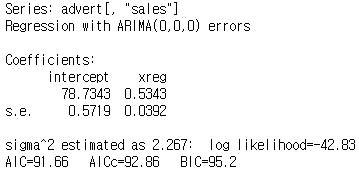
|  |
| --- |
| Breusch-Godfrey test 의 귀무가설  이고 유의확률이 0.02856 으로 0.05 보다 작으므로, 유의미한 autocorrelation (serial correlation) 이 있다고 판단한다. 그림 ACF 에서도 Lag 2 까지 기준치 이상을 볼수 있다. |

|  |
| --- |
| 자기상관함수란 [**시계열 데이터**](https://freshrimpsushi.tistory.com/900)의 자기상관성을 파악하기 위한 함수로, 같은 변수라도 어떤 시차를 가지고 스스로와 비슷한지에 관심을 둔다. 서로 다른 변수의 상관관계에 관심을 가지는 [**회귀분석**](https://freshrimpsushi.tistory.com/548)의 아이디어와는 달리, 그 스스로가 시차 kk 를 두고 YtYt 와 Yt−kYt−k 로 나뉘어 두 변수와 같이 취급된다. |

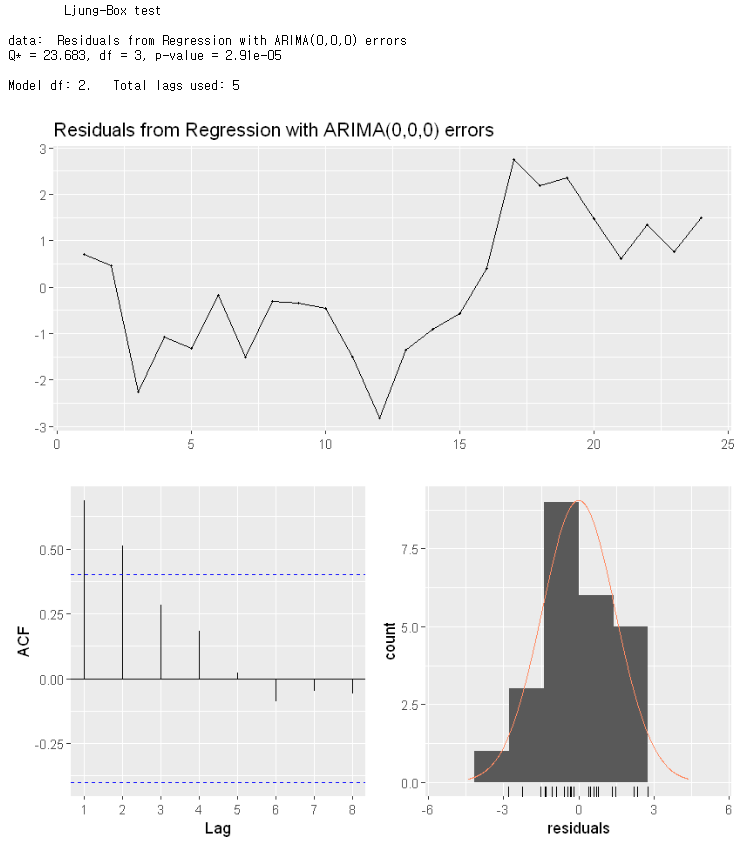
<https://freshrimpsushi.tistory.com/1209>

**d. tslm() 함수를 사용하는 대신에 아래의 함수를 사용하면 어떤 차이가 있습니까?**

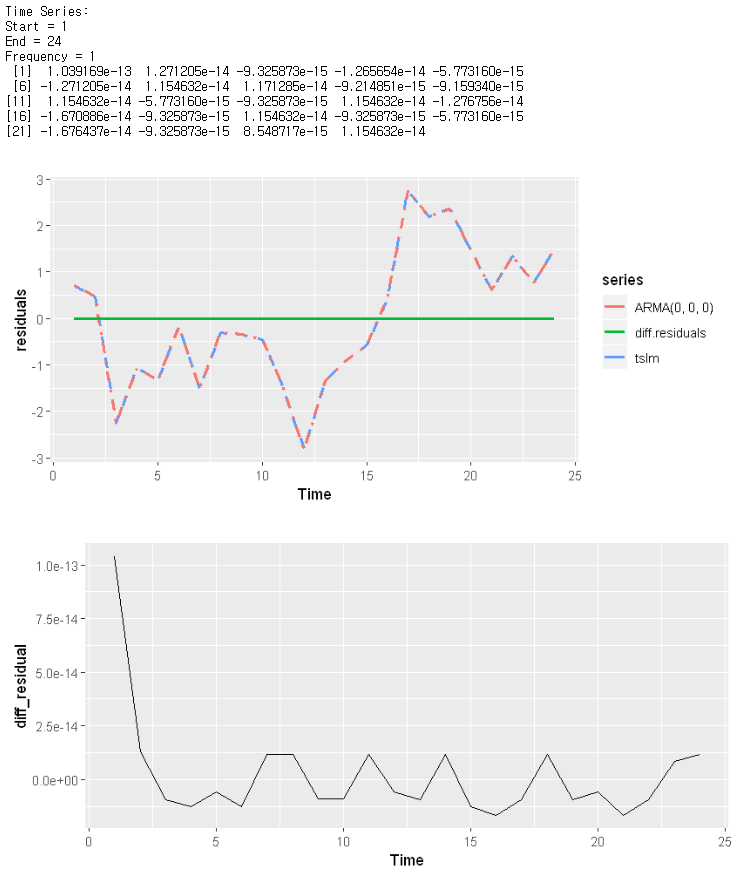
|  |
| --- |
| fit\_Arima <- Arima(advert[,'sales'], xreg = advert[,'advert'], order = c(0, 0, 0))  fit\_Arima |



|  |
| --- |
| checkresiduals(fit\_Arima) |



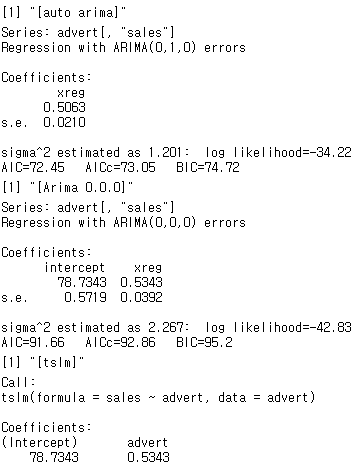
|  |
| --- |
| diff\_residual <- fit\_tslm$residuals - fit\_Arima$residuals  print(diff\_residual)  p1 <- autoplot(fit\_tslm$residuals, series = "tslm", linetype = 'dashed', size = 1) +  autolayer(fit\_Arima$residuals, series = "ARMA(0, 0, 0)", linetype = 'dotdash', size = 1) +  autolayer(diff\_residual, series = "diff.residuals", linetype = 'solid', size = 1) +  ylab("residuals")  p2 <- autoplot(diff\_residual)  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1),  c(2))) |



|  |
| --- |
| tslm 과 거의 같다고 봐도 무방하다. 아마 으로 고정 시켜서 더 그런 결과가 나왔을 가능성이 있다. residuals 의 비교를 보더라고 거의 차이가 없다. coefficient 들도 같다. order = c(0, 0, 0) 이 아닌 다른 것을 사용하던가, auto.arima 를 사용하면 달라질 것이라고 생각한다. |

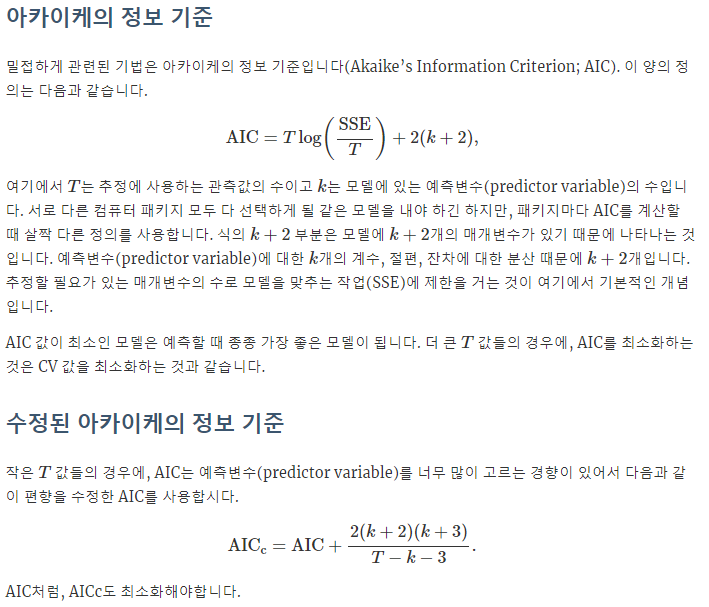
**e. auto.arima()를 이용하여 모델을 다시 맞춰봅시다. 오차 모델이 추정된 매개변수에 얼마나 많은 차이를 만들어냅니까? 오차에 대해 어떤 ARIMA 모델을 골라야합니까?**

|  |
| --- |
| fit\_auto.arima <- auto.arima(advert[, "sales"], xreg = advert[, "advert"])  print('[auto arima]')  fit\_auto.arima  print('[Arima 0.0.0]')  fit\_Arima  print('[tslm]')  fit\_tslm |



|  |
| --- |
| AIC, AICc 를 볼때, 과 tslm 은 같다고 보고, AIC = 91.66, AICc = 92.86 과 auto.arima 를 사용하면 AIC = 72.45, AICc = 73.05 로 많이 줄었다. 따라서 auto.arima 가 선택해준 을 모델을 골라야 한다. |

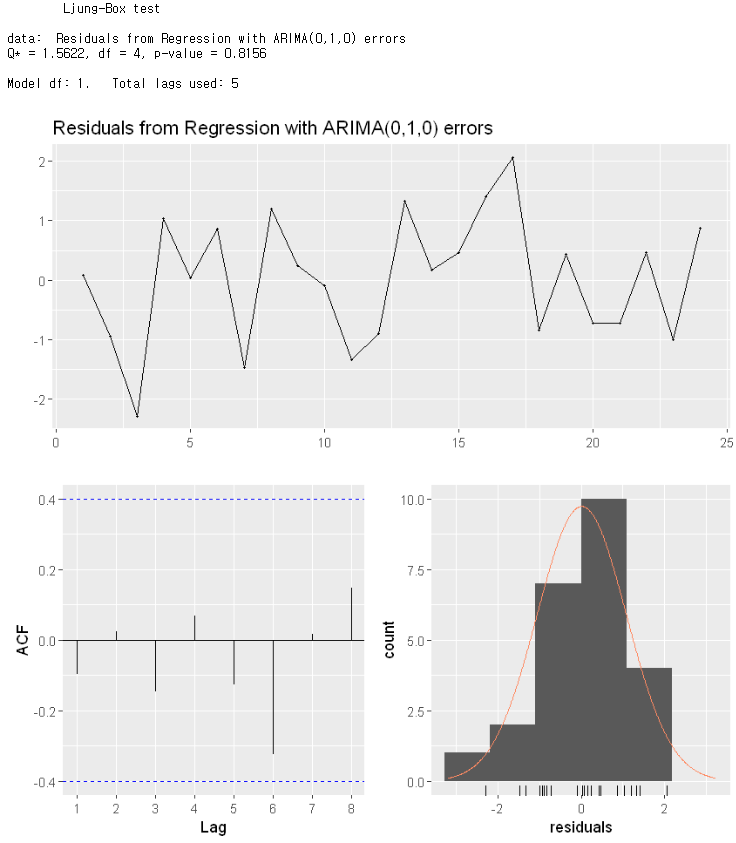
AIC 가 SSE (Sum of Squares Error) 를 사용하므로, 오차 와 관련 매개 변수로 생각하였다.



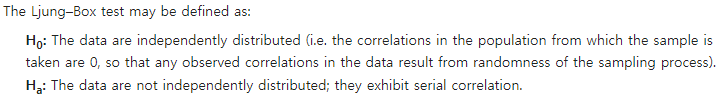
<https://otexts.com/fppkr/selecting-predictors.html>

**f. 맞춘 모델의 잔차를 확인해봅시다.**

|  |
| --- |
| checkresiduals(fit\_auto.arima) |



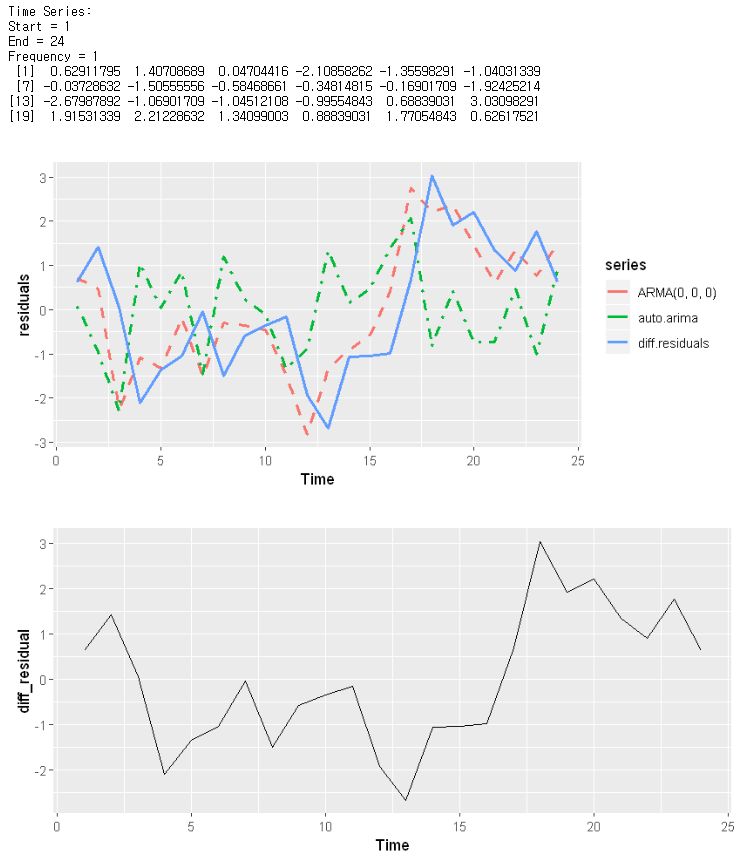
|  |
| --- |
| Ljung-Box test 의 p-value 가 0.8156 으로 귀무가설을 기각하지 못하므로 데이터가 상관 관계가 없이 독립적 분포를 따르고 있어서 자기 상관관계 (serial correlation) 이 없다. 잔차도 우상향 하는 느낌이 없어졌고, ACF 도 안정적이다. |



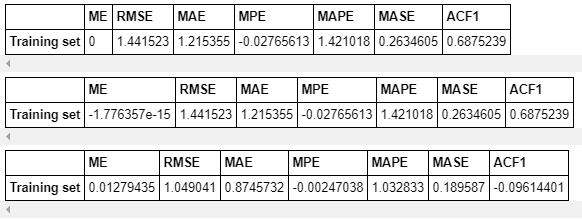
<https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test>

**잔차의 차이를 한번 보자.**

|  |
| --- |
| *# http://www.sthda.com/english/wiki/ggplot2-line-types-how-to-change-line-types-of-a-graph-in-r-software*  diff\_residual <- fit\_Arima$residuals - fit\_auto.arima$residuals  print(diff\_residual)  p1 <- autoplot(fit\_Arima$residuals, series = "ARMA(0, 0, 0)", linetype = 'dashed', size = 1) +  autolayer(fit\_auto.arima$residuals, series = "auto.arima", linetype = 'dotdash', size = 1) +  autolayer(diff\_residual, series = "diff.residuals", linetype = 'solid', size = 1) + ylab("residuals")  p2 <- autoplot(diff\_residual)  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1),  c(2))) |



|  |
| --- |
| accuracy(fit\_tslm)  accuracy(fit\_Arima)  accuracy(fit\_auto.arima) |



각 수치들이 대체적으로 auto.arima 가 좋다.

**g.다음 6달에 대해 광고비용이 정확히 1달에 10 단위라고 가정하여, 다음 6달에 대한 판매량 예측값을 예측구간과 함께 얻어봅시다.**

|  |
| --- |
| *# https://otexts.com/fppkr/dynamic-regression-forecasting.html*  *# rep(10, 6) : 1달에 10 단위, 6 달의 예측값을 계산*  p1 <- autoplot(forecast(fit\_Arima, xreg = rep(10, 6))) + xlab("월")  p2 <- autoplot(forecast(fit\_auto.arima, xreg = rep(10, 6))) + xlab("월")  grid.arrange(p1, p2,  layout\_matrix = rbind(  c(1),  c(2))) |

