Homework2

seongbin

2020 9 17

0.Data check

시계열 분석을 하기에 앞서 데이터의 분포를 살표본다.

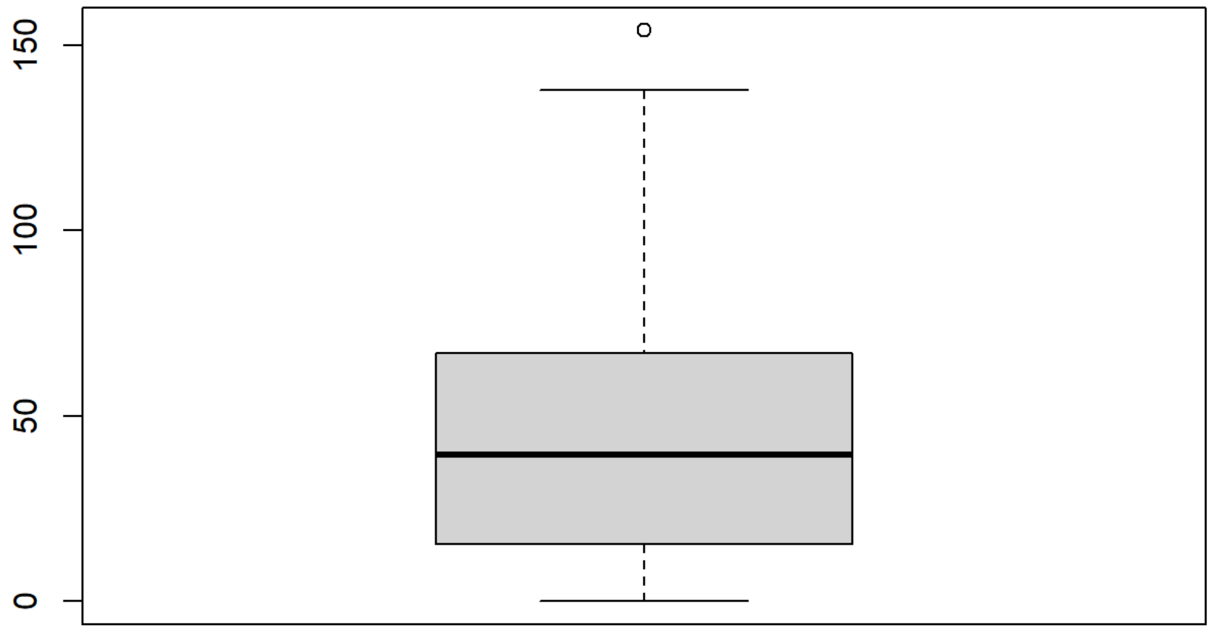
```
X<- read.csv("C://Users//stat//Desktop//학교//2-2//시계열//spot.csv")
head(X)
```

```
##   Spot
## 1  101
## 2   82
## 3   66
## 4   35
## 5   31
## 6    7
```

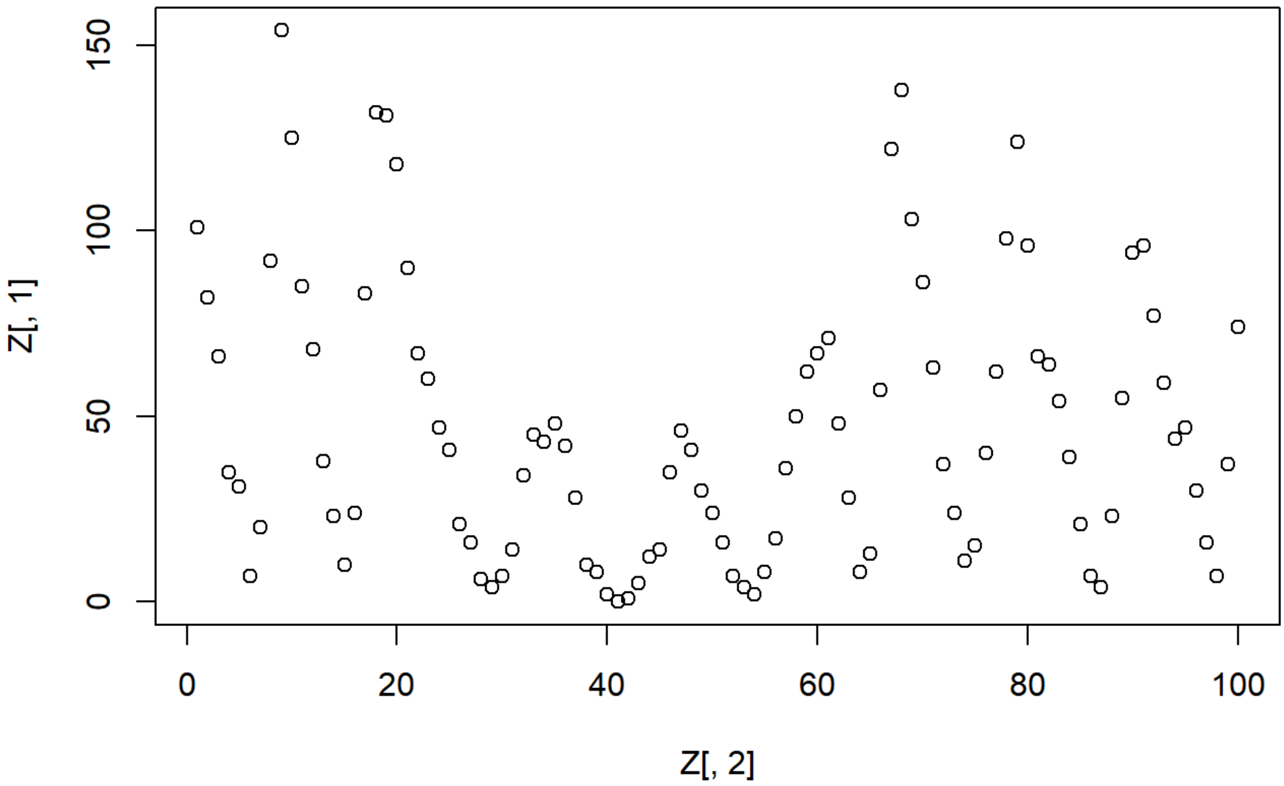
```
T<-as.matrix(X)
dim(T)
```

```
## [1] 100   1
```

```
boxplot(T[,1])
```



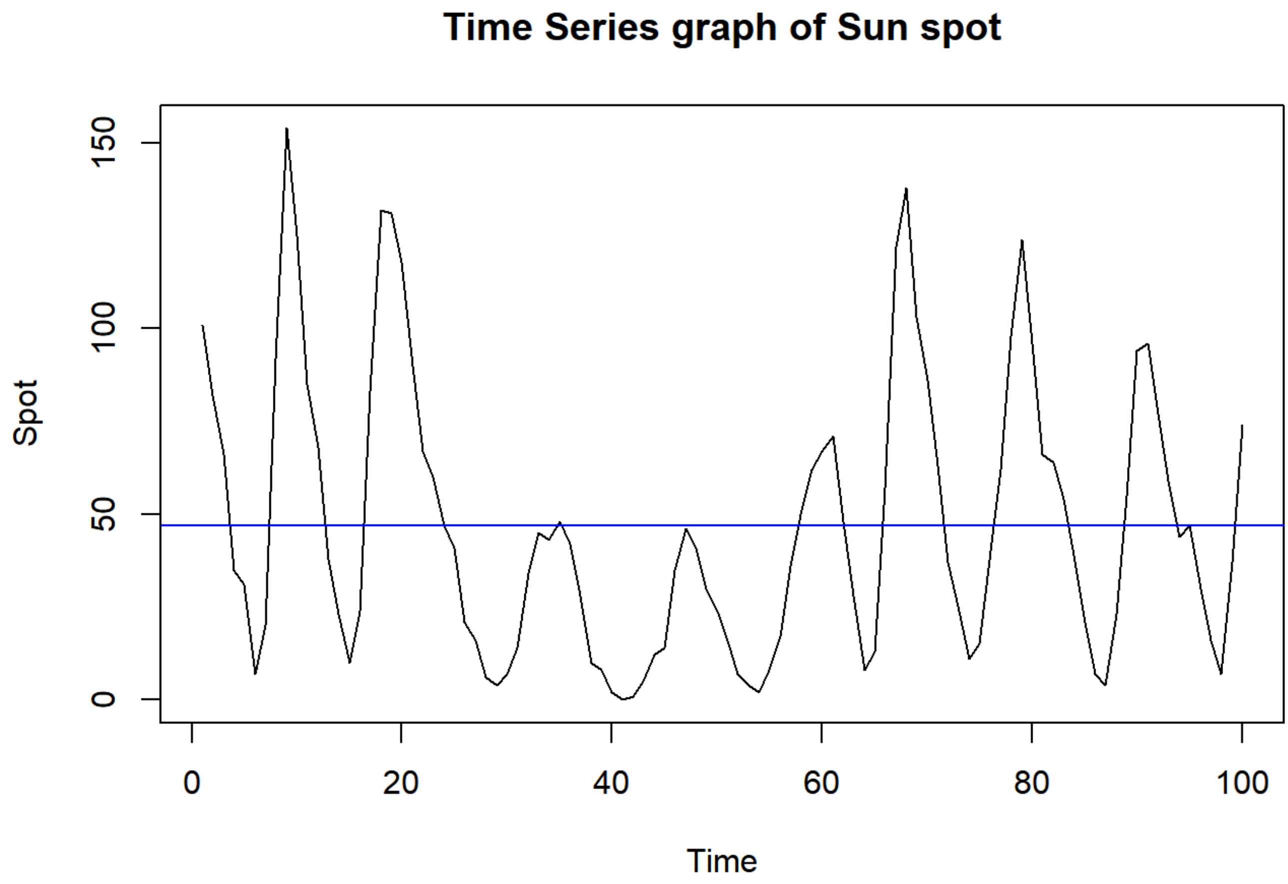

```
Y=seq(1,100,1)
Z<-cbind(T,Y)
plot(Z[,2],Z[,1])
```



SOL1

Draw time series graph of sun spot

```
T.ts=ts(data=T,frequency=1)
plot(T.ts, mai ="Time Series graph of Sun spot")
abline(h=mean(T.ts[,1]),col="blue")
```



Do you see any cycle or seasonal effect?

```
tseries::kpss.test(T.ts,null="Level")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Warning in tseries::kpss.test(T.ts, null = "Level"): p-value greater than
## printed p-value
```

```
## 
##  KPSS Test for Level Stationarity
## 
## data:  T.ts
## KPSS Level = 0.15966, Truncation lag parameter = 4, p-value = 0.1
```

```
tseries::kpss.test(T.ts,null="Trend")
```

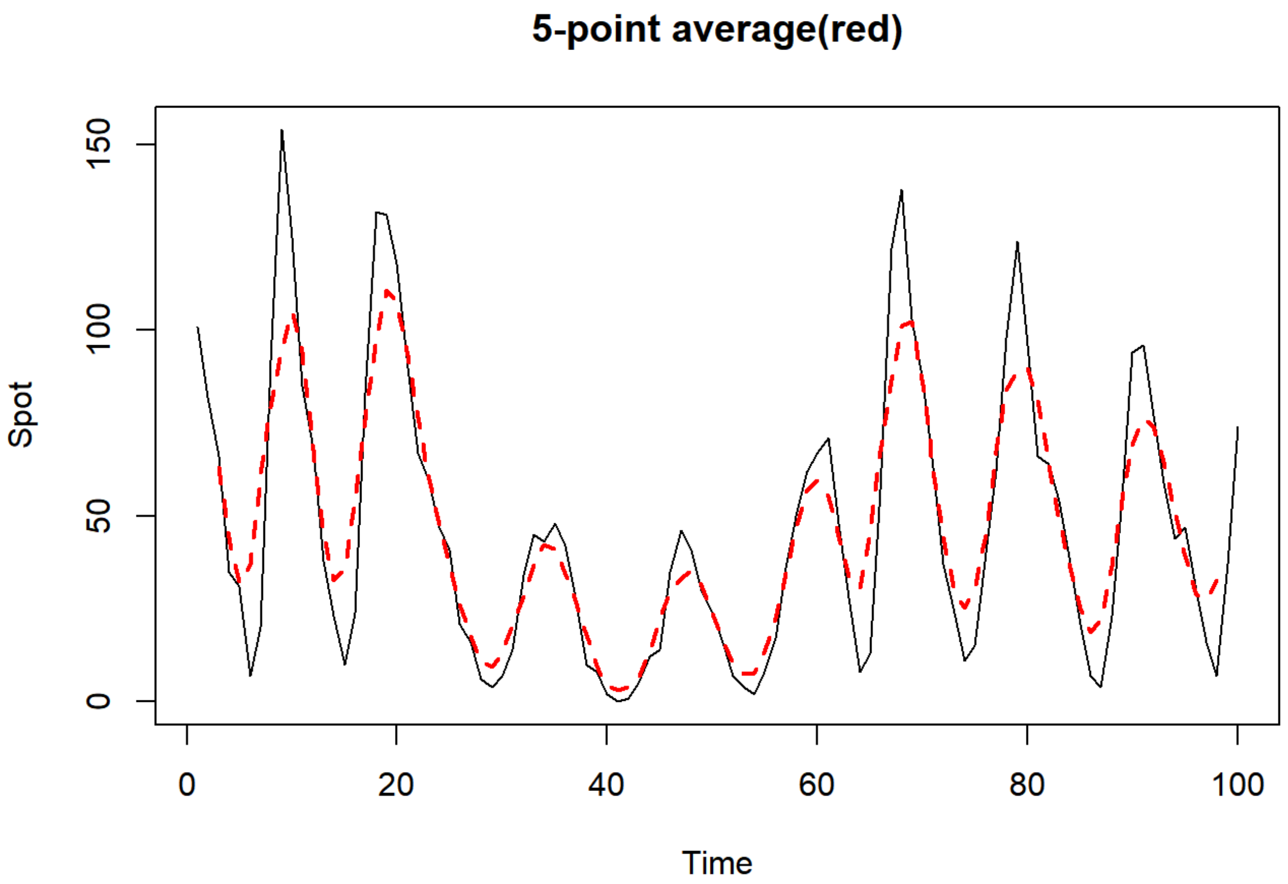
```
##
##  KPSS Test for Trend Stationarity
##
## data:  T.ts
## KPSS Trend = 0.15776, Truncation lag parameter = 4, p-value = 0.0402
```

kpss.test의 p-value 판단 결과(유의수준0.05기준), Trend를 고려하지 않았을 경우 데이터는 비정상 시계열로 판단된다.반면,Trend(추세&계절성)를 고려했을 경우 정상적인 시계열로 판단할 수 있다. =>Trend가 있다고 판단된다.

SOL2

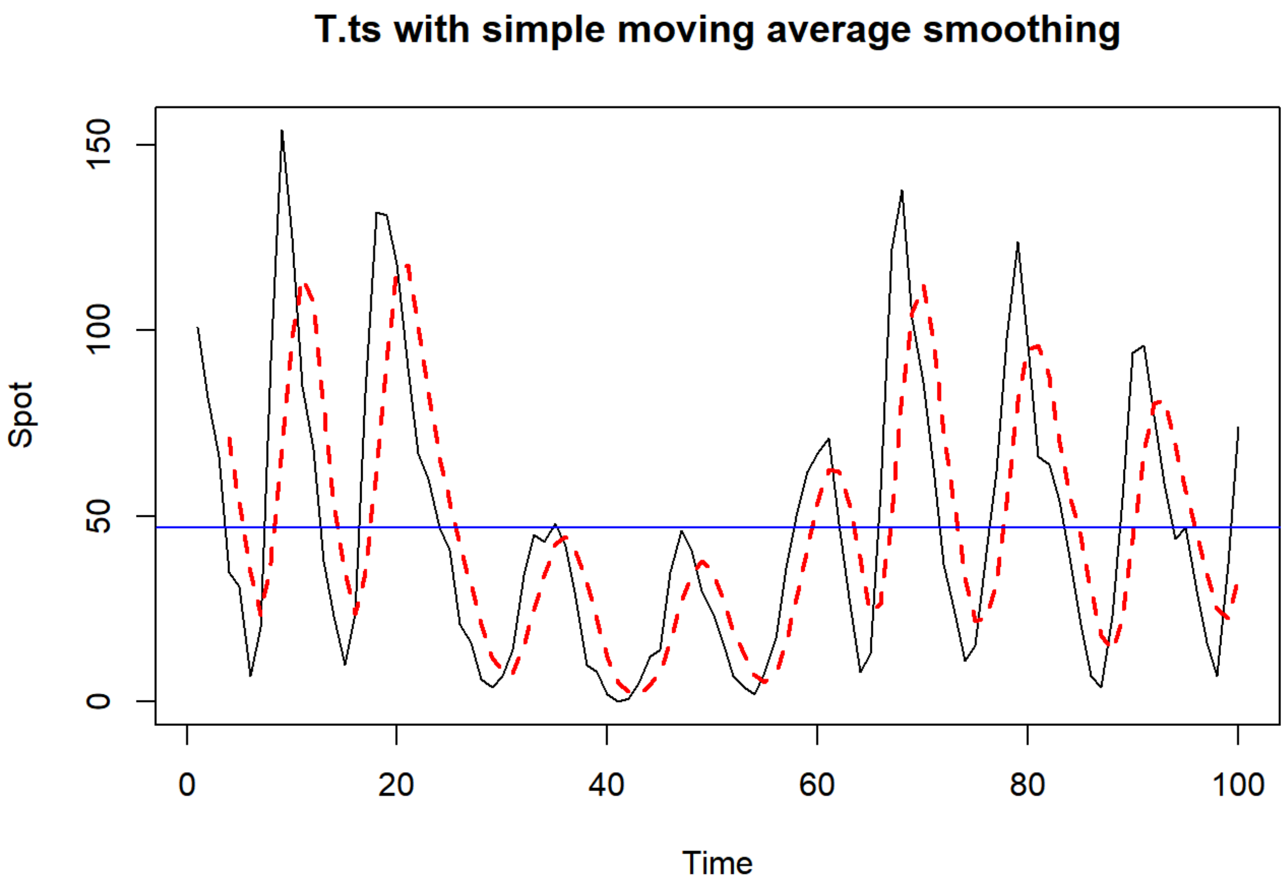
try 5-point moving average smoothing.

```
m5=filter(T.ts, filter=rep(1/5,5))
plot(T.ts,main="5-point average(red)")
lines(m5,col="red",lty=2, lwd=2)
```



Draw the plot of original graph in black, 4 point MA smoothing in red, mean value in blue.

```
ff=filter(T.ts, filter=rep(1,4)/4, method="convolution", sides=1)
plot(T.ts, main="T.ts with simple moving average smoothing")
lines(ff, col="red",lty=2, lwd=2)
abline(h=mean(T),col="blue")
```



SOL3

Check the residual plot

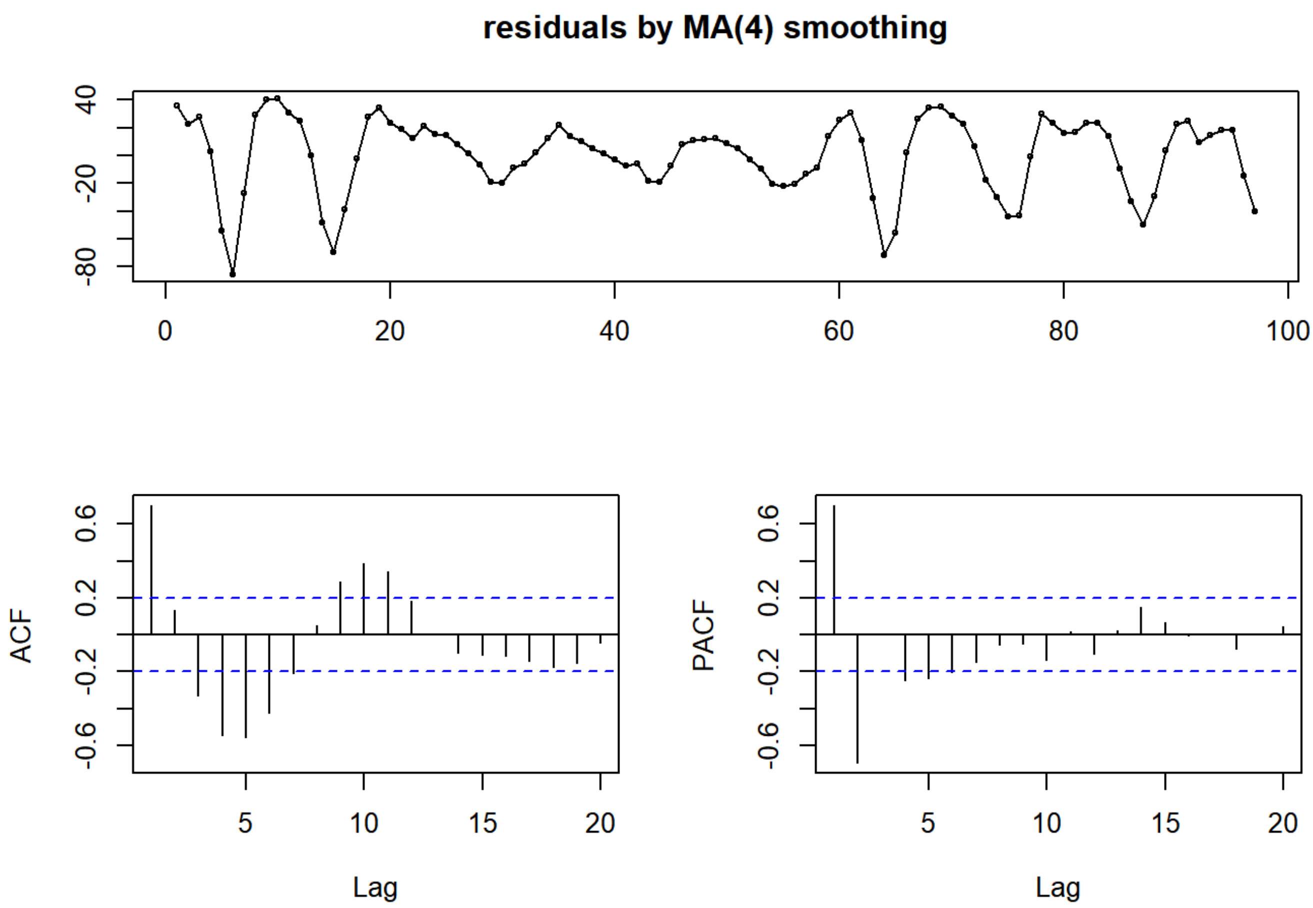
```
head(ff,10)
```

```
##       [,1]
##  [1,]    NA
##  [2,]    NA
##  [3,]    NA
##  [4,] 71.00
##  [5,] 53.50
##  [6,] 34.75
##  [7,] 23.25
##  [8,] 37.50
##  [9,] 68.25
## [10,] 97.75
```

```
#4개의 point로 MA를 시행하였으므로 앞의 3관측치를 제외
res=ff[-1:-3,]-T.ts[-1:-3,]
head(res,10)
```

```
##  [1]  36.00  22.50  27.75   3.25 -54.50 -85.75 -27.25  29.00  40.00  41.00
```

```
library(forecast)
tsdisplay(res, main="residuals by MA(4) smoothing")
```



check the stationary

ACF와 PACF를 보면 신뢰수준(표준편차 2배)을 벗어나는 값이 다수 존재하므로 정상성이 존재하지 않는다고 판단된다.

the test the independence assumption

```
Box.test(res)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 47.341, df = 1, p-value = 5.964e-12
```

Box.test 결과 p-value가 유의수준 0.05충분히 크므로 잔차들간의 독립성이 존재한다고 볼 수 있다. # Carefully interpret the residual analysis. MA(4)에 대한 잔차분석 결과, 정상성은 존재하지 않으며 잔차들 끼리의 독립성은 존재한다고 판단할 수 있다. ## SOL4 # Fit the simple exponential smoothing with alpha=0.1 and with the optimized alpha.

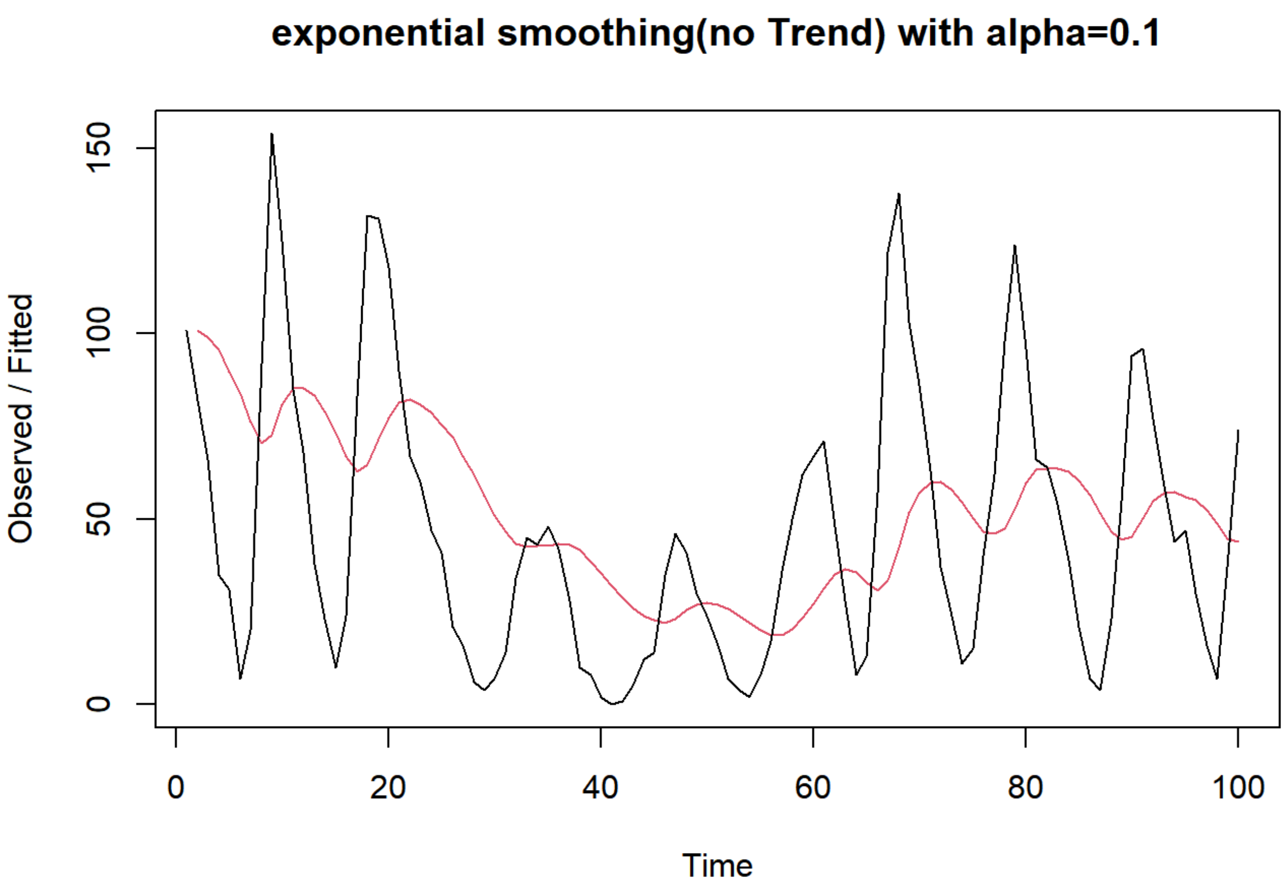
```
alpha=0.1 #exponential
 ho=HoltWinters(T.ts, alpha=0.1, beta=F, gamma=F)#exponential smoothing#(beta=F, gamma=F): no trend and no seasonal effect
 ho
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = T.ts, alpha = 0.1, beta = F, gamma = F)
##
## Smoothing parameters:
##  alpha: 0.1
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 46.92938
```

```
head(ho$fitted) #fitting value
```

```
## Time Series:
## Start = 2
## End = 7
## Frequency = 1
##          xhat     level
## 2 101.00000 101.00000
## 3  99.10000  99.10000
## 4  95.79000  95.79000
## 5  89.71100  89.71100
## 6  83.83990  83.83990
## 7  76.15591  76.15591
```

```
plot(ho,main = "exponential smoothing(no Trend) with alpha=0.1")
```



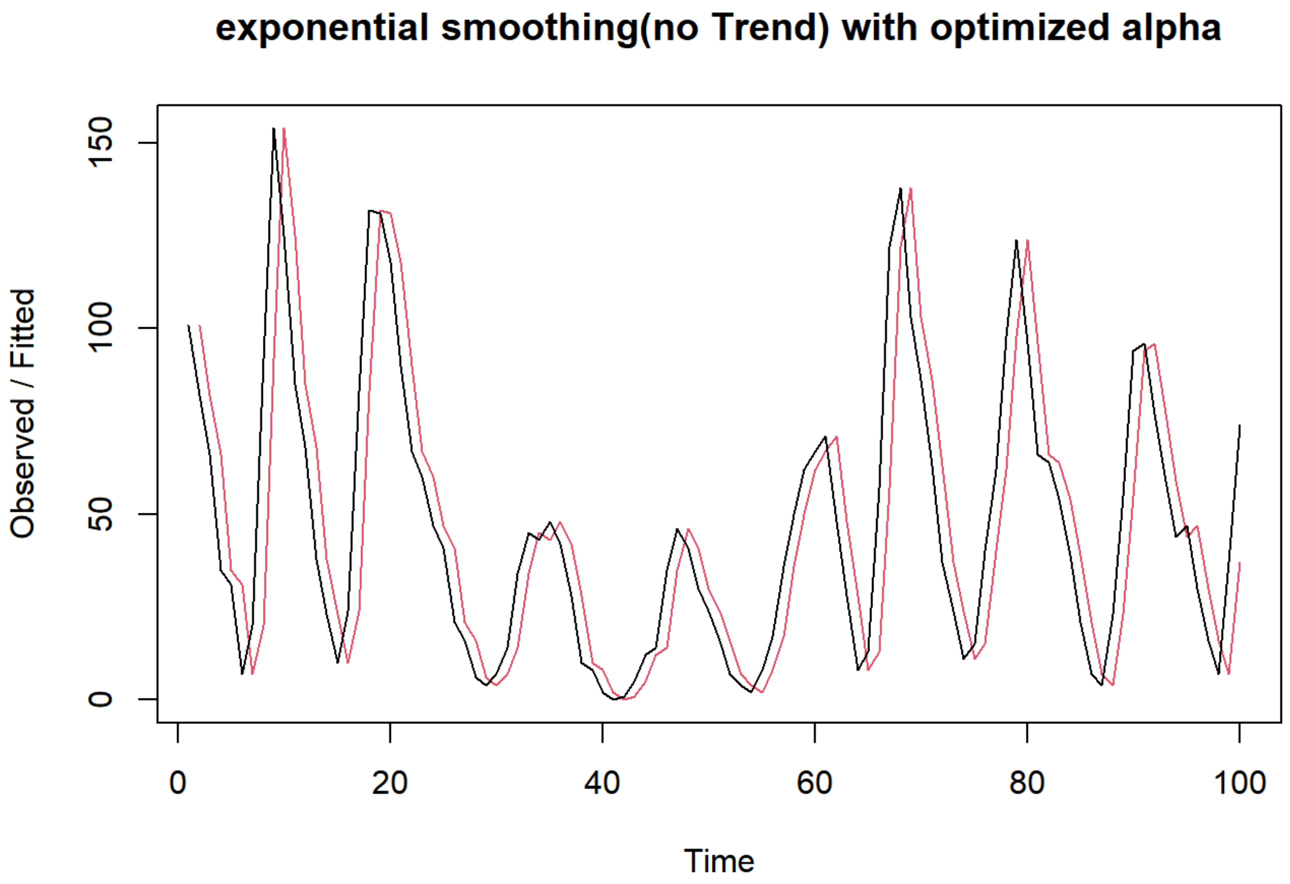

```
ha=HoltWinters(T.ts,beta=F,gamma=F) #exponential smoothing
ha
```

```
## Holt-Winters exponential smoothing without trend and without seasonal compo
nent.
##
## Call:
## HoltWinters(x = T.ts, beta = F, gamma = F)
##
## Smoothing parameters:
##  alpha: 0.9999339
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 73.99755
```

```
head(ha$fitted) #fitting value
```

```
## Time Series:
## Start = 2
## End = 7
## Frequency = 1
##         xhat      level
## 2 101.000000 101.000000
## 3  82.001256  82.001256
## 4  66.001058  66.001058
## 5  35.002049  35.002049
## 6  31.000265  31.000265
## 7   7.001587   7.001587
```

```
plot(ha,main="exponential smoothing(no Trend) with optimized alpha") #the red
line is the fitted value
```



If you think we need a trend, or seasonal, or both try them

처음 1번에서 Ts데이터는 Trend가 고려될 경우에 정상을 나타낸다. 그러므로 계절성과 추세를 고려한 모델링을 해본다.

또한, 계절성을 고려한 지수 평활법을 사용하기 위해서 frequency값을 4로 임의로 설정하고 진행한다.

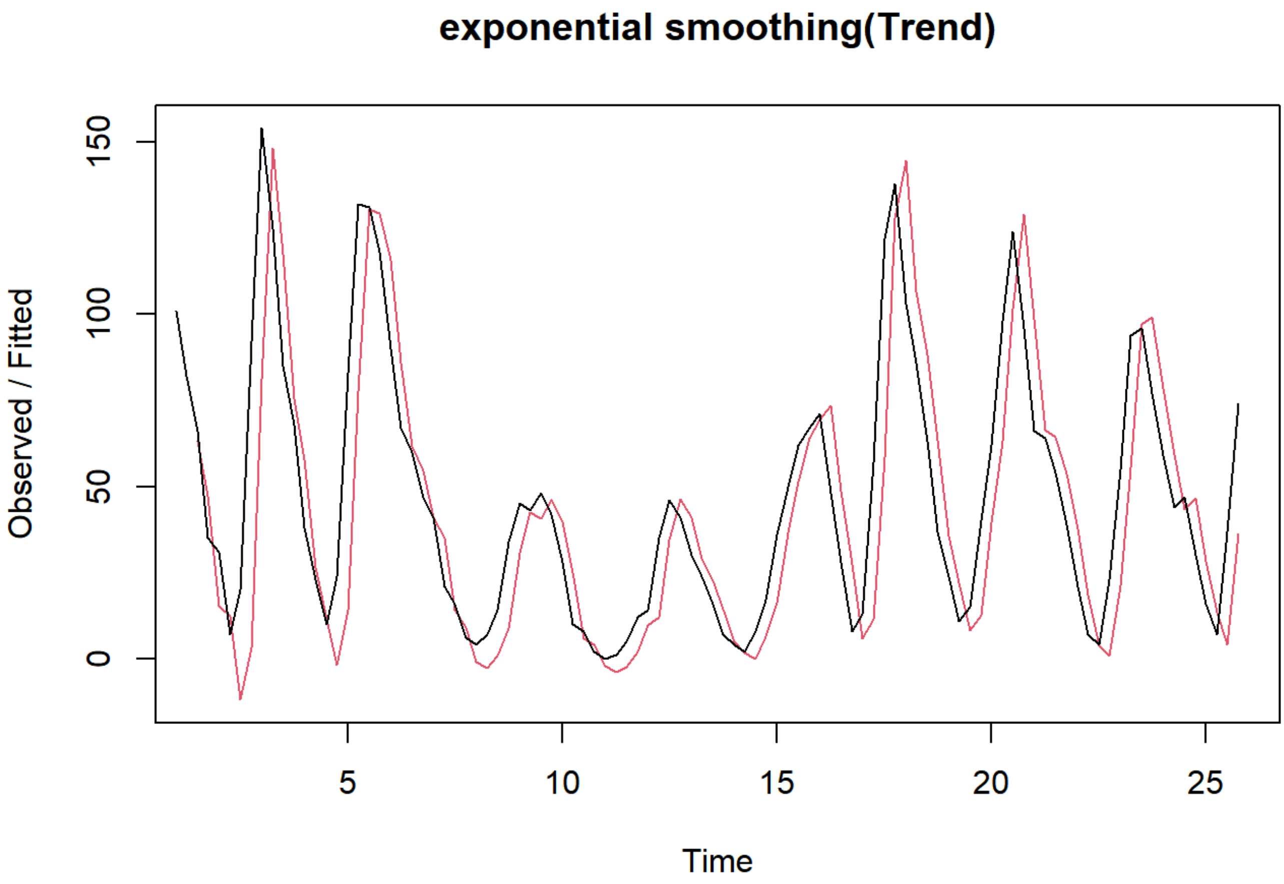
```
T.ts=ts(data=T,frequency=4)
Ta=HoltWinters(T.ts,gamma=F) #exponential smoothing(trend)
Ta
```

```
## Holt-Winters exponential smoothing with trend and without seasonal componen
t.
##
## Call:
## HoltWinters(x = T.ts, gamma = F)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0.0678801
##  gamma: FALSE
##
## Coefficients:
##        [,1]
## a 74.000000
## b  1.870588
```

```
head(Ta$fitted)
```

```
##            xhat level     trend
## 1 Q3  63.000000    82 -19.00000
## 1 Q4  47.203640    66 -18.79636
## 2 Q1  15.375256    35 -19.62474
## 2 Q2  12.435865    31 -18.56413
## 2 Q3 -11.933122     7 -18.93312
## 2 Q4   3.234502    20 -16.76550
```

```
plot(Ta,main="exponential smoothing(Trend)")
```



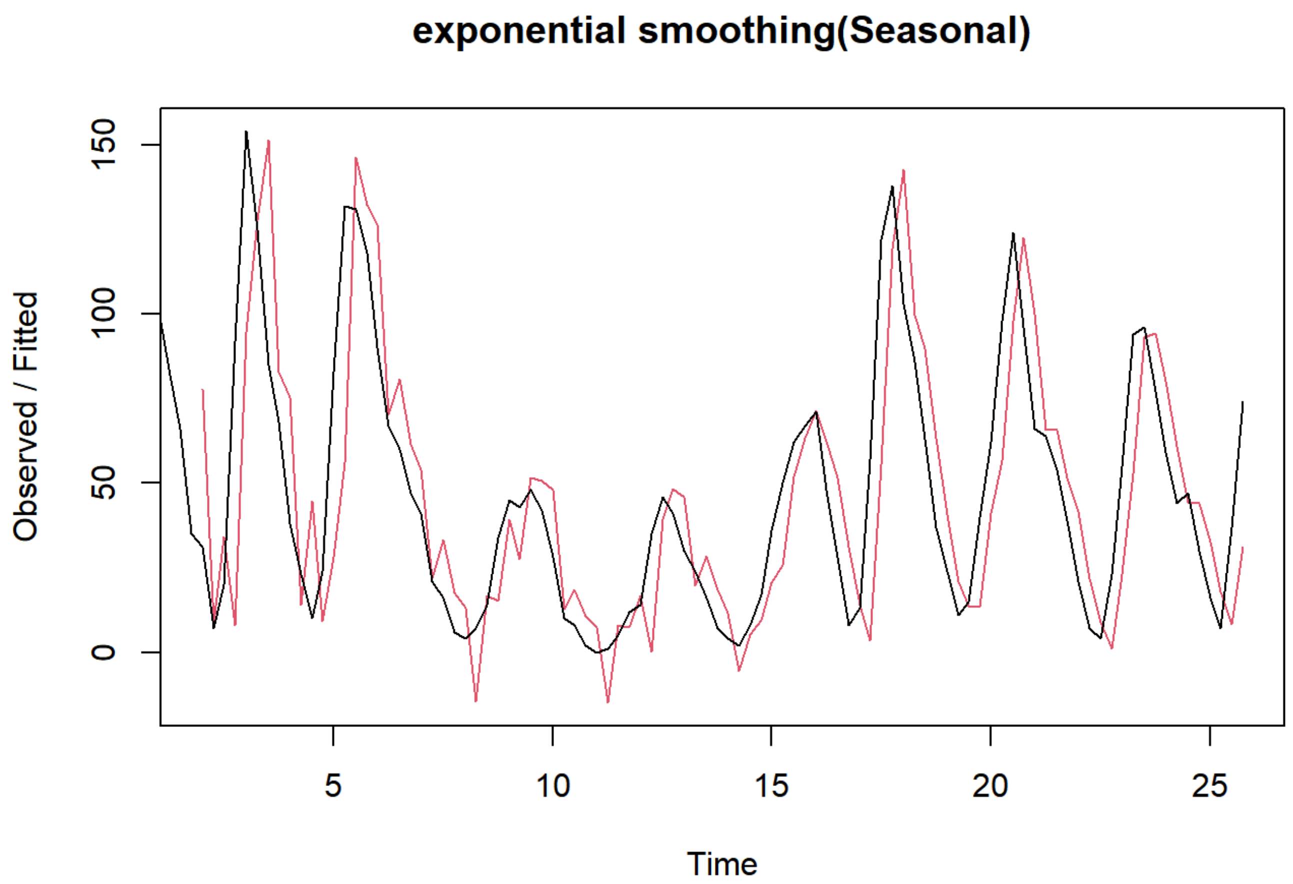

```
S.ts=ts(data=T,frequency=4)
Sa=HoltWinters(S.ts,beta=F) #exponential smoothing(Seasonal)
Sa
```

```
## Holt-Winters exponential smoothing without trend and with additive seasonal
component.
##
## Call:
## HoltWinters(x = S.ts, beta = F)
##
## Smoothing parameters:
##  alpha: 0.9350651
##  beta : FALSE
##  gamma: 1
##
## Coefficients:
##          [,1]
## a  74.3403045
## s1 -1.5697735
## s2 -0.3844468
## s3  2.6566979
## s4 -0.3403045
```

```
head(Sa$fitted)
```

```
##              xhat     level      season
## 2 Q1  77.812500  69.12500   8.687500
## 2 Q2   8.664767  25.35227 -16.687500
## 2 Q3  34.233102  23.79560  10.437500
## 2 Q4   8.049226  10.48673  -2.437500
## 3 Q1  94.633894  88.98616   5.647733
## 3 Q2 127.701731 144.49733 -16.795602
```

```
plot(Sa,main="exponential smoothing(Seasonal)")
```



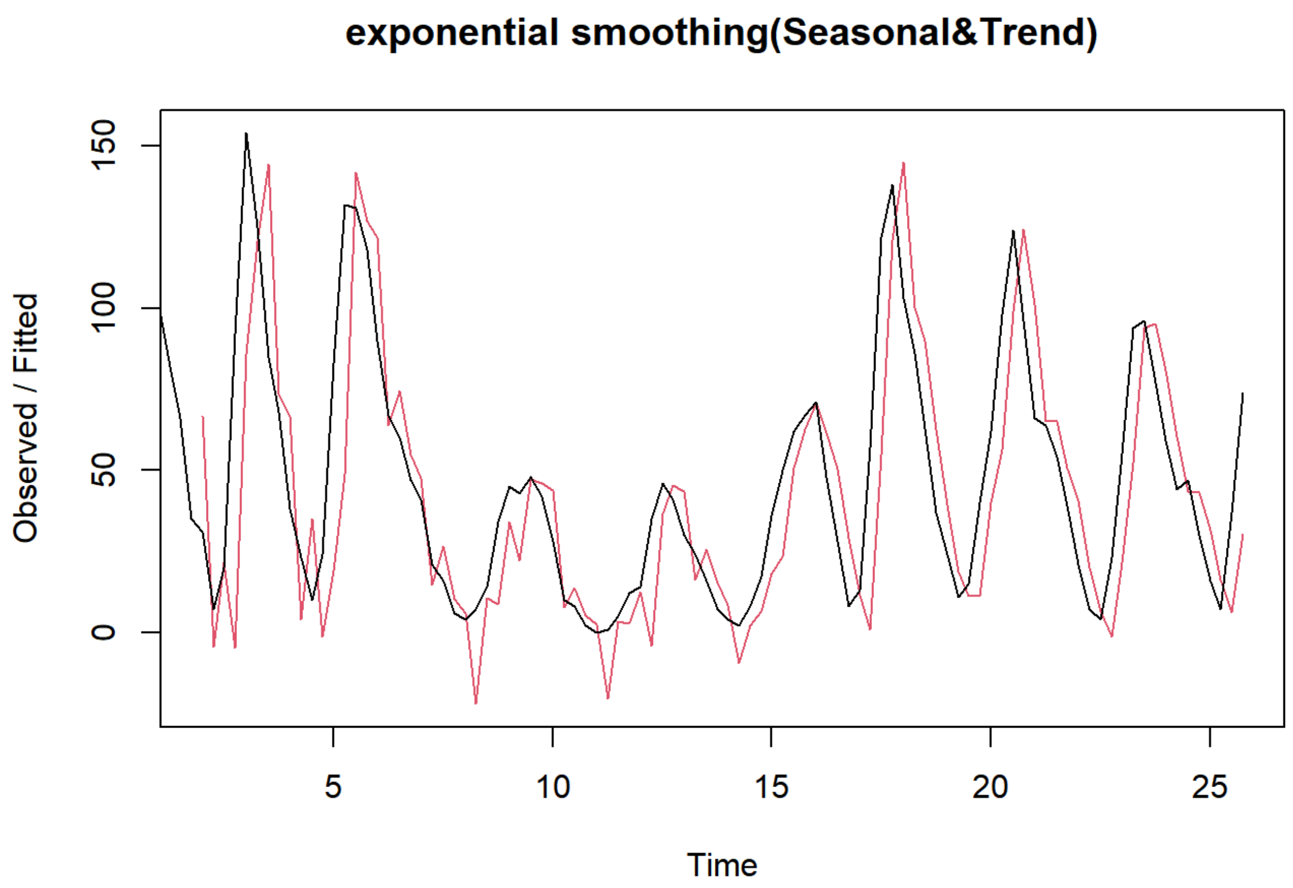

```
S.ts=ts(data=T,frequency=4)
TSa=HoltWinters(S.ts) #exponential smoothing(Trend&Seasonal)
TSa
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal compone
nt.
##
## Call:
## HoltWinters(x = S.ts)
##
## Smoothing parameters:
##  alpha: 0.9370146
##  beta : 0.03377394
##  gamma: 1
##
## Coefficients:
##          [,1]
## a  68.8328871
## b   0.4755106
## s1  4.2076887
## s2  5.3628840
## s3  8.2509240
## s4  5.1671129
```

```
head(TSa$fitted)
```

```
##             xhat      level      trend     season
## 2 Q1  66.737500  69.125000 -11.075000   8.687500
## 2 Q2  -4.330034  24.563439 -12.205973 -16.687500
## 2 Q3  21.563959  22.973874 -11.847415  10.437500
## 2 Q4  -4.673403   9.661006 -11.896909  -2.437500
## 3 Q1  85.947534  88.348491  -8.837517   6.436561
## 3 Q2 120.619373 143.277130  -6.683883 -15.973874
```

```
plot(TSa,main="exponential smoothing(Seasonal&Trend)")
```



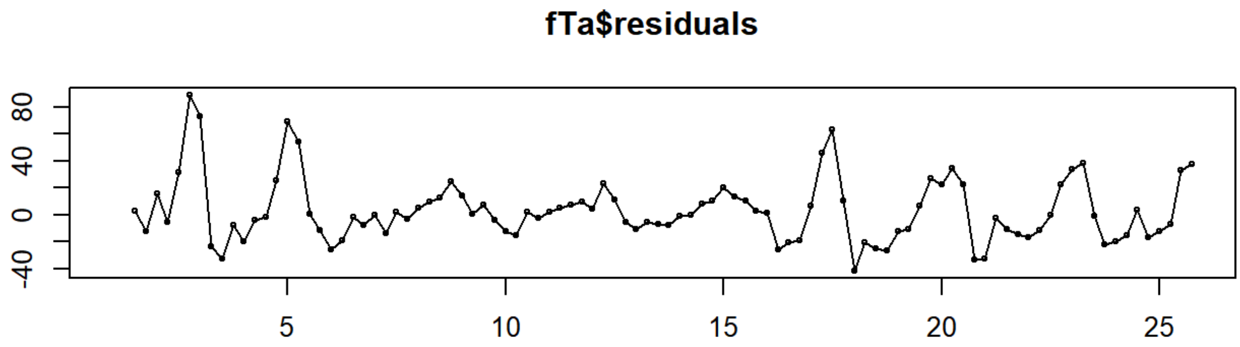
Please address all the moeling and show how you find the best exponential smoothing model for spot data.

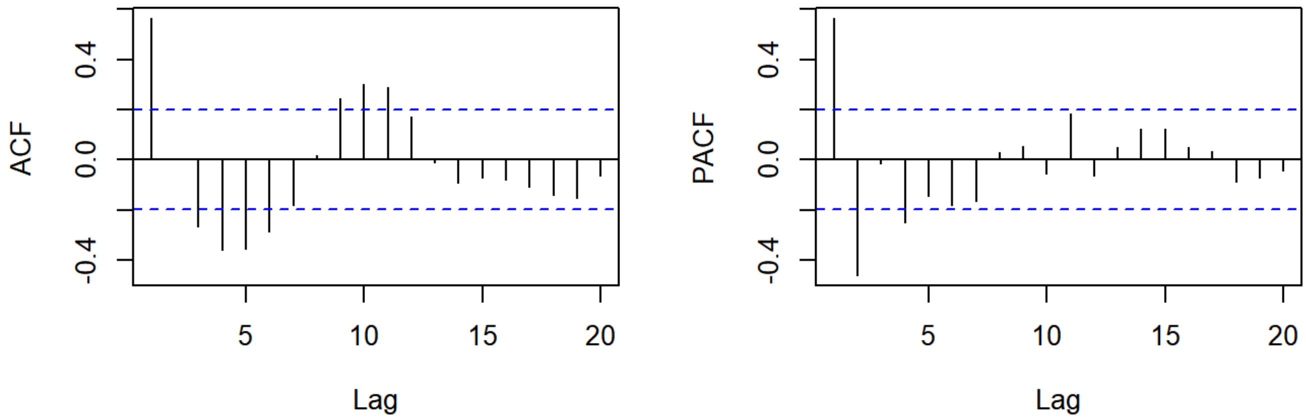
RMSE를 비교하여 최적의 지수 평활법 모델을 선정한다.

```
library(tseries)
library(forecast)
fTa=forecast(Ta)
names(fTa)
```

```
## [1] "method"    "model"     "level"     "mean"      "lower"     "upper"
## [7] "x"         "series"    "fitted"    "residuals"
```

```
tsdisplay(fTa$residuals)
```






```
Box.test(fTa$residuals,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  fTa$residuals
## X-squared = 30.996, df = 1, p-value = 2.585e-08
```

```
tseries::kpss.test(fTa$residuals,null="Level")
```

```
## Warning in tseries::kpss.test(fTa$residuals, null = "Level"): p-value great
er
## than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  fTa$residuals
## KPSS Level = 0.12067, Truncation lag parameter = 4, p-value = 0.1
```

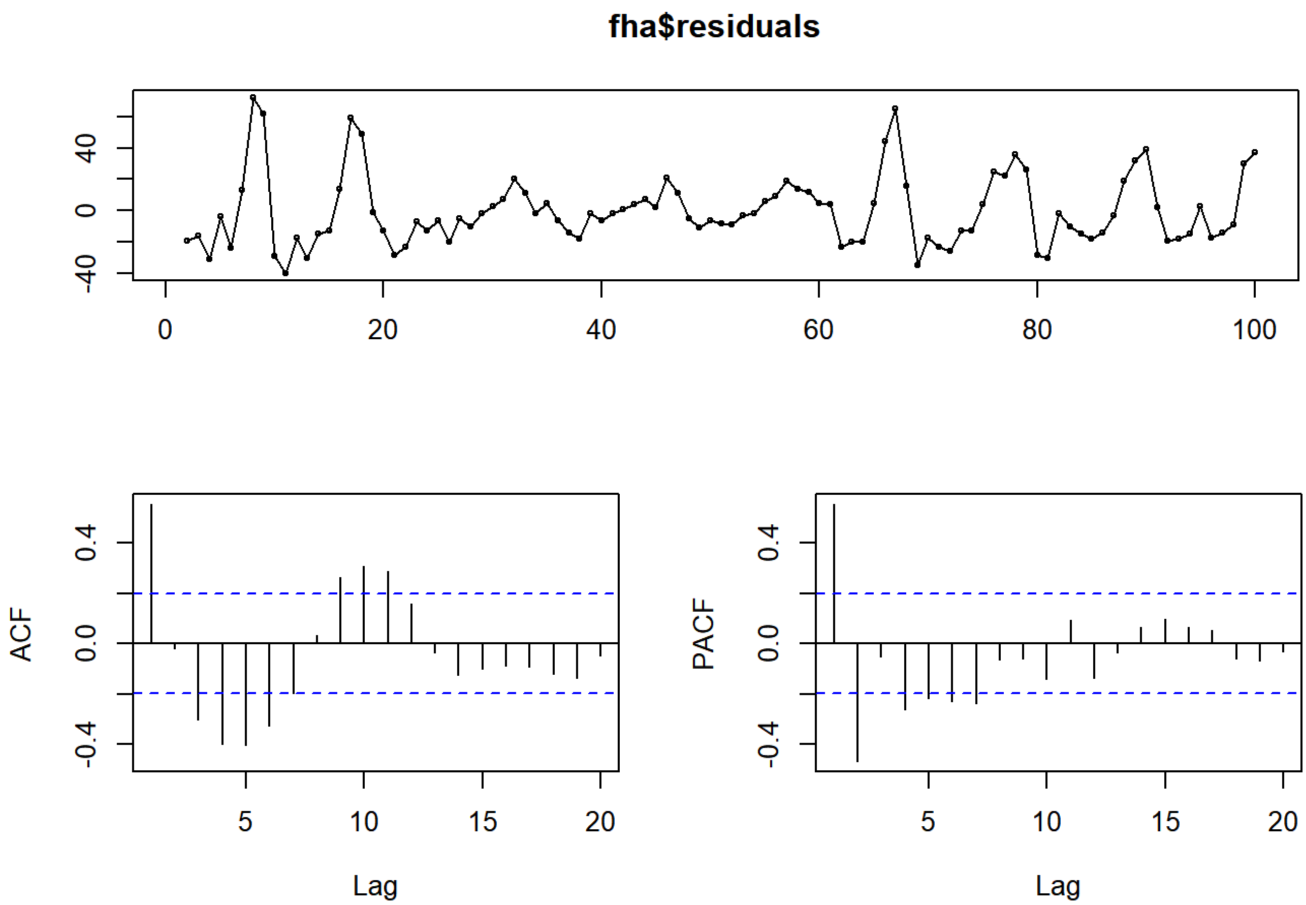
```
accuracy(fTa)
```

```
##                    ME     RMSE      MAE MPE MAPE      MASE      ACF1
## Training set 3.137373 23.61944 16.89736 Inf  Inf 0.3529475 0.5623951
```

```
fha=forecast(ha)
names(fha)
```

```
##  [1] "method"    "model"     "level"     "mean"      "lower"     "upper"
##  [7] "x"         "series"    "fitted"    "residuals"
```

```
tsdisplay(fha$residuals)
```




```
Box.test(fha$residuals,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  fha$residuals
## X-squared = 30.077, df = 1, p-value = 4.153e-08
```

```
tseries::kpss.test(fha$residuals,null="Level")
```

```
## Warning in tseries::kpss.test(fha$residuals, null = "Level"): p-value great
er
## than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  fha$residuals
## KPSS Level = 0.042049, Truncation lag parameter = 4, p-value = 0.1
```

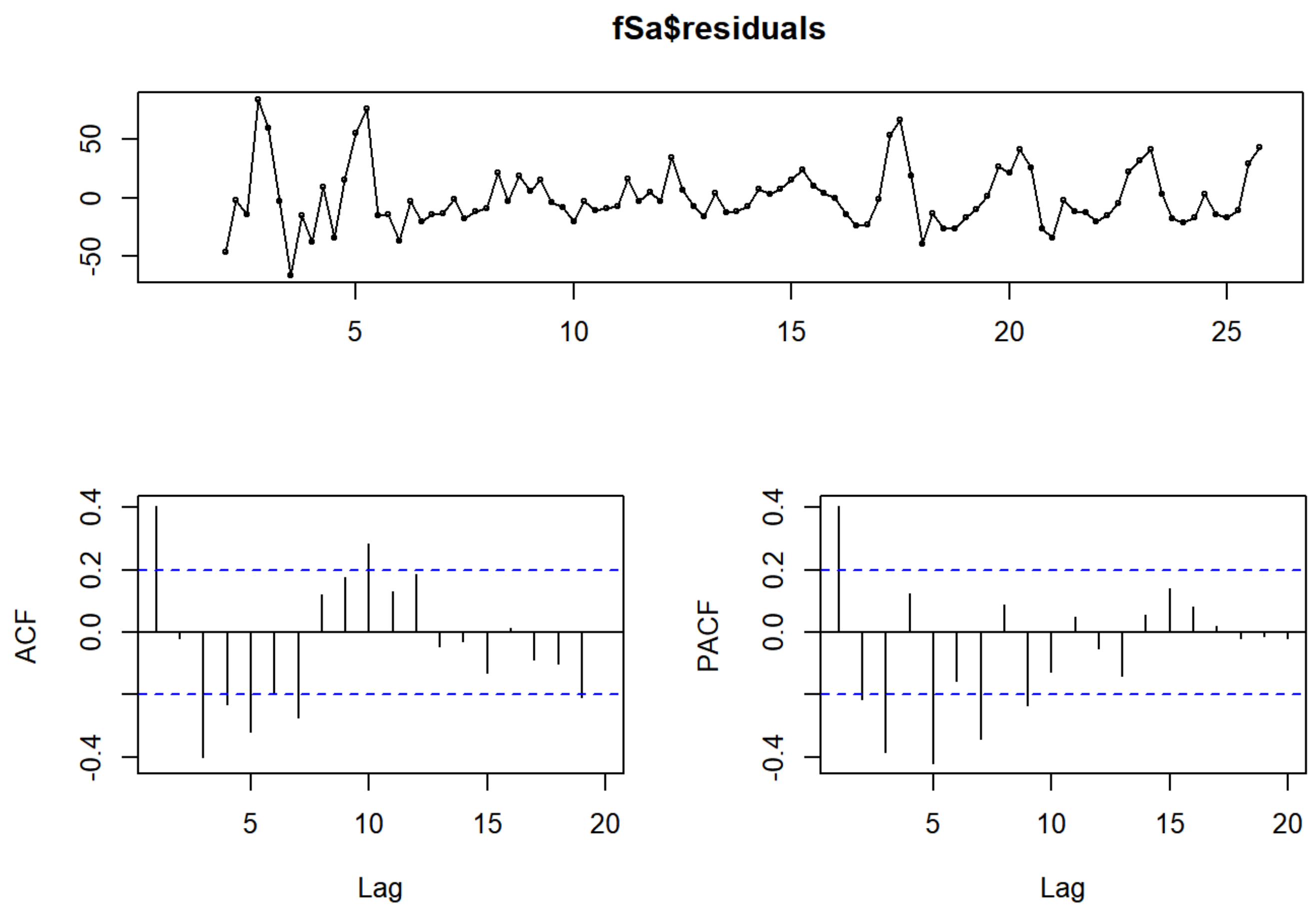
```
accuracy(fha)
```

```
##                    ME     RMSE      MAE  MPE MAPE     MASE      ACF1
## Training set -0.27277 22.45279 17.14213 -Inf  Inf 1.000042 0.5511841
```

```
fSa=forecast(Sa)
names(fSa)
```

```
##  [1] "method"    "model"     "level"     "mean"      "lower"     "upper"
##  [7] "x"         "series"    "fitted"    "residuals"
```

```
tsdisplay(fSa$residuals)
```




```
Box.test(fSa$residuals,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  fSa$residuals
## X-squared = 15.538, df = 1, p-value = 8.085e-05
```

```
tseries::kpss.test(fSa$residuals,null="Level")
```

```
## Warning in tseries::kpss.test(fSa$residuals, null = "Level"): p-value great
er
## than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  fSa$residuals
## KPSS Level = 0.02891, Truncation lag parameter = 4, p-value = 0.1
```

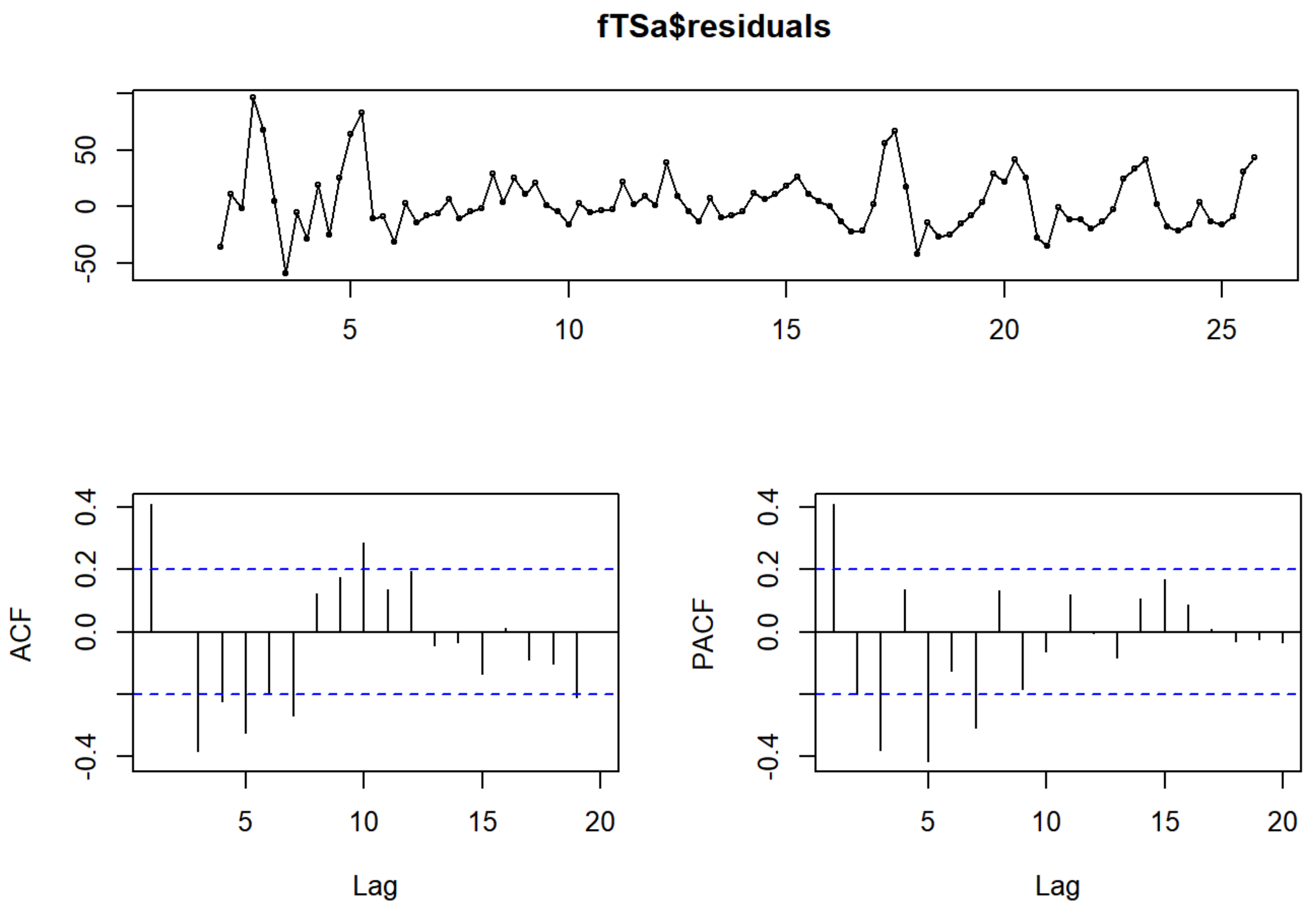
```
accuracy(fSa)
```

```
##                       ME     RMSE      MAE  MPE MAPE      MASE      ACF1
## Training set 0.05809873 25.79089 19.20619 -Inf  Inf 0.4011737 0.402316
```

```
fTSa=forecast(TSa)
names(fTSa)
```

```
##  [1] "method"    "model"     "level"     "mean"      "lower"     "upper"
##  [7] "x"         "series"    "fitted"    "residuals"
```

```
tsdisplay(fTSa$residuals)
```




```
Box.test(fTSa$residuals,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  fTSa$residuals
## X-squared = 16.099, df = 1, p-value = 6.01e-05
```

```
tseries::kpss.test(fTSa$residuals,null="Level")
```

```
## Warning in tseries::kpss.test(fTSa$residuals, null = "Level"): p-value grea
ter
## than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  fTSa$residuals
## KPSS Level = 0.065248, Truncation lag parameter = 4, p-value = 0.1
```

```
accuracy(fTSa)
```

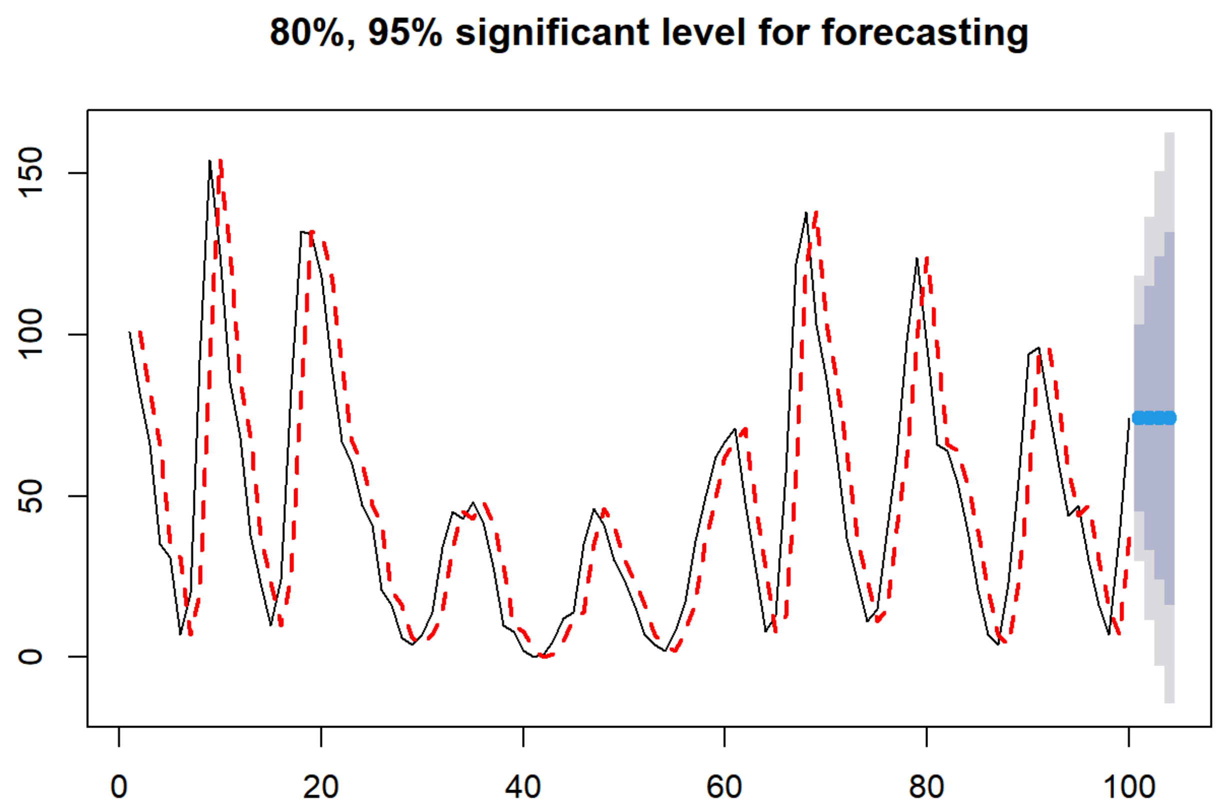
```
##                   ME     RMSE      MAE  MPE MAPE      MASE      ACF1
## Training set 3.80191 26.57515 19.03591 -Inf  Inf 0.3976169 0.4095143
```

잔차분석 결과 model 전부 비정상성을 가진다고 판단된다. 하지만, 모델중에서 계절성과 추세를 고려하지 않은 최적의 alpha 지수 평활법 모댈이 가장 작은 RMSE를 보이므로 최적의 모델로 선정한다.

From your best model, find the forecast of next 4 points.

MA(이동평균법)의 경우 예측의 목적 보다는, 분해법에서 ’계절조정’에 주로 사용된다. 그러므로 지수 평활법 중 최적 모델인 “Exponential smoothing(no Trend) with optimized alpha”로 예측을 실시한다.

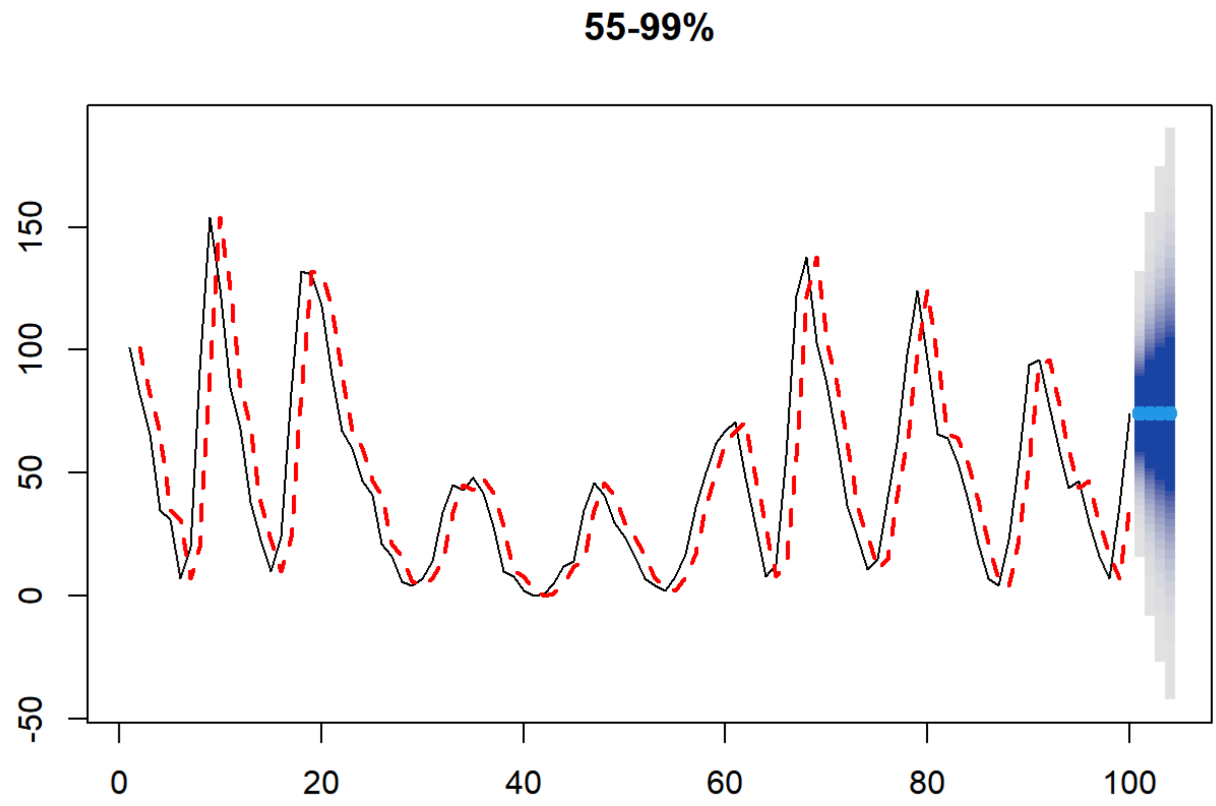
```
F=forecast(ha, h=4)
plot(F,main="80%, 95% significant level for forecasting")
lines(F$fitted, col="red", lty=2, lwd=2)
```




```
FF=forecast(ha, h=4,fan=T)
```

```
## Warning in if (fan) {: length > 1 이라는 조건이 있고, 첫번째 요소만이 사용
될 것
## 입니다
```

```
plot(FF,main="55-99%")
lines(FF$fitted,col="red", lty=2, lwd=2)
```


55-99%
150
100
50
0
-50
0
20
40
60
80
100