Timeseires_HW3

김민국

2020-04-24

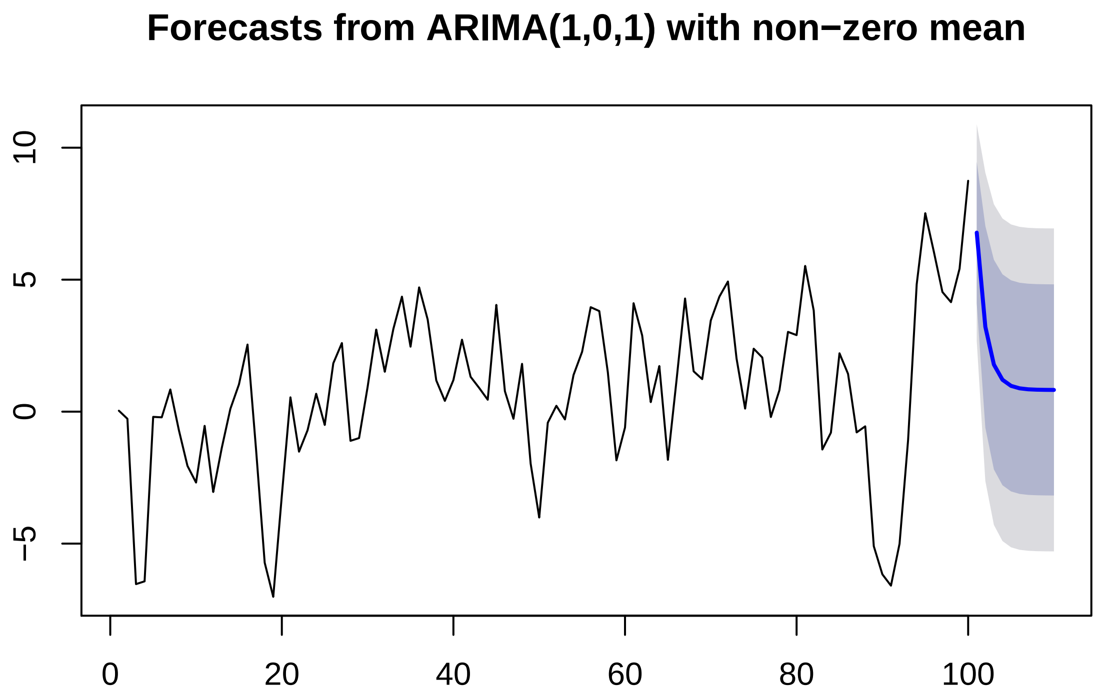
```r
#### 6번

set.seed(200423)

data_6 <- arima.sim(model = list(ar = c(0.5), ma = c(2)), n = 100)
fit_6 <- arima(data_6, order = c(1,0,1), include.mean = T)
fit_6$coef
```

```
##       ar1       ma1 intercept
## 0.4015951 0.6098252 0.8211402
```

```r
pre_6 <- forecast(fit_6, h=10)
pre_6
```

```
##     Point Forecast      Lo 80    Hi 80     Lo 95     Hi 95
## 101      6.7819526  4.0969158 9.466989  2.675543 10.888362
## 102      3.2149733 -0.6039864 7.033933 -2.625621  9.055568
## 103      1.7824919 -2.1891435 5.754127 -4.291600  7.856584
## 104      1.2072143 -2.7884981 5.202927 -4.903700  7.318129
## 105      0.9761857 -3.0233963 4.975768 -5.140647  7.093018
## 106      0.8834057 -3.1168000 4.883611 -5.234381  7.001192
## 107      0.8461458 -3.1541605 4.846452 -5.271795  6.964086
## 108      0.8311823 -3.1691402 4.831505 -5.286783  6.949148
## 109      0.8251731 -3.1751520 4.825498 -5.292796  6.943142
## 110      0.8227598 -3.1775657 4.823085 -5.295210  6.940730
```

```r
plot(pre_6)
```




```r
#### 7번

set.seed(200423)

###### Error term ~ normal

data_n_7 <- arima.sim(model = list(ar = c(0.7)), n = 100)
mle_n_7 <- Arima(data_n_7, order = c(1,0,0), method = "ML")
mle_n_7
```

```
## Series: data_n_7
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.6984  0.6076
## s.e.  0.0708  0.3297
##
## sigma^2 estimated as 1.056:  log likelihood=-143.92
## AIC=293.84   AICc=294.09   BIC=301.66
```

```r
mle_n_7$coef
```

```
##       ar1 intercept 
## 0.6983875 0.6075945
```

```r
mle_n_7$var.coef
```

```
##                    ar1    intercept
## ar1       5.006080e-03 7.319763e-05
## intercept 7.319763e-05 1.086808e-01
```

```r
lse_n_7 <- Arima(data_n_7, order = c(1,0,0), method = "CSS")
lse_n_7
```

```
## Series: data_n_7 
## ARIMA(1,0,0) with non-zero mean 
## 
## Coefficients:
##          ar1    mean
##       0.7014  0.6703
## s.e.  0.0714  0.3418
## 
## sigma^2 estimated as 1.049:  part log likelihood=-143.78
```

```r
lse_n_7$coef
```

```
##       ar1 intercept 
## 0.7014161 0.6703269
```

```r
lse_n_7$var.coef
```

```
##                   ar1   intercept
## ar1       0.005097064 0.001338192
## intercept 0.001338192 0.116824955
```

```r
###### Error term ~ t(4)

data_t_7 <- arima.sim(model = list(ar = c(0.7)), n = 100,
                      rand.gen = function(n,...) rt(n,4))
mle_t_7 <- Arima(data_t_7, order = c(1,0,0), method = "ML")
mle_t_7
```

```
## Series: data_t_7 
## ARIMA(1,0,0) with non-zero mean 
## 
## Coefficients:
```

```
##          ar1    mean
##       0.6844  0.2219
## s.e.  0.0720  0.4624
##
## sigma^2 estimated as 2.265:  log likelihood=-182.08
## AIC=370.16   AICc=370.41   BIC=377.98
```

```r
mle_t_7$coef
```

```
##       ar1 intercept
## 0.6844076 0.2219436
```

```r
mle_t_7$var.coef
```

```
##                    ar1    intercept
## ar1        0.005185172 -0.001141223
## intercept -0.001141223  0.213859308
```

```r
lse_t_7 <- Arima(data_t_7, order = c(1,0,0), method = "CSS")
lse_t_7
```

```
## Series: data_t_7
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.6912  0.2304
## s.e.  0.0730  0.4851
##
## sigma^2 estimated as 2.265:  part log likelihood=-182.26
```

```r
lse_t_7$coef
```

```
##       ar1 intercept
## 0.6911722 0.2304491
```

```r
lse_t_7$var.coef
```

```
##                    ar1    intercept
## ar1        0.005333994 -0.001084524
## intercept -0.001084524  0.235275113
```

-> error term이 t분포를 따를 때보다 normal 분포를 따른다고 가정했을 때 추정 값이 True 값에 근접한 결과를 얻을 수 있었습니다.
-> 같은 error term의 가정을 했을 때는 mle 추정값이 lse 추정값보다 작은 standard error 값을 조금 더 정확한 추정을 하고 있다고 생각할 수 있습니다.

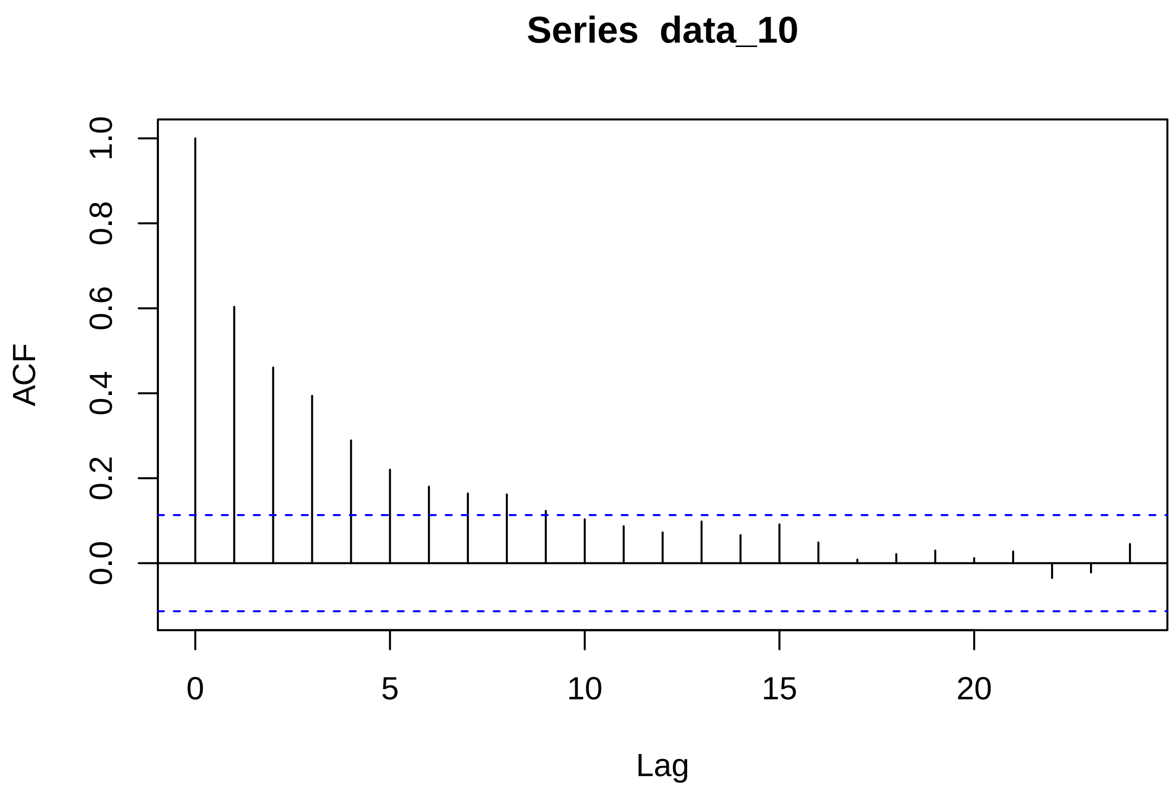
```r
#### 8번

set.seed(200423)

data_8 <- arima.sim(model = list(ar = c(-0.2,0.48)), n = 100,
                    rand.gen = function(n,...) rt(n,4))
fit_8 <- arima(data_8, order = c(2,0,0))
esti <- fit_8$coef
covar <- solve(fit_8$var.coef)
margin <- qchisq(0.95,2)

##### CI = tr((true - esti))%*%covar%*%(true-esti)<=margin
```

-> 추정값이 자유도가 2인 카이제곱 분포를 따르므로 이에 맞게 신뢰영역을 구해주어야 합니다.
-> 카이제곱 분포의 신뢰영역 공식에 맞게 참값을 true, 추정값을 esti, 추정값의 분산행렬의 역행렬을 covar, 자유도가 2인 카이제곱분포에서 95%의 확률을 갖는 값을 margin이라는 변수로 설정했습니다.
-> 그러면 우리가 원하는 신뢰영역은 {true : tr((true - esit)) %% *covar* %%(true - eist) <= margin} 으로 얻을 수 있습니다.

```r
#### 10번

set.seed(200423)

data_10 <- arima.sim(model = list(ar=c(0.7), ma=c(-0.2)), n = 300)
acf_10 <- acf(data_10)

head(acf_10$acf)
```

```
## [1] 1.0000000 0.6035043 0.4605766 0.3940694 0.2889177 0.2199886
```

```r
plot(acf_10)
```



-> 300개의 자료를 생성하여 acf 값을 구했습니다.

```r
#### 12번

set.seed(200423)

data_12 <- arima.sim(model = list(ma=c(0.5)), n = 300)
fit_12 <- Arima(data_12, order = c(0,0,1))
fit_12
```

```
## Series: data_12
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.5094  0.0414
## s.e.  0.0544  0.0876
##
## sigma^2 estimated as 1.019:  log likelihood=-427.72
## AIC=861.44   AICc=861.52   BIC=872.56
```

```r
pre12 <- forecast(fit_12, h = 30)
pre12
```

```
##     Point Forecast     Lo 80     Hi 80     Lo 95    Hi 95
## 301    -0.72347579 -2.017448 0.5704962 -2.702435 1.255484
## 302     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 303     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 304     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 305     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 306     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 307     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 308     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 309     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 310     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 311     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 312     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 313     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 314     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 315     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 316     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 317     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 318     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 319     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 320     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 321     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 322     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 323     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 324     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 325     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 326     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 327     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 328     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 329     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
## 330     0.04140279 -1.410775 1.4935807 -2.179512 2.262317
```