Time Series Forecasting for currency rate between NZD and CNY

Hang_YU

March 5, 2017

Introduction

In this project, the time series analysis and forecasting is applied on the currency exchange rate between NZD and CNY. Fifteen-months data (01/01/2016 - 01/03/2017) are obtained from Reserve Band of New Zealand (RBNZ) through Quandl. An arima model with fine-grained parameters is applied for time series modeling and forecasting.

Data collection and cleaning

First, the dependent packages are loaded. Quandl is used to collect RBNZ data, Reshape2 is used to tidy up data, Forecast is for prediction, DoParallel is for parallel computing, and TSeries is for time-series stationarity check.

```
library(Quandl)#for data collection
library(reshape2)#to melt
library(forecast)
library(doParallel)
library(tseries)#adf.test for stationary check
```

Data of Exchange rate between NZD and CNY from 01/01/2016 to 01/03/2017 are collected and investigated.

```
ExRate <- data.frame(Quandl("RBNZ/B1_D", collapse = "daily",start_date="2016-1-01",end_date="2017-3-01",type = '
names(ExRate)
```

```
##  [1] "Date"                     "TWI...17.currency.basket"
##  [3] "United.States.dollar"     "UK.pound.sterling"
##  [5] "Australian.dollar"        "Japanese.yen"
##  [7] "European.euro"            "Canadian.dollar"
##  [9] "South.Korean.won"         "Chinese.renminbi"
## [11] "Malaysian.ringgit"        "Hong.Kong.dollar"
## [13] "Indonesian.rupiah"        "Thai.baht"
## [15] "Singapore.dollar"         "New.Taiwan.dollar"
## [17] "Indian.rupee"             "Philippine.peso"
## [19] "Vietnamese.dong"          "Base.June.1979.100"
```

We first check if there is any NA observations.

```
any(is.na(ExRate))
```

```
## [1] FALSE
```

The initial data frame is not in a tidy format. Therefore, ExRate data frame is reshaped to contain only "date","currency", and "rate" columns using melt() in Reshape2.

```
ExRate <- melt(ExRate,id="Date")
names(ExRate) <- c("Date","Currency","Rate")
```

At last, only CNY data is extracted and transferred to time series with a frequency of 30 (monthly), as a new data frame DT.

```
ExRate <- subset(ExRate,Currency=="Chinese.renminbi",select = c("Rate"))
DT_total <- ts(ExRate,frequency = 30)
```

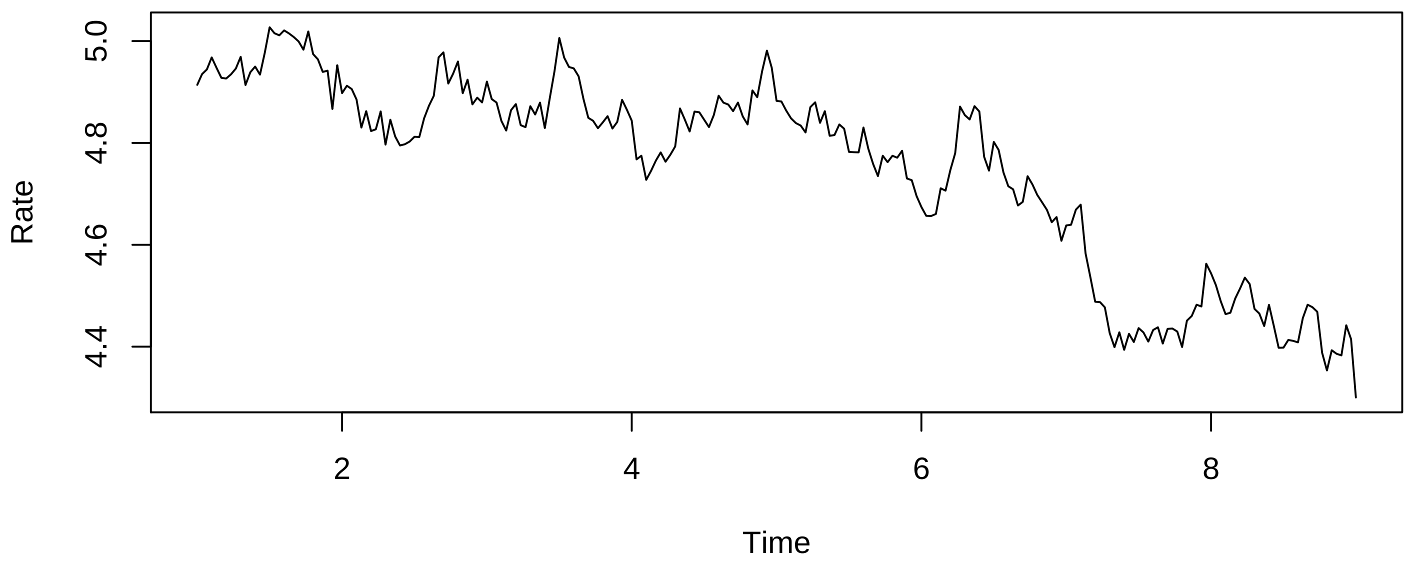
Exploratory data analysis

The DT data is splitted into a training set and a test set.

```
DT <- window(DT_total,start=1,end=9)
DT_test <- window(DT_total,start=9)
```

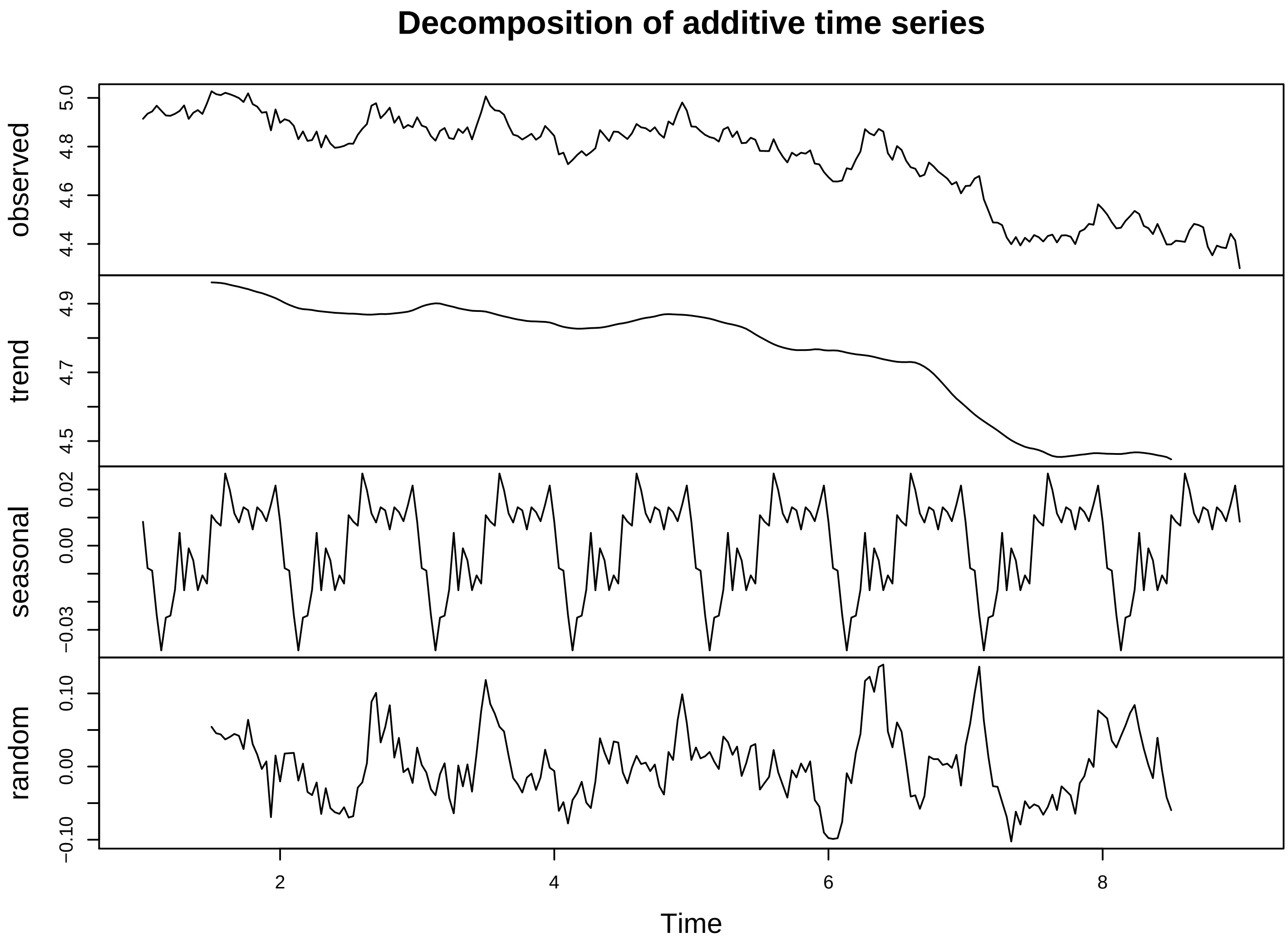
To explore the composition of time series (seasonality, trend, random, and cycle), we first plot the decomposed components of DT. In order to achieve an accurate decomposition, we first plot DT to see if the components show an additive feature.

```
plot(DT)
```



Since the variance along time line does not show an obvious relation with currency values, we choose the additive model for decomposition.

```
plot(decompose(DT,type="additive"))
```



From the plots, DT data shows down trend for the whole year and strong seasonality every month.

Time series modeling

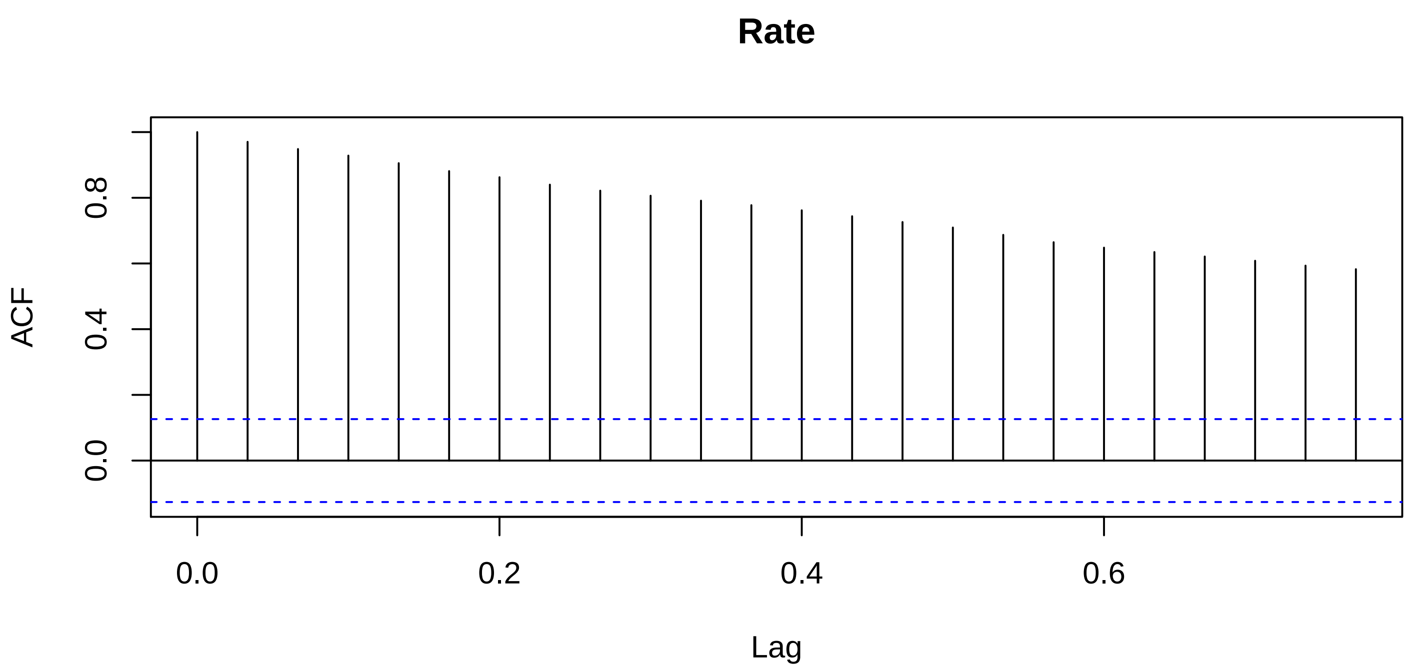
Then, we commence the forecasting for DT. SARIMA (Seasonal AutoRegressive Intergreted Moving Average) is adopted here due to the seasonality observed. We first check the stationarity of DT using adf.test().

```
adf.test(DT)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  DT
## Dickey-Fuller = -2.4256, Lag order = 6, p-value = 0.3968
## alternative hypothesis: stationary
```

Since $p > 0.05$, we accept the null hyphothesis of non-stationarity. This is as we observed before due to the trend of data. Another way to check stationarity is to plot the acf, which shows the autocorrelation of time lags, of DT.

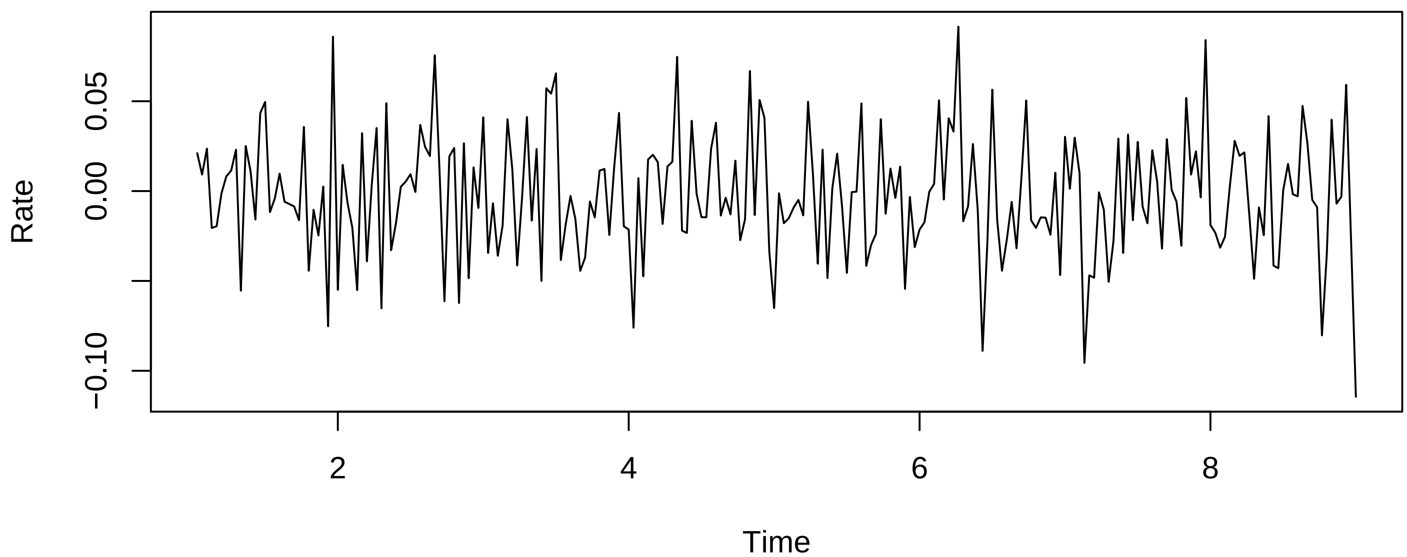
```
acf(DT)
```



Once it shows a slowly decreased trend it indicates non-stationarity that always comes with correlated lags.

To make the time series stationary, we apply first-order differencing, after which DT becomes stationary. Therefore, d in ARIMA becomes 1.

```
plot(diff(DT))
```



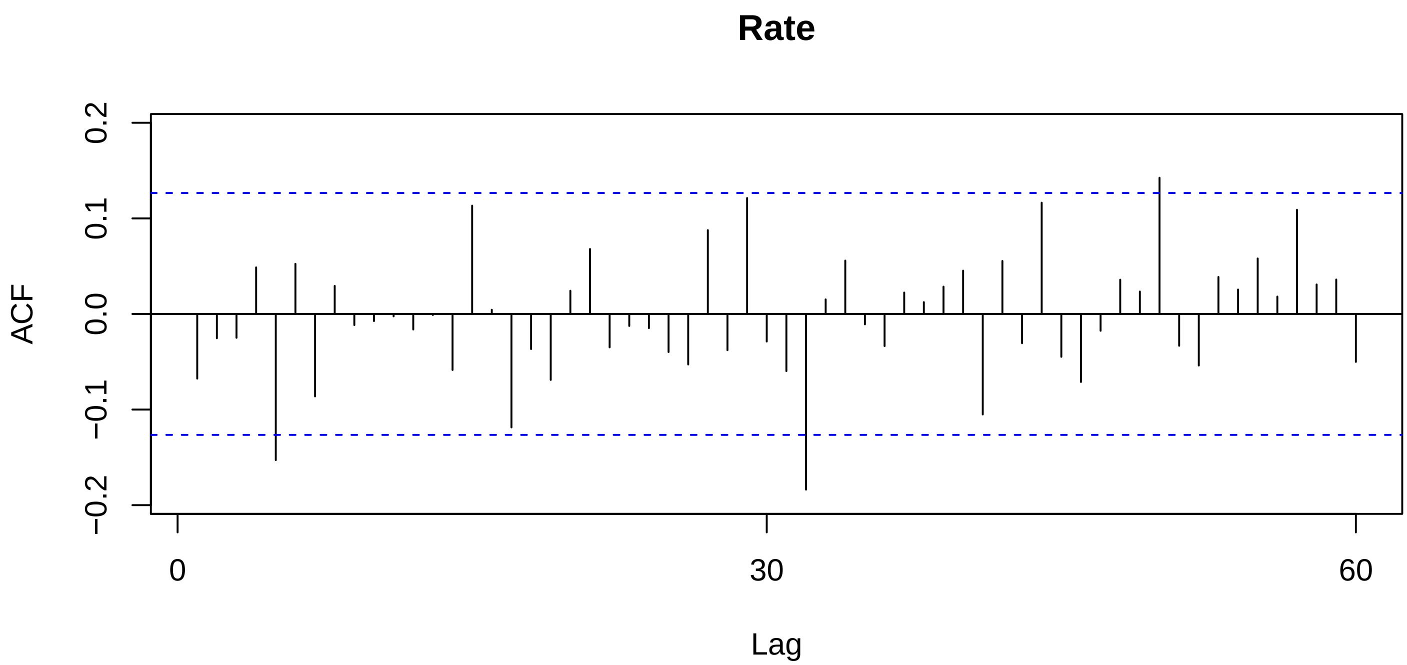

```
adf.test(diff(DT))
```

```
## Warning in adf.test(diff(DT)): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
```

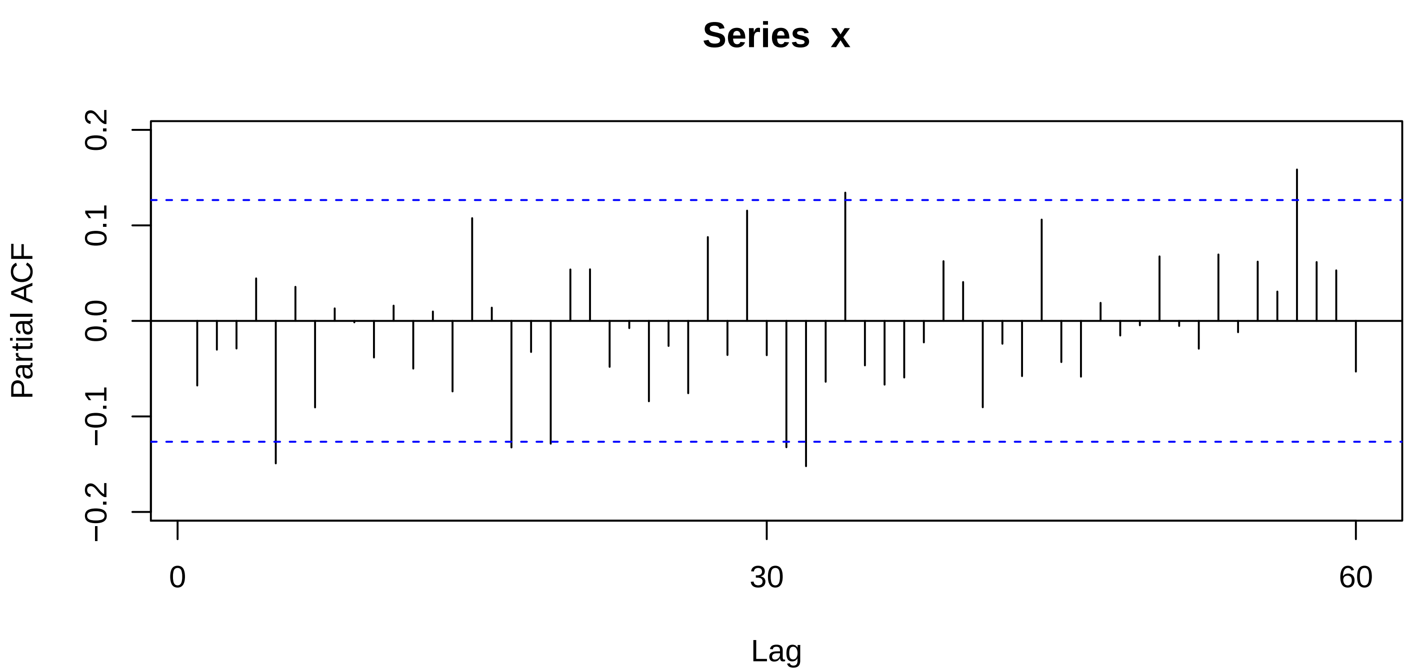
```
##
## data:  diff(DT)
## Dickey-Fuller = -6.7199, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
Acf(diff(DT))
```



No spikes after 0 of acf(), which suggests that the current point (0) is not significantly related to previous lags, implies an order of 0 (q=0) for the MA part of ARIMA. To determine the order of AR part (p), we plot pacf which tells the pure correlation between the current point and the lags that is not explained by previous lags.

```r
Pacf(diff(DT))
```



As shown in the graph, pacf() basically shows an uniformly distributed correlation with no significant spikes along time lags, which indicates p=0.

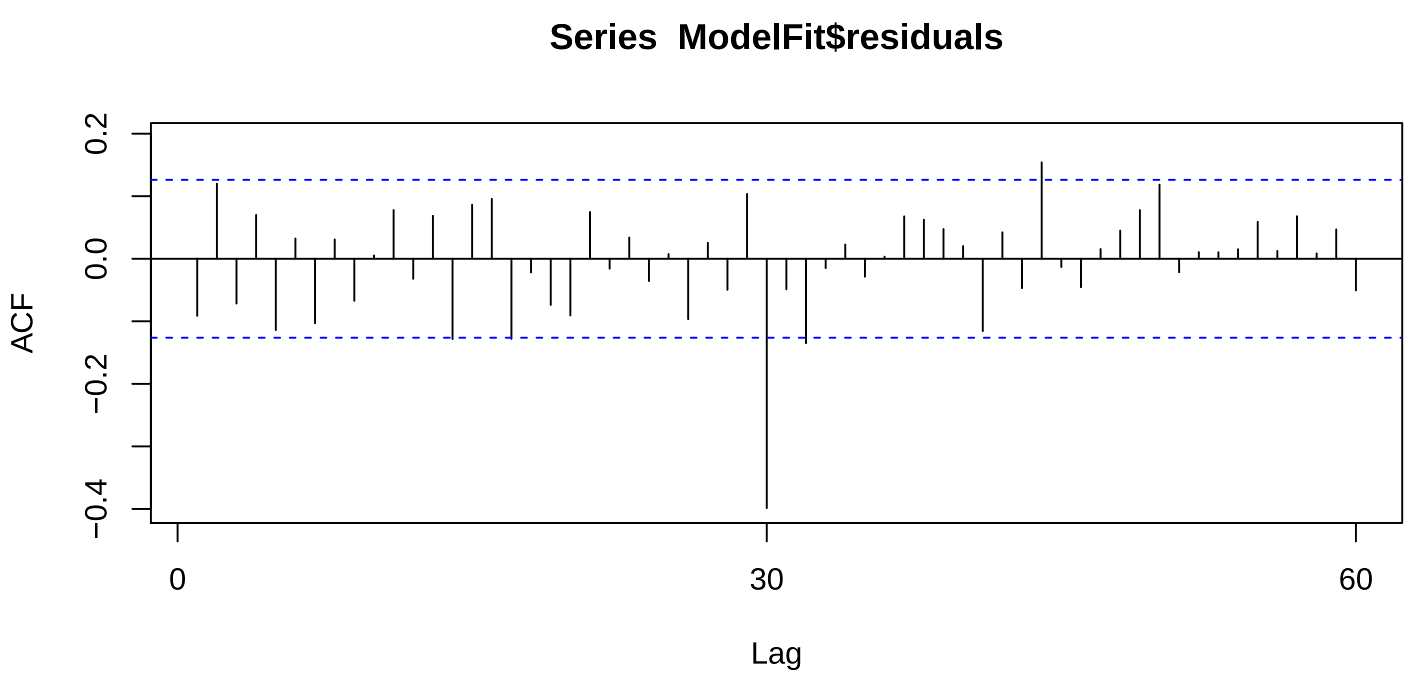
At this stage, our model is determined to be ARIMA(0,1,0) with seasonal components.

```
ModelFit <- Arima(DT,c(0,1,0),seasonal = c(0,1,0))
summary(ModelFit)
```

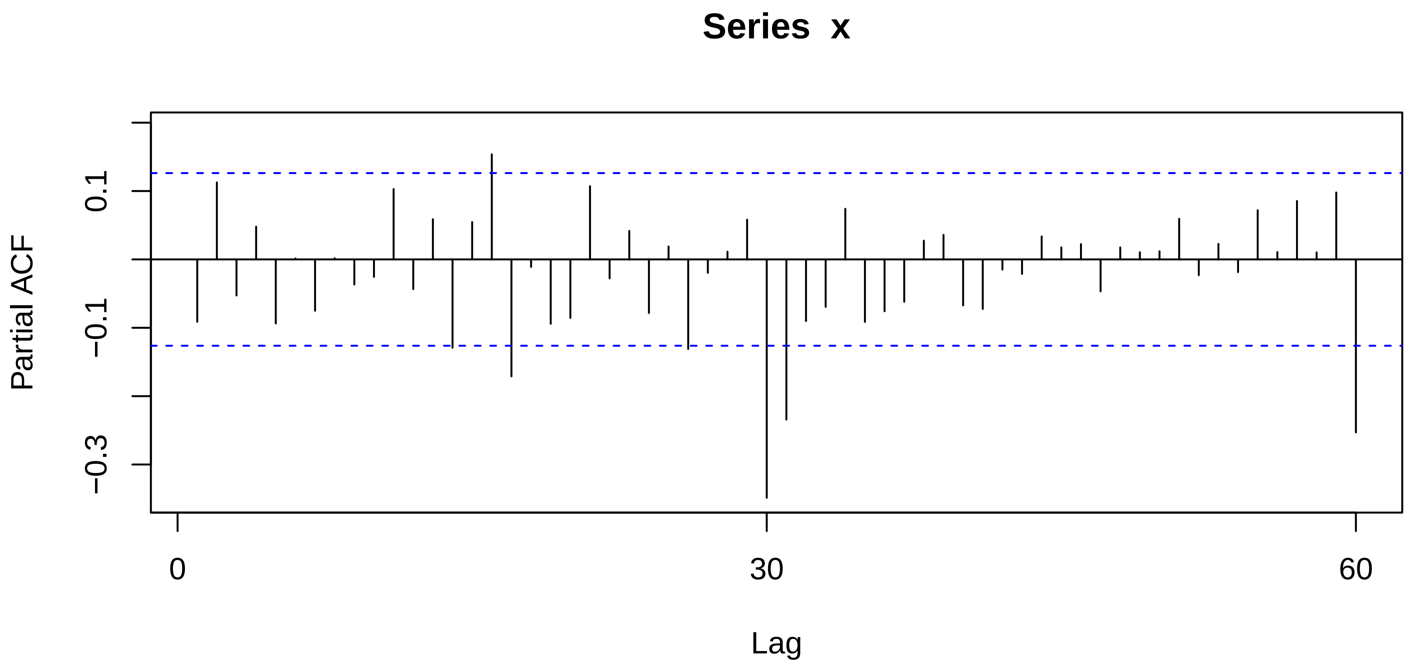
```
## Series: DT
## ARIMA(0,1,0)(0,1,0)[30]
##
## sigma^2 estimated as 0.002351:  log likelihood=337.76
## AIC=-673.53   AICc=-673.51   BIC=-670.18
##
## Training set error measures:
##                        ME       RMSE       MAE         MPE      MAPE
## Training set -0.001026109 0.04525664 0.0339917 -0.02443497 0.7207453
##                   MASE        ACF1
## Training set 0.330091 -0.09118602
```

We check the residuals:

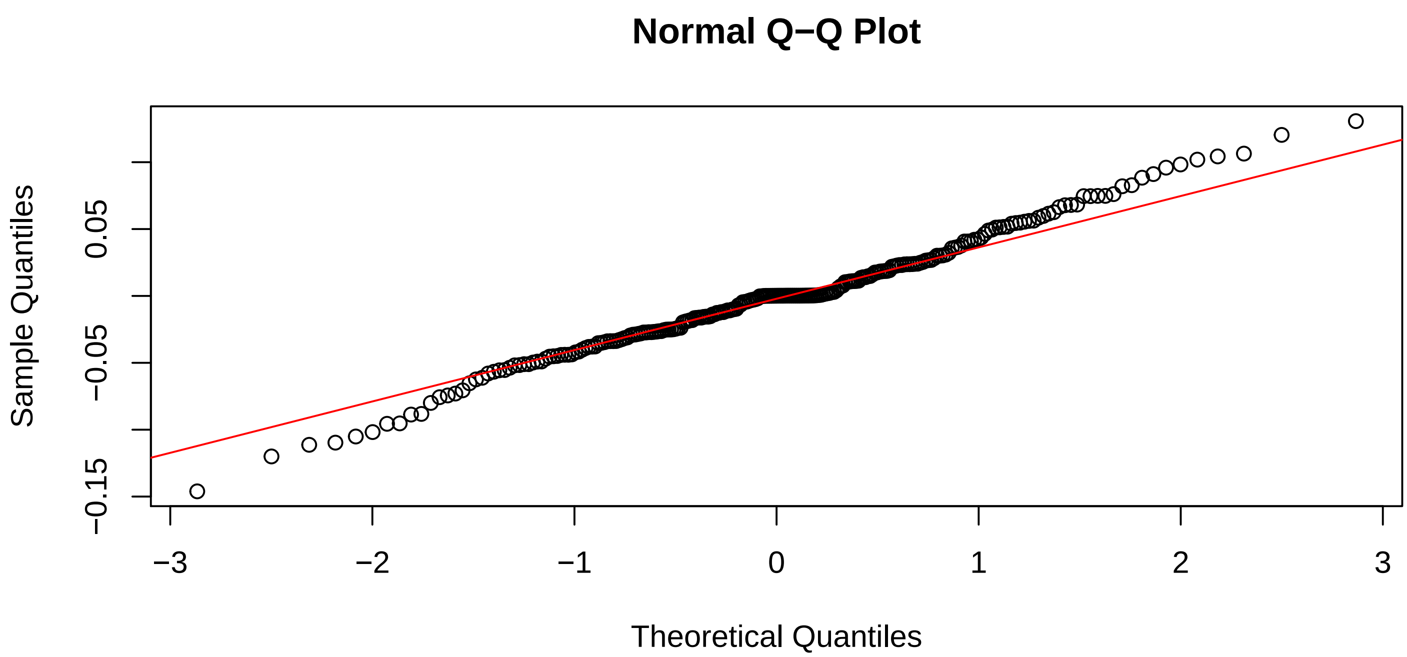
```
Acf(ModelFit$residuals)
```




```
Pacf(ModelFit$residuals)
```




```
qqnorm(ModelFit$residuals)
qqline(ModelFit$residuals,col="red")
```



Residuals are uniformly along lags like white noise. Besides, they are normally distributed. By checking Box.test (H0: cor(residuals)=0), it is proven to be a good model with unrelated residuals along time.

```
Box.test(ModelFit$residuals)
```

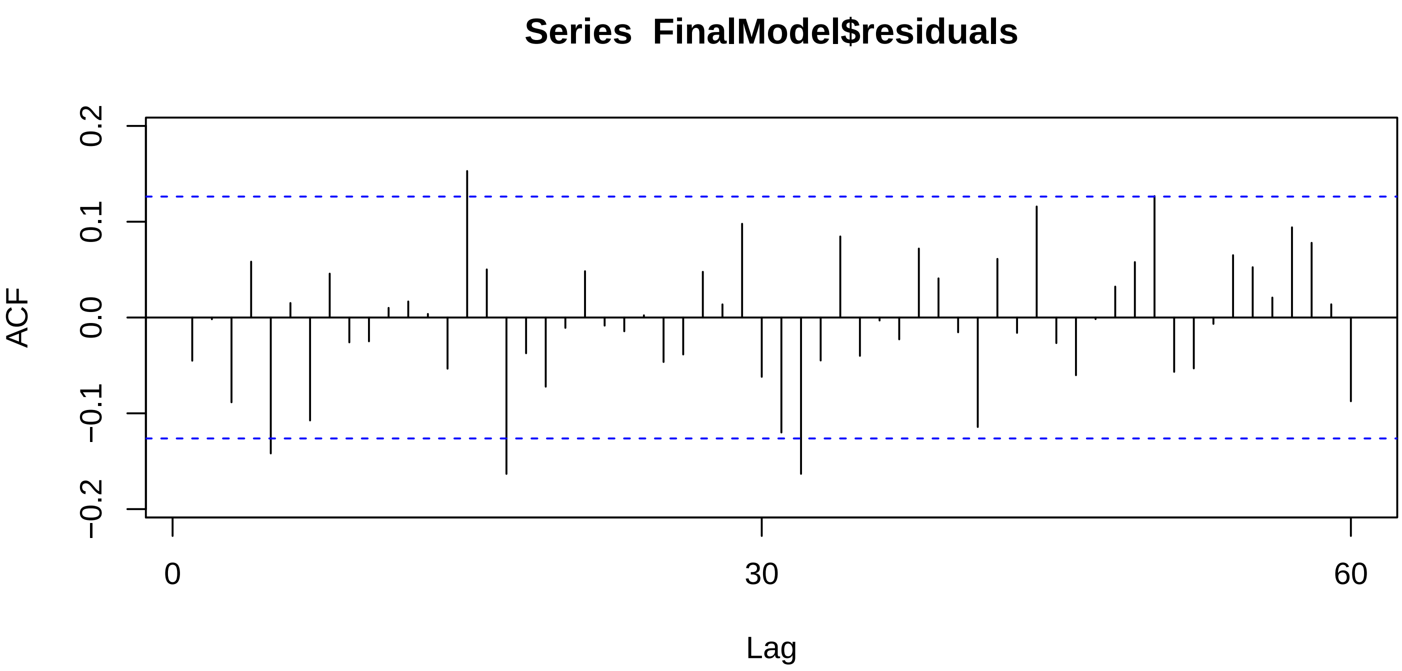
```
##
##  Box-Pierce test
##
## data:  ModelFit$residuals
## X-squared = 2.0039, df = 1, p-value = 0.1569
```

To verify the final model, we use arima.auto() with seasonality which select the model with the lowest AIC among all models with different combinations of p,d, and q.

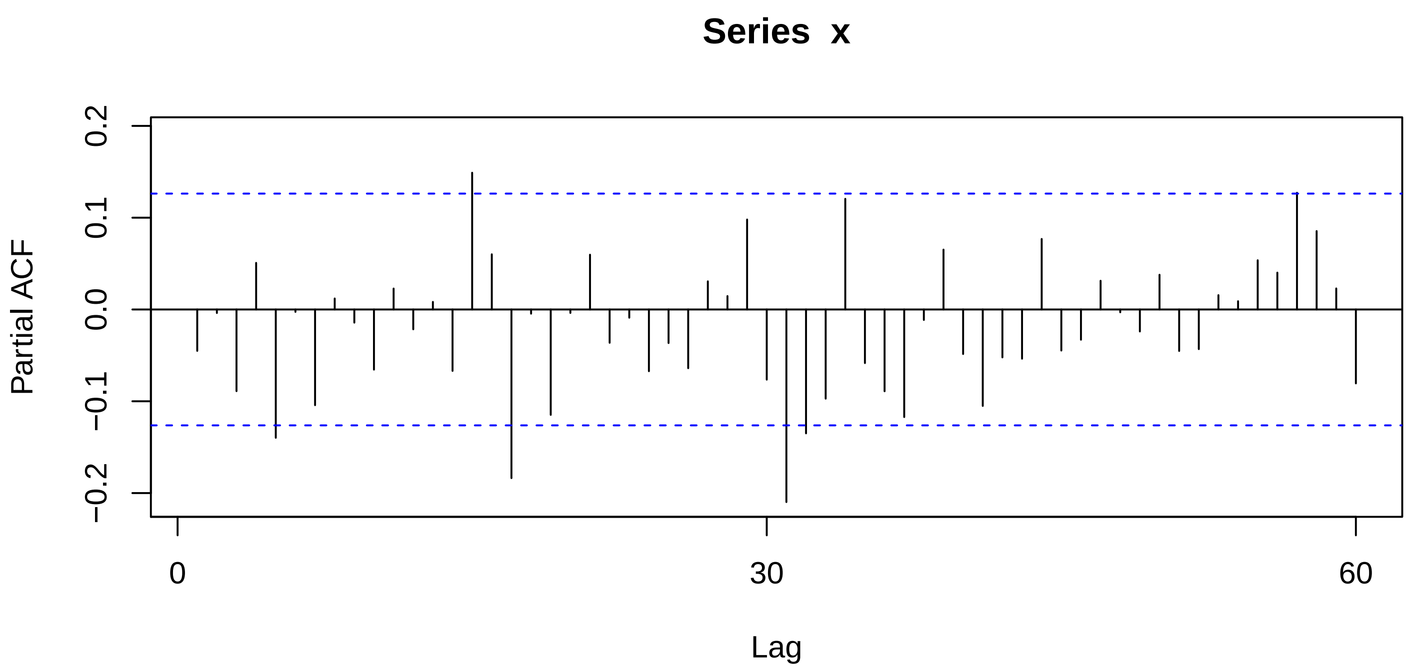
```
FinalModel <- auto.arima(DT,seasonal = TRUE,max.order=10,D=1)
summary(FinalModel)
```

```
## Series: DT
## ARIMA(0,1,0)(2,1,0)[30]
##
## Coefficients:
##          sar1     sar2
##       -0.6288  -0.3571
## s.e.   0.0710   0.0779
##
## sigma^2 estimated as 0.001626:  log likelihood=369.84
## AIC=-733.67   AICc=-733.55   BIC=-723.63
##
## Training set error measures:
##                        ME       RMSE        MAE         MPE      MAPE
## Training set -0.001300539 0.03745858 0.02857493 -0.03017577 0.6052389
##                   MASE        ACF1
## Training set 0.2774891 -0.04508099
```

```
Acf(FinalModel$residuals)
```




```
Pacf(FinalModel$residuals)
```

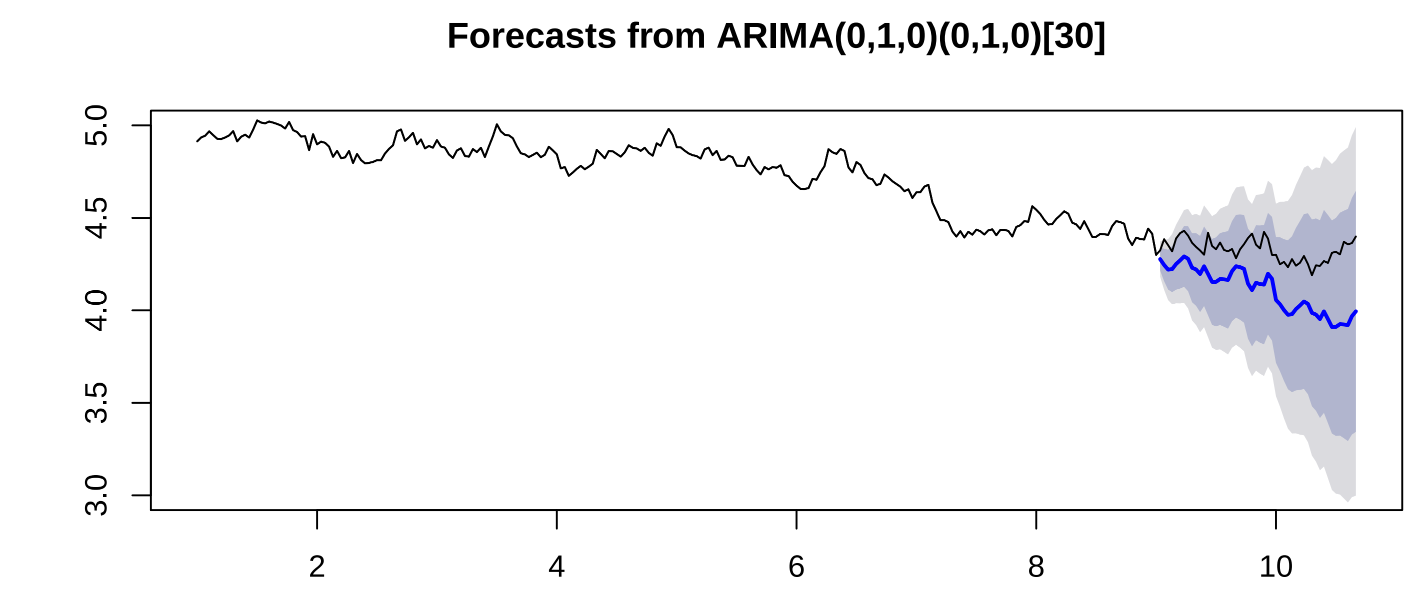


Auto ARIMA drops the seasonal components since it selects models based on AIC. However it does not mean that the optimal model contains no seasonality.

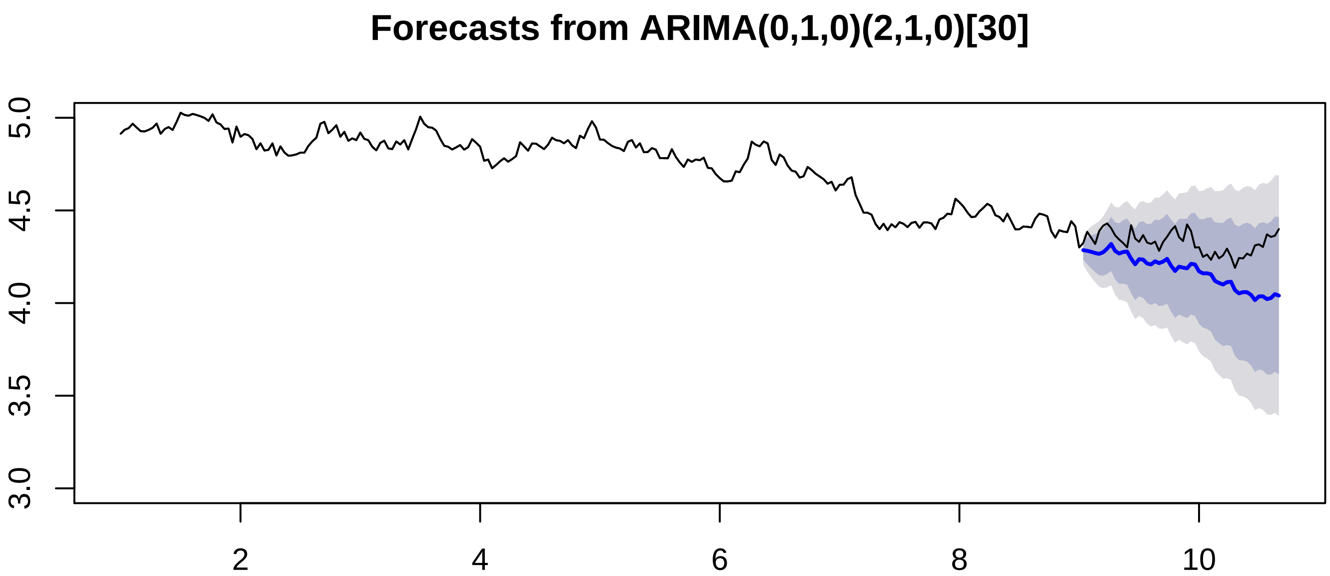
Forecasting

At last, we forecast the currency rate using both ModelFit and FinalModel.

```
plot(forecast(ModelFit,h=50),ylim=c(3,5));lines(DT_total)
```




```
plot(forecast(FinalModel,h=50),ylim=c(3,5));lines(DT_total)
```




```r
accuracy(forecast(ModelFit),DT_test)
```

```
##                         ME       RMSE       MAE         MPE      MAPE
## Training set -0.001026109 0.04525664 0.0339917 -0.02443497 0.7207453
## Test set      0.217282000 0.23991756 0.2172820  5.02391459 5.0239146
##                   MASE        ACF1 Theil's U
## Training set 0.330091 -0.09118602        NA
## Test set     2.110010  0.79230940  5.549407
```

```r
accuracy(forecast(FinalModel),DT_test)
```

```
##                         ME       RMSE        MAE         MPE      MAPE
## Training set -0.001300539 0.03745858 0.02857493 -0.03017577 0.6052389
## Test set      0.153251528 0.17354376 0.15325153  3.53695229 3.5369523
##                    MASE        ACF1 Theil's U
## Training set 0.2774891 -0.04508099        NA
## Test set     1.4882145  0.73326239  4.005613
```

Due to the high accuracy, we choose the model selected by auto.arima() as the final model.