# Gold Price Forecasting Using ARIMA Model (Replicated Version)

Radbeh Heravi

2023-01-09

## Introduction

This study gives an inside view of the application of ARIMA time series model to forecast the future Gold price in Indian browser based on past data from November 2003 to January 2014 to mitigate the risk in purchases of gold. This study is based on secondary monthly data for Gold price which is collected from Multi Commodity Exchange of India Ltd (MCX) ranging from November 2003 to January 2014. MCX is a commodity future exchange based in India which started its operations from November 2003.

## Step 0

In this step data is prepared to start replication. Also our initial guess from considering merely ACF and PACF is that p=1. However, this could be misleading since we have not checked for the unit root yet.

```
# packages
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(forecast)
```

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

```
library(urca)
library(ggplot2)
library(pander)

# remove previous stuff
rm(list=ls(all=TRUE))
ls()
```

```
## character(0)
```

```
# set path
setwd("C:/Users/asus/Desktop/R projects")
getwd()
```

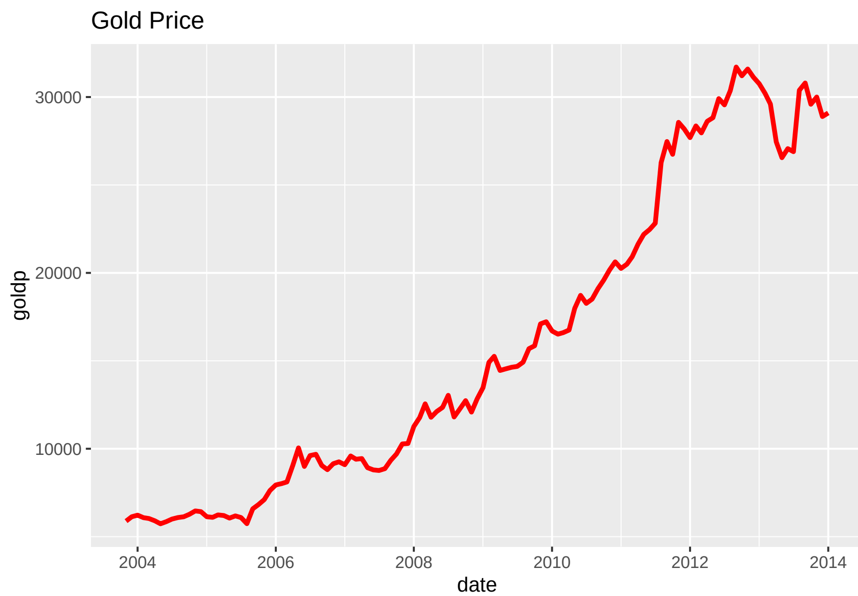
```
## [1] "C:/Users/asus/Desktop/R projects"
```

```r
# import data
mydata <- read.csv("C:/Users/asus/Desktop/goldp.csv",sep=",",header = TRUE)

# preparing data
mydata <- as.data.frame(mydata)
class(mydata)
```

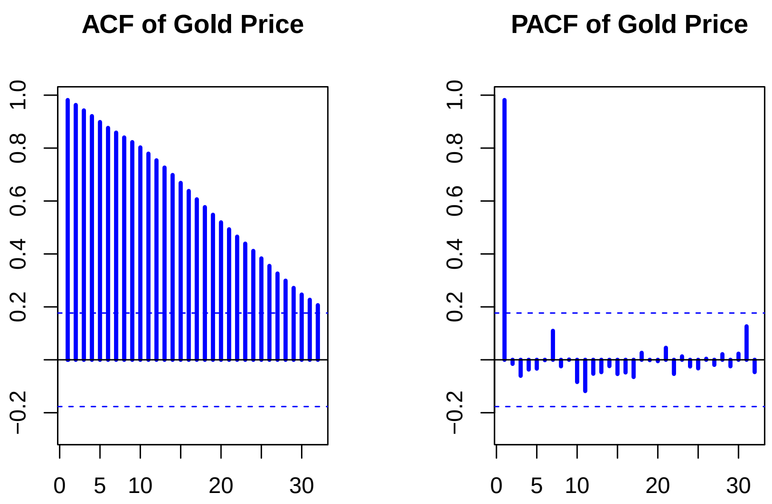
```
## [1] "data.frame"
```

```r
mydata$goldp <- as.ts(mydata$goldp)
mydata$date <- as.Date(mydata$date)

# plot data for goldp
ggplot(mydata, aes(x = date, y = goldp)) +
  geom_line(col="red" , size=1.2) +
  ggtitle("Gold Price")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```r
# plot acf for goldp
layout(matrix(c(1,2),1,2))
Acf(mydata$goldp,lag.max=32,ann=FALSE, main="ACF of Gold Price", col= "blue", lwd=3)
Pacf(mydata$goldp,lag.max=32,ann=FALSE, main="PACF of Gold Price", col= "blue", lwd=3)
```




## Step 1

Having conducted unit root test, the data is not stationary. Probably it is I(1) with drift. The unit root test of the first difference of the data rejects the null hypothesis at the 5 percent level meaning that it is a stationary process.

```
# step 1: is the data stationary?-------------------------------
# set path
setwd("C:/Users/asus/Desktop/R projects")
getwd()
```

```
## [1] "C:/Users/asus/Desktop/R projects"
```

```
# load my unit root tests
source("urtests.r")

# unit root test for goldp
test1 <- ur.test(mydata$goldp,trend="ct",method="adf.gls")
print.ur.test(test1)
```

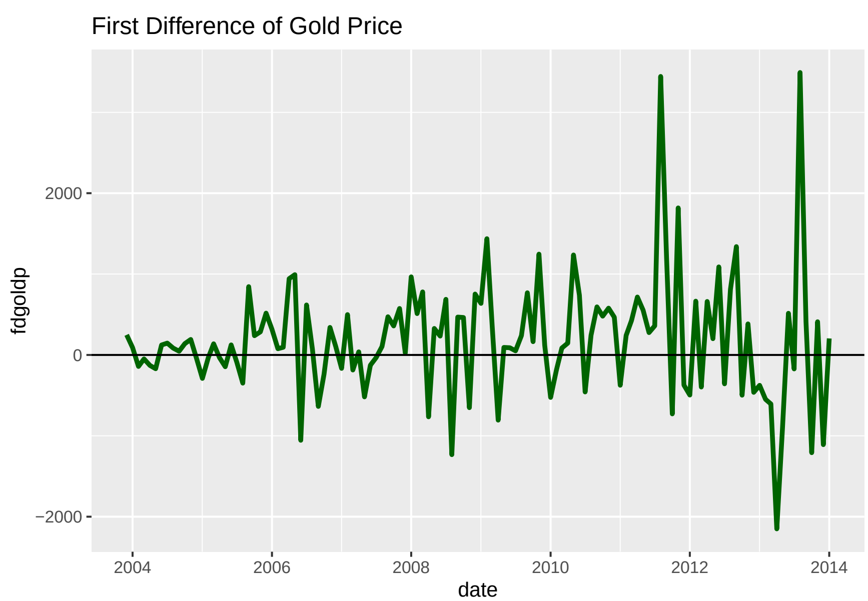
```
##
##  Test for Unit Root: ADF Test with GLS Detrending
##
##  Null Hypothesis: there is a unit root
##  Test Statistic: -1.304
##  Critical Values (.01,.05,.10): -3.42 -2.91 -2.62
##
##  Lag Order Selection Rule: MAIC
##  Selected Lag Order: 4
##  Estimated Coefficient: 0.9670
```

```
# result: it is not. Thus it is I(1) with drift

# first diff of goldp
mydata$fdgoldp <- c(NA, diff(mydata$goldp))
```

```r
# unit root test of first diff of goldp
test2 <- ur.test(mydata$fdgoldp[2:123],trend="c",method="adf.gls")
print.ur.test(test2)
```

```
##
##  Test for Unit Root: ADF Test with GLS Detrending
##
##  Null Hypothesis: there is a unit root
##  Test Statistic: -2.193
##  Critical Values (.01,.05,.10): -2.58 -1.98 -1.62
##
##  Lag Order Selection Rule: MAIC
##  Selected Lag Order: 11
##  Estimated Coefficient: 0.0475
```

```r
# result: it is stationary at the 5 percent level

# plot data for fdgoldp
ggplot(mydata[2:123,], aes(x = date, y = fdgoldp)) +
  geom_line(col="dark green", lwd=1.2) +
  geom_hline(yintercept=0) +
  ggtitle("First Difference of Gold Price")
```




## Step 2

Now it is time to determine the best model for the future prediction. It is crystal clear that all the models have statistical issues because of poor precision of the estimated coefficients other than ARIMA(1,1,1) which is suitable for our goal.

```r
# Step 2: finding the best ARIMA model-------------------------------
# estimate ARIMA(1,0,1)
model101 <- Arima(mydata$goldp,order=c(1,0,1),method="ML",include.drift = TRUE)
pander(summary(model101))
```

Call: Arima(y = mydata$goldp, order = c(1, 0, 1), include.drift = TRUE, method = "ML")

Table 1: Coefficients

| | ar1 | ma1 | intercept | drift |
|---|---|---|---|---|
| | 0.9515 | 0.07021 | 3233 | 210.3 |
| **s.e.** | 0.02763 | 0.09717 | 2173 | 27.85 |

sigma^2 estimated as 527950: log likelihood = -984.11, aic = 1978.23

```
# result: it is not verified, because ma(1) and intercept are not statistically
# significant.

# estimate ARIMA(1,0,2)
model102 <- Arima(mydata$goldp,order=c(1,0,2),method="ML",include.drift = TRUE)
pander(summary(model102))
```

Call: Arima(y = mydata$goldp, order = c(1, 0, 2), include.drift = TRUE, method = "ML")

Table 2: Coefficients

| | ar1 | ma1 | ma2 | intercept | drift |
|---|---|---|---|---|---|
| | 0.9541 | 0.08362 | -0.04588 | 3228 | 209.9 |
| **s.e.** | 0.02781 | 0.1007 | 0.1292 | 2186 | 28.01 |

sigma^2 estimated as 532209: log likelihood = -984.09, aic = 1980.18

```
# result: it is not verified, because ma(1), ma(2) and intercept are not statistically
# significant

# estimate ARIMA(1,0,3)
model103 <- Arima(mydata$goldp,order=c(1,0,3),method="ML",include.drift = TRUE)
pander(summary(model103))
```

Call: Arima(y = mydata$goldp, order = c(1, 0, 3), include.drift = TRUE, method = "ML")

Table 3: Coefficients

| | ar1 | ma1 | ma2 | ma3 | intercept | drift |
|---|---|---|---|---|---|---|
| | 0.9166 | 0.1904 | 0.02298 | 0.2504 | 2693 | 216.8 |
| **s.e.** | 0.04482 | 0.1065 | 0.1217 | 0.101 | 1886 | 25.09 |

sigma^2 estimated as 512424: log likelihood = -981.35, aic = 1976.7

```
# result: it is not verified, because ma(1), ma(2) and intercept are not statistically
# significant

# estimate ARIMA(1,1,1)
model111 <- Arima(mydata$goldp,order=c(1,1,1),method="ML",include.drift = TRUE)
pander(summary(model111))
```

Call: Arima(y = mydata$goldp, order = c(1, 1, 1), include.drift = TRUE, method = "ML")

Table 4: Coefficients

| | ar1 | ma1 | drift |
|---|---|---|---|
| | -0.7358 | 0.8709 | 190.7 |
| **s.e.** | 0.1156 | 0.07649 | 69.29 |

sigma^2 estimated as 517164: log likelihood = -974.19, aic = 1956.38

```
# result: Goooooood

# estimate ARIMA(1,1,2)
model112 <- Arima(mydata$goldp,order=c(1,1,2),method="ML",include.drift = TRUE)
pander(summary(model112))
```

Call: Arima(y = mydata$goldp, order = c(1, 1, 2), include.drift = TRUE, method = "ML")

Table 5: Coefficients

| | ar1 | ma1 | ma2 | drift |
|---|---|---|---|---|
| | -0.6782 | 0.7761 | -0.06757 | 190.8 |
| **s.e.** | 0.151 | 0.1598 | 0.09642 | 65.35 |

sigma^2 estimated as 519405: log likelihood = -973.95, aic = 1957.9

```
# result: it is not verified, because ma(2) and intercept are not statistically
# significant.

# estimate ARIMA(1,1,3)
model113 <- Arima(mydata$goldp,order=c(1,1,3),method="ML",include.drift = TRUE)
pander(summary(model113))
```

Call: Arima(y = mydata$goldp, order = c(1, 1, 3), include.drift = TRUE, method = "ML")

Table 6: Coefficients

| | ar1 | ma1 | ma2 | ma3 | drift |
|---|---|---|---|---|---|
| | -0.4873 | 0.5979 | 0.03958 | 0.1934 | 189.6 |
| **s.e.** | 0.2038 | 0.1974 | 0.12 | 0.1077 | 77.95 |

sigma^2 estimated as 513024: log likelihood = -972.72, aic = 1957.44

```
# result: it is not verified, because ma(2) and ma(3) are not statistically
# significant.
# final result of this part: ARIMA(1,1,1) is picked.
```

The table, depicted below, compares different models based on their statistical significance of the standard errors of the coefficients.

| Model | Intercept | Drift | AR(1) | MA(1) | MA(2) | MA(3) |
|---|---|---|---|---|---|---|
| ARIMA(1,0,1) | insignificant | significant | significant | insignificant | ____ | ____ |
| ARIMA(1,0,2) | insignificant | significant | significant | insignificant | insignificant | ____ |
| ARIMA(1,0,3) | insignificant | significant | significant | insignificant | insignificant | significant |

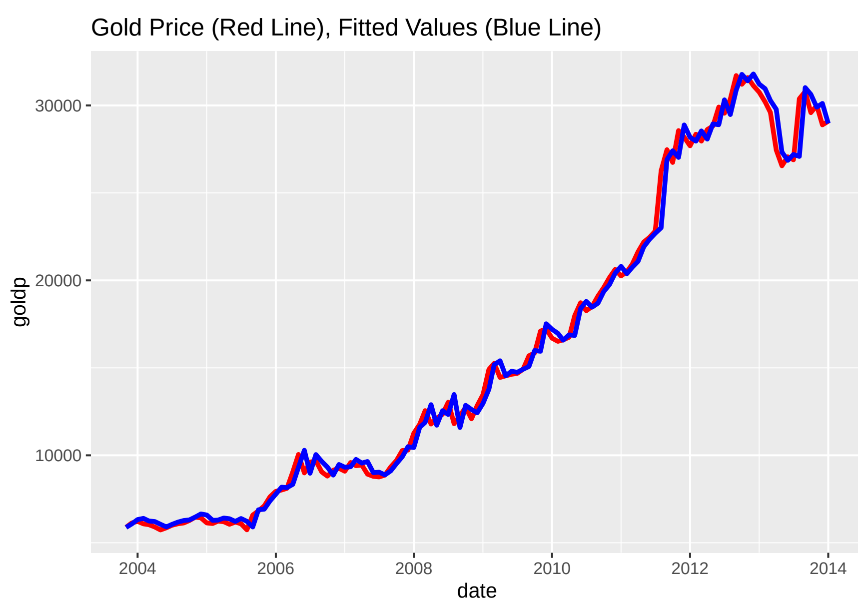
| Model | Intercept | Drift | AR(1) | MA(1) | MA(2) | MA(3) |
|---|---|---|---|---|---|---|
| ARIMA(1,1,1) | _____ | significant | significant | significant | _____ | _____ |
| ARIMA(1,1,2) | _____ | significant | significant | significant | insignificant | _____ |
| ARIMA(1,1,3) | _____ | significant | significant | significant | insignificant | insignificant |

### Step 3

This step is to predict the following 6 months (2014-02 till 2014-07) by implementing our selected model from previous section, ARIMA(1,1,1). It should be noted that 6-step ahead prediction method is taken into consideration.

```
# step 3: forecasting-------------------------------
# add a column of fitted values to mydata
mydata$fit <- model111$fitted

# plot goldp and fit values
ggplot(mydata, aes(x = date)) +
  geom_line(aes(y = goldp), col="red", size=1.2) +
  geom_line(aes(y = fit), col="blue", size=1.2) +
  ggtitle("Gold Price (Red Line), Fitted Values (Blue Line)")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

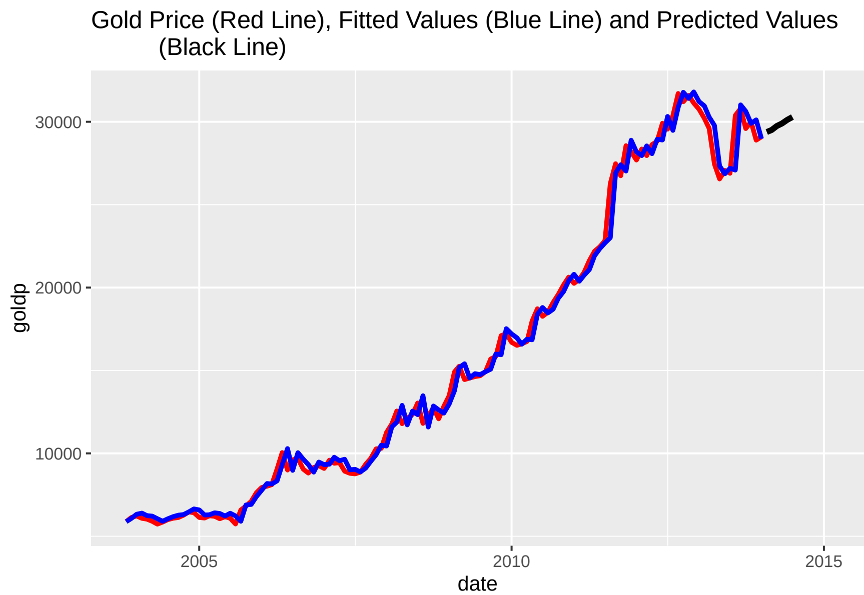



```
# 6-step ahead forecasts
# ARIMA(1,0,1)
pred101 <- forecast(model101,h=6)
pred101 <- as.ts(pred101$mean,start=c(2014,2),end=c(2014,7),frequency=12)

# ARIMA(1,0,2)
pred102 <- forecast(model102,h=6)
pred102 <- as.ts(pred102$mean,start=c(2014,2),end=c(2014,7),frequency=12)

# ARIMA(1,0,3)
pred103 <- forecast(model103,h=6)
```

```r
pred103 <- as.ts(pred103$mean,start=c(2014,2),end=c(2014,7),frequency=12)

# ARIMA(1,1,1) <---- This is our selected model
pred111 <- forecast(model111,h=6)
pred111 <- as.ts(pred111$mean,start=c(2014,2),end=c(2014,7),frequency=12)

# ARIMA(1,1,2)
pred112 <- forecast(model112,h=6)
pred112 <- as.ts(pred112$mean,start=c(2014,2),end=c(2014,7),frequency=12)

# ARIMA(1,1,3)
pred113 <- forecast(model113,h=6)
pred113 <- as.ts(pred113$mean,start=c(2014,2),end=c(2014,7),frequency=12)
```

```r
# making a suitable data frame for ARIMA(1,1,1) to plot goldp, fitted and predicted
# values.
date <- c(mydata$date,"2014-02-01"
          ,"2014-03-01"
          ,"2014-04-01"
          ,"2014-05-01"
          ,"2014-06-01"
          ,"2014-07-01"
          ,"2014-08-01"
          ,"2014-09-01"
          ,"2014-10-01"
          ,"2014-11-01"
          ,"2014-12-01"
          ,"2015-01-01"
          ,"2015-02-01"
)
date <- as.Date(date)
goldp <- ts(c(mydata$goldp, rep(NA,13)), start = c(2003,11), end = c(2015,2),
            frequency = 12)
fit <- ts(c(mydata$fit, rep(NA,13)), start = c(2003,11), end = c(2015,2),
          frequency = 12)
prediction <- ts(c(rep(NA,123), pred111, rep(NA,7)), start = c(2003,11), end = c(2015,2),
               frequency = 12)
mydata1 <- data.frame(date, goldp, fit, prediction)

# plot goldp, fitted and forecasted values
ggplot(mydata1, aes(x = date)) +
  geom_line(aes(y = goldp), col="red", size=1.2) +
  geom_line(aes(y = fit), col="blue", size=1.2) +
  geom_line(aes(y = prediction), col="black", size=1.5) +
  ggtitle("Gold Price (Red Line), Fitted Values (Blue Line) and Predicted Values
          (Black Line)")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
# Comparing observed and estimated values
# observed values during 2014-02 till 2014-07
obs <- c(29482.91,29670.43,28514.64,27812.81,26813.15,27867.11)
article.pred <- c(29386.40,29614.93,29850.13,30009.72,30224.71,30399.11)
obs.vs.pred <- data.frame(date[124:129], obs, article.pred, pred111)
colnames(obs.vs.pred) <- c("Date","Real Observations",
                           "Predictions of The Article","My Predictions")
pander(obs.vs.pred)
```

| Date | Real Observations | Predictions of The Article | My Predictions |
|---|---|---|---|
| 2014-02-01 | 29483 | 29386 | 29384 |
| 2014-03-01 | 29670 | 29615 | 29503 |
| 2014-04-01 | 28515 | 29850 | 29746 |
| 2014-05-01 | 27813 | 30010 | 29898 |
| 2014-06-01 | 26813 | 30225 | 30118 |
| 2014-07-01 | 27867 | 30399 | 30287 |

### Step 4 (Appendix)

This part is for fit statistics.

```
# step 4: fit statistics (appendix)--------------------------------
# r-squared
e101 <- mydata$goldp - model101$fitted
r101 <- 1-(sum(e101^2)/sum((mydata$goldp-mean(mydata$goldp))^2))

e102 <- mydata$goldp - model102$fitted
r102 <- 1-(sum(e102^2)/sum((mydata$goldp-mean(mydata$goldp))^2))

e103 <- mydata$goldp - model103$fitted
r103 <- 1-(sum(e103^2)/sum((mydata$goldp-mean(mydata$goldp))^2))

e111 <- mydata$goldp - model111$fitted
r111 <- 1-(sum(e111^2)/sum((mydata$goldp-mean(mydata$goldp))^2))
```

```
e112 <- mydata$goldp - model112$fitted
r112 <- 1-(sum(e112^2)/sum((mydata$goldp-mean(mydata$goldp))^2))

e113 <- mydata$goldp - model113$fitted
r113 <- 1-(sum(e113^2)/sum((mydata$goldp-mean(mydata$goldp))^2))

r <- rbind(r101,r102,r103,r111,r112,r113)

# rmse, mape, mae
ac101 <- accuracy(pred101, obs)
ac102 <- accuracy(pred102, obs)
ac103 <- accuracy(pred103, obs)
ac111 <- accuracy(pred111, obs)
ac112 <- accuracy(pred112, obs)
ac113 <- accuracy(pred113, obs)

ac <- rbind(ac101,ac102,ac103,ac111,ac112,ac113)

# normalized bic
bic101 <- log(sum(model101$residuals^2)/123) + (length(model101$coef)/123)*log(123)
bic102 <- log(sum(model102$residuals^2)/123) + (length(model102$coef)/123)*log(123)
bic103 <- log(sum(model103$residuals^2)/123) + (length(model103$coef)/123)*log(123)
bic111 <- log(sum(model111$residuals^2)/123) + (length(model111$coef)/123)*log(123)
bic112 <- log(sum(model112$residuals^2)/123) + (length(model112$coef)/123)*log(123)
bic113 <- log(sum(model113$residuals^2)/123) + (length(model113$coef)/123)*log(123)

bic <- rbind(bic101,bic102,bic103,bic111,bic112,bic113)

# Ljung q statistics
bt101 <- Box.test(model101$resid,lag=18,type="Ljung-Box",fitdf=length(model101$coef))
bt102 <- Box.test(model102$resid,lag=18,type="Ljung-Box",fitdf=length(model102$coef))
bt103 <- Box.test(model103$resid,lag=18,type="Ljung-Box",fitdf=length(model103$coef))
bt111 <- Box.test(model111$resid,lag=18,type="Ljung-Box",fitdf=length(model111$coef))
bt112 <- Box.test(model112$resid,lag=18,type="Ljung-Box",fitdf=length(model112$coef))
bt113 <- Box.test(model113$resid,lag=18,type="Ljung-Box",fitdf=length(model113$coef))

bt <- rbind(bt101$p.value,bt102$p.value,bt103$p.value,bt111$p.value,bt112$p.value
            ,bt113$p.value)
```

Although the fit statistics of mine are not in line with those of the original article; to the best of my knowledge, the conducted calculations are flawless.

Table 9: Table continues below

| | R-squared | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|---|
| **ARIMA(1,0,1)** | 0.9932 | -1477 | 1977 | 1584 | -5.376 | 5.736 |
| **ARIMA(1,0,2)** | 0.9932 | -1523 | 2006 | 1591 | -5.537 | 5.767 |
| **ARIMA(1,0,3)** | 0.9935 | -1444 | 1989 | 1618 | -5.264 | 5.854 |
| **ARIMA(1,1,1)** | 0.9934 | -1463 | 1944 | 1551 | -5.321 | 5.621 |
| **ARIMA(1,1,2)** | 0.9934 | -1483 | 1959 | 1557 | -5.393 | 5.643 |
| **ARIMA(1,1,3)** | 0.9935 | -1330 | 1845 | 1508 | -4.852 | 5.453 |

| | Normalized BIC | Ljung-Box p-value |
|---|---|---|
| **ARIMA(1,0,1)** | 13.3 | 0.2514 |
| **ARIMA(1,0,2)** | 13.34 | 0.1898 |
| **ARIMA(1,0,3)** | 13.33 | 0.5765 |
| **ARIMA(1,1,1)** | 13.24 | 0.5668 |
| **ARIMA(1,1,2)** | 13.28 | 0.547 |
| **ARIMA(1,1,3)** | 13.29 | 0.7778 |