MA585-HW7

Ranfei Xu

2022/3/29

1

```r
knitr::include_graphics("hw7-1.jpg")
```

1. Suppose an AR(1) is fitted to a time series data of length $n = 144$ and the estimated values of the parameters are $\mu = 99, \phi = 0.6$ and $\sigma = 1$. Assume the last three values of the time series are $X_{144} = 100, X_{143} = 100$, $X_{142} = 99$. Compute the forecasts and 95% forecast intervals for the next four values.

for MA(1): $X_t = \mu + \phi(X_{t-1} - \mu) + e_t$, we have

$\hat{X}_n(1) = E(X_{n+1} \mid X_n, X_{n-1})$

$= E(\mu + \phi(X_n - \mu) + e_{n+1} \mid X_n, X_{n-1}, \cdots)$

$= \mu + \phi(X_n - \mu)$

$\hat{X}_n(2) = E(X_{n+2} \mid X_n, X_{n-1}, \cdots)$

$= E(\mu + \phi(X_{n+1} - \mu) + e_{n+2} \mid X_n, X_{n-1}, \cdots)$

$= \mu + \phi(\hat{X}_n(1) - \mu)$

$= \mu + \phi^2(X_n - \mu)$

……

$\hat{X}_n(l) = \mu + \phi^l(X_n - \mu)$

$\hat{X}_{145} = \hat{X}_{144}(1) = \mu + \phi(X_{144} - \mu) = 99 + 0.6 \times (100 - 99) = 99.6$

$\hat{X}_{146} = \hat{X}_{144}(2) = \mu + \phi^2(X_{144} - \mu) = 99 + 0.6^2 = 99.36$

$\hat{X}_{147} = \hat{X}_{144}(3) = \mu + \phi^3(X_{144} - \mu) = 99 + 0.6^3 = 99.216$

$\hat{X}_{148} = \hat{X}_{144}(4) = \mu + \phi^4(X_{144} - \mu) = 99 + 0.6^4 = 99.1296$

For forecast interval we have $\hat{X}_n(l) = 1.96\sigma\sqrt{\frac{1-\phi^{2l}}{1-\phi^2}}$

so the forcast intervals are

$X_{145} = \hat{X}_{145} \pm 1.96\sigma = (97.64, 101.56)$

$X_{146} = \hat{X}_{146} \pm 1.96\sigma\sqrt{\frac{1-\phi^4}{1-\phi^2}} = 99.36 \pm 1.96 \times \sqrt{1.36} = (97.07, 101.65)$

$X_{147} = \hat{X}_{147} \pm 1.96\sigma\sqrt{\frac{1-\phi^6}{1-\phi^2}} = 99.216 \pm 1.96 \times \sqrt{1.4896} = (96.82, 101.61)$

$X_{148} = \hat{X}_{148} \pm 1.96\sigma\sqrt{\frac{1-\phi^8}{1-\phi^2}} = 99.1296 \pm 1.96 \times \sqrt{1.536} = (96.70, 101.56)$

2

```r
knitr::include_graphics("hw7-2.jpg")
```

2. Suppose the annual sales of a company (in millions of $) follow an AR(2) model given by $X_t = 5 + 1.1X_{t-1} - 0.5X_{t-2} + e_t$ with $\sigma^2 = 2$.

   (a) If the sales of the company in 2011, 2012 and 2013 were $9 million, $11 million and $10 million respectively, forecast sales for 2014 and 2015.

   (b) Construct 95% forecast intervals for 2014 and 2015.

   (c) If the sales in 2014 turns out to be $12 million, update your forecast for 2015.

(a) forecast the sales for 2014:

$\hat{X}_{2014} = \hat{X}_{2013}(1) = E(X_{2014} \mid X_{2013}, X_{2012}, \cdots)$ ☆

$= E(5 + 1.1X_{2013} - 0.5X_{2012} + e_{2014} \mid X_{2013}, X_{2012}, \cdots)$

$= 5 + 1.1X_{2013} - 0.5X_{2012} + 0$ independent & $E(X|X) = X$

$= 5 + 1.1 \times 10 - 0.5 \times 11$

$= 10.5$

$\hat{X}_{2015} = \hat{X}_{2013}(2) = E(5 + 1.1X_{2014} - 0.5X_{2013} + e_{2015} \mid X_{2013}, X_{2012}, \cdots)$ ☆

$= 5 + 1.1E(X_{2014} \mid X_{2013}, X_{2012}, \cdots) - 0.5X_{2013} + 0$

$= 5 + 1.1\hat{X}_{2013}(1) - 0.5X_{2013}$

$= 5 + 1.1 \times 10.5 - 0.5 \times 10$

$= 11.55$

(b) 1st step: transfer AR process into $MA(\infty)$

$X_t = e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \cdots$

$= (1 + \psi_1 B + \psi_2 B^2 + \cdots)e_t$

for AR(2): $X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} = e_t$

$(1 - \phi_1 B - \phi_2 B^2)X_t = e_t$

$\because$ coefficient consistency:

$\Rightarrow (1 - \phi_1 B - \phi_2 B^2)(1 + \psi_1 B + \psi_2 B^2 + \cdots) = 1$ ☆

$\Rightarrow \begin{cases} \psi_1 = \phi_1 = 1.1 \\ \psi_2 = \phi_1\psi_1 - \phi_2\psi_0 = 0 \\ \psi_j = \phi_1\psi_{j-1} - \phi_2\psi_{j-2} \end{cases}$

2nd step: the 95% forecast interval is given by $\hat{X}_n(\ell) \pm 1.960\sqrt{\sum_{j=0}^{\ell-1}\psi_j^2}$ ☆

$\Rightarrow$ for X_{2014}: $\hat{X}_{2013}(1) \pm 1.96 \times \sqrt{2} \times \sqrt{\psi_0^2} = 10.5 \pm 2.77 = (7.73, 13.27)$

$(\ell = 1)$ $\quad \psi_0 = 1$

for X_{2015}: $\hat{X}_{2013}(2) \pm 1.96 \times \sqrt{2} \times \sqrt{\psi_0^2 + \psi_1^2} = 11.55 \pm 2.91 = (8.64, 14.41)$

$(\ell = 2)$ $\quad \psi_0 = 1, \ \psi_1 = 1.1$

(c) update of ARMA equation: $\hat{X}_{n+1}(\ell-1) = \hat{X}_n(\ell) + \psi_{\ell-1}(X_{n+1} - \hat{X}_n(1))$ ☆

$\hat{X}_{2015} = \hat{X}_{2014}(1) = \hat{X}_{2013}(2) + \psi_1(X_{2014} - \hat{X}_{2013}(1)) = 11.55 + 1.1 \times (12 - 10.5) = 13.2$

$\psi_1 = 1.1, \ X_{2014} = 12$

3

Simulate an AR(2) process with phi_1 = 1.5, phi_2 = -0.75, and mu = 100. Simulate 100 values, but set aside the last 10 values to compare forecasts to actual values.

(a) Using the first 90 observations in the series, find the MLE of the model parameters. Are the estimates comparable to the true values?

```
set.seed(585)
AR_sim = arima.sim(n=100, model = list(order = c(2,0,0), ar=c(1.5, -0.75))) + 100
train <- AR_sim[1:90]
test  <- AR_sim[91:100]
AR_auto = auto.arima(train)
summary(AR_auto)
```

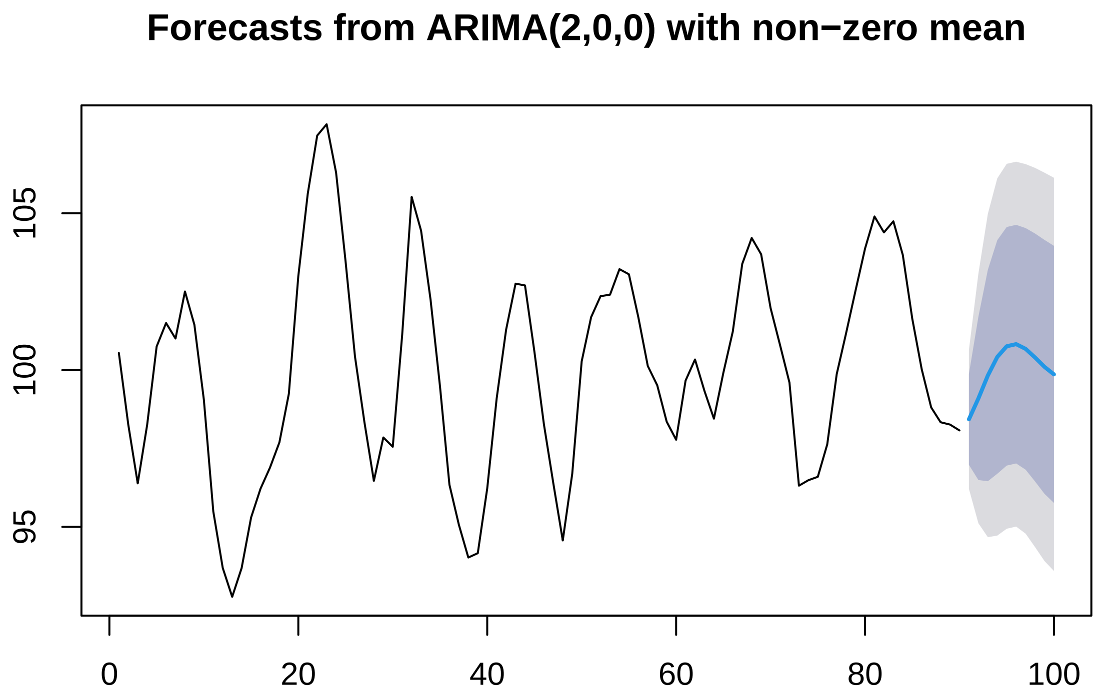
```
## Series: train
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      mean
##       1.4889  -0.7414  100.0287
## s.e.  0.0686   0.0682    0.4726
##
## sigma^2 = 1.326:  log likelihood = -140.33
## AIC=288.66   AICc=289.13   BIC=298.66
##
## Training set error measures:
##                      ME     RMSE       MAE         MPE      MAPE      MASE
## Training set -0.0109079 1.132156 0.8605699 -0.02413761 0.8627662 0.5722674
##                     ACF1
## Training set -0.03287086
```

```
# ar(train , method = "mle") # another method
```

Only use the obs. in training set, we got the result with MLE of the model parameters. And we can see the estimates are 1.4889 and -0.7414 which both not much different from the true values.

(b) Use the fitted model to forecast the 10 future values and obtain 95% forecast intervals.

```
# fit the model and apply forecast function
fit = arima(train,order = c(2,0,0))
forecast = forecast(fit, h = 10)
plot(forecast)
```




```
cat('The 95% forecasts intervals for 10 future values are \n lower:', as.numeric(forecast$lower[, '95%']
```

```
## The 95% forecasts intervals for 10 future values are
##  lower: 96.21094 95.11639 94.67225 94.72225 94.94133 95.00953 94.78456 94.35434 93.91081 93.59433
##  upper: 100.6489 103.0762 104.9794 106.1136 106.576 106.6445 106.5676 106.4471 106.2944 106.1315
```

(c) What percentage of the observed values are covered by the forecast intervals?

```
df = cbind(test, forecast$lower[, '95%'], forecast$upper[, '95%']) %>% as.data.frame()
colnames(df) = c('observed' , 'lower',  'upper')
df %<>%  mutate(covered_by_CI = if_else(observed>lower & observed<upper, "TRUE", "FALSE"))
df
```

```
##     observed    lower    upper covered_by_CI
## 1   97.68991 96.21094 100.6489          TRUE
## 2   98.52554 95.11639 103.0762          TRUE
## 3   99.61145 94.67225 104.9794          TRUE
## 4  101.09501 94.72225 106.1136          TRUE
## 5  101.69769 94.94133 106.5760          TRUE
## 6   99.22366 95.00953 106.6445          TRUE
## 7   97.49167 94.78456 106.5676          TRUE
## 8   97.90582 94.35434 106.4471          TRUE
## 9   98.56417 93.91081 106.2944          TRUE
## 10  97.80984 93.59433 106.1315          TRUE
```

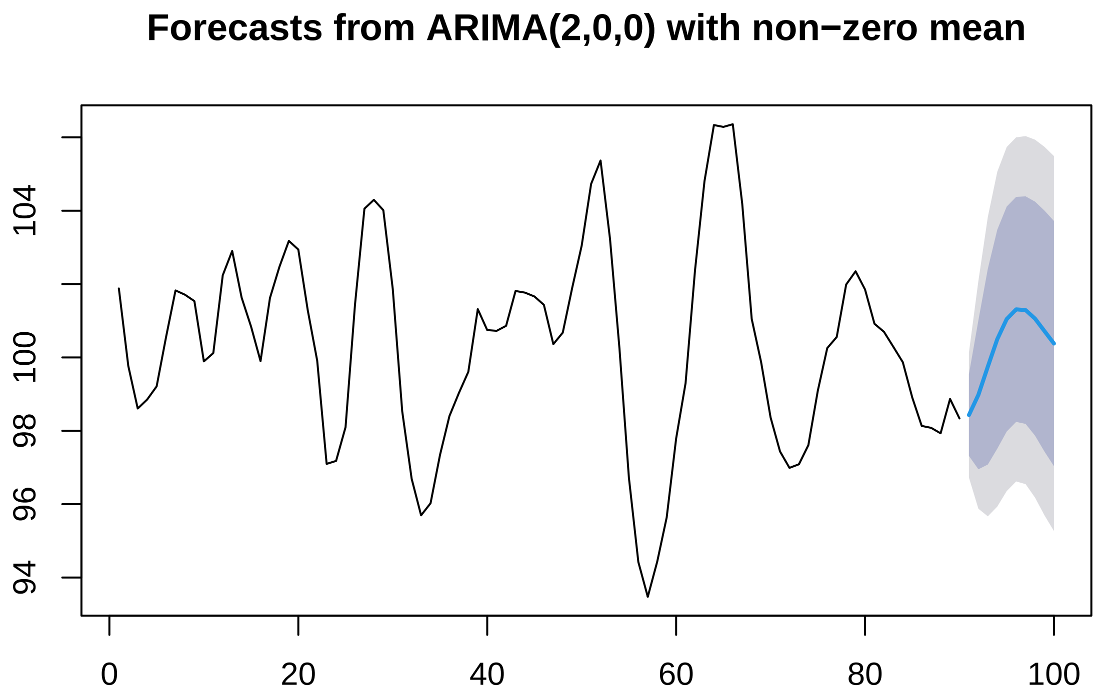
According to the table, we can see the percentage of the observed values are covered by the forecast intervals is 100%.

(d) Simulate a new sample data of the same size from the sample model and repeat steps (a),(b) and (c)

```
set.seed(1)  # simulate a new sample
AR_sim = arima.sim(n=100, model = list(order = c(2,0,0), ar=c(1.5, -0.75))) + 100
train <- AR_sim[1:90]
test  <- AR_sim[91:100]
AR_auto = auto.arima(train)
summary(AR_auto)
```

```
## Series: train
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      mean
##       1.5237  -0.7674  100.3976
## s.e.  0.0660   0.0654    0.3762
##
## sigma^2 = 0.7819:  log likelihood = -116.68
## AIC=241.35   AICc=241.82   BIC=251.35
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.01044327 0.8693758 0.6952826 -0.01801496 0.6912748 0.6260269
##                     ACF1
## Training set -0.09329523
```

```
# fit the model and apply forecast function
fit = arima(train,order = c(2,0,0))
forecast = forecast(fit, h = 10)
plot(forecast)
```




```
cat('The 95% forecasts intervals for 10 future values are \n lower:', as.numeric(forecast$lower[, '95%']
```

```
## The 95% forecasts intervals for 10 future values are
##  lower: 96.72572 95.87571 95.6684 95.9335 96.35694 96.61971 96.54595 96.17741 95.69371 95.26725
##  upper: 100.1336 102.0867 103.831 105.0612 105.7367 106.0008 106.0339 105.9362 105.7408 105.4902
```

```
df = cbind(test, forecast$lower[, '95%'], forecast$upper[, '95%']) %>% as.data.frame()
colnames(df) = c('observed' , 'lower',  'upper')
df %<>%  mutate(covered_by_CI = if_else(observed>lower & observed<upper, "TRUE", "FALSE"))
df
```

```
##    observed    lower    upper covered_by_CI
## 1  98.65899 96.72572 100.1336          TRUE
## 2  97.70001 95.87571 102.0867          TRUE
## 3  97.25479 95.66840 103.8310          TRUE
## 4  97.07890 95.93350 105.0612          TRUE
## 5  97.02516 96.35694 105.7367          TRUE
## 6  97.67167 96.61971 106.0008          TRUE
## 7  96.82428 96.54595 106.0339          TRUE
## 8  98.15924 96.17741 105.9362          TRUE
## 9  97.95569 95.69371 105.7408          TRUE
## 10 97.85056 95.26725 105.4902          TRUE
```

By setting another set.seed, we generate a new sample for the original model, and get a similar but different estimates, which are also not much different from the true values. The result of the percentage of the observed values are covered by the forecast intervals is also 100%.

4

```r
knitr::include_graphics("hw7-4.jpg")
```

4 Consider a MA(1) process given by $\mu = 5, \theta = 0.6$ and $\sigma = 0.1$. Suppose a sample realization of $n = 5$ is given by (starting from 1 ending at 5) $4.16, 5.76, 5.77, 4.02, 3.67$. Find the forecast of the sixth and seventh observations and construct 95% forecast intervals.

MA(1) : $X_t = \mu + e_t + \theta e_{t-1}$

$$\begin{cases} \hat{X}_n(1) = \mu + \theta \tilde{e}_n \\ \hat{X}_n(2) = \mu \end{cases} \qquad \mu = 5,\ \theta = 0.6.$$

t	$\hat{X}_{t-1}(1)$	$\tilde{e}_t = X_t - \hat{X}_{t-1}(l)$
0	x	0
1	$5 + 0.6 \times 0 = 5$	$4.16 - 5 = -0.84$: $\tilde{e}_1$
2	$5 + 0.6 \times (-0.84) = 4.496$	$5.76 - 4.496 = 1.264$: $\tilde{e}_2$
3	$5 + 0.6 \times 1.264 = 5.7584$	$5.77 - 5.7584 = 0.0116$: $\tilde{e}_3$
4	$5 + 0.6 \times 0.0116 \approx 5.01$	$4.02 - 5.01 = -0.99$ $\tilde{e}_4$
5	$5 + 0.6 \times (-0.99) = 4.406$	$3.67 - 4.406 \approx -0.736$ $\tilde{e}_5$

So $\hat{X}_6 = \hat{X}_5(1) = \mu + \theta \tilde{e}_5 = 5 - 0.6 \times 0.736 = 4.5584$

$\hat{X}_7 = \hat{X}_5(2) = \mu = 5$

the forecast interval should be

for X_6 : $\hat{X}_5(1) \pm 1.96\sigma = 4.5584 \pm 1.96 \times 0.1 = (4.3624, 4.7544)$

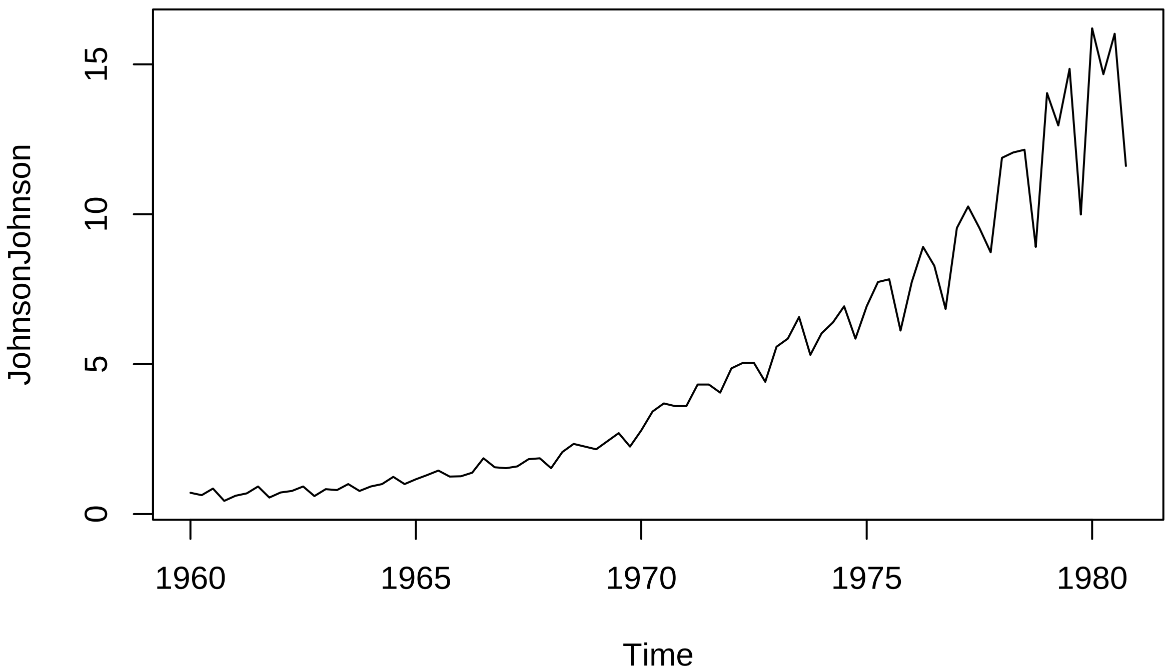
X_7 : $\hat{X}_5(2) \pm 1.96\sigma\sqrt{1+\theta^2} = 5 \pm \pm 1.96 \times 0.1 \times 1.17 = (4.7707, 5.2293)$

5

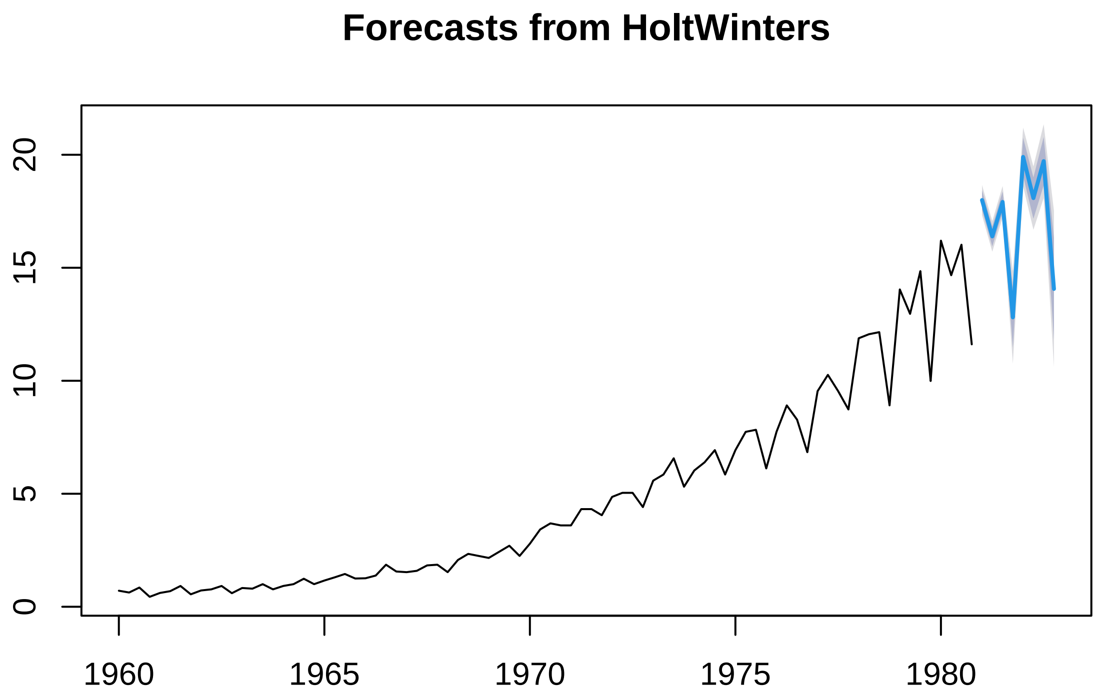
Consider the Johnson and Johnson Data from the 'HW_6'.

(a) holt-winter forecast

```r
rm(list = ls())
library(forecast)
data("JohnsonJohnson")
plot(JohnsonJohnson)
```

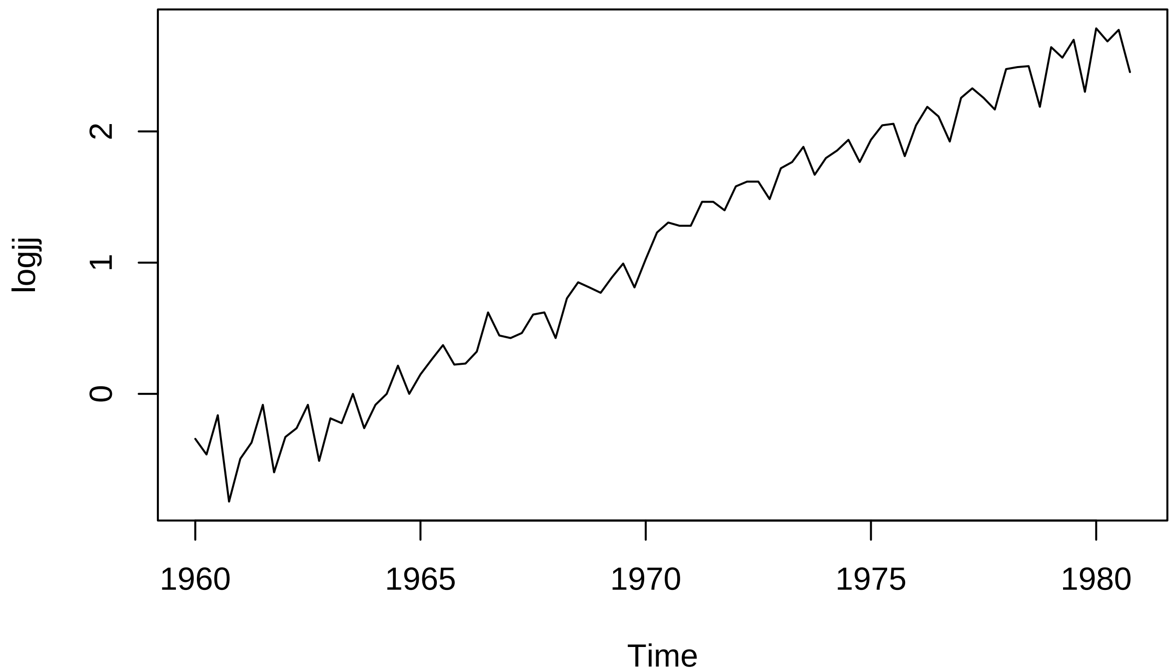



```r
fit <- HoltWinters(JohnsonJohnson, seasonal = "multiplicative")
hwfast <- forecast(fit, h=8)
# hwfast
plot(hwfast)
```



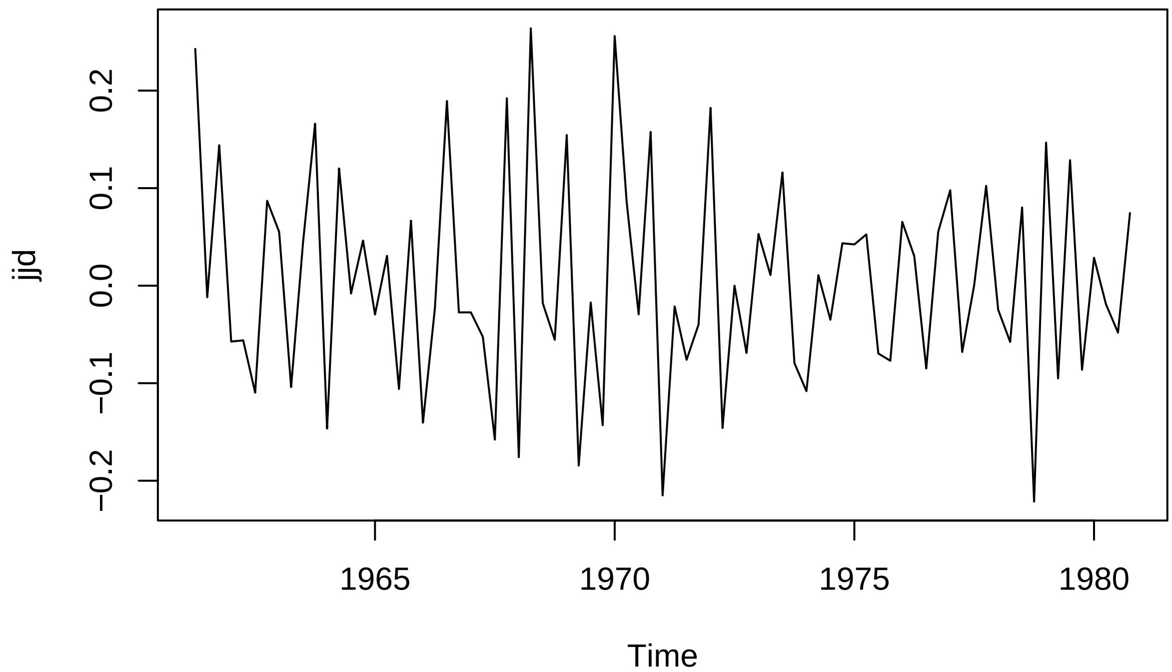
(b) identifying the ARIMA model and forecast for the next eight values

```
logjj <- log(JohnsonJohnson) # variance stabilizing transformation
plot(logjj)
```

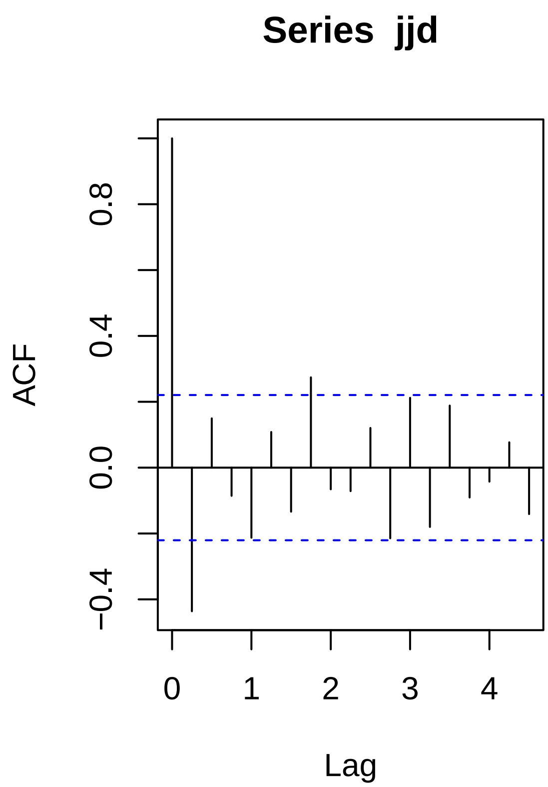


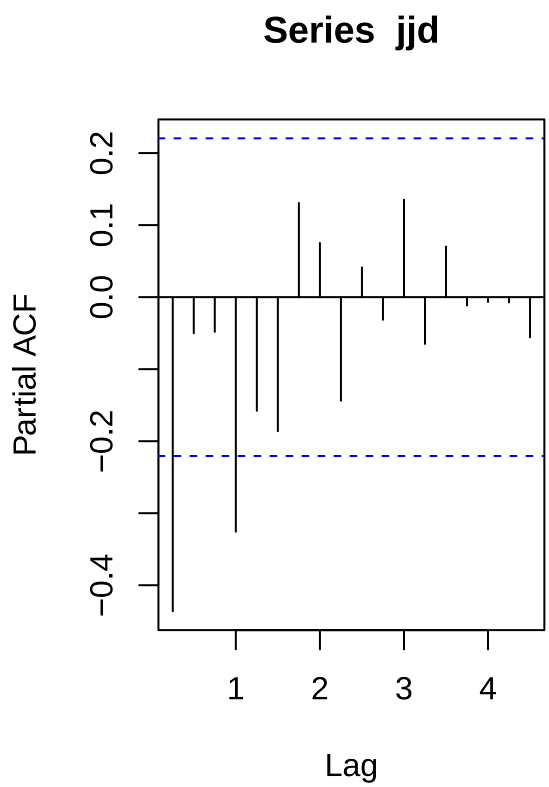

```r
# j1 <- diff(logjj, lag=4) # remove the seasonal component
# plot(j1)
# j2 <- diff(j1) # remove the trend component
# plog(j2)

jjd <- diff(diff(logjj, lag=4))
plot(jjd)
```

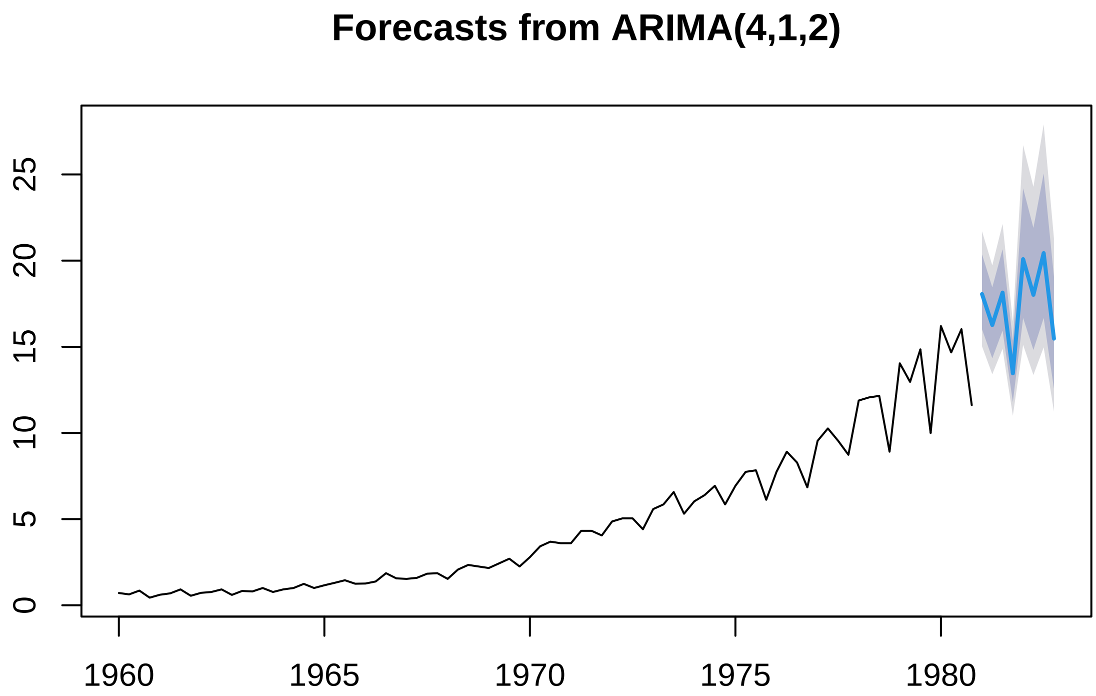



```
par(mfrow=c(1,2))
acf(jjd)
pacf(jjd)
```

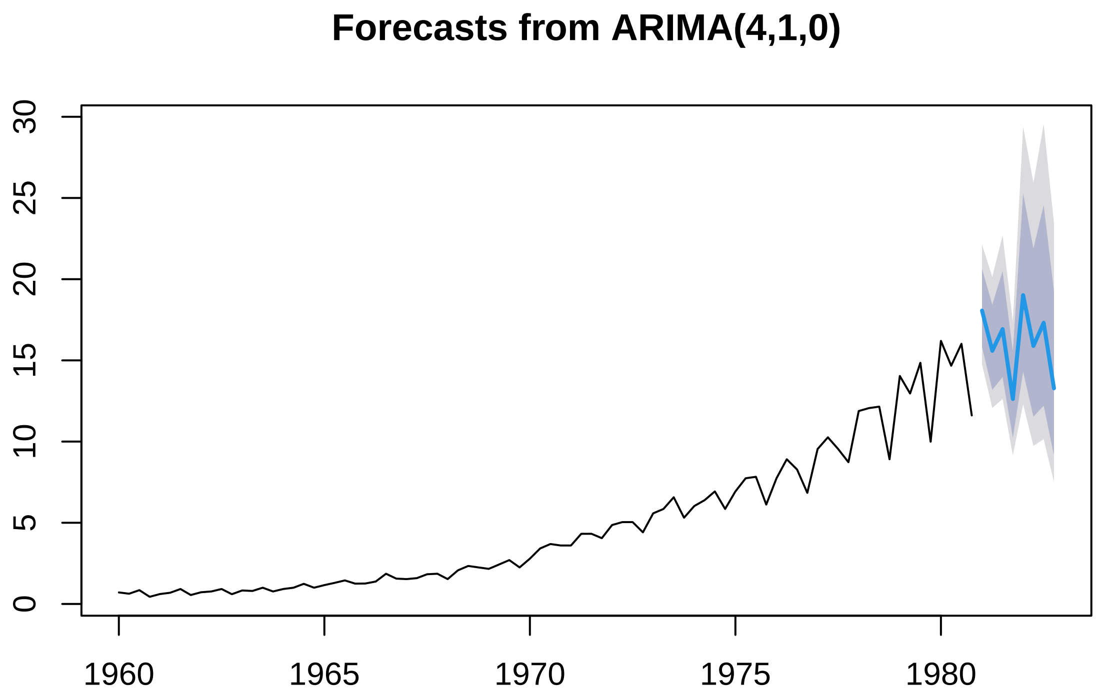





```r
# could be ARIMA(4,1,2) or ARIMA(4,1,0)
# Try ARIMA(4,1,2)
par(mfrow=c(1,1))
fit2 <- Arima(JohnsonJohnson, order = c(4,1,2), lambda = 0) # don't worry about the lambda
armafcast <- forecast(fit2, h=8)
# armafcast
plot(armafcast)
```




```
# Try ARIMA(4,1,0)
fit3 <- Arima(JohnsonJohnson, order = c(4,1,0), lambda = 0)
armafcast <- forecast(fit3, h=8)
#armafcast
plot(armafcast)
```



(c) Set aside the last eight observations in the data set as the validations sample and using the remaining data as the training sample, predict the eight observations. Compute RMSE, MAE and MAPE criteria of forecast comparison. What is your conclusion?

```
train = JohnsonJohnson[1:(length(JohnsonJohnson)-8)]
train = ts(train, frequency = 4, start = c(1960, 1))
test = JohnsonJohnson[(length(JohnsonJohnson)-8+1): length(JohnsonJohnson)]
test = ts(test, frequency = 4, start = c(1979, 1))

# ARIMA forecast performance of JJ Data
# Try ARIMA(4,1,2)
fit3 <- Arima(train, order = c(4,1,2), lambda = 0) # don't worry about the lambda
arimafcast <- forecast(fit3, h=8)
err = test - arimafcast$mean  # errors
mae = mean(abs(err))  # mean absolute error
rmse = sqrt(mean(err^2)) # root mean square error
mape = mean (abs(err/test*100)) # mean absolute percentage error
cat('\n(c):The RMSE, MAE and MAPE criteria of ARIMA(4,1,2) forecast')
```

```
##
## (c):The RMSE, MAE and MAPE criteria of ARIMA(4,1,2) forecast
```

```
cat('\nMAE:', mae)
```

```
##
## MAE: 0.7063004
```

```r
cat('\nRMSE:', rmse)
```

```
##
## RMSE: 0.8387923
```

```r
cat('\nMAPE:', mape)
```

```
##
## MAPE: 5.103268
```

```r
# Try ARIMA(4,1,0)
fit4 <- Arima(train, order = c(4,1,0), lambda = 0) # don't worry about the lambda
arimafcast <- forecast(fit4, h=8)
err = test - arimafcast$mean  # errors
mae = mean(abs(err))  # mean absolute error
rmse = sqrt(mean(err^2)) # root mean square error
mape = mean (abs(err/test*100)) # mean absolute percentage error
cat('\n(c):The RMSE, MAE and MAPE criteria of ARIMA(4,1,0) forecast')
```

```
##
## (c):The RMSE, MAE and MAPE criteria of ARIMA(4,1,0) forecast
```

```r
cat('\nMAE:', mae)
```

```
##
## MAE: 2.609401
```

```r
cat('\nRMSE:', rmse)
```

```
##
## RMSE: 2.859139
```

```r
cat('\nMAPE:', mape)
```

```
##
## MAPE: 18.19924
```