Next-Day Bitcoin Price Forecast

Nihad & Vikram

2024-06-12

Requirement

Abstract: This study analyzes forecasts of Bitcoin price using the autoregressive integrated moving average (ARIMA) and neural network autoregression (NNAR) models. Employing the static forecast approach, we forecast next-day Bitcoin price both with and without re-estimation of the forecast model for each step. For cross-validation of forecast results, we consider two different training and test samples. In the first training-sample, NNAR performs better than ARIMA, while ARIMA outperforms NNAR in the second training-sample. Additionally, ARIMA with model re-estimation at each step outperforms NNAR in the two test-sample forecast periods. The Diebold Mariano test confirms the superiority of forecast results of ARIMA model over NNAR in the test-sample periods. Forecast performance of ARIMA models with and without re-estimation are identical for the estimated test-sample periods. Despite the sophistication of NNAR, this paper demonstrates ARIMA enduring power of volatile Bitcoin price prediction.

Keywords: ARIMA; artificial neural network; Bitcoin; cryptocurrency; static forecast

Code setup

Link to code repo Github

Required libraries

```
library(xts)
library(quantmod)
library(ggthemes)
library(dygraphs)
library(tidyverse)
library(urca)
```

```r
library(tseries)
library(forecast)
library(dplyr)
```

Required sub source files

```r
source("../code/config.R")
source("../code/utils.R")
source("../code/model_executor.R")
```

Read bitcoin csv daily price

```r
quotes_bitcoin <- read_csv("../data/Bitcoindata.csv",
                            col_select = c(Date,Close))
```

We can examine structure of the resulting object:

```r
head(quotes_bitcoin)
```

```
# A tibble: 6 x 2
  Date       Close
  <chr>      <dbl>
1 04/10/2018 6548.
2 03/10/2018 6457.
3 02/10/2018 6500
4 01/10/2018 6571.
5 30/09/2018 6598.
6 29/09/2018 6579.
```

```r
tail(quotes_bitcoin)
```

```
# A tibble: 6 x 2
  Date       Close
  <chr>      <dbl>
1 06/01/2012  6
2 05/01/2012  6.65
3 04/01/2012  5.57
4 03/01/2012  5.29
5 02/01/2012  5
6 01/01/2012  5
```

```r
glimpse(quotes_bitcoin)
```

```
Rows: 2,466
Columns: 2
$ Date  <chr> "04/10/2018", "03/10/2018", "02/10/2018", "01/10/2018", "30/09/2~
$ Close <dbl> 6547.56, 6456.77, 6500.00, 6571.20, 6597.81, 6579.38, 6610.76, 6~
```

Let's also check the class of the Date column:

```r
class(quotes_bitcoin$Close)
```

```
[1] "numeric"
```

lets check structure of the whole dataset

```r
str(quotes_bitcoin)
```

```
tibble [2,466 x 2] (S3: tbl_df/tbl/data.frame)
 $ Date : chr [1:2466] "04/10/2018" "03/10/2018" "02/10/2018" "01/10/2018" ...
 $ Close: num [1:2466] 6548 6457 6500 6571 6598 ...
 - attr(*, "spec")=
  .. cols(
  ..   ...1 = col_skip(),
  ..   Date = col_character(),
  ..   Open = col_skip(),
  ..   High = col_skip(),
  ..   Low = col_skip(),
  ..   Close = col_double(),
  ..   `Volume (BTC)` = col_skip(),
  ..   `Volume (Currency)` = col_skip(),
  ..   `Weighted Price` = col_skip()
  .. )
```

Let's transform column 'Date' into type date:

```r
quotes_bitcoin$Date <- as.Date(quotes_bitcoin$Date, format = "%d/%m/%Y")
```

We have to give the format in which date is originally stored: * %y means 2-digit year, * %Y means 4-digit year * %m means a month * %d means a day

```r
class(quotes_bitcoin$Date)
```

```
[1] "Date"
```

```r
head(quotes_bitcoin)
```

```
# A tibble: 6 x 2
  Date       Close
  <date>     <dbl>
1 2018-10-04 6548.
2 2018-10-03 6457.
3 2018-10-02 6500
4 2018-10-01 6571.
5 2018-09-30 6598.
6 2018-09-29 6579.
```

```r
glimpse(quotes_bitcoin)
```

```
Rows: 2,466
Columns: 2
$ Date  <date> 2018-10-04, 2018-10-03, 2018-10-02, 2018-10-01, 2018-09-30, 201~
$ Close <dbl> 6547.56, 6456.77, 6500.00, 6571.20, 6597.81, 6579.38, 6610.76, 6~
```

Now R understands this column as dates

Creating xts objects

```r
quotes_bitcoin <-
  xts(quotes_bitcoin[, -1], # data columns (without the first column with date)
      quotes_bitcoin$Date)  # date/time index
```

Lets see the result:

```r
head(quotes_bitcoin)
```

```
           Close
2012-01-01  5.00
2012-01-02  5.00
```

```
2012-01-03  5.29
2012-01-04  5.57
2012-01-05  6.65
2012-01-06  6.00
```

```
str(quotes_bitcoin)
```

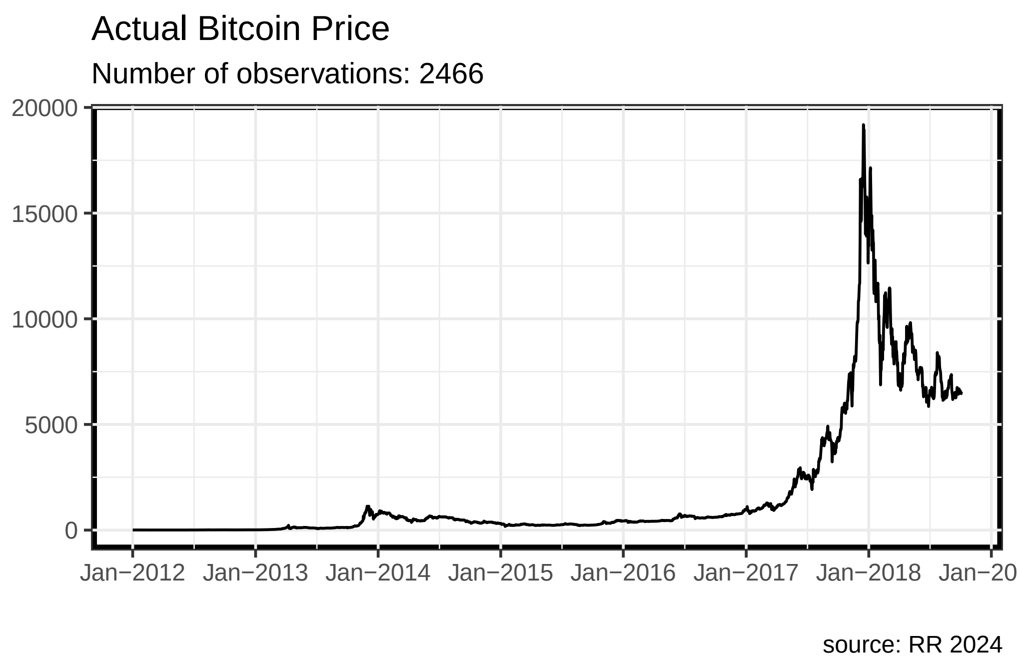
```
An xts object on 2012-01-01 / 2018-10-04 containing:
  Data:    double [2466, 1]
  Columns: Close
  Index:   Date [2466] (TZ: "UTC")
```

Basic graphs

Finally, let's use the ggplot2 package to produce nice visualization. The ggplot2 package expects data to be in long format, rather than wide format. Hence, first we have to convert the tibble to a long tibble:

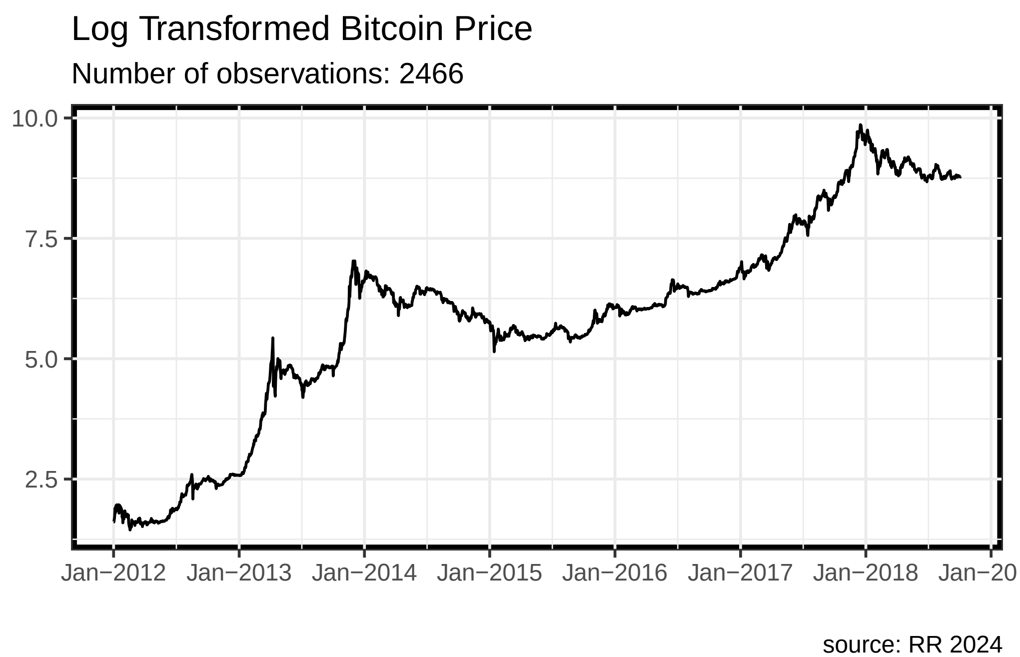
Plotting Actual Bitcoin Price

```
tibble(df = quotes_bitcoin) %>%
  ggplot(aes(zoo::index(quotes_bitcoin), df)) +
  geom_line() +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%b-%Y")+
  labs(
    title = "Actual Bitcoin Price",
    subtitle = paste0("Number of observations: ", length(quotes_bitcoin)),
    caption = "source: RR 2024",
    x="",
    y=""
  ) +
  theme(panel.background = element_rect(fill = "transparent",color = "black",linewidth = 2
```



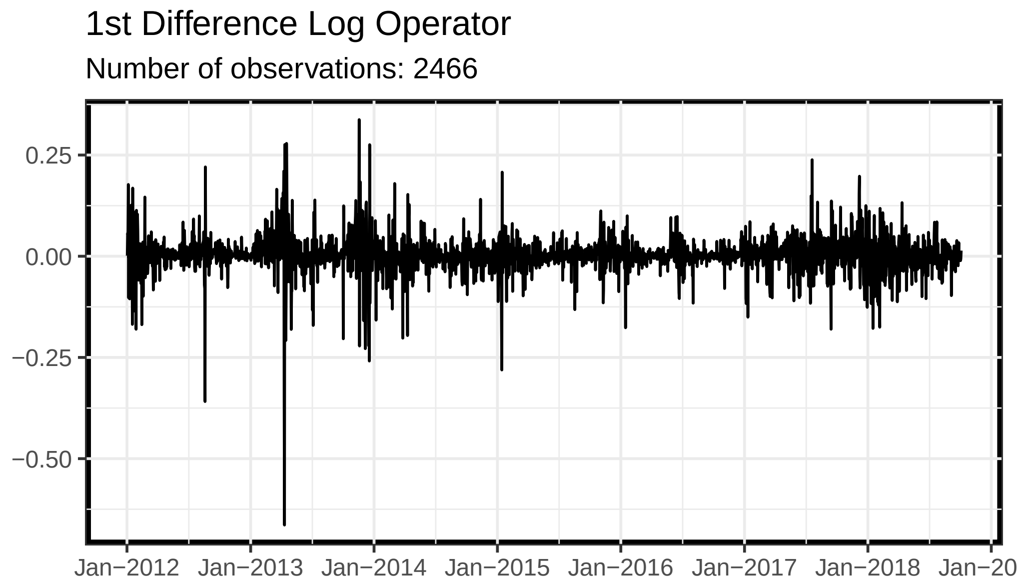
Plotting Log Transformed Bitcoin Price

```
tibble(df = quotes_bitcoin) %>%
  ggplot(aes(zoo::index(quotes_bitcoin), log(quotes_bitcoin))) +
  geom_line() +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%b-%Y")+
  labs(
    title = "Log Transformed Bitcoin Price",
    subtitle = paste0("Number of observations: ", length(quotes_bitcoin)),
    caption = "source: RR 2024",
    x="",
    y=""
  ) +
  theme(panel.background = element_rect(fill = "transparent",color = "black",linewidth = 2
```



Plotting 1st Difference Log Operator

```
tibble(df = quotes_bitcoin) %>%
  ggplot(aes(zoo::index(quotes_bitcoin), periodReturn(quotes_bitcoin, period="daily", type
  geom_line() +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%b-%Y")+
  labs(
    title = "1st Difference Log Operator",
    subtitle = paste0("Number of observations: ", length(quotes_bitcoin)),
    caption = "source: RR 2024",
    x="",
    y=""
  ) +
  theme(panel.background = element_rect(fill = "transparent",color = "black",linewidth = 2
```



source: RR 2024

Table 1. Stationary test of data.

First in-sample window (500 days)

```
                         Data        Training_Sample          ADF_Test
1               Original data 01/01/2012~14/05/2013 -1.849 ( 0.642 )
2        Log transformed data 01/01/2012~14/05/2013 -1.521 ( 0.781 )
3 1st difference log operator 01/01/2012~14/05/2013 -9.743 ( 0.010 )
             PP_Test
1  -12.235 ( 0.427 )
2   -3.828 ( 0.896 )
3 -497.980 ( 0.010 )
```

Second in-sample window (2000 days)

```
                         Data        Training_Sample           ADF_Test
1               Original data 01/01/2012~25/06/2017   0.617 ( 0.990 )
2        Log transformed data 01/01/2012~25/06/2017  -1.378 ( 0.842 )
3 1st difference log operator 01/01/2012~25/06/2017 -11.478 ( 0.010 )
              PP_Test
1     5.162 ( 0.990 )
```

2	-3.367 (0.918)
3	-2103.646 (0.010)

ADF. Augmented Dicky-Fuller test; PP. Phillips-Perron test. p-values in parenthesis, p-value less than 0.05 confirms stationary

Table 2. Training-sample forecast performance.

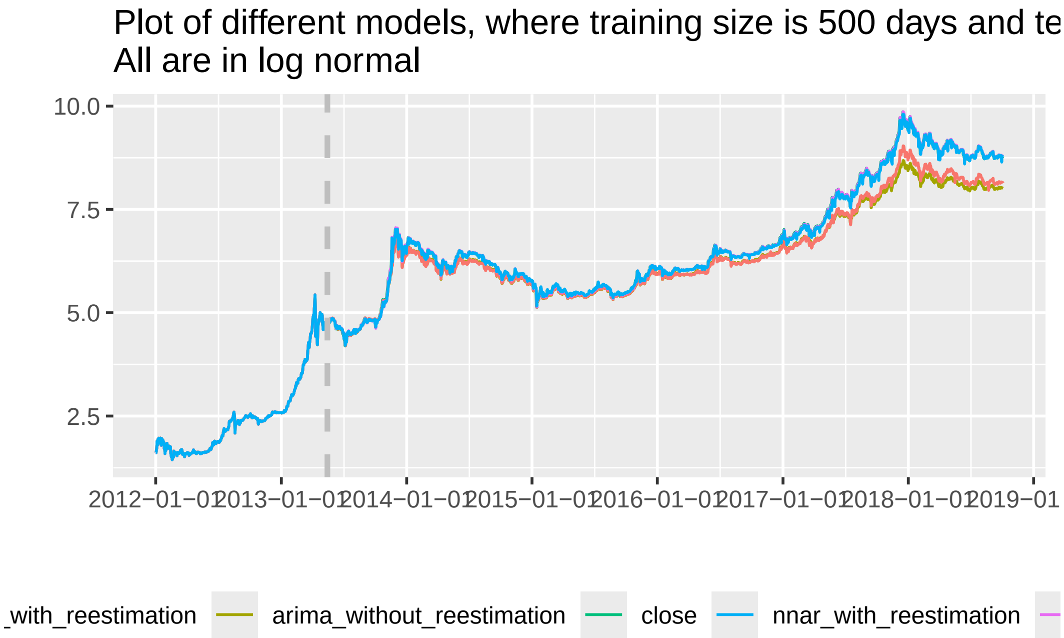
First training-sample window (500 days)

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA (4,1,0)	01/01/2012~14/05/2013	0.063	1.317	0.033
2	NNAR (2,1)	01/01/2012~14/05/2013	0.058	1.266	0.032

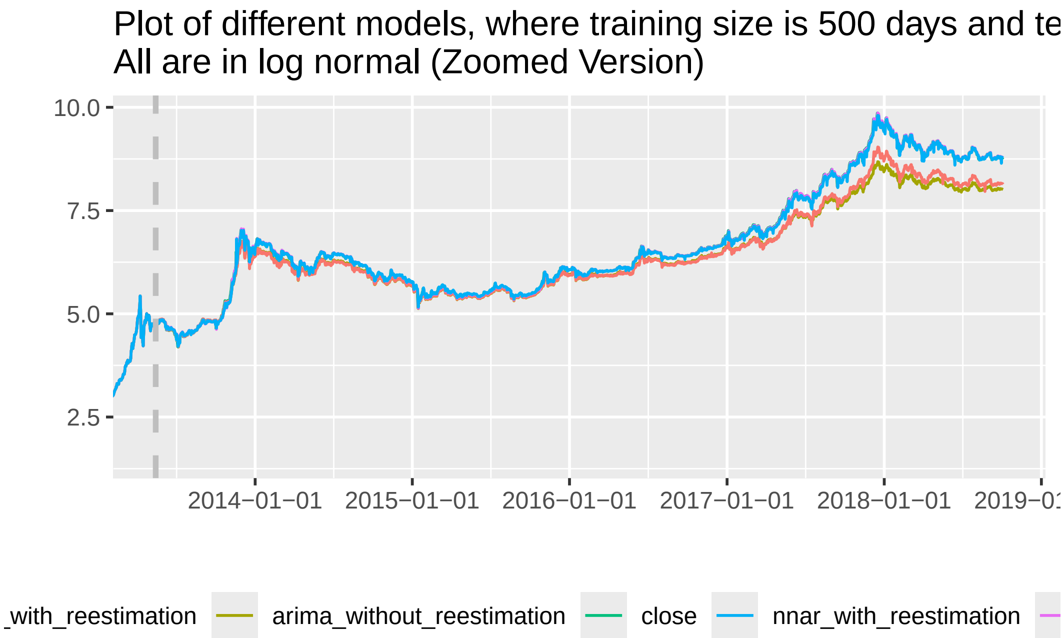
Second training-sample window (2000 days)

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA (4,1,1)	01/01/2012~25/06/2017	0.048	0.645	0.027
2	NNAR (1,2)	01/01/2012~25/06/2017	0.048	0.641	0.027

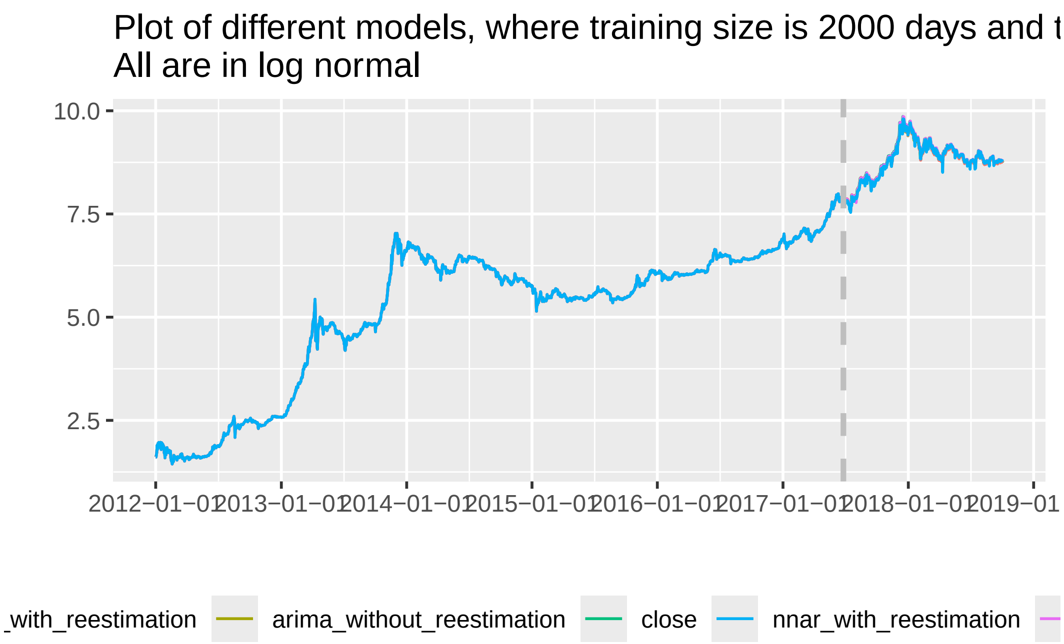
(a) Actual and forecasted Bitcoin price (training sample:500 days, test-sample:1966 days)



(b) Concentrated view on the forecast period (test-sample:1966 days)



(c) Actual and forecasted Bitcoin price (training sample:2000 days, test-sample:466 days)



(d) Concentrated view on the forecast period (test-sample:466 days)

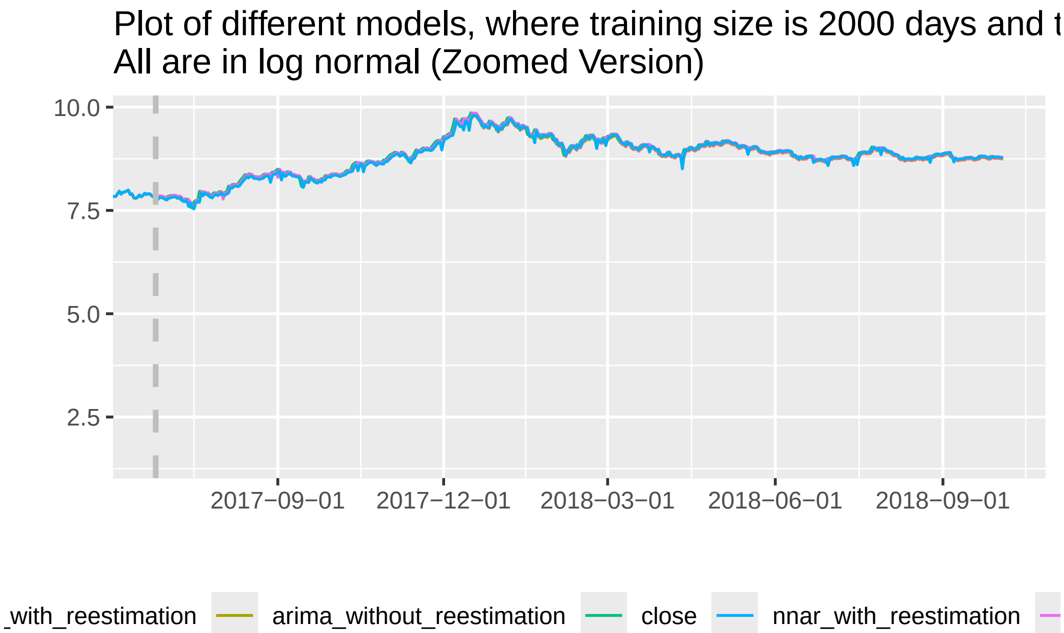


Table 3. Test-sample static forecast performance.

First test sample

First test-sample window (1966 days) Forecast without re-estimation at each step

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA (4,1,0)	15/05/2013~04/10/2018	0.373	2.924	0.230
2	NNAR (2,1)	15/05/2013~04/10/2018	0.042	0.357	0.024

Forecast with re-estimation at each step

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA	15/05/2013~04/10/2018	0.312	2.668	0.205
2	NNAR	15/05/2013~04/10/2018	0.050	0.425	0.029

Second test sample

Second test-sample window (466 days) Forecast without re-estimation at each step

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA (4,1,1)	26/06/2017~04/10/2018	0.026	0.098	0.009
2	NNAR (1,2)	26/06/2017~04/10/2018	0.022	0.078	0.007

Forecast with re-estimation at each step

	Forecast_Model	Training_Sample	RMSE	MAPE	MAE
1	ARIMA (4,1,1)	26/06/2017~04/10/2018	0.026	0.097	0.009
2	NNAR (1,2)	26/06/2017~04/10/2018	0.031	0.106	0.009

Table 4. DM test of forecast results.

First test-sample window (1966 days)

	Models_Compared	DM_Statistics	p_Value
1	ARIMA vs. NNAR (re-estimation)	-37.724	3.062208e-246
2	ARIMA vs. NNAR (without re-estimation)	-34.225	2.223566e-210
3	ARIMA (re-estimation) vs. ARIMA (without re-estimation)	18.317	2.281731e-70
4	NNAR (re-estimation) vs. NNAR (without re-estimation)	-18.115	5.935986e-69

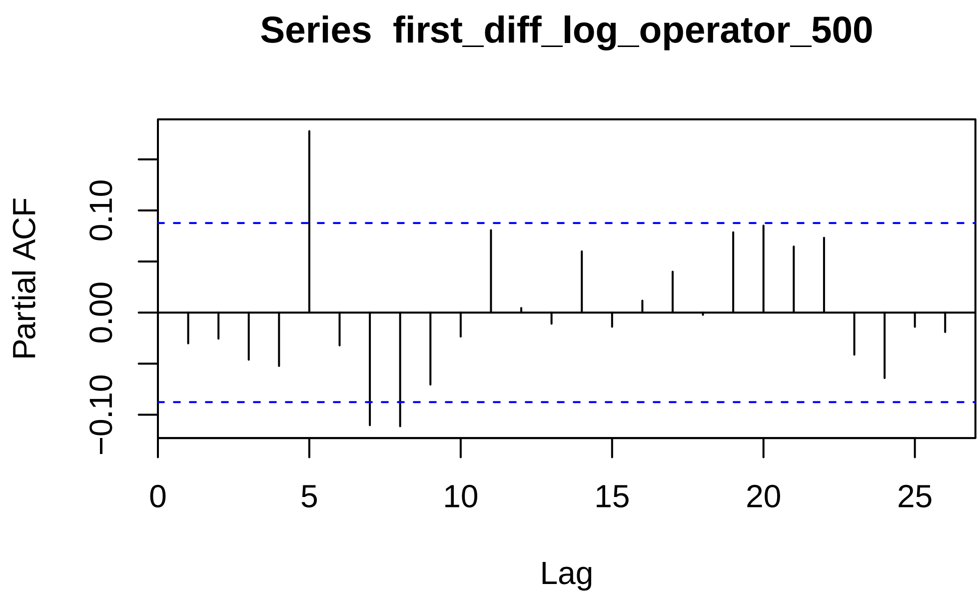
Second test-sample window (466 days)

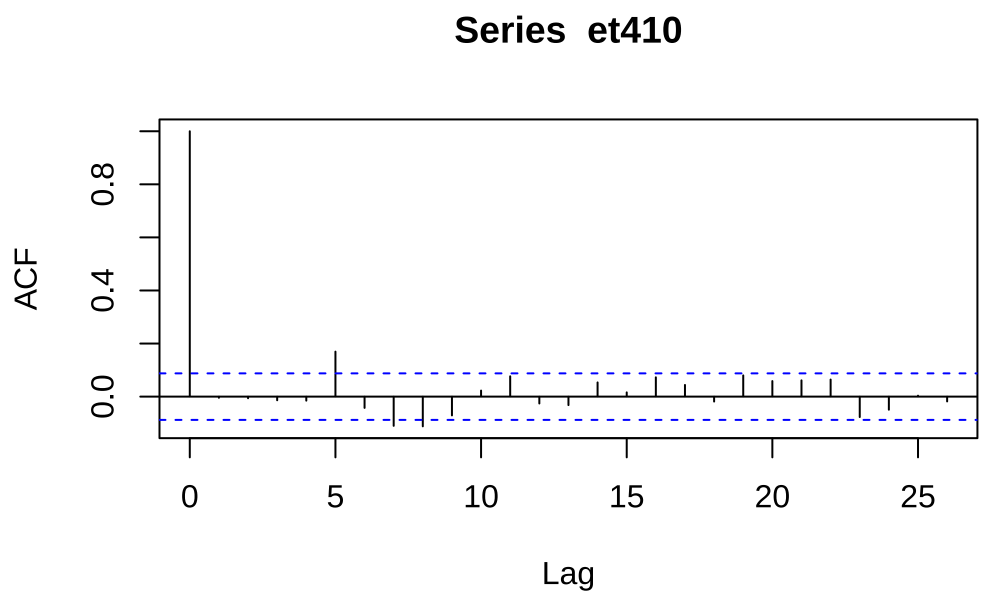
	Models_Compared	DM_Statistics	p_Value
1	ARIMA vs. NNAR (re-estimation)	1.036	3.004223e-01
2	ARIMA vs. NNAR (without re-estimation)	-19.023	2.136618e-75
3	ARIMA (re-estimation) vs. ARIMA (without re-estimation)	6.177	
4	NNAR (re-estimation) vs. NNAR (without re-estimation)	-13.003	

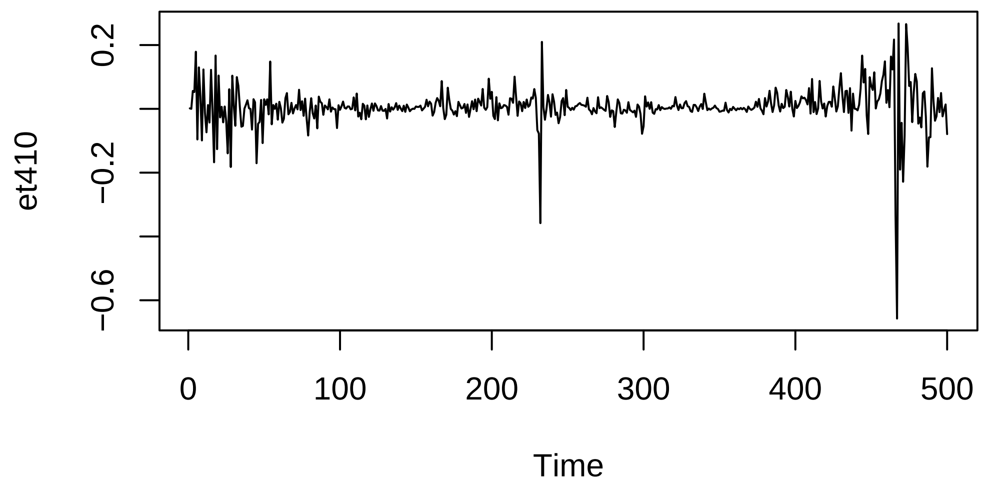
```
3 7.611747e-10
4 1.943571e-37
```

$p < 0.05$ indicates that forecast results of the first method is better than the second method.

Ljung-Box testing for used ARIMA models

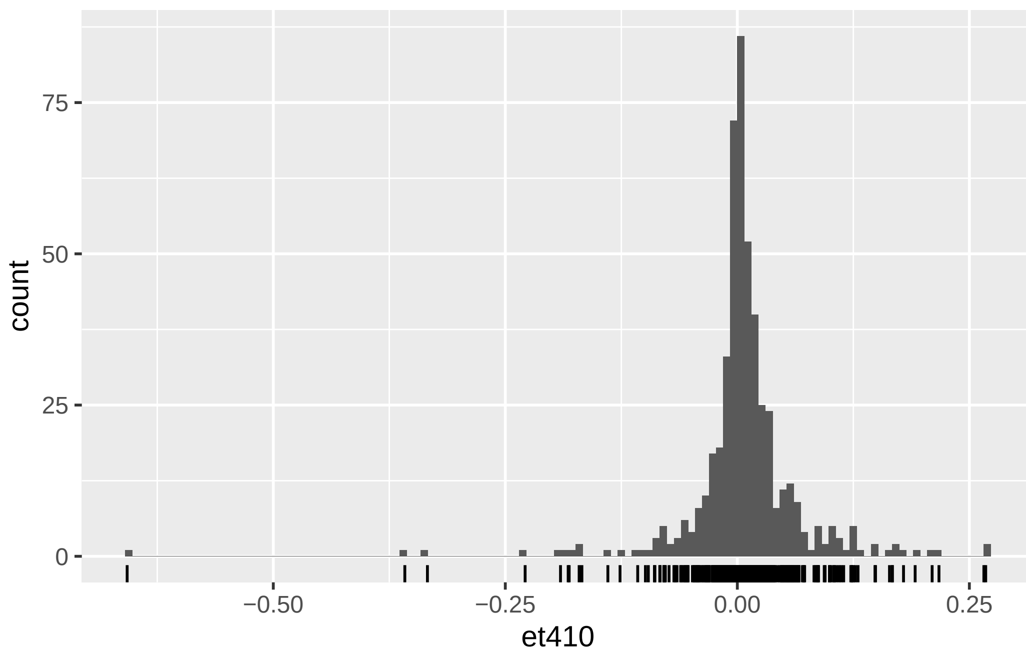


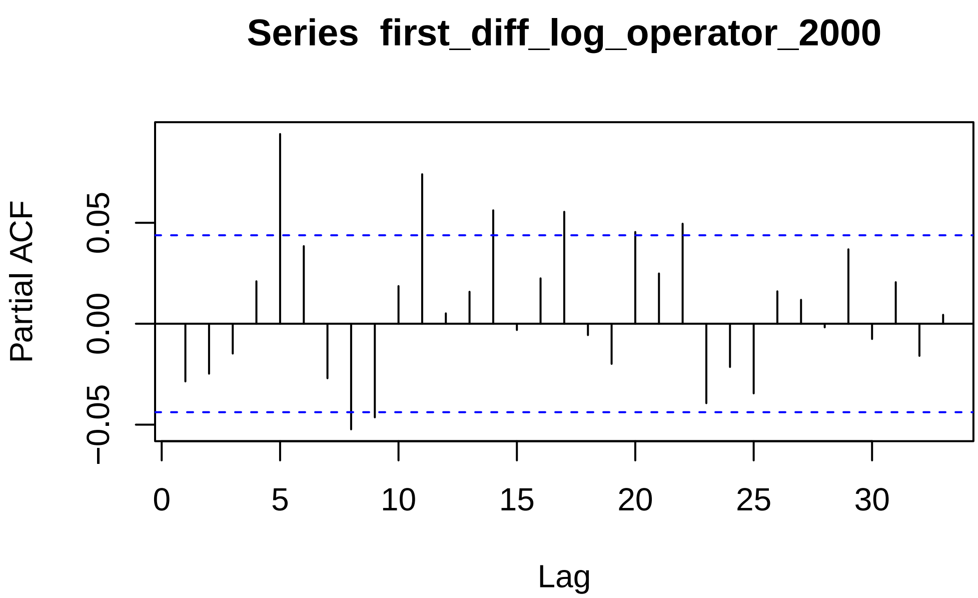


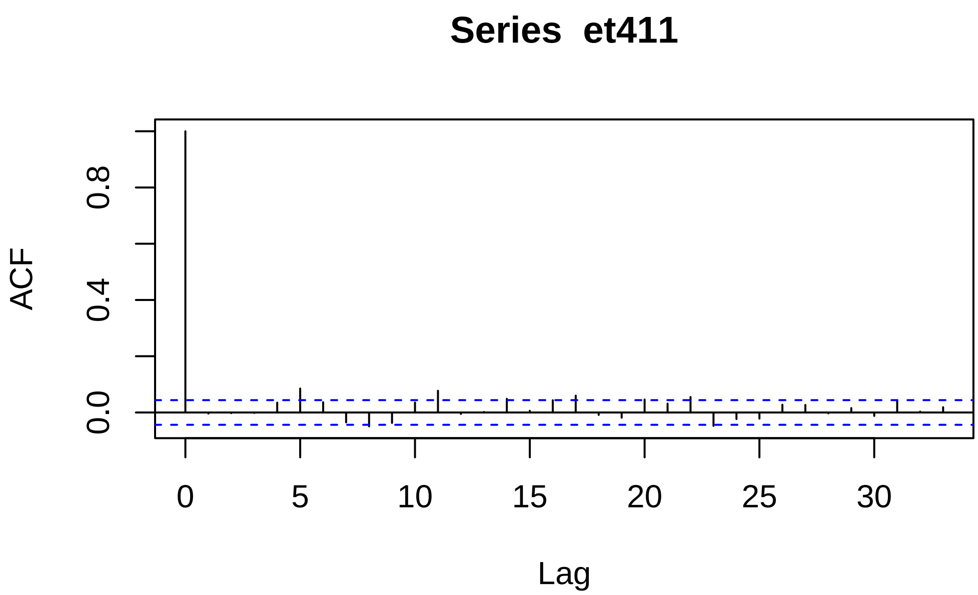


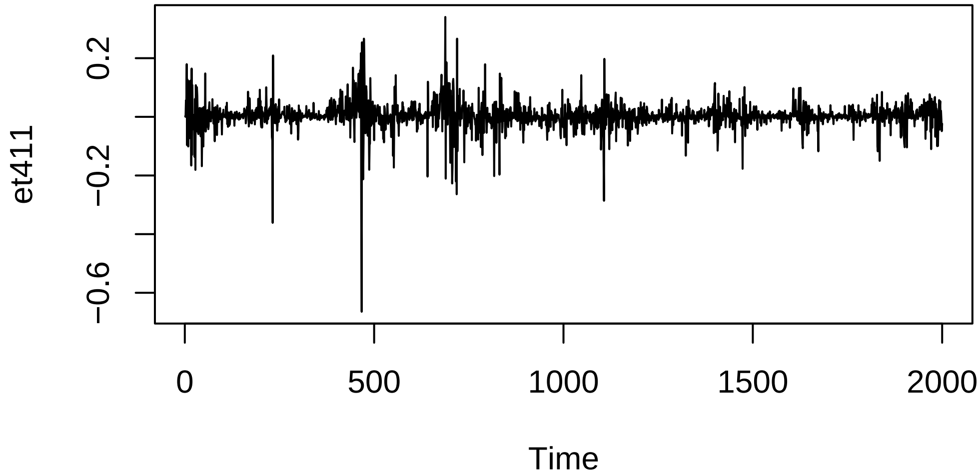

```
Box-Pierce test
```

```
data:  et410
X-squared = 27.863, df = 4, p-value = 0.00001329
```



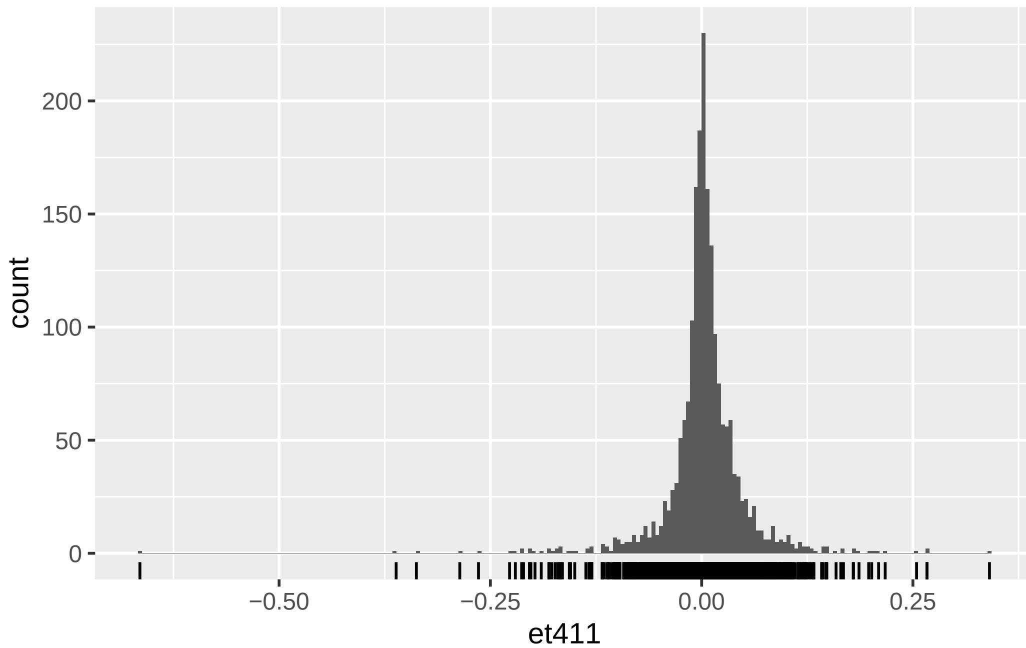






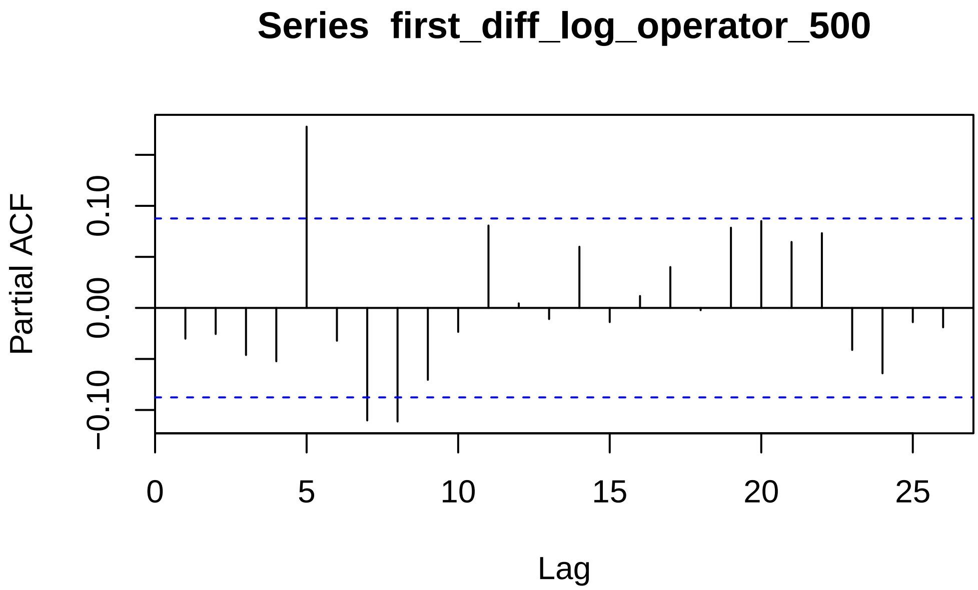

```
Box-Pierce test
```

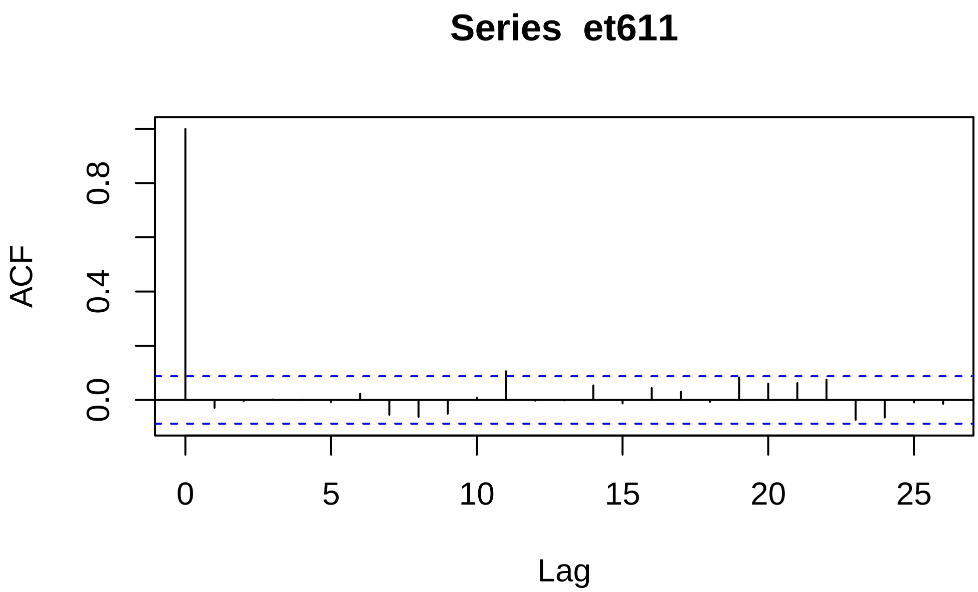
```
data:  et411
X-squared = 27.005, df = 3, p-value = 0.000005873
```

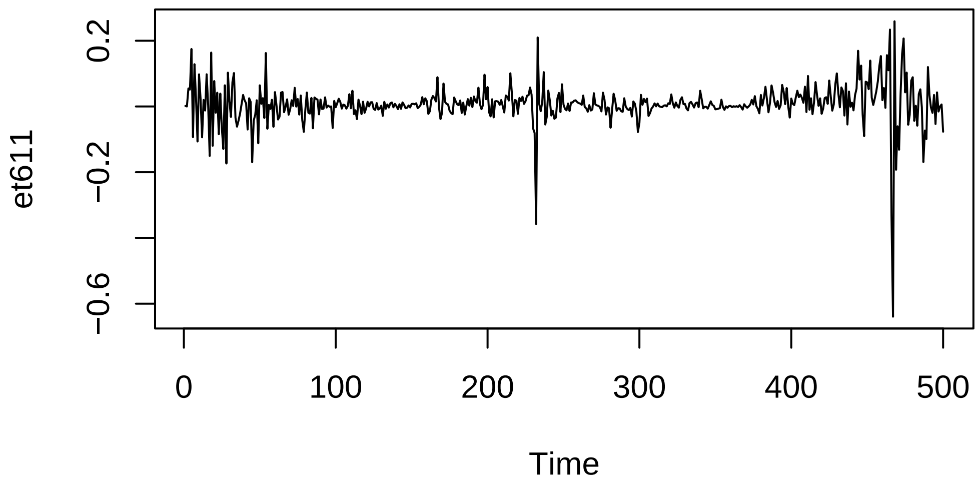


Conclusion

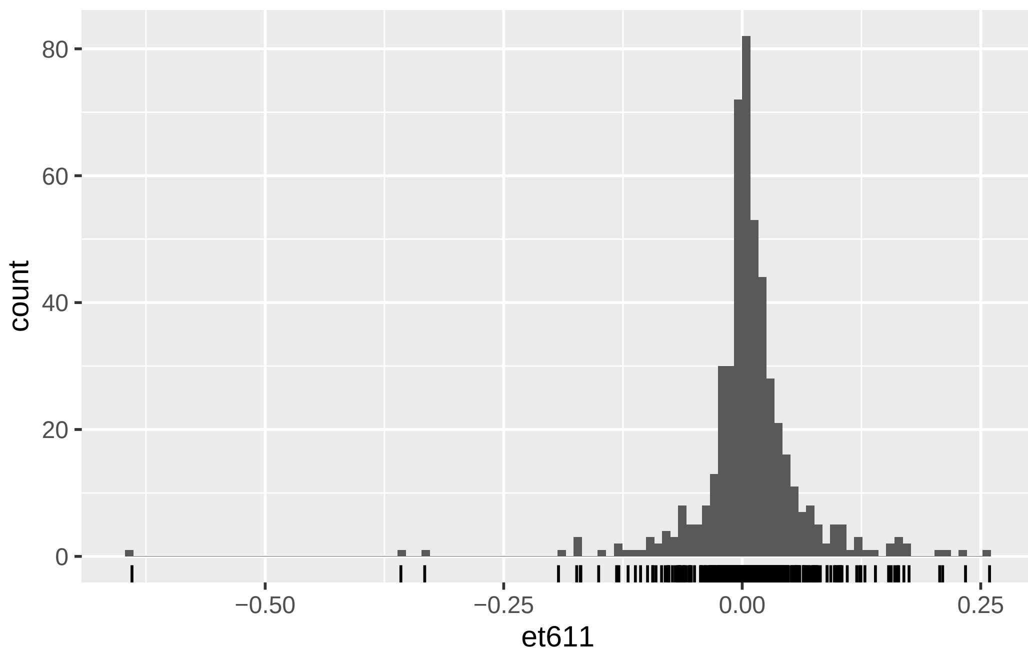
Proposed improved solution for 500 training data set - ARIMA models (6,1,1)


Series first_diff_log_operator_500
Partial ACF
−0.10
0.00
0.10
0
5
10
15
20
25
Lag


Series et611
ACF
0.0
0.4
0.8
0
5
10
15
20
25
Lag

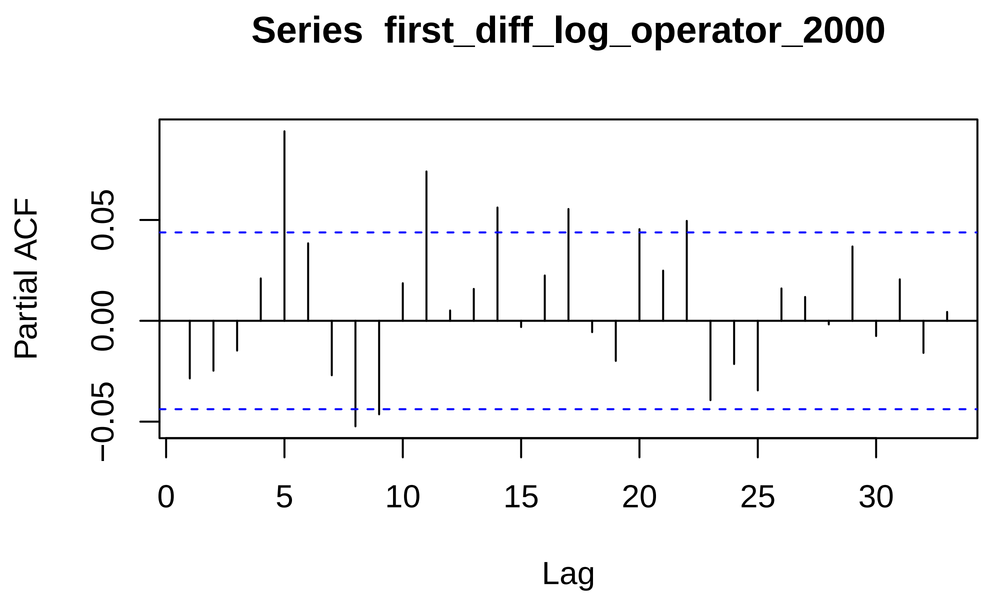


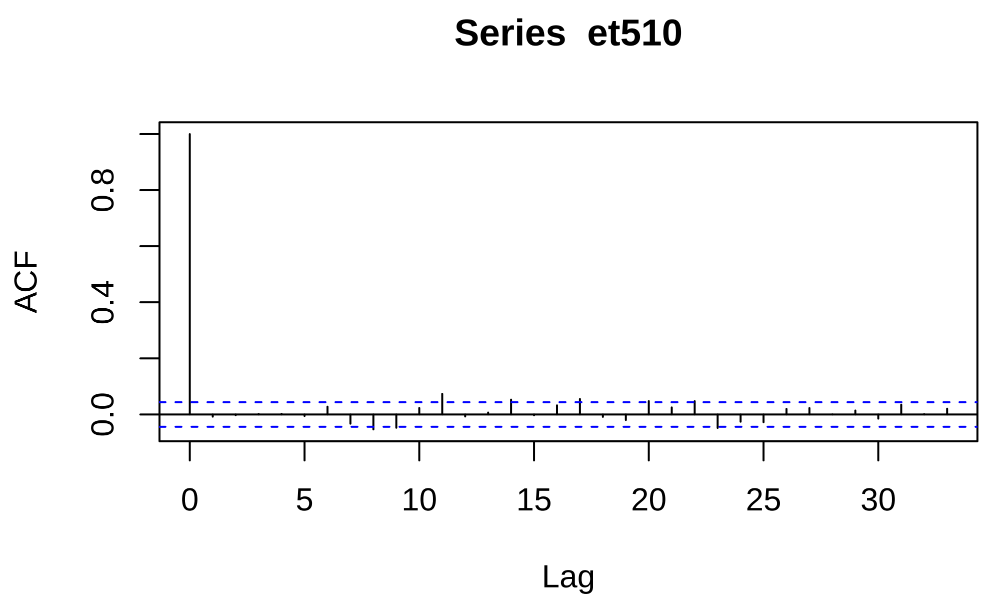

```
    Box-Pierce test

data:  et611
X-squared = 5.5026, df = 3, p-value = 0.1385
```

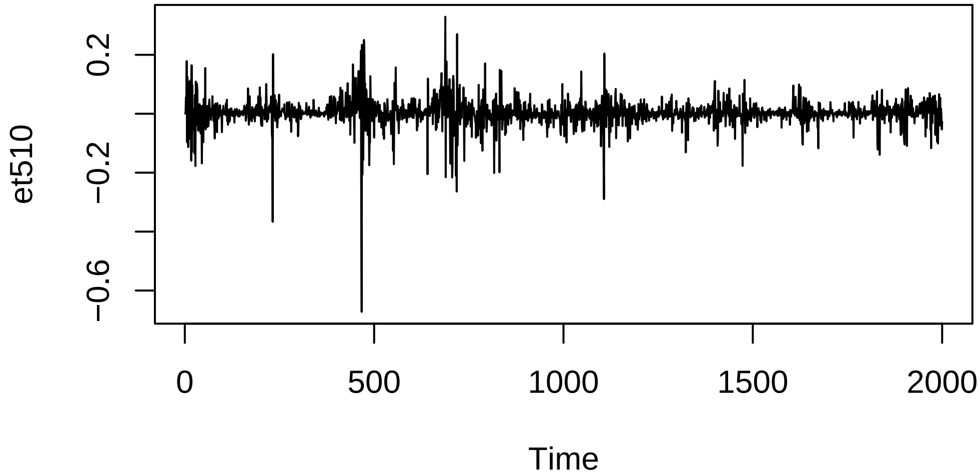



```
                     ME       RMSE        MAE       MPE     MAPE     MASE
Training set 0.006948153 0.06167435 0.03395994 0.2132751 1.364582 1.020184
                    ACF1
Training set -0.02906356
```

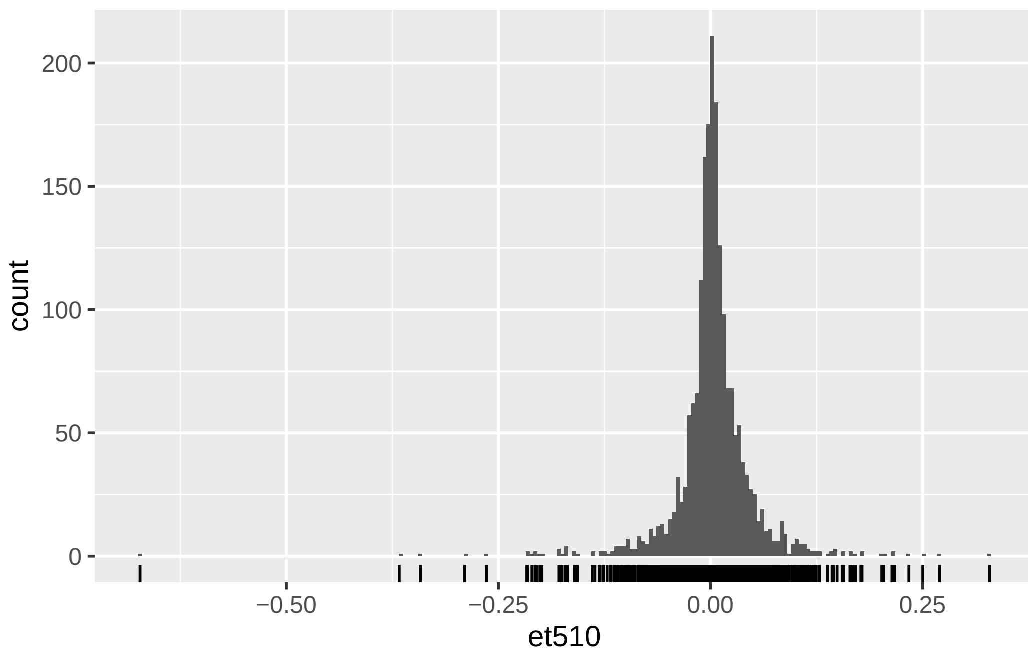
Proposed improved solution for 2000 training data set - ARIMA models (5,1,1)


Series first_diff_log_operator_2000
Partial ACF
0.05
0.00
−0.05
0
5
10
15
20
25
30
Lag


Series et510
ACF
0.8
0.4
0.0
0
5
10
15
20
25
30
Lag




```
    Box-Pierce test

data:  et510
X-squared = 3.942, df = 2, p-value = 0.1393
```




```
                       ME       RMSE        MAE        MPE      MAPE      MASE
Training set 0.002875649 0.04777167 0.02727576 0.06537294 0.6468885 0.9993979
                     ACF1
Training set -0.007808208
```