Economic Forecasting

Forecasting with non-stationary ARIMA models

Sebastian Fossati

University of Alberta | E493 | 2023

Outline

Forecasting with an $I(1)$ process

Consider an integrated process of order one ($I(1)$). Since $y'_t = y_t - y_{t-1}$, the level y_t may be represented as

$$y_t = y_{t-1} + y'_t$$

Similarly, the level at time $t + h$ may be represented as

$$y_{t+h} = y_t + y'_{t+1} + \cdots + y'_{t+h}$$

Forecasting with an $I(1)$ process

Forecasting from an $I(1)$ process follows directly from writing y_{T+h} as

$$y_{T+h} = y_T + y'_{T+1} + \cdots + y'_{T+h}$$

Then

$$\begin{aligned} y_{T+h|T} &= y_T + y'_{T+1|T} + y'_{T+2|T} + \cdots + y'_{T+h|T} \\ &= y_T + \sum_{s=1}^{h} y'_{T+s|T} \end{aligned}$$

Notice that forecasting an I(1) process proceeds from the most recent observation.

Computing point forecasts: ARIMA(1,1,0)

ARIMA(1,1,0) forecasts

$$(1 - \phi_1 B)(y'_t - \mu) = \varepsilon_t, \quad \varepsilon_t \sim \mathrm{N}(0, \sigma^2)$$

Assume μ, ϕ_1, and σ^2 are known.

Using the chain-rule of forecasting, the h-step ahead forecast of y'_{T+h} based on information at time T is

$$y'_{T+h|T} = \mu + \phi_1^h (y'_T - \mu)$$

Computing point forecasts: ARIMA(1,1,0)

ARIMA(1,1,0) forecasts

$$(1 - \phi_1 B)(y'_t - \mu) = \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}(0, \sigma^2)$$

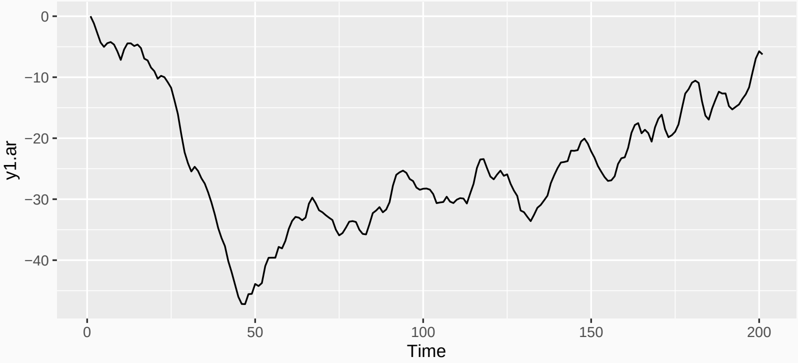
Assume μ, ϕ_1, and σ^2 are known.

Then, the h-step ahead forecast of y_{T+h} is

$$\begin{aligned} y_{T+h|T} &= y_T + \sum_{s=1}^{h} \left[\mu + \phi_1^s (y'_T - \mu)\right] \\ &= y_T + h\mu + (y'_T - \mu) \sum_{s=1}^{h} \phi_1^s \end{aligned}$$

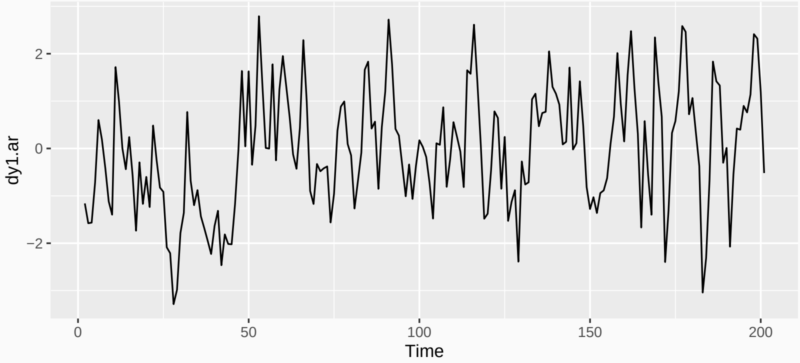
Example: ARIMA(1,1,0)

```r
# simulate ARIMA(1,1,0)
y1.ar <- arima.sim(list(order = c(1,1,0), ar = 0.5), n = 200)
autoplot(y1.ar)
```



Example: ARIMA(1,1,0)

```
# plot stationary first difference
dy1.ar <- diff(y1.ar)
autoplot(dy1.ar)
```



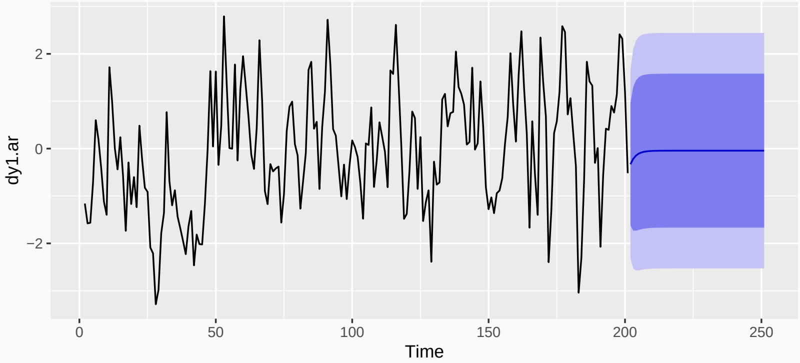
Example: ARIMA(1,1,0)

```r
# fit AR(1) to first difference
(fit1 <- Arima(dy1.ar, order = c(1,0,0)))
```

```
## Series: dy1.ar
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
##        0.603   -0.043
## s.e.   0.056    0.178
##
## sigma^2 = 1.02:  log likelihood = -285.3
## AIC=576.5   AICc=576.7   BIC=586.4
```

Example: ARIMA(1,1,0)

```r
# plot forecasts
autoplot(forecast(fit1, h = 50), main = "")
```



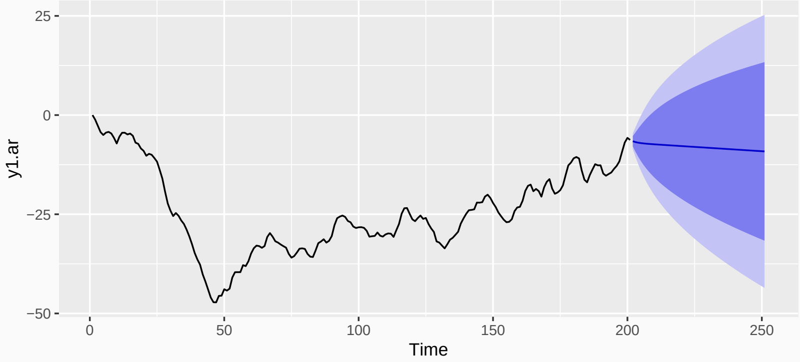
Example: ARIMA(1,1,0)

```r
# fit ARIMA(1,1,0)
(fit2 <- Arima(y1.ar, order = c(1,1,0), include.drift = TRUE))
```

```
## Series: y1.ar
## ARIMA(1,1,0) with drift
##
## Coefficients:
##          ar1   drift
##        0.603  -0.043
## s.e.  0.056   0.178
##
## sigma^2 = 1.02:  log likelihood = -285.3
## AIC=576.5   AICc=576.7   BIC=586.4
```

Example: ARIMA(1,1,0)

```r
# plot forecasts
autoplot(forecast(fit2, h = 50), main = "")
```



Understanding ARIMA models

Remarks:

- if $c = 0$ and $d = 0$, the long-term forecasts will go to zero
- if $c \neq 0$ and $d = 0$, the long-term forecasts will go to the mean of the data
- if $c = 0$ and $d = 1$, the long-term forecasts will go to a non-zero constant
- if $c \neq 0$ and $d = 1$, the long-term forecasts will follow a straight line

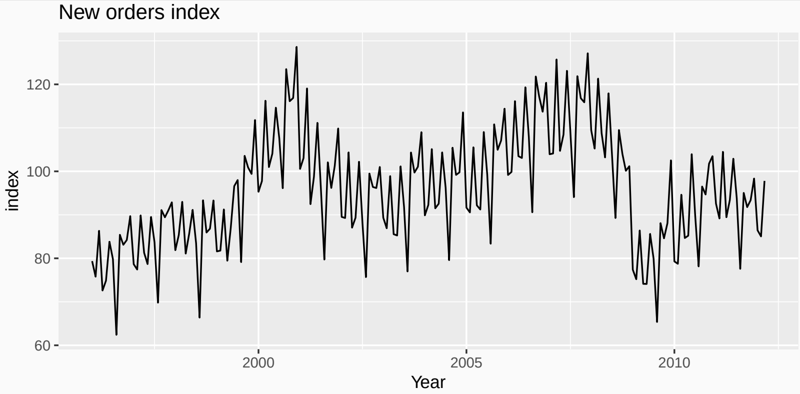
Understanding ARIMA models

Remarks:

- if $d = 0$, the long-term forecast standard deviation will go to the standard deviation of the historical data
- if $d \neq 0$, the prediction intervals will increase in size
- the higher the value of d, the more rapidly the prediction intervals increase in size

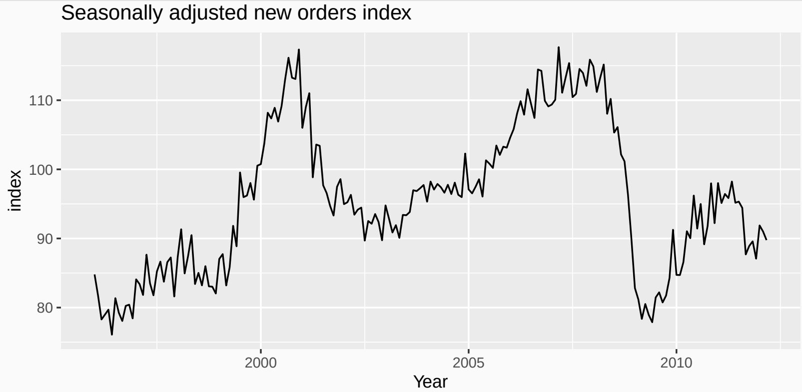
Seasonally adjusted electrical equipment

```r
# plot electrical equipment index
autoplot(elecequip) + xlab("Year") + ylab("index") +
  ggtitle("New orders index")
```



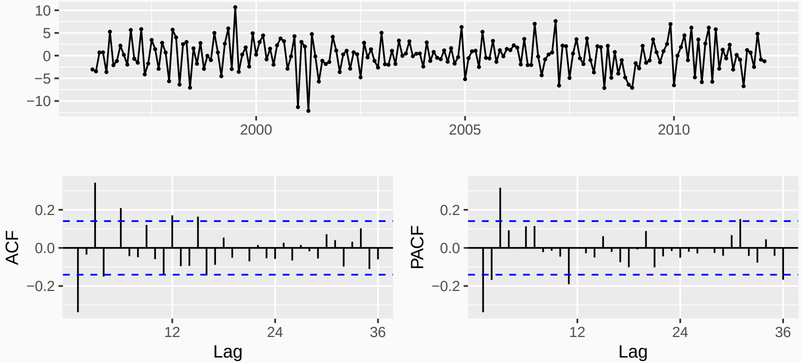
Seasonally adjusted electrical equipment

```r
# plot electrical equipment index
eeadj <- seasadj(stl(elecequip, s.window = "periodic"))
autoplot(eeadj) + xlab("Year") + ylab("index") +
  ggtitle("Seasonally adjusted new orders index")
```



Seasonally adjusted electrical equipment

```r
# plot electrical equipment index, first difference
ggtsdisplay(diff(eeadj))
```



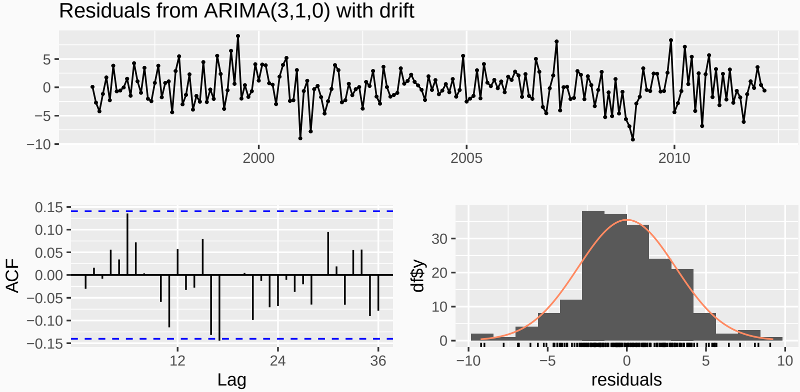
Seasonally adjusted electrical equipment

```r
# fit model to first difference
(fit <- Arima(eeadj, order=c(3,1,0), include.drift = TRUE))
```

```
## Series: eeadj
## ARIMA(3,1,0) with drift
##
## Coefficients:
##           ar1     ar2    ar3  drift
##        -0.342  -0.043  0.318  0.030
## s.e.    0.068   0.073  0.068  0.207
##
## sigma^2 = 9.69:  log likelihood = -493.8
## AIC=997.6   AICc=997.9   BIC=1014
```

Seasonally adjusted electrical equipment

```
# check residuals
checkresiduals(fit, test = FALSE)
```



Seasonally adjusted electrical equipment

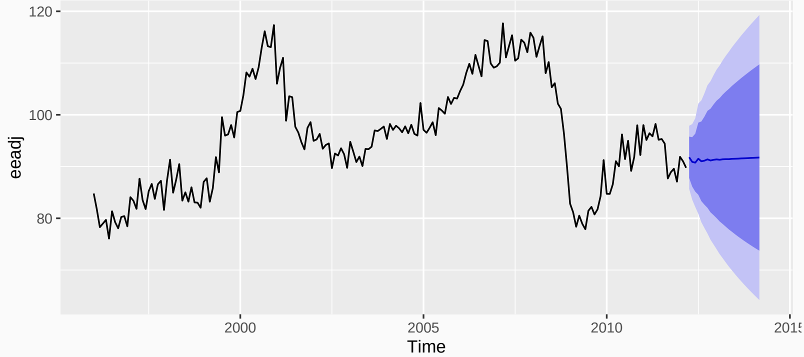
```r
# check residuals
checkresiduals(fit, plot = FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0) with drift
## Q* = 24, df = 21, p-value = 0.3
##
## Model df: 3.   Total lags used: 24
```

- ARIMA(3,1,0) model looks like white noise

Seasonally adjusted electrical equipment

```r
# plot forecasts
autoplot(forecast(fit), main = "")
```



Outline

1. Forecasting with non-stationary time series
2. Example: US output growth

Example: US output growth

The file `us_macro_quarterly.csv` contains quarterly data on several macroeconomic series for the United States.

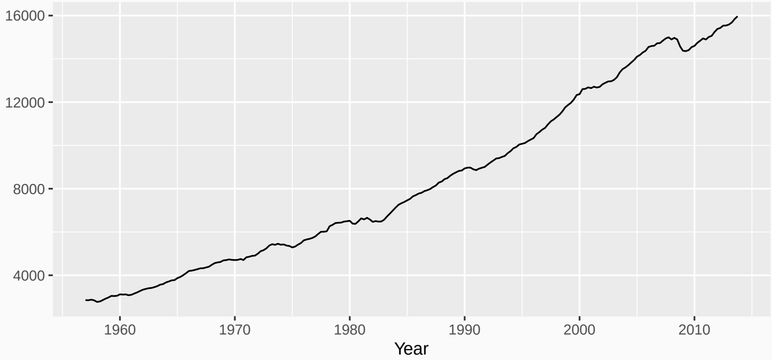
The variable `GDPC96` is the index of real GDP for the period 1957Q1 to 2013Q4.

```
# read quarterly data
data <- read.csv("data/us_macro_quarterly.csv", header = TRUE)

# get time series
rgdp.lv <- ts(data$GDPC96, start = c(1957,1), freq = 4)
rgdp.gr <- ts(
  400*diff(log(data$GDPC96)),
  start = c(1957,2),
  freq = 4
)
```

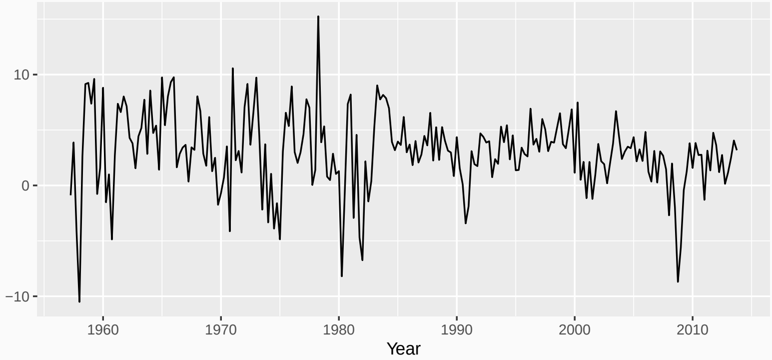
Example: US output growth

```
# plot RGDP
autoplot(rgdp.lv) + xlab("Year") + ylab(" ")
```



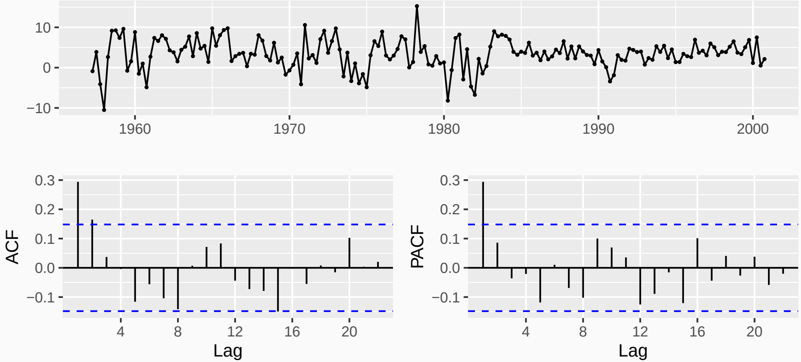
Example: US output growth

```r
# plot RGDP growth
autoplot(rgdp.gr) + xlab("Year") + ylab(" ")
```



Example: US output growth

```r
# in-sample (training set): 1970Q1-2000Q4
yt <- window(rgdp.gr, end = c(2000,4))
ggtsdisplay(yt)
```



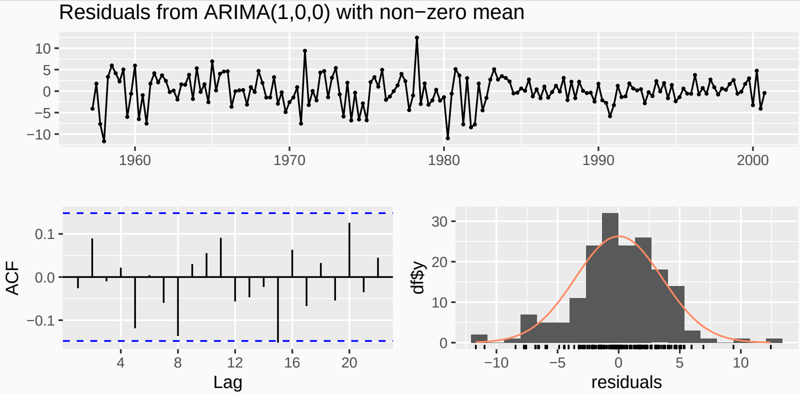
Example: US output growth

```r
# AR(1), in-sample
(model10 <- Arima(yt, order = c(1,0,0)))
```

```
## Series: yt
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##        0.295   3.398
## s.e.   0.072   0.380
##
## sigma^2 = 12.8:  log likelihood = -470.2
## AIC=946.4   AICc=946.5   BIC=955.9
```

Example: US output growth

```r
# AR(1), in-sample
checkresiduals(model10, lag = 10, test = FALSE)
```



Example: US output growth

```r
# AR(1), in-sample
checkresiduals(model10, lag = 10, plot = FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 9.1, df = 9, p-value = 0.4
##
## Model df: 1.   Total lags used: 10
```

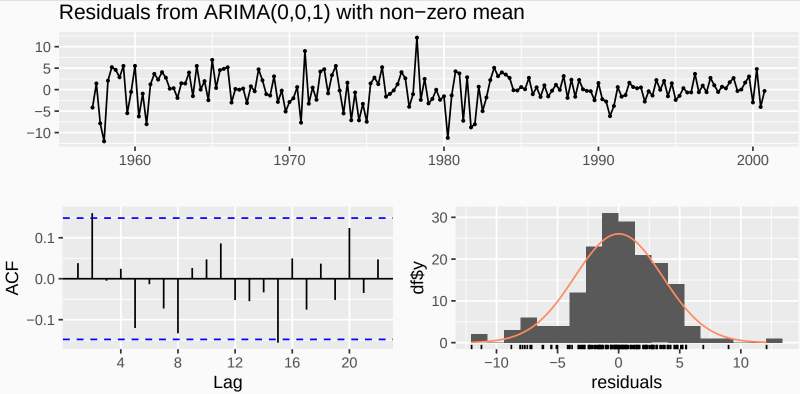
Example: US output growth

```r
# MA(1), in-sample
(model01 <- Arima(yt, order = c(0,0,1)))
```

```
## Series: yt
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##        0.238   3.404
## s.e.   0.064   0.335
##
## sigma^2 = 13:  log likelihood = -471.8
## AIC=949.6   AICc=949.8   BIC=959.1
```

Example: US output growth

```r
# MA(1), in-sample
checkresiduals(model01, lag = 10, test = FALSE)
```



Example: US output growth

```r
# MA(1), in-sample
checkresiduals(model01, lag = 10, plot = FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with non-zero mean
## Q* = 12, df = 9, p-value = 0.2
##
## Model df: 1.   Total lags used: 10
```

Example: US output growth

```r
# aics
c(AIC(model10),AIC(model01))
```

```
## [1] 946.4 949.6
```

```r
# bics
c(BIC(model10),BIC(model01))
```

```
## [1] 955.9 959.1
```

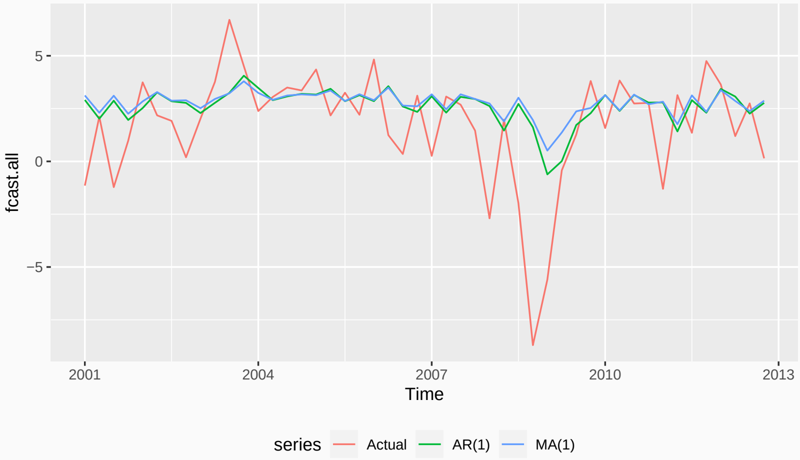
- AR(1) shows the best in-sample fit

Example: US output growth

```r
# end of training set, 2000Q4
n.end <- 2000.75

# test set: 2001Q1 - 2012Q4
# set matrix for storage, 48 obs in test set
pred <- matrix(rep(NA,144),48,3)
# loop
for(i in 1:48){
  tmp0 <- 1970
  tmp1 <- n.end+(i-1)*1/4
  tmp <- window(rgdp.gr,tmp0,tmp1)
  pred[i,1] <- window(rgdp.gr,tmp1+1/4,tmp1+1/4) # actual
  # compute forecasts
  pred[i,2] <- forecast(Arima(tmp,order=c(1,0,0)),h=1)$mean # AR(1)
  pred[i,3] <- forecast(Arima(tmp,order=c(0,0,1)),h=1)$mean # MA(1)
}
```

Example: US output growth



Example: US output growth

Choose forecasting model with the smallest RMSE computed using time series cross-validation.

```
# compute rmse
rmse <- rep(NA,2)
for(m in 1:2){rmse[m] <- sqrt(mean((pred[,1]-pred[,1+m])^2))}
# display rmse
rmse
```

```
## [1] 2.57 2.69
```

- AR(1) shows the best out-of-sample performance (RMSE)