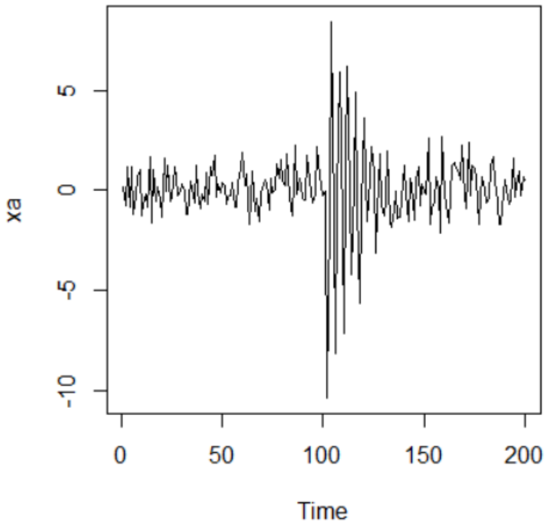
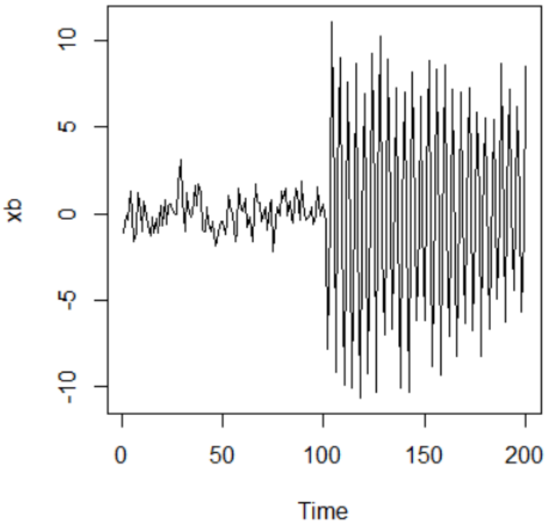
A)

A-1) 1.2

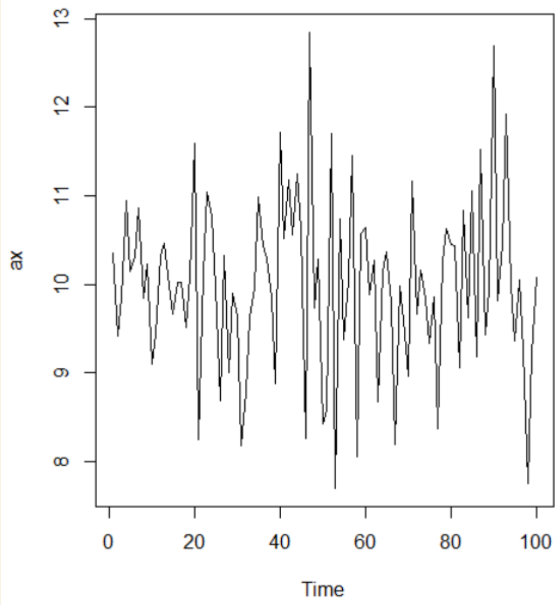
1-2-(a)



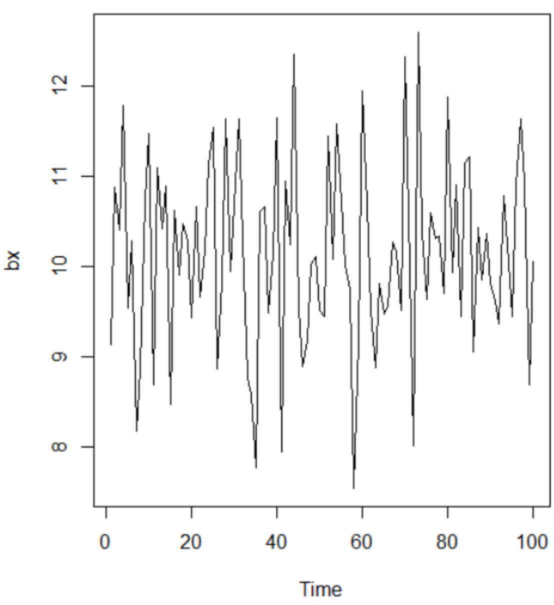
1-2-(b)



1-2-(c)-(a)



1-2-(c)-(b)



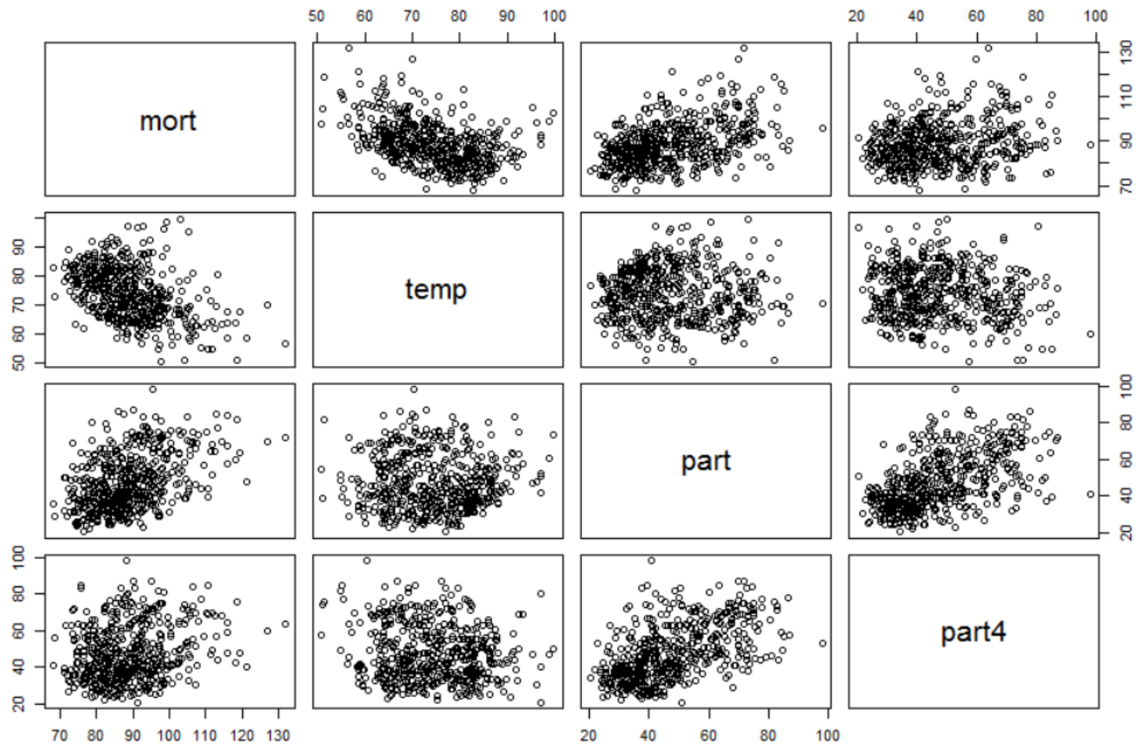
A-2) 2.2

(a)

```
Call:
lm(formula = mort ~ trend + temp1 + temp2 + part + part4)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.4935  -4.3268  -0.5336   3.5860  29.1889 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.852e+03  1.964e+02  14.520  < 2e-16 ***
trend       -1.405e+00  9.943e-02 -14.135  < 2e-16 ***
temp1       -4.923e-01  3.148e-02 -15.639  < 2e-16 ***
temp2        2.281e-02  2.782e-03   8.199 2.04e-15 ***
part         3.057e-01  2.213e-02  13.813  < 2e-16 ***
part4       -9.328e-02  2.235e-02  -4.174 3.52e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.284 on 502 degrees of freedom
Multiple R-squared:  0.609,	Adjusted R-squared:  0.6051 
F-statistic: 156.4 on 5 and 502 DF,  p-value: < 2.2e-16
```

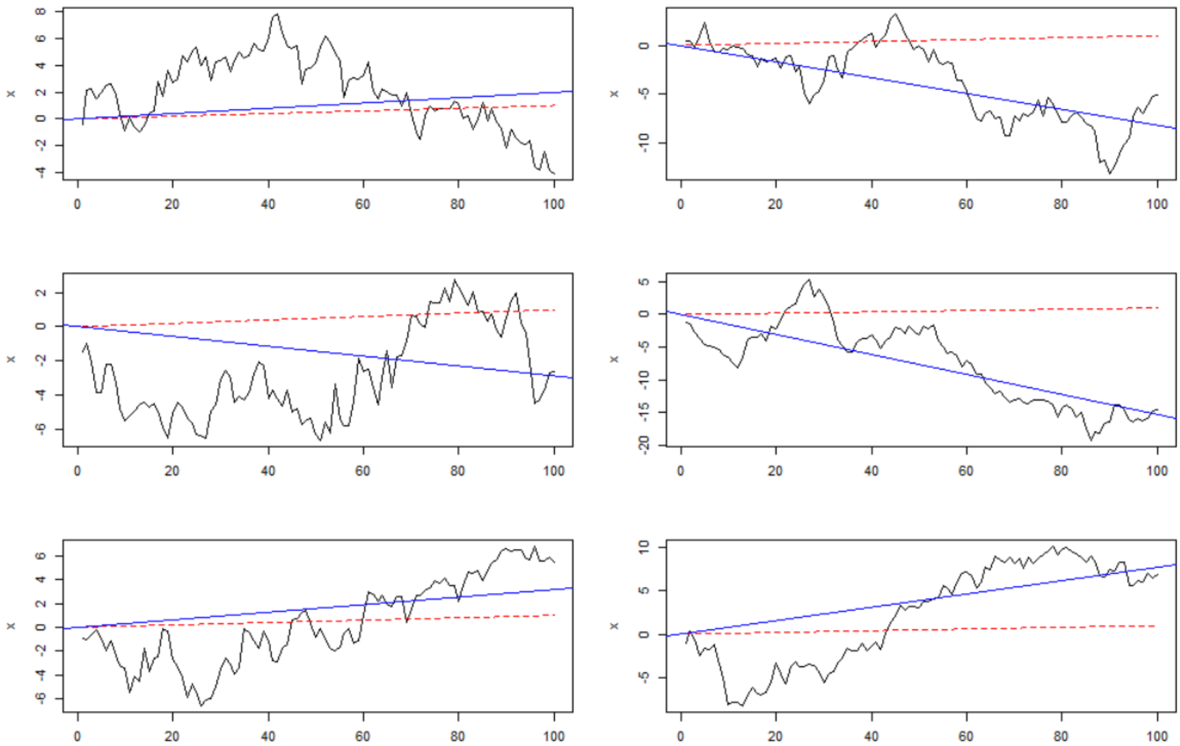
p –value 값을 통해 회귀식이 유의하다는 것을 알 수 있다. 또한 각 변수들 역시 유의하며 Adjusted R-squared 값을 통해 이 회귀식이 전체의 60퍼센트 정도를 설명함을 알 수 있다.

(b)



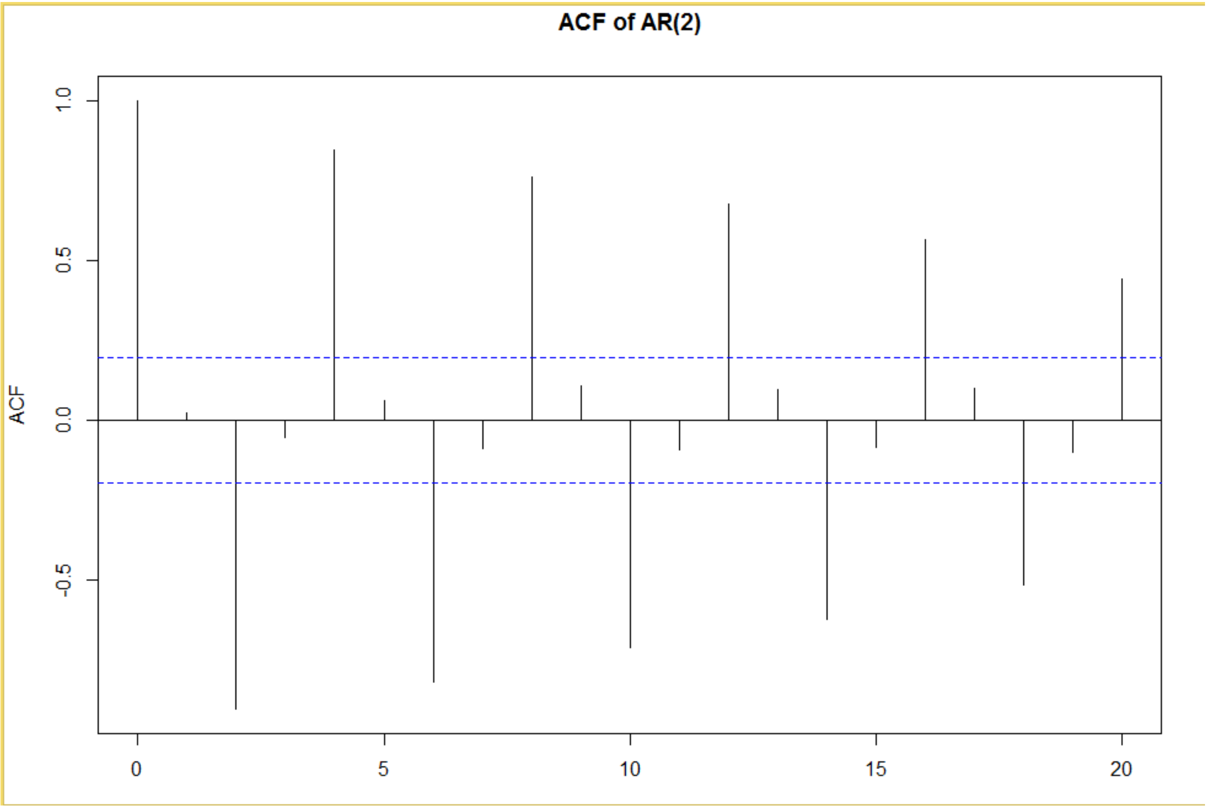

```
> cor(cbind(mort,part), cbind(mort,part4))
          mort     part4
mort 1.0000000 0.2122061
part 0.4438713 0.5464271
```

A-3) 2-3



평균(빨간선) 회귀선(파랑) / 평균은 점차 증가하는 추세이고, 이 추세는 함수 반복 횟수에 상관없이 동일 한데 (rnorm 에서 평균을 0.01로 지정해 줬으니 어쩌면 당연한 결과) 회귀선은 샘플이 어떻게 나타나느냐에 따라 굉장히 민감하게 반응한다. (매번 달라진다)

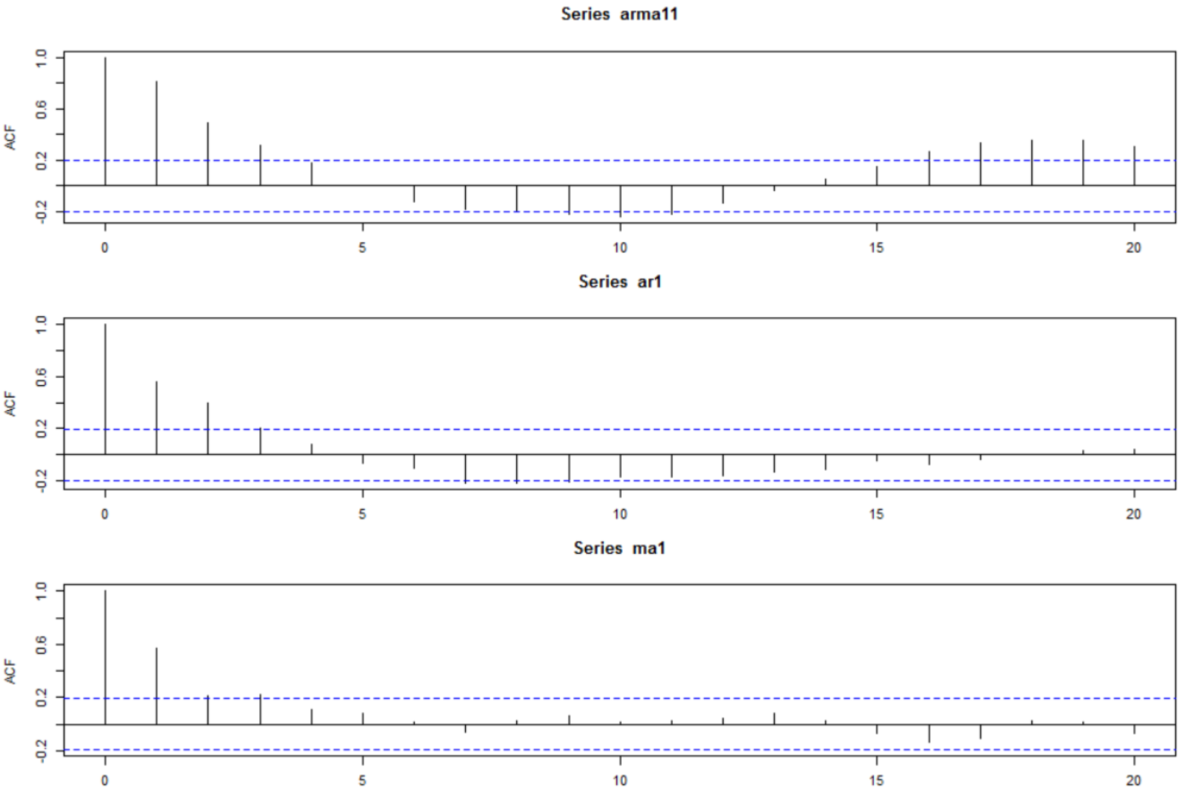
A-4) 3.6

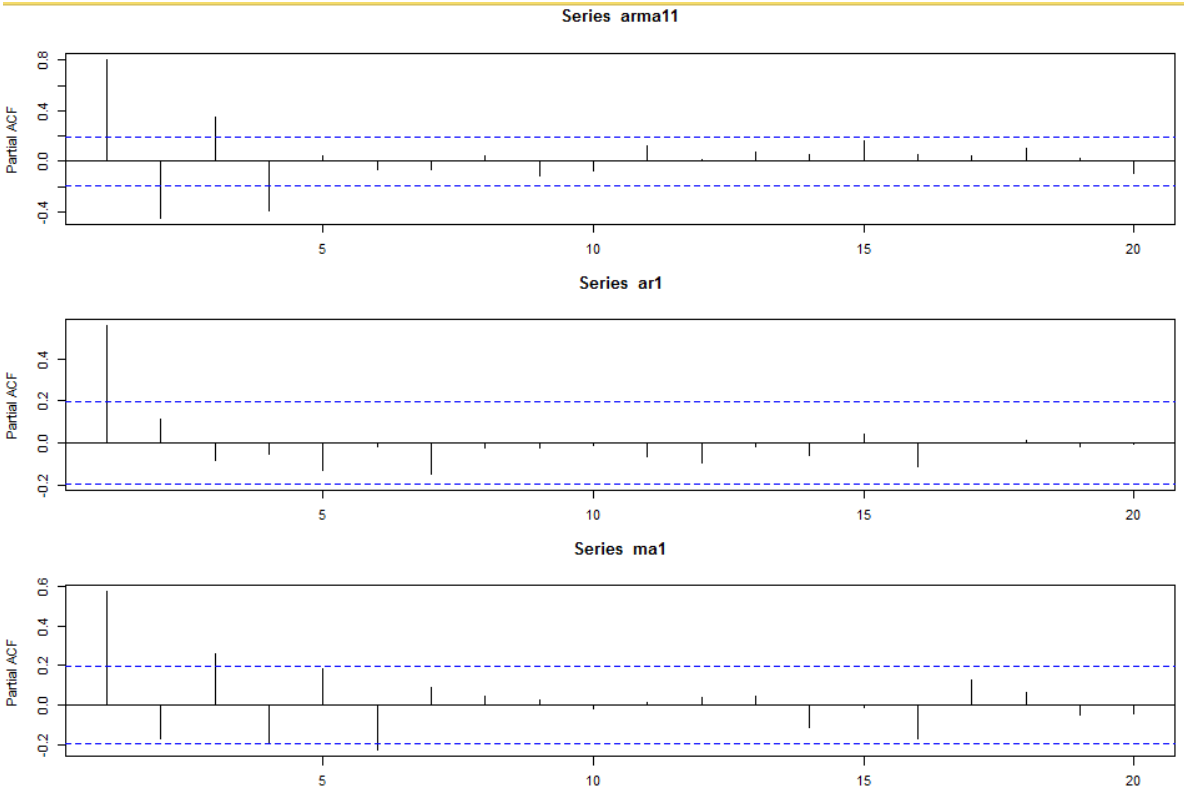



```
> polyroot(aa)[1]
[1] 1.054093+0i
```

ACF plot을 통해 현재시점의 데이터는 짝수번째 시차에 대한 데이터의 영향을 받고있다고 해석할 수 있다.

A-5) 3-9


Series arma11
ACF
Series ar1
ACF
Series ma1
ACF


Series arma11
Partial ACF
Series ar1
Partial ACF
Series ma1
Partial ACF

3-19

(a) AR(1) 모형 : $X_t = \phi X_{t-1} + Z_t \rightarrow X_{t-1} = \phi^{-1} X_t - \phi^{-1} Z_t$

X_1이 여기서는 제일 가까운 관측값이므로 $\therefore X_t^n = \phi^{1-t} X_1 \quad (t \le 1)$

(b) $\hat{w}_t(\phi) = X_t^n - \phi X_{t-1}^n = \phi^{1-t} x_1 - \phi(\phi^{1-(t-1)} x_1) = x_1 \phi^{1-t} (1-\phi^2)$

(c) $\hat{w}_t(\phi) = x_1 (1-\phi^2) \phi^{1-t}$

$\hat{w}_t^2(\phi) = x_1^2 (1-\phi^2)^2 \phi^{2-2t}$

$\sum_{t=-\infty}^{1} \hat{w}_t^2(\phi) = x_1^2 (1-\phi^2)^2 (\phi^0 + \phi^2 + \phi^4 + \cdots)$

$= x_1^2 (1-\phi^2)^2 \dfrac{1}{1-\phi^2} = x_1^2 (1-\phi^2)$

(d) $\sum_{t=-\infty}^{n} \hat{w}_t^2(\phi) = x_1^2 (1-\phi^2)^2 (\phi^{2-2n} + \phi^{-2n+4} + \phi^{-2n+6} + \cdots)$

$= x_1^2 (1-\phi^2)^2 \times \dfrac{\phi^{2-2n}}{1-\phi^2} = x_1 (1-\phi^2) \phi^{2-2n} = S(\phi)$

(e) r_t가 어떤걸 의미하는지 잘 모르겠어요ㅠㅠ

A-7)3-21

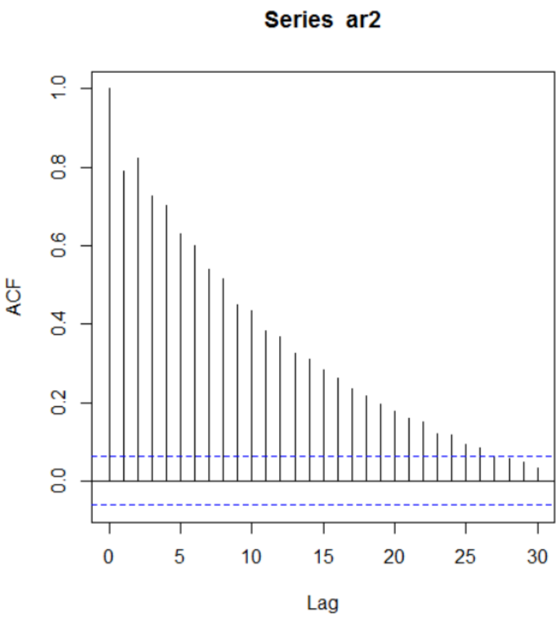
```
> library(stats4)
> mle(mle_function, list(sigma = 1, theta = 1, pii = 1), method = "CG")

Call:
mle(minuslogl = mle_function, start = list(sigma = 1, theta = 1,
    pii = 1), method = "CG")

Coefficients:
       sigma         theta           pii
-0.000182625   0.688490854   0.841240542
```

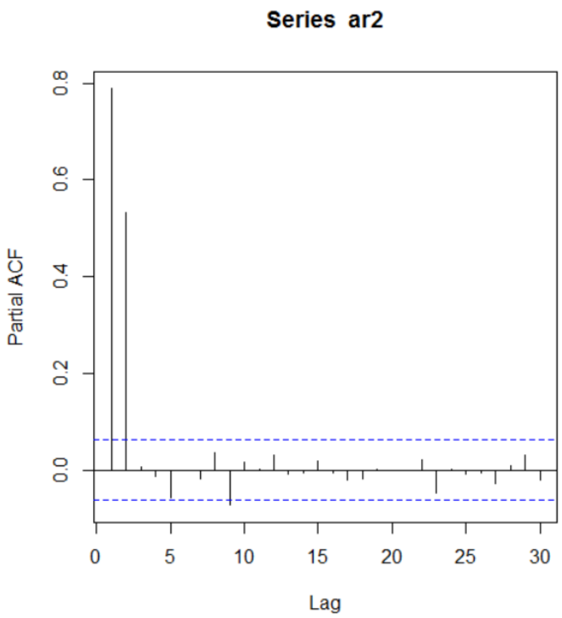
B) Generating 1000 observations form AR(2) model

```
> ar2=arima.sim(list(order=c(2,0,0), ar=c(0.4,0.5)),n=1000) # 어떻게 해야하지?
> head(ar2)
Time Series:
Start = 1
End = 6
Frequency = 1
[1] 0.2063819 0.5477018 0.6371883 0.6589994 2.5863528 0.2147617
> str(ar2)
 Time-Series [1:1000] from 1 to 1000: 0.206 0.548 0.637 0.659 2.586 ...
```

Xt=0.4Xt-1-0.5Xt-2+Zt (Zt~WN) 인 임의의 모델을 생성했다.

B-i) Draw time series plot, sample ACF and PACF plot.





B-ii) Fit the model under AR(2) and under AR(4) and explain the result. What do you think is the problem when you fit bigger model then necessary?

```
> new_ar2

Call:
arima(x = ar2, order = c(2, 0, 0))

Coefficients:
         ar1     ar2  intercept
      0.4442  0.4639    -0.1869
s.e.  0.0279  0.0280     0.3365

sigma^2 estimated as 0.9832:  log likelihood = -1411.28,  aic = 2830.56
> new_ar4

Call:
arima(x = ar2, order = c(4, 0, 0))

Coefficients:
         ar1     ar2     ar3      ar4  intercept
      0.4411  0.4663  0.0123  -0.0119    -0.1876
s.e.  0.0316  0.0345  0.0346   0.0317     0.3349

sigma^2 estimated as 0.983:  log likelihood = -1411.18,  aic = 2834.37
```

문제점: coefficients 값을 보아 ar3, ar4가 유의한 변수가 아닐 뿐더러 AIC값 역시 AR(2)와 비교했을 때 커졌다. 이를 통해 AR(2)가 더 적합하다는 걸 파악할 수 있다. (유의성은 ar의 계수와 s.e.의 값을 통해 알 수 있다)

B-iii) Fit the model under MA(p) varying p from 1 to 10. What do you observe? Explain your observation

```
> ex.arima = function(x){
+
+   for(x in 1:10){
+     print(arima(ar2, order=c(0,0,x)))
+   }
+ }
> ex.arima(1)

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1  intercept
      0.4533      0.117
s.e.  0.0202      0.066

sigma^2 estimated as 2.061:  log likelihood = -1780.64,  aic = 3567.29

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2  intercept
      0.4677  0.5818     0.1142
s.e.  0.0298  0.0229     0.0758

sigma^2 estimated as 1.371:  log likelihood = -1577,  aic = 3162

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3  intercept
      0.5185  0.6231  0.2872     0.1135
s.e.  0.0336  0.0248  0.0293     0.0853

sigma^2 estimated as 1.237:  log likelihood = -1525.67,  aic = 3061.33

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4  intercept
      0.4978  0.7444  0.3706  0.2744     0.1118
s.e.  0.0316  0.0318  0.0308  0.0270     0.0967

sigma^2 estimated as 1.125:  log likelihood = -1478.46,  aic = 2968.92

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4     ma5  intercept
      0.4984  0.7533  0.4521  0.3281  0.2013     0.1101
s.e.  0.0322  0.0335  0.0337  0.0289  0.0298     0.1059

sigma^2 estimated as 1.076:  log likelihood = -1456.15,  aic = 2926.29

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1    ma2     ma3     ma4     ma5     ma6  intercept
      0.4934  0.758  0.4669  0.4072  0.2589  0.1402     0.1092
s.e.  0.0320  0.035  0.0387  0.0345  0.0316  0.0288     0.1140

sigma^2 estimated as 1.051:  log likelihood = -1444.43,  aic = 2904.87

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7  intercept
      0.4702  0.7489  0.4746  0.4277  0.3193  0.1770  0.1444     0.1075
s.e.  0.0328  0.0352  0.0399  0.0364  0.0349  0.0297  0.0329     0.1205

sigma^2 estimated as 1.031:  log likelihood = -1434.84,  aic = 2887.68
```

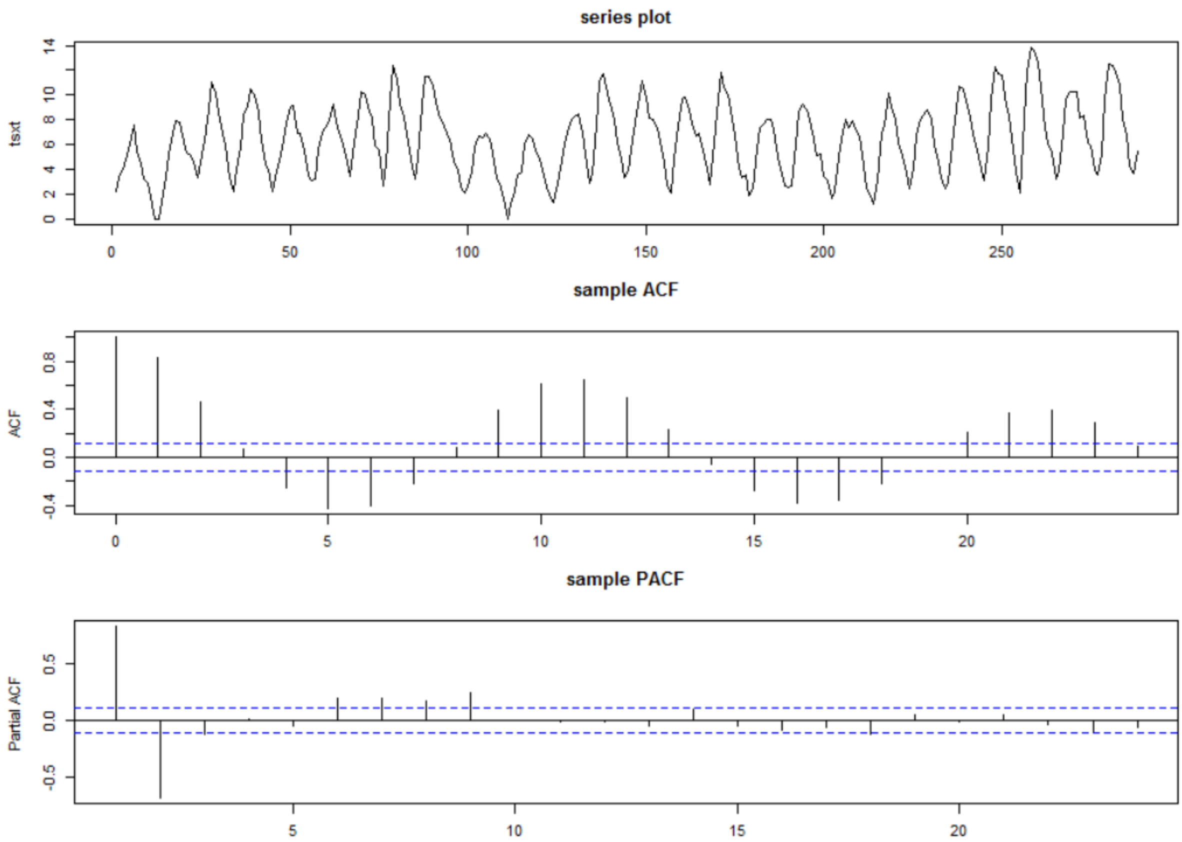
```
Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8  intercept
      0.4592  0.7395  0.4696  0.4348  0.3385  0.2700  0.1953  0.1431     0.1058
s.e.  0.0321  0.0357  0.0421  0.0413  0.0369  0.0351  0.0331  0.0308     0.1283

sigma^2 estimated as 1.009:  log likelihood = -1424.04,  aic = 2868.09

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8     ma9  intercept
      0.4549  0.7323  0.4530  0.4253  0.3643  0.3040  0.2733  0.1990  0.1242     0.1044
s.e.  0.0318  0.0355  0.0421  0.0418  0.0381  0.0363  0.0386  0.0342  0.0319     0.1361

sigma^2 estimated as 0.994:  log likelihood = -1416.62,  aic = 2855.24

Call:
arima(x = ar2, order = c(0, 0, x))

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8     ma9    ma10
      0.4506  0.7254  0.4465  0.4369  0.3782  0.3268  0.3018  0.2556  0.1593  0.0883
s.e.  0.0319  0.0350  0.0409  0.0415  0.0404  0.0383  0.0403  0.0395  0.0344  0.0326
      intercept
         0.1029
s.e.     0.1431

sigma^2 estimated as 0.9867:  log likelihood = -1412.9,  aic = 2849.81
```

모형이 커질수록 정확도는 높아진다. (log likelihood 값과 AIC값 모두가 감소함을 알 수 있다)

분산 역시 작아진다. 하지만 AIC가 낮아진다고 무조건적으로 모형이 큰 게 좋은게 아니므로 적당한 기준을 세워야 할 것이다. 보면 점점 MA값이 커질수록 설명력이 낮아지는데 이를 통해 MA(4)정도까진 선택해도 될 것 같다고 볼 수 있다.

C)

C-i) Draw time series plot, sample ACF and PACF plot



C-ii) By considering the sample ACF and sample PACF, decide wich of the following would e appropriate for this data: AR(1), AR(2), MA(1), MA(2). Use the data to estimate the parameters of the model that you choose.

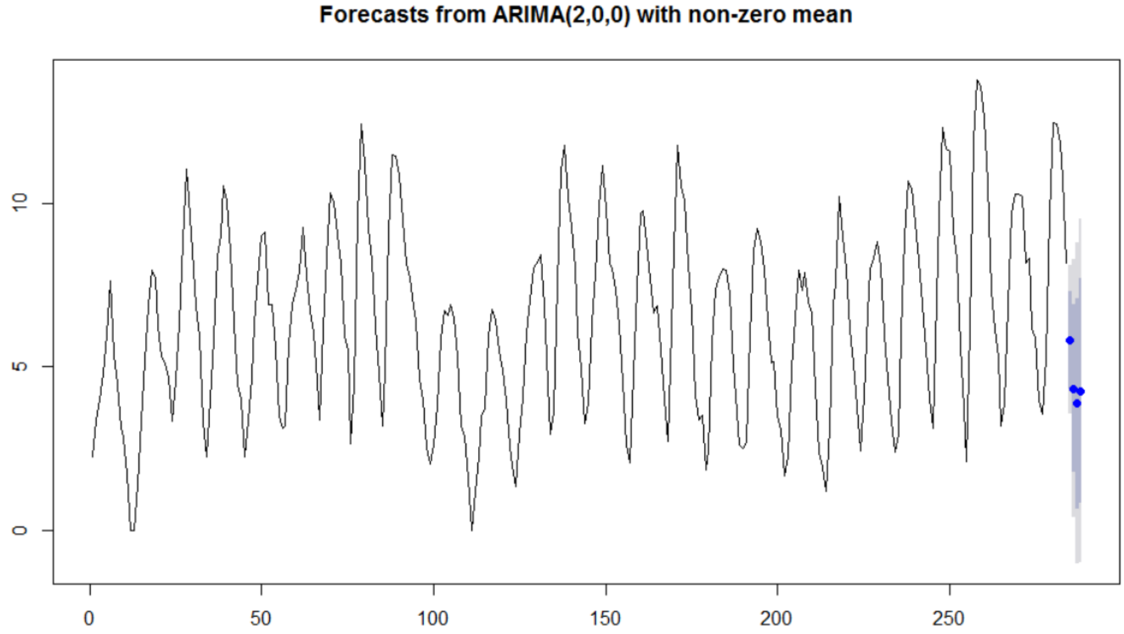
```
> ar1

Call:
arima(x = tsxt, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.8292     6.2503
s.e.  0.0327     0.5488

sigma^2 estimated as 2.613:  log likelihood = -547.53,  aic = 1101.07
> ar2

Call:
arima(x = tsxt, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      1.4034  -0.6928     6.3352
s.e.  0.0423   0.0423     0.2371

sigma^2 estimated as 1.357:  log likelihood = -453.83,  aic = 915.66

> ma1

Call:
arima(x = tsxt, order = c(0, 0, 1))

Coefficients:
         ma1  intercept
      0.8239     6.3205
s.e.  0.0263     0.1947

sigma^2 estimated as 3.292:  log likelihood = -580.79,  aic = 1167.58
> ma2

Call:
arima(x = tsxt, order = c(0, 0, 2))

Coefficients:
         ma1     ma2  intercept
      1.2458  0.7845     6.3257
s.e.  0.0392  0.0354     0.2380

sigma^2 estimated as 1.788:  log likelihood = -493.61,  aic = 995.23
```

Conclusion : AIC가 제일 낮은 AR(2)가 제일 적합하다고 결론 지을 수 있다. 이때의 식은 $X_t=1.4034*X_{t-1}-0.6928*X_{t-2}+Z_t$ (Z_t~WN)이다.

C-iii) Using your fitted model, calculate forecasts $X_{n \wedge n+h}$ for h=1,2,3,4. Calculate the 95% prediction intervals (assuming Gussian noise)

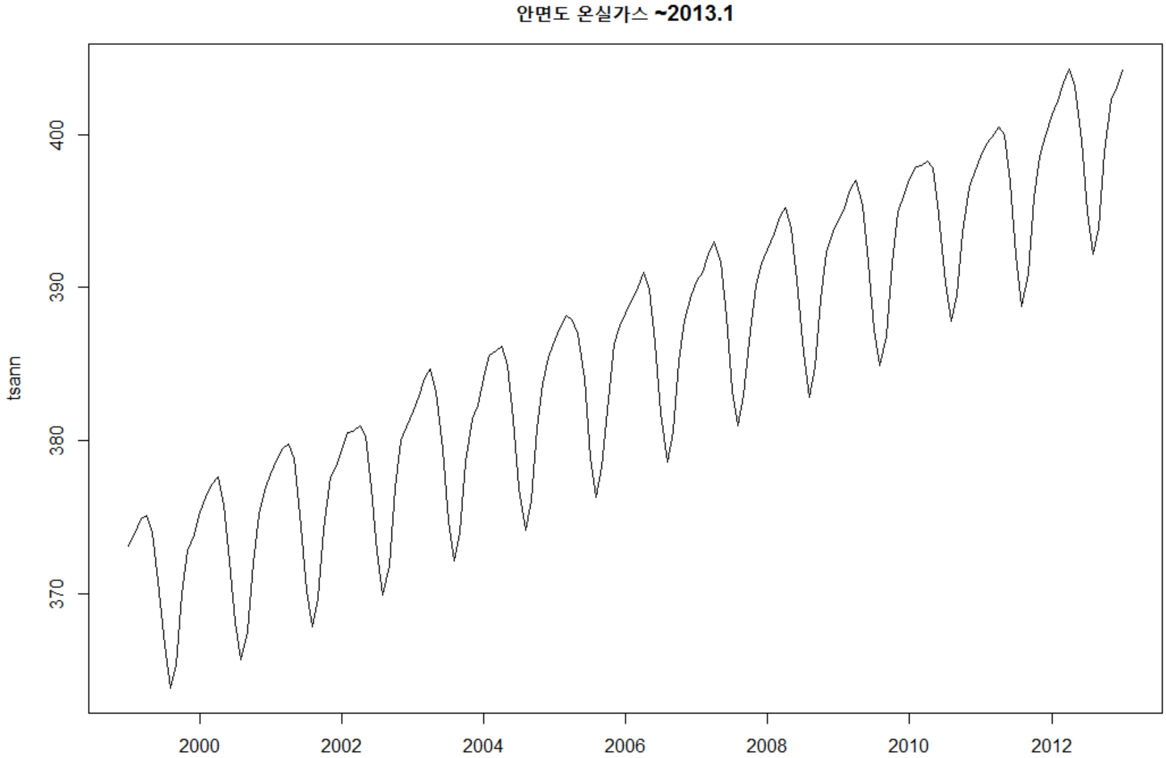
```
> forecast(ar2, h = 4)
    Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
289       6.893965 5.401190  8.38674 4.610963  9.176967
290       7.758360 5.185937 10.33078 3.824178 11.692542
291       7.945363 4.743830 11.14690 3.049041 12.841685
292       7.608919 4.181640 11.03620 2.367349 12.850489
```

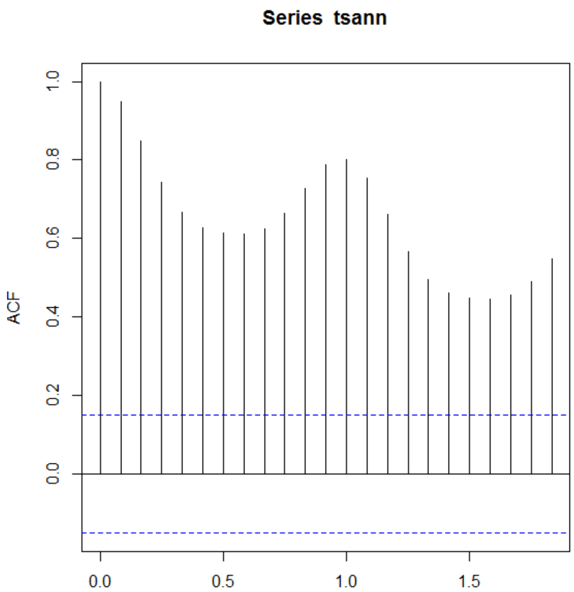
C-iv) Use the observation from 1700 to 1983 for fitting the model. Plot all of the data, and your forecasts and prediction intervals for the last four years. (Don't forget to undo the square root transformation by taking the square of your predictions)

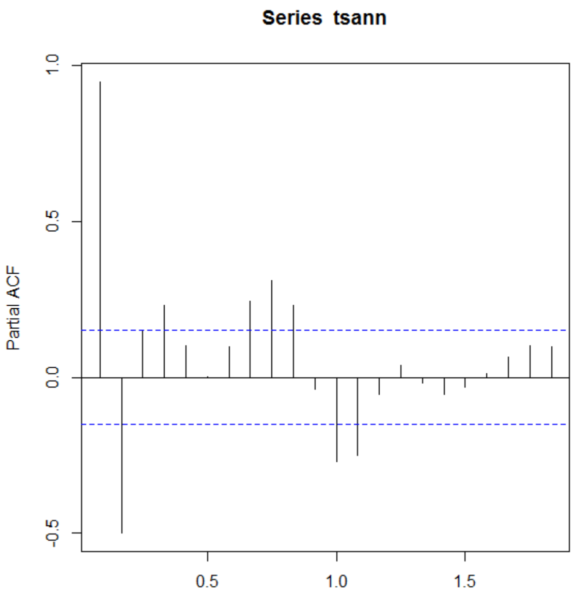



```
> forcasting = (as.data.frame(forecast(fit3, h = 4))[, 1])^2
> forcasting
[1] 33.84347 18.76252 14.98035 18.10609
```

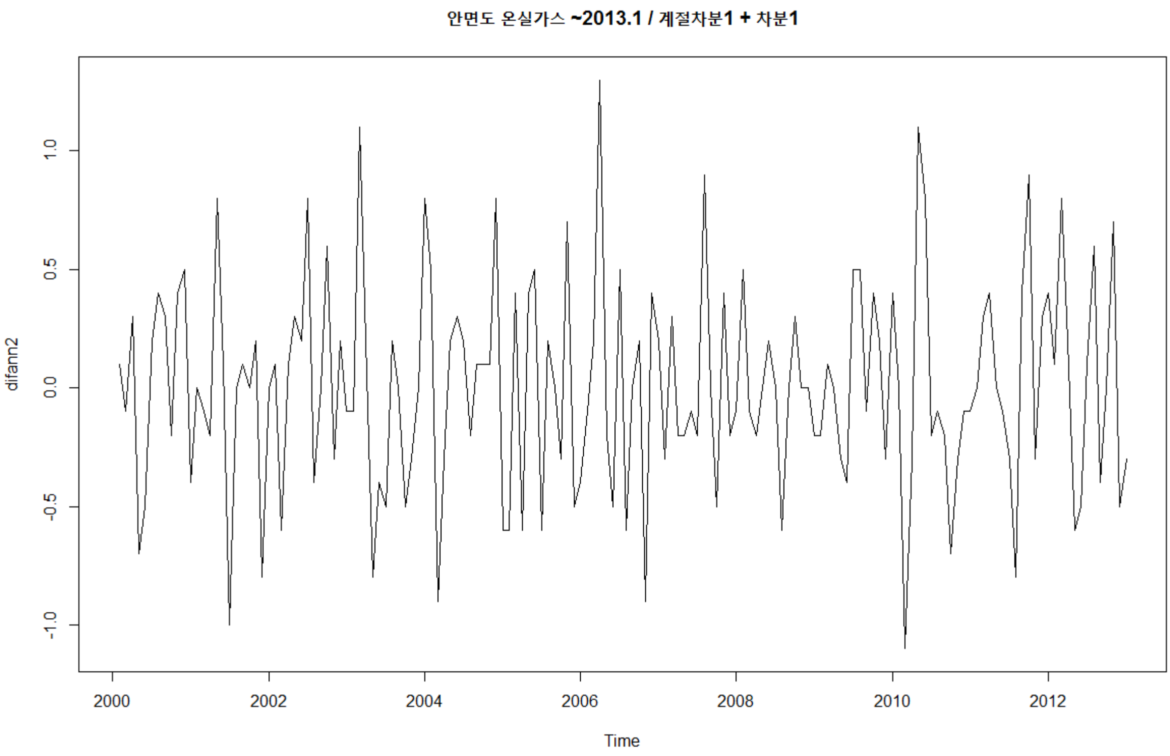
D) By using seasonal ARIMA models. Forecast monthly averages for year 2014. Compare your forecast with the observed values. Do anything you can and make a readable answer.



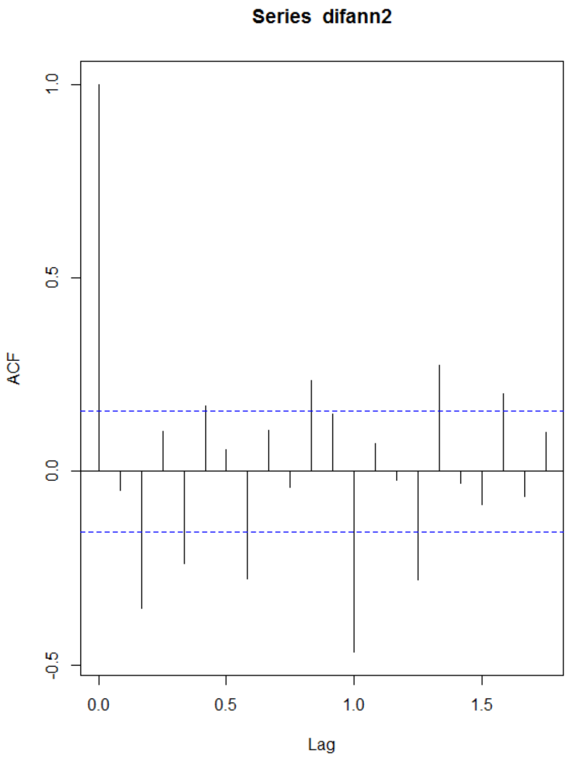


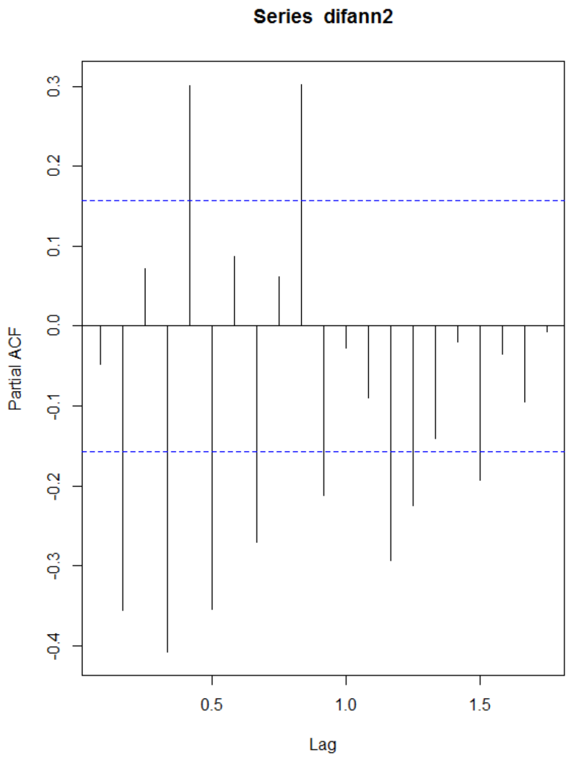


Trend , 주기성 존재 -> differncing 으로 주기와 트렌드 제거하는 것이 필요하다



트렌드 제거, 주기 제거 확인. (계절차분 한번 + 차분한번)



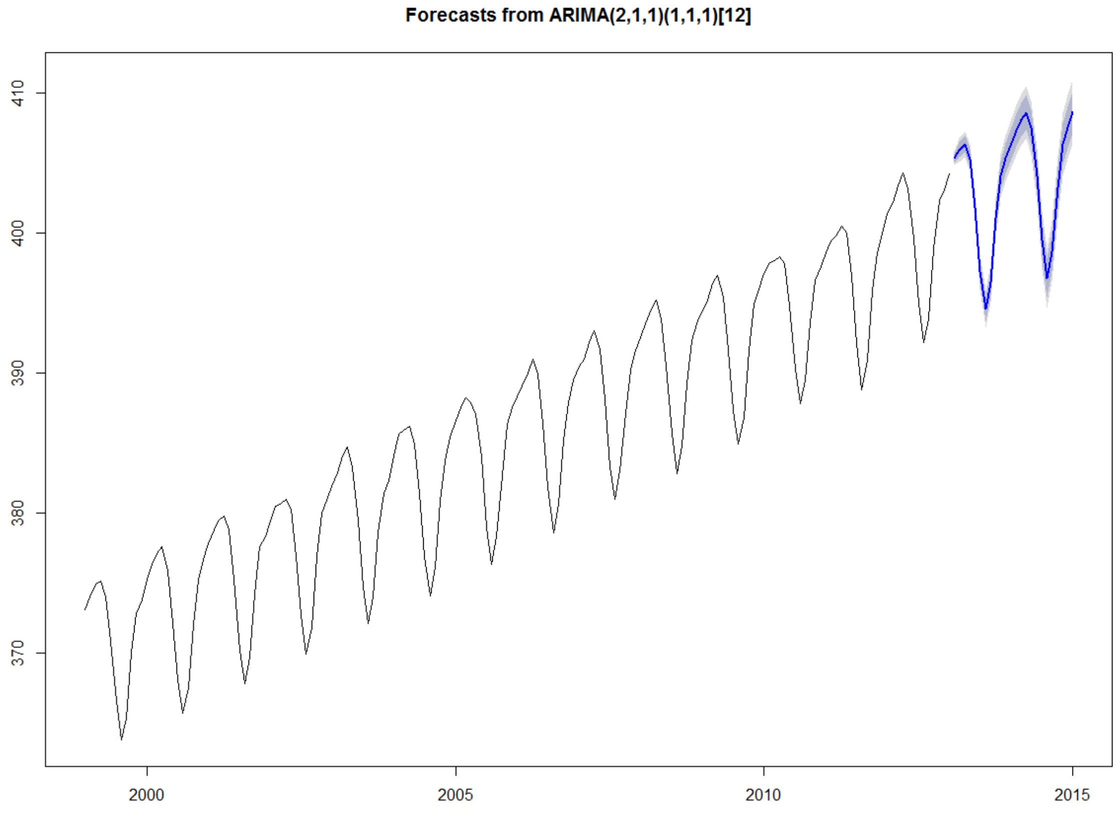


```
> auto.arima(tsann)
Series: tsann
ARIMA(3,0,2)(2,1,0)[12] with drift

Coefficients:
          ar1     ar2     ar3     ma1     ma2     sar1     sar2   drift
      -0.4381  0.1188  0.5061  1.7300  0.8882  -0.6787  -0.4322  0.1820
s.e.   0.0791  0.0824  0.1004  0.0528  0.0471   0.0939   0.0838  0.0036

sigma^2 estimated as 0.07308:  log likelihood=-7.03
AIC=32.05   AICc=33.28   BIC=59.56
```

이를 기반으로 여러가지를 fitting 해본 결과

```
> fit7 = arima(ts.train[, 2], order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1)));fit7 #얘가 베스트임(aic 기준)

Call:
arima(x = ts.train[, 2], order = c(2, 1, 1), seasonal = list(order = c(1, 1,
    1)))

Coefficients:
          ar1      ar2     ma1    sar1     sma1
      -0.4895  -0.4125  0.8433  -0.107  -0.9992
s.e.   0.0868   0.0758  0.0498   0.093   0.2359

sigma^2 estimated as 0.05975:  log likelihood = -19.45,  aic = 50.9
```

Arima(2,1,1)*(1,1,1)[12] 가 AIC기준으로 가장 좋은 적합모델임을 확인했다. (사실 모든모델을 다 비교해 보진 못했으므로 완벽하다고 볼 수는 없지만 auto.arima를 통해 추정한 값 보다도 훨씬 낮은 aic값임을 확인했다)



```
> result
2014년 예측평균 2014년 원자료평균
      404.8313        404.6417
```

실제값과 결과값을 비교했을 때 정말 근소한 차이를 보이므로 거의 정확히 예측했다고 볼 수 있을 것 같다.