Tutorial 8 Answers

Return to the monthly Bank Accepted Bill interest rates data in BAB3mth.csv.

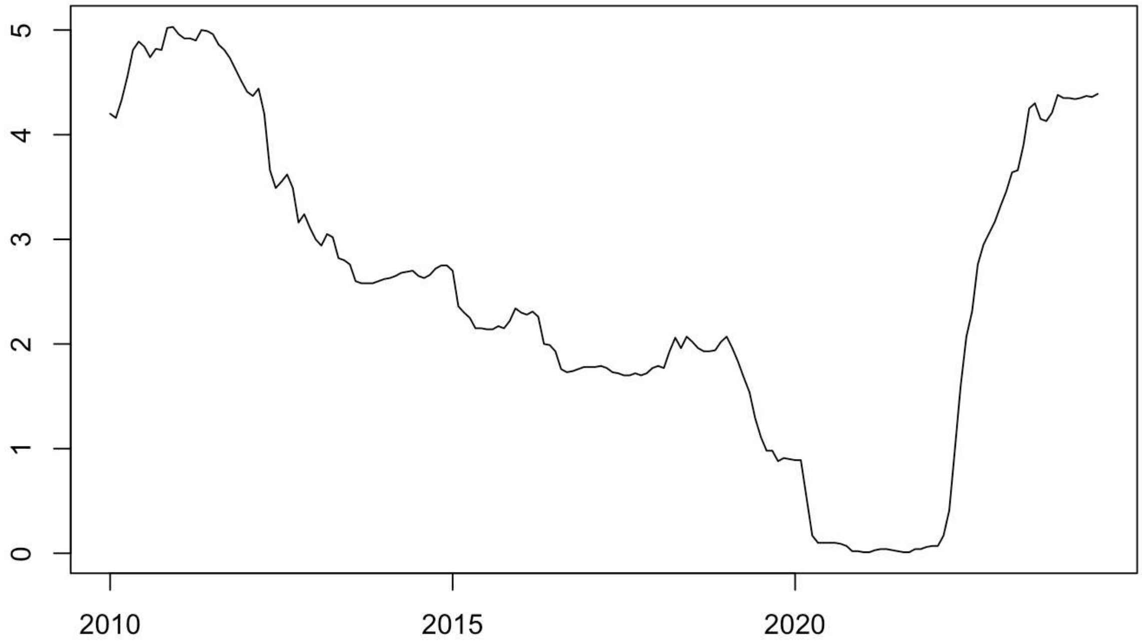
```
# Read data, define ts() and diff() objects
dt <- read.csv("BAB3mth.csv")
Y <- ts(dt$BAB3, start=c(2010,1), end=c(2025,6), frequency=12)
DY <- diff(Y)

# Forecast evaluation sample: Jul-2024 to Jun-2025
Yf <- window(Y, start=c(2024,7), end=c(2025,6))
DYf <- window(DY, start=c(2024,7), end=c(2025,6))

# Estimation sample: Jan-2010 to Jun-2024
Y <- window(Y, end=c(2024,6))
DY <- window(DY, end=c(2024,6))
```

1. Here is the plot of the interest rate over the estimation sample:

```
plot(Y)
```



While the clear overall trend is less obvious with the most recent observations included, we will include a trend in the unit root test:

```
library(urca)
adf_Y <- ur.df(Y, type="trend", lags=13, selectlags="AIC")
print(adf_Y@testreg$coefficients)
```

```
                 Estimate   Std. Error    t value     Pr(>|t|)
(Intercept) -0.0163896943 0.0338318086 -0.4844463 6.287575e-01
z.lag.1     -0.0060908370 0.0069181827 -0.8804100 3.800091e-01
tt           0.0003156819 0.0002335712  1.3515445 1.785030e-01
z.diff.lag1  0.6248054557 0.0788913131  7.9198258 4.409966e-13
z.diff.lag2 -0.1863696752 0.0921214936 -2.0230857 4.479430e-02
z.diff.lag3  0.2107555795 0.0801427020  2.6297539 9.411742e-03
```

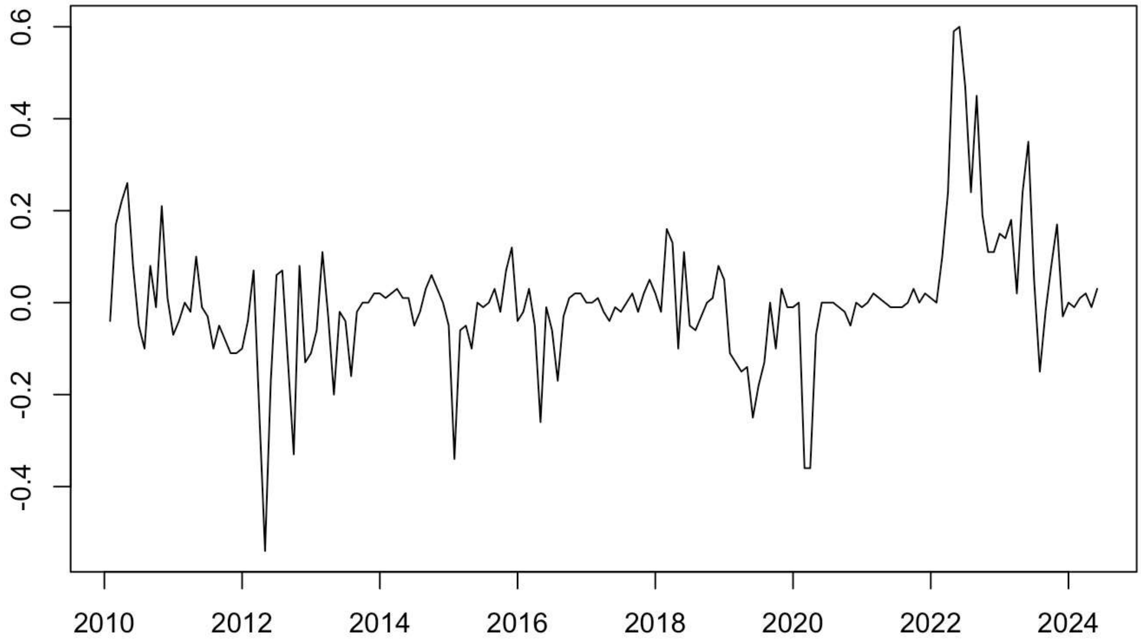
```
cat(paste0("ADF p-value = ",
    round(punitroot(adf_Y@teststat[1], trend="ct", statistic="t"),3)))
```

```
ADF p-value = 0.957
```

The AIC selects 3 lagged differences for the testing equation. The ADF p-value is not less than 0.05, so the unit root null hypothesis is not rejected.

We therefore also check the first difference series:

```
plot(DY)
```




```
library(urca)
adf_DY <- ur.df(DY, type="drift", lags=13, selectlags="AIC")
print(adf_DY@testreg$coefficients)
```

```
                 Estimate  Std. Error     t value     Pr(>|t|)
(Intercept) -0.0007273731 0.008617478 -0.08440673 9.328420e-01
z.lag.1     -0.2994327333 0.074266925 -4.03184504 8.656326e-05
z.diff.lag1 -0.0491609296 0.082934002 -0.59277171 5.541978e-01
z.diff.lag2 -0.2283908528 0.078177265 -2.92144849 4.005041e-03
```

```
cat(paste0("ADF p-value = ",
    round(punitroot(adf_DY@teststat[1], trend="c", statistic="t"),3)))
```

```
ADF p-value = 0.001
```

The ADF p-value for the differenced series is less than 0.05, so we reject the unit root null hypothesis and conclude that a single difference is sufficient.

2. Code:

```
library(forecast)
```

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

```
## ARMA search
# Maximum AR and MA orders
pmax <- 4
qmax <- 4

# Storage for AIC and LB p-values
AICc <- matrix(nrow=1+pmax, ncol=1+qmax)
rownames(AICc) <- paste0("p=", 0:pmax)
colnames(AICc) <- paste0("q=", 0:qmax)
LBp <- AICc

for (p in 0:pmax){
  for (q in 0:qmax){
    eq <- Arima(DY, order=c(p,0,q))
    AICc[p+1,q+1] <- eq$aicc
    LBp[p+1,q+1] <- Box.test(eq$residuals, lag=12,
                              type="Ljung-Box", fitdf=p+q)$p.value
  }
}
```

The models that show evidence $(p < 0.05)$ of residual autocorrelation are as follows:

```
print(1*(LBp<0.05))
```

```
    q=0 q=1 q=2 q=3 q=4
p=0   1   1   1   1   0
p=1   0   0   0   0   0
p=2   0   0   0   0   0
p=3   0   0   0   0   0
p=4   0   0   0   0   0
```

The AICc is minimised for an ARMA(1,2) model, and since this passes the residual autocorrelation test it becomes the preferred model.

```
print(1*(AICc==min(AICc)))
```

```
    q=0 q=1 q=2 q=3 q=4
p=0   0   0   0   0   0
p=1   0   0   1   0   0
p=2   0   0   0   0   0
p=3   0   0   0   0   0
p=4   0   0   0   0   0
```

The required forecasts are computed as follows:

```
eq <- Arima(DY, order=c(1,0,2))
```

```
hmax <- length(DYf)
DYf_AR1MA2 <- forecast(eq, h=hmax)$mean
```

```
         Jul     Aug     Sep     Oct     Nov     Dec
2024 0.02613 0.01452 0.01351 0.01259 0.01177 0.01102

         Jan     Feb     Mar     Apr     May     Jun
2025 0.01035 0.00974 0.00918 0.00869 0.00824 0.00783
```

3. The AR-only search has exactly the same logic as the ARMA search.

```
# AR search
pmax <- 12
AICc <- matrix(nrow=1+pmax, ncol=1)
rownames(AICc) <- paste0("p=", 0:pmax)
LBp <- AICc
colnames(AICc) <- paste0("AICc")
colnames(LBp) <- "LBp"

for (p in 0:pmax){
  eq <- Arima(DY, order=c(p,0,0))
  AICc[p+1,1] <- eq$aicc
  LBp[p+1,1] <- Box.test(eq$residuals, lag=p+12,
                     type="Ljung-Box", fitdf=p)$p.value
}
print(1*(t(LBp)<0.05))
```

```
    p=0 p=1 p=2 p=3 p=4 p=5 p=6 p=7 p=8 p=9 p=10 p=11 p=12
LBp   1   0   0   0   0   0   0   0   0   0    0    0    0
```

```
print(1*(t(AICc)==min(AICc)))
```

```
     p=0 p=1 p=2 p=3 p=4 p=5 p=6 p=7 p=8 p=9 p=10 p=11 p=12
AICc   0   0   0   1   0   0   0   0   0   0    0    0    0
```

```
eq <- Arima(DY, order=c(3,0,0))
DYf_AR3 <- forecast(eq, h=hmax)$mean
```

```
         Jul     Aug     Sep     Oct     Nov     Dec
2024 0.02495 0.01010 0.00873 0.00920 0.00698 0.00524

         Jan     Feb     Mar     Apr     May     Jun
2025 0.00456 0.00398 0.00339 0.00298 0.00271 0.00248
```

4. The direct forecasting model search follows the exact structure of that in the self-directed activities:

```
# Direct forecasting

pmax <- hmax+4

AICc <- matrix(nrow=1+pmax, ncol=hmax)
```

```
rownames(AICc) <- paste0("p=",0:pmax)
colnames(AICc) <- paste0("h=",1:hmax)
LBp <- AICc

# Loop over all forecast horizons
for (h in 1:hmax){

  # Loop over all AR lag orders
  for (p in c(0,h:pmax)){

    # Estimate AR(p) zeros has same length as
    # parameters in AR(p)+intercept
    zeros <- rep(NA,p+1)

    # coefficients on first h-1 lags set to zero
    if (p>0 & h>1){
      zeros[1:(h-1)] <- 0
    }

    # Estimate model with coefficient
    # restrictions imposed
    eq <- Arima(DY, order=c(p,0,0),
            fixed=zeros)
    AICc[1+p,h] <- eq$aicc

    # Ljung-Box statistic include 12 lags beyond AR order
    LB <- Box.test(eq$residuals,
               lag=p+12,
               type="Ljung-Box")$statistic

    # Remove first h-1 lags from LB
    if (h>1){
      LB <- LB-
            Box.test(eq$residuals,
             lag=h-1,
             type="Ljung-Box")$statistic
    }

    # p-value for LB test
    LBp[1+p,h] <- pchisq(LB, df=12,
                       lower.tail=FALSE)

  }
}

cat("\nLjung-Box significant:\n")
```

```
Ljung-Box significant:
```

```
print(1*(LBp<0.05))
```

```
     h=1 h=2 h=3 h=4 h=5 h=6 h=7 h=8 h=9 h=10 h=11 h=12
p=0    1   1   1   1   1   1   1   1   1    0    0    0
```

```
p=1    0  NA  NA  NA  NA  NA  NA  NA  NA   NA   NA   NA
p=2    0   1  NA  NA  NA  NA  NA  NA  NA   NA   NA   NA
p=3    0   0   1  NA  NA  NA  NA  NA  NA   NA   NA   NA
p=4    0   0   0   0  NA  NA  NA  NA  NA   NA   NA   NA
p=5    0   0   0   0   0  NA  NA  NA  NA   NA   NA   NA
p=6    0   0   0   0   0   1  NA  NA  NA   NA   NA   NA
p=7    0   0   0   0   0   0   0  NA  NA   NA   NA   NA
p=8    0   0   0   0   0   0   0   0  NA   NA   NA   NA
p=9    0   0   0   0   0   0   0   0   0   NA   NA   NA
p=10   0   0   0   0   0   0   0   0   0    0   NA   NA
p=11   0   0   0   0   0   0   0   0   0    0    0   NA
p=12   0   0   0   0   0   0   0   0   0    0    0    0
p=13   0   0   0   0   0   0   0   0   0    0    0    0
p=14   0   0   0   0   0   0   0   0   0    0    0    0
p=15   0   0   0   0   0   0   0   0   0    0    0    0
p=16   0   0   0   0   0   0   0   0   0    0    0    0
```

```r
# Select p for each h
ph <- (0:pmax)[apply(AICc, 2, which.min)]

names(ph) <- paste0("h=",1:h)
print(ph)
```

```
 h=1  h=2  h=3  h=4  h=5  h=6  h=7  h=8  h=9 h=10 h=11 h=12 
   3    4    4    4    8    8    8    8    9   10   12   12 
```

Here is the direct forecasting computation based on the chosen models:

```r
## Direct forecasting - forecasts

# Storage for forecasts
DYf_Direct <- ts(start=c(2024,7), end=c(2025,6), frequency=12)

for (h in 1:hmax){

  # Estimate chosen model for each h
  zeros <- c(rep(0,h-1),
         rep(NA,ph[h]-h+2))

  eq <- Arima(DY,
          order=c(ph[h],0,0),
          fixed=zeros)

  # Forecast h steps ahead
  DYf_Direct[h] <- forecast(eq, h=h)$mean[h]
}

DYf_ <- cbind(DYf_AR1MA2, DYf_AR3, DYf_Direct)
print(round(DYf_,5))
```

```
         DYf_AR1MA2 DYf_AR3 DYf_Direct
Jul 2024    0.02613 0.02495    0.02495
Aug 2024    0.01452 0.01010    0.01340
```

```
Sep 2024    0.01351 0.00873    0.00544
Oct 2024    0.01259 0.00920    0.01192
Nov 2024    0.01177 0.00698    0.00952
Dec 2024    0.01102 0.00524    0.01087
Jan 2025    0.01035 0.00456    0.00278
Feb 2025    0.00974 0.00398    0.00975
Mar 2025    0.00918 0.00339    0.00846
Apr 2025    0.00869 0.00298    0.00688
May 2025    0.00824 0.00271    0.00232
Jun 2025    0.00783 0.00248    0.00755
```

5. RMSE calculations:

```
RMSE <- sqrt(apply((DYf_-DYf)^2, 2, mean))
```

```
AR1MA2    AR3 Direct
0.1088 0.1051 0.1069
```

There are not large differences between these RMSEs, but the AR(3) model is best with direct AR forecasting next, ARMA(1,2) third.

6. Conversion of forecasts for differences to levels

```
n <- length(Y)
Yf_AR1MA2 <- ts(Y[n]+cumsum(DYf_AR1MA2),
              start=c(2024,7), end=c(2025,6), frequency=12)
Yf_AR3 <- ts(Y[n]+cumsum(DYf_AR3),
              start=c(2024,7), end=c(2025,6), frequency=12)
Yf_Direct <- ts(Y[n]+cumsum(DYf_Direct),
              start=c(2024,7), end=c(2025,6), frequency=12)

Yf_ <- cbind(Yf_AR1MA2, Yf_AR3, Yf_Direct)
```

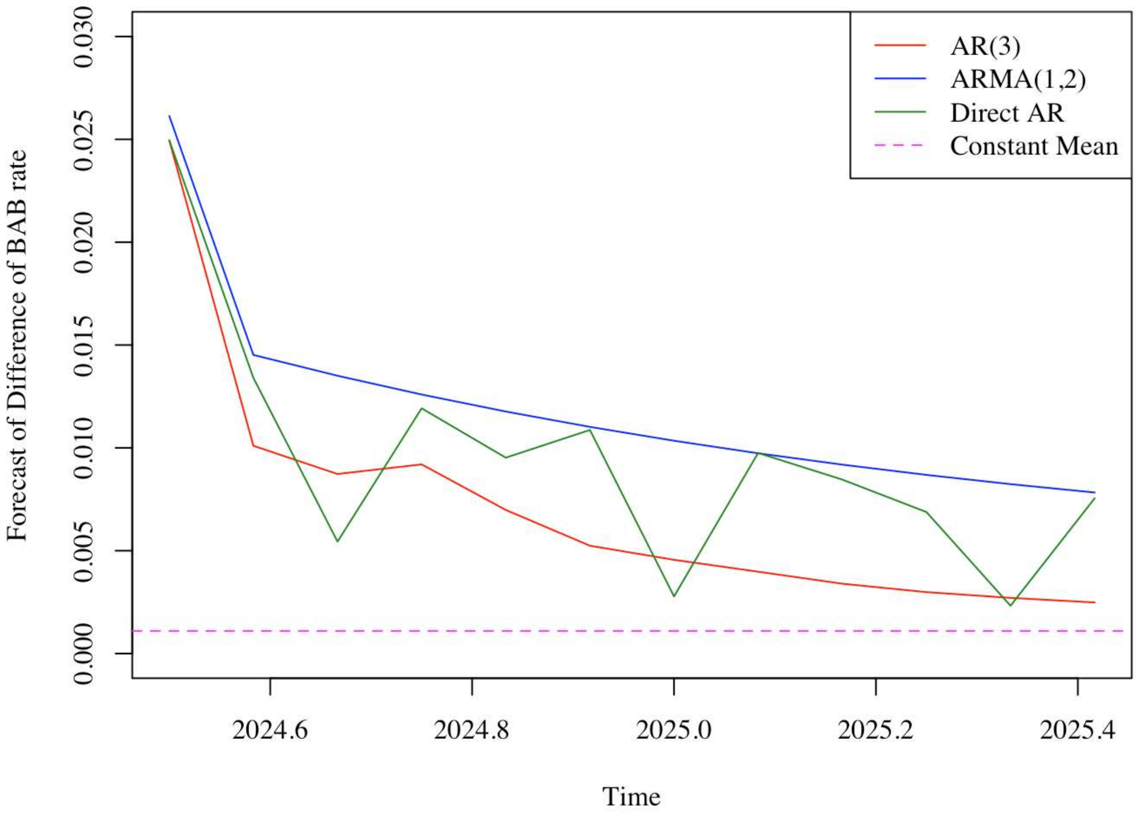
```
         Yf_AR1MA2  Yf_AR3 Yf_Direct
Jul 2024   4.41613 4.41495   4.41495
Aug 2024   4.43065 4.42505   4.42835
Sep 2024   4.44416 4.43378   4.43379
Oct 2024   4.45675 4.44297   4.44571
Nov 2024   4.46852 4.44995   4.45524
Dec 2024   4.47954 4.45520   4.46611
Jan 2025   4.48988 4.45975   4.46888
Feb 2025   4.49962 4.46373   4.47863
Mar 2025   4.50880 4.46713   4.48709
Apr 2025   4.51749 4.47011   4.49397
May 2025   4.52573 4.47282   4.49629
Jun 2025   4.53356 4.47530   4.50384
```

```
RMSE <- sqrt(apply((Yf_-Yf)^2, 2, mean))
```

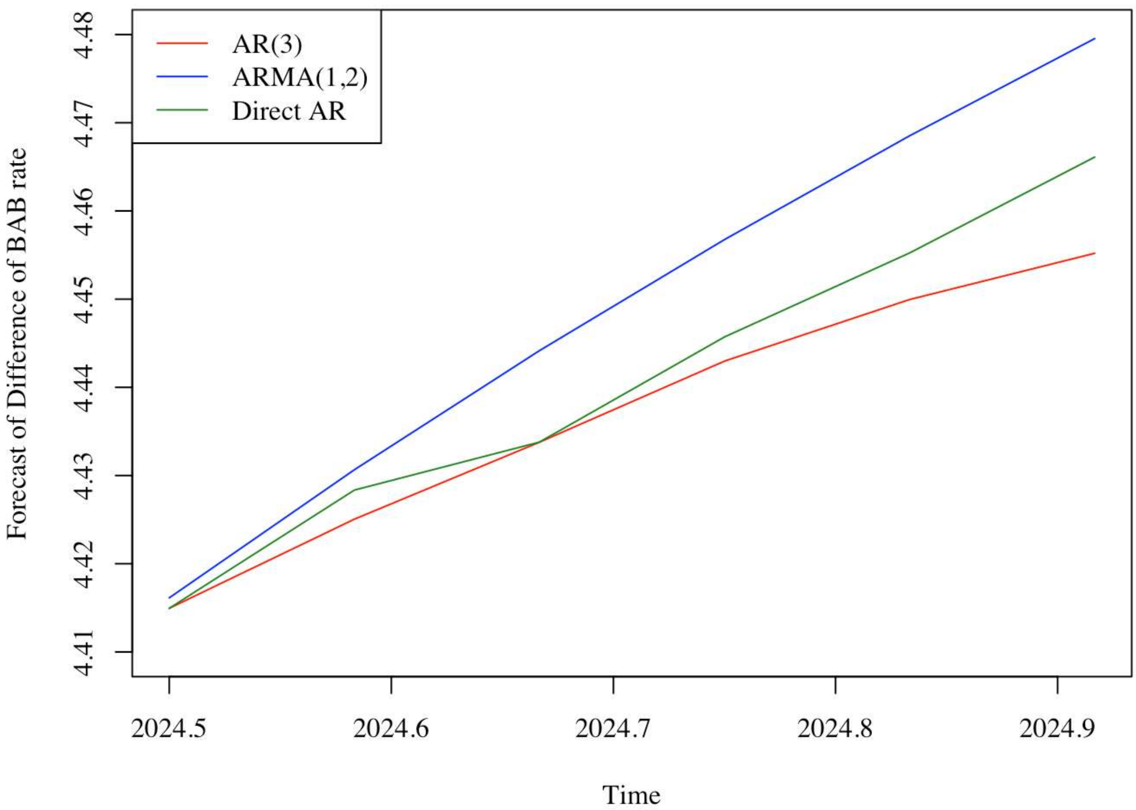
```
AR1MA2    AR3 Direct
0.3886 0.3570 0.3716
```

Again, not surprisingly, the forecast accuracy is similar acros the models with AR(3) best followed by Direct AR and then ARMA(1,2).

7. Here is the plot of the forecasts of the *first differences* of the BAB interest rates.



Here is the plot of the forecasts of the *levels* of the BAB interest rates.



- The three forecasts of the differences have broadly similar shapes. Note all three are converging towards the unconditional constant mean. This is a property of h-step-ahead forecasts from stationary models as h increases.

- There is nothing particularly systematic about the ARMA(1,2) forecasts being greater than the AR(3) forecasts. In a different application it may have been the reverse, or some crossing over. The main finding is the two sets of forecasts are quite similar (note the scale, the magnitudes of differences are quiet small).

- The more jagged forecasts from the direct AR approach reflect the fact they are computed from different models for each h. In some cases the lag orders differ across h (see above), but all cases we re-estimate the model for each h with zeros imposed for the first $h-1$ coefficients, and these differing coefficients across h are producing the jaggedness of the forecasts. This is not necessarily an advantage or disadvantage of the direct approach, simply a characteristic. It might be argued the separate estimates for each h are explicitly finding the best coefficient values at that h (as opposed to using those $h = 1$ for every h), but it may also be argued that the separate estimates are adding more estimation error to the process.

- The forecasts for the *levels*, computed from the forecasts for the differences using `cumsum`, can be seen to diverge from each other as h increases. This is a characteristic of forecasts from different *non-stationary* models — note that the ARMA(3,0), ARMA(1,2) and direct AR(p_h) models are stationary in the *first differences* and when we convert back to levels they become ARIMA(3,1,0), ARIMA(1,1,2), ARIMA(p_h,1,0) models for the levels. Even though the forecasts from the three stationary models for the difference are very similar and converge to the unconditional mean, all initial differences for smaller h persist and accumulate in the levels forecasts.