COLLEGE OF BUSINESS
LeBow
Drexel University

Business Conditions and Forecasting
FIN-642-001
Professor: Merrill Liechty, Ph.D.

FORECASTING CHINA / U.S. FOREIGN EXCHANGE RATE

Group 8
Linh Truong, Han Xiao, Yue 'Alex' Fu,
Quang Ho, Zhicheng Liu

Date: June 11, 2020

I. Introduction

As the relationship between China and the United States has never been out of date as a topic of interest for many discussions and studies, this project aims to exercise various time series methods and focuses on China / U.S. foreign exchange rate after China became the second-largest economy in the world. The exchange rate measures the currency's value of one country against another. Over time, the rate changes due to various factors such as supply and demand in the market, the difference between two economies' systems and growth via nominal interest rate, and inflation, etc. Despite the limitations, it is not uncommon to apply time series analysis towards the forecasts of the currency exchange rate.

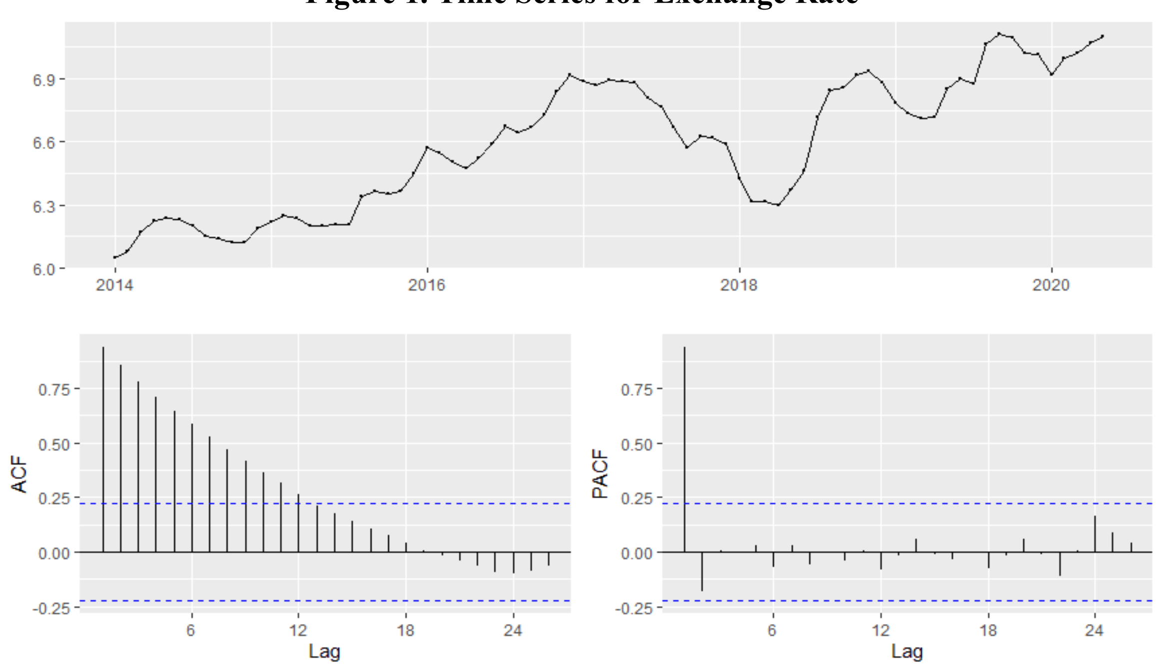
II. Data Description & Exploration

The China / U.S. foreign exchange rate dataset is obtained from the Federal Reserve Bank of St. Louis. Originally, the forex quote is the daily buying rate for cable transfers payable in New York at noon. This rate is stated as the price of one U.S. Dollar (USD) in terms of the Chinese Yuan (CNY). However, to avoid non-trading days, our dataset adjusts to the monthly average from January 2014 to May 2020.

From the time plot (Figure 1), there is a noticeable upward trend during 2014-2017 and after 2018. The seasonal pattern does not show up quite as clearly. Given the short time length, which is less than ten years, of our dataset, cyclicality is hardly apparent here. Our autocorrelation coefficient (ACF) decreases as the number of lags increases. There are significant and positive autocorrelations for small lags, which also indicates that this time series contains a trend and is not stationary. To use an ARIMA model, we need to make the data stationary through differencing later. Our Partial Correlation Function (PCAF) has only a significant spike at lag 1. Once again, the non-stationary process is confirmed.

Our time series decomposition shows the strength of our trend and seasonality. The strength of the trend is validated with FT close

Figure 1. Time Series for Exchange Rate

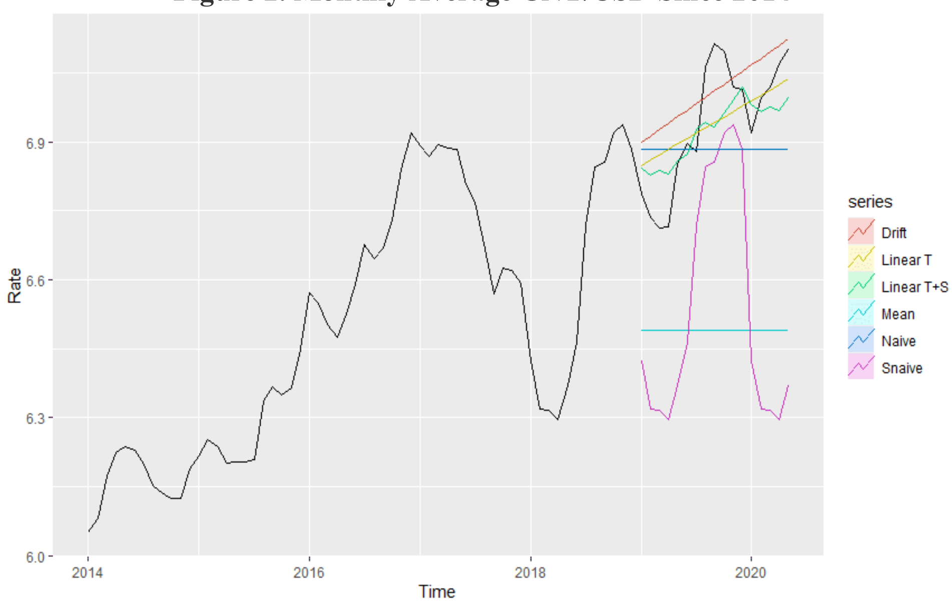


to 1 at 0.9319. Meanwhile, FS with a value of 0.1758, which suggests a weak seasonal component. Since the *BoxCox.lambda* function in R calculates λ to be approximately 2, we believe that it is not necessary to perform any transformations. For model assessment later, we will split our data into two sets: training and test, with the percentage of 78 and 22, respectively. The training dataset consists of monthly average forex rates in 2014-2018. In the test set, there are 17 months of data which starts in 2019. To develop a forecasting model, we have tried various approaches which can be classified into three main categories: simple benchmarks, exponential smoothing, and ARIMA.

III. Forecasting Model

A) Simple Forecasting Methods:

Figure 2. Monthly Average CNY/USD Since 2014

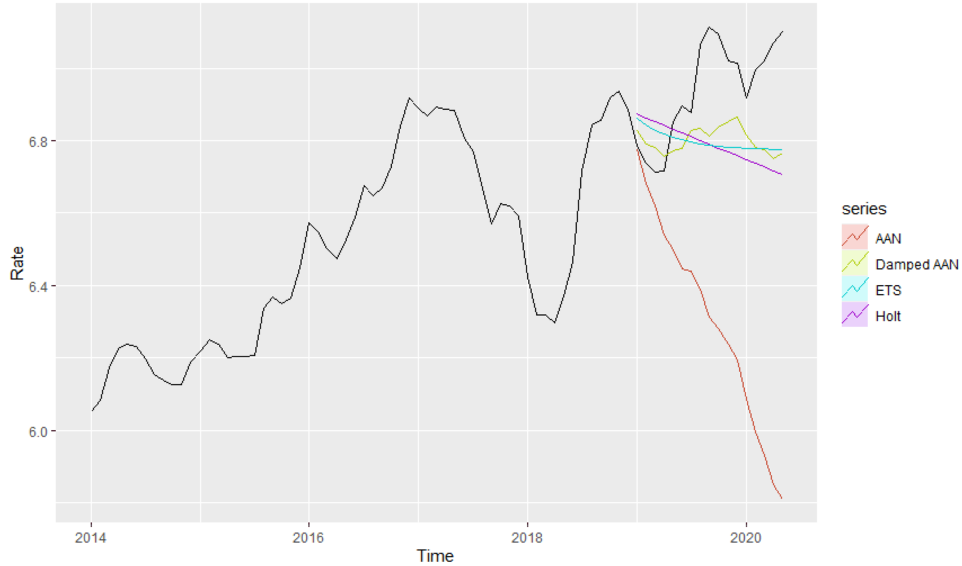


Some simple forecasting practices are average, naïve, seasonal naïve, and drift methods. The average method plainly assigns the mean of historical data to future forecasts. While naïve forecasts are based on the last observation, seasonal naïve forecasts use the last value of the same month in the previous year instead, and the drift method allows naïve values to change over time. As mentioned above that our forex dataset has a strong trend component, and vague seasonality, we run two different time series regression models: one contains only a trend, the other includes both trend and season.

From Figure 2, the average, naïve, drift, and regression applied to trend appear in a linear movement. Mean and naïve forecasts are flat, but mean is much below the actuals than naïve forecasts. Linear trend and drift methods can capture the upward tendency. On the other hand, linear regression applied to trend and season can layout both ups and downs during the overall escalation. Aside from underprediction, seasonal naïve mimics the course of the forex rates quite well. To measure how close these forecasts are to the observed values, we compute the RMSE, MAE, MAPE, and MASE. The comprehensive comparison will be presented in Section IV, along with other methods. As a result, the linear regression with trend and season as predictors gives the lowest errors.

B) Exponential Smoothing Model:

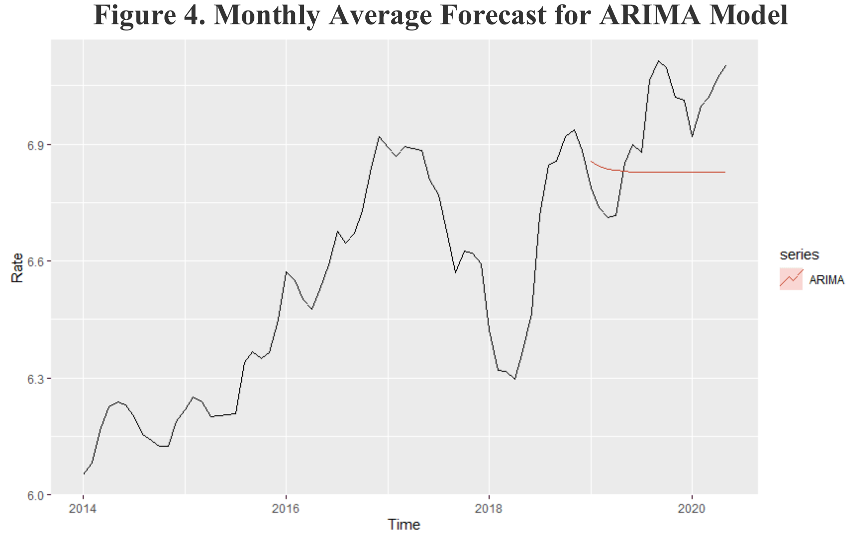
Figure 3. Monthly Average Forecast for Exponential Smoothing Model



With rules and regulations changing through the years, there is a possibility that the more current our historical data is, the more relevant it is to forecast the future. Exponential smoothing can help deal with this. For data with a trend, we think that Holt's linear pattern, additive damped trend methods, and ETS model can be appropriate. However, based on Figure 3, all of them go downwards, which do not resemble the actuals at all. This result is to be expected because we already know our data has a strong trend, but with weak seasonality, and ETS model has been proven with poor performance in this kind of scenario. The weak performance might be due to the forex rates decrease several months before our forecasting period, and exponential smoothing takes that into account with more weights. The damped AAN model looks like the most accurate one here, and the numbers of RMSE, MAE, MAPE, and MASE in Section IV also agree with that.

C) ARIMA Model:

Figure 4. Monthly Average Forecast for ARIMA Model



ARIMA model is another popular time series forecasting method along with exponential smoothing and focuses on the autocorrelations. We used the *auto.arima* function in R and obtained an ARIMA(1,1,0) model, where the value of the autoregressive term and the non-seasonal difference are ones, and the value of lagged forecast error is 0. It is also equivalent to a differenced first-order autoregressive model, which reflects the weak seasonality feature of our data. The model has a minimum of AICc(-161.66).

IV. Performance Comparison

Table 1: Model Performance Matrix

Section	Forecast Method	RMSE	MAE	MAPE	MASE
Simple	Mean	0.4701	0.4499	6.4460	1.6261
	Snaive	0.4594	0.4070	5.8643	1.4710
	Naïve	0.1477	0.1302	1.8685	0.4705
	Drift	0.1150	0.0983	1.4311	0.3553
	Linear T+S	0.0902	0.0757	1.0899	0.2738
	Linear T	0.0968	0.0787	1.1381	0.2846
Exponential	Damped AAN	0.1918	0.1635	2.3301	0.5907
	AAN	0.7533	0.6413	9.1441	2.3179
	Holt	0.2340	0.2067	2.9543	0.7472
	ETS	0.2121	0.1895	2.7068	0.6847
ARIMA	ARIMA (1,1,0)	0.1782	0.1582	2.2610	0.5716

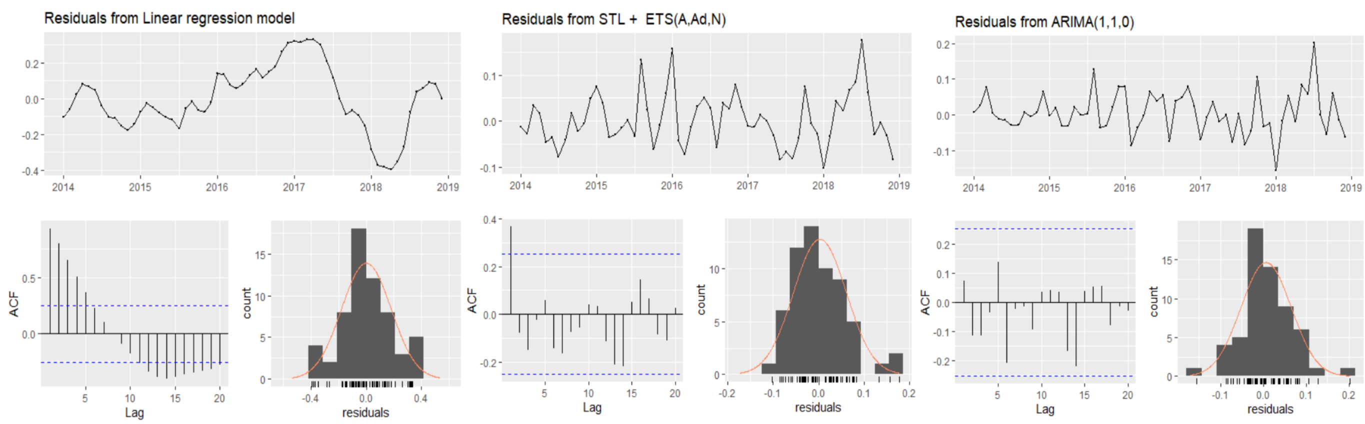
In this section, we present the general results of all forecast methods under four different measurements. As you can see in Table 1, the Linear regression model with trend and seasonality provides the best performance compared with other simple forecast methods. But a linear regression model only with the trend has a close performance, which could be explained by the weak seasonality of our data. Among all exponential smoothing models, Damped AAN outperforms other approaches with the lowest errors. However, as we mentioned in the previous section, exponential smoothing models have been proven to have poor performance on data with weak seasonality. Thus, it's not surprising that most of them can't well capture the variation of our data. For the ARIMA model, it has similar performance with Damped AAN and can't accurately reflect the variations of the data, which could be explained by the stationary of our data since the ARMIA model can essentially capture the linear relationship. We choose the best performing method among each section and compare their forecast accuracy and errors in the next part.

V. Model Selection

To further explore how good the forecasting method is, we also check whether the residuals of linear regression, damped AAN, and ARIMA resemble white noise. For linear regression, the lags are statistically significant with a decreasing trend, and the variance of residuals is inconsistent as well, which means the residuals don't look like white noise, and it also doesn't satisfy the assumptions. For the damped AAN, the significant spike only exists at lag 1 in the

ACF. The residual of damped AAN looks like white noise. The time plot of the residuals indicates that the variation of residuals stays the same in most of the time. But a little left-skewed distribution might cause the prediction interval to be inaccurate. For ARIMA, there are no significant spikes in the ACF, which means there is no significant correlation in the residuals series. The histogram of ARIMA's residuals suggests that the left tail is a little bit longer. And the corresponding residuals look like white noise.

Even though linear regression still has correlations between residuals, which means that there is information left, we think this model happens to be the best in terms of RMSE in our case.



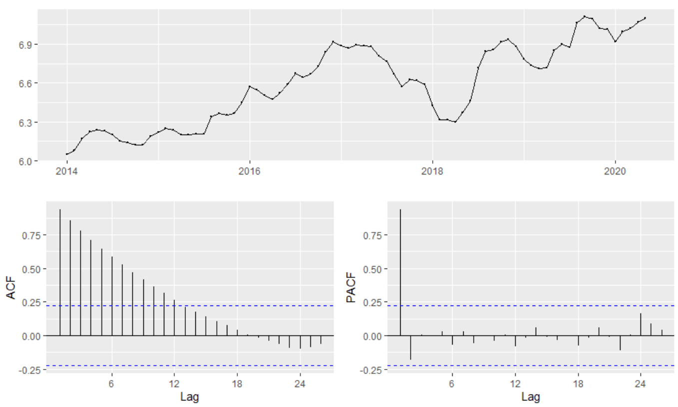
VI. Conclusion

We try to use some simple models to predict the exchange rate between Chinese RMB and the U.S dollar based on historical records. It can be used for exchange arbitrage. Our chosen forecast model is quite simple and not fully effective. We assume the model to work in a short period of time and a normal relation between the two countries. For improvement, we might need more variables related to the foreign exchange rate other than time factors such as interest rate and inflation. Besides, a robust forecast model for time series data like interest rate should have the ability to capture the suddenly changing factors between two countries, such as a trade war or trade policies, which might make interest rates change rapidly. Finally, there are many factors influencing the exchange rate that we cannot forecast statistically. However, the outcome that we predicted has allowed us to understand the limitations of our forecasting models more.

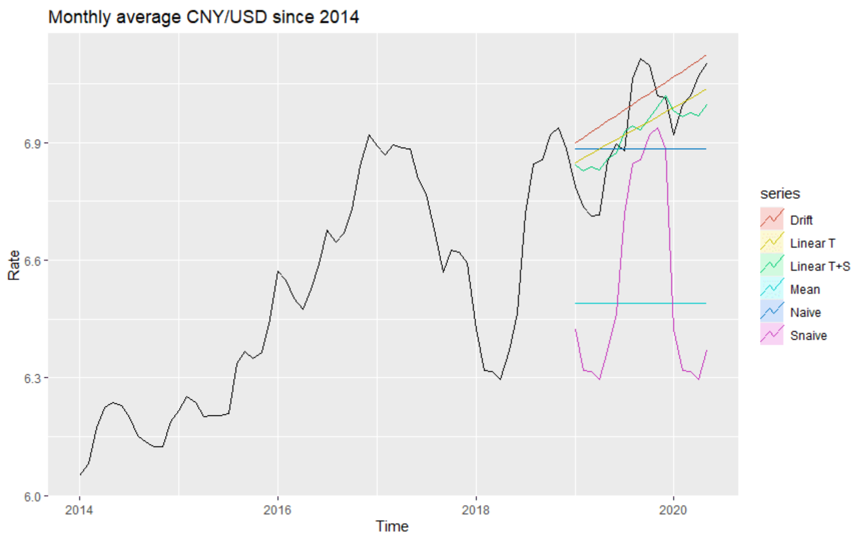
VII. APPENDIX

R codes and outputs

```
library(fpp2)
## Loading required package: ggplot2
## Loading required package: forecast
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
## Loading required package: fma
## Loading required package: expsmooth
DEXCHUS <- read.csv("C:/Users/Documents/FIN642/DEXCHUS.csv")
DEXCHUS <- ts(DEXCHUS$DEXCHUS, start=2014, frequency=12)
ggtsdisplay(DEXCHUS)
```




```
BoxCox.lambda(DEXCHUS)
## [1] 1.999924
fit <- stl(DEXCHUS, s.window = "periodic") # Decomposition
max(0,1-var(remainder(fit))/var(remainder(fit)+trendcycle(fit))) # FT
## [1] 0.9319371
max(0,1-var(remainder(fit))/var(remainder(fit)+seasonal(fit))) # FS
## [1] 0.1758402
train <- window(DEXCHUS, end = c(2018,12)) # make training data for DEXCHUS
test <- window(DEXCHUS, start = c(2019,1)) # make test data for DEXCHUS
h <- length(test)
h # number of months to forecast
## [1] 17
fc0 <- meanf(train, h = h) # Average method
fc1 <- snaive(train, h = h) # Seasonal naive method
fc2 <- naive(train, h = h) # Naive method
fc3 <- rwf(train, h = h, drift=TRUE) # Drift method
fc4 <- forecast(tslm(train ~ trend + season), h = h) # Linear regression
fc5 <- forecast(tslm(train ~ trend), h = h) # Linear regression
autoplot(DEXCHUS) +
  ggtitle("Monthly average CNY/USD since 2014") +
  ylab("Rate") +
  autolayer(fc0, series="Mean", PI = FALSE) +
  autolayer(fc1, series="Snaive", PI = FALSE) +
  autolayer(fc2, series="Naive", PI = FALSE) +
  autolayer(fc3, series="Drift", PI = FALSE) +
  autolayer(fc4, series="Linear T+S", PI = FALSE) +
  autolayer(fc5, series="Linear T", PI = FALSE)
```



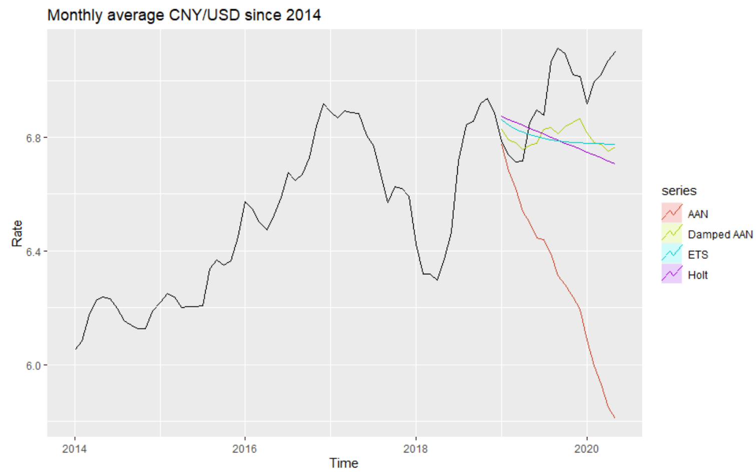

```
fc0.acc <- accuracy(fc0,test)[2,]
fc1.acc <- accuracy(fc1,test)[2,]
fc2.acc <- accuracy(fc2,test)[2,]
fc3.acc <- accuracy(fc3,test)[2,]
fc4.acc <- accuracy(fc4,test)[2,]
fc5.acc <- accuracy(fc5,test)[2,]
table <- rbind(fc0.acc, fc1.acc, fc2.acc, fc3.acc, fc4.acc, fc5.acc)
rownames(table) <- c("Mean", "Snaive", "Naive", "Drift", "Linear T+S", "Linear T")
table[,c(2,3,5,6)]
```

```
                 RMSE        MAE     MAPE      MASE
Mean       0.47008833 0.44992263 6.446030 1.6260844
Snaive     0.45935164 0.40702388 5.864280 1.4710422
Naive      0.14765655 0.13017118 1.868466 0.4704572
Drift      0.11495329 0.09831241 1.431103 0.3553150
Linear T+S 0.09016218 0.07574966 1.089856 0.2737701
Linear T   0.09682317 0.07873640 1.138142 0.2845645
```

```
ets(train)
```

```
ETS(M,Ad,N)
Call:
 ets(y = train)
  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.6577
    phi   = 0.8
  Initial states:
    l = 6.0384
    b = 0.0078
  sigma:  0.0098
      AIC      AICc       BIC
-78.47034 -76.88544 -65.90427
```

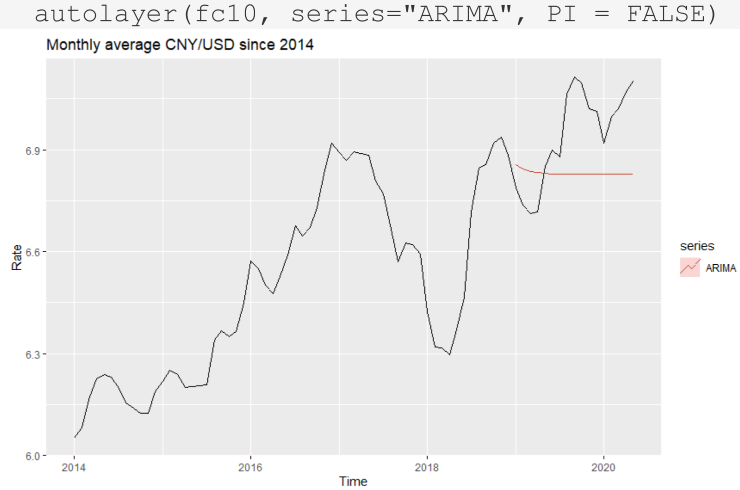
```
fc6 <- stlf(train, h = h, etsmodel = "AAN", damped = T) # Damped AAN
fc7 <- stlf(train, h = h, etsmodel = "AAN", damped = F) # AAN
fc8 <- holt(train, h = h) # Holt's linear trend
fc9 <- forecast(ets(train), h = h) #ETS
autoplot(DEXCHUS) +
  ggtitle("Monthly average CNY/USD since 2014") +
  ylab("Rate") +
  autolayer(fc6, series="Damped AAN", PI = FALSE) +
  autolayer(fc7, series="AAN", PI = FALSE) +
  autolayer(fc8, series="Holt", PI = FALSE) +
  autolayer(fc9, series="ETS", PI = FALSE)
```




```
fc6.acc <- accuracy(fc6,test)[2,]
fc7.acc <- accuracy(fc7,test)[2,]
fc8.acc <- accuracy(fc8,test)[2,]
fc9.acc <- accuracy(fc9,test)[2,]
table <- rbind(fc6.acc, fc7.acc, fc8.acc, fc9.acc)
rownames(table) <- c("Damped AAN", "AAN", "Holt", "ETS")
table[,c(2,3,5,6)]
```

	RMSE	MAE	MAPE	MASE
Damped AAN	0.1917569	0.1634503	2.330122	0.5907325
AAN	0.7533465	0.6413376	9.144143	2.3178854
Holt	0.2339502	0.2067474	2.954296	0.7472145
ETS	0.2121335	0.1894636	2.706787	0.6847483

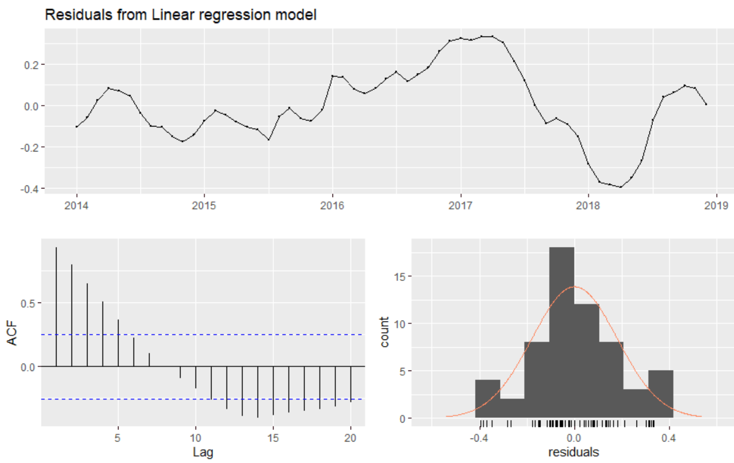
```
fc10 <- forecast(auto.arima(train), h = h) # ARIMA
autoplot(DEXCHUS) +
  ggtitle("Monthly average CNY/USD since 2014") +
  ylab("Rate") +
  autolayer(fc10, series="ARIMA", PI = FALSE)
```



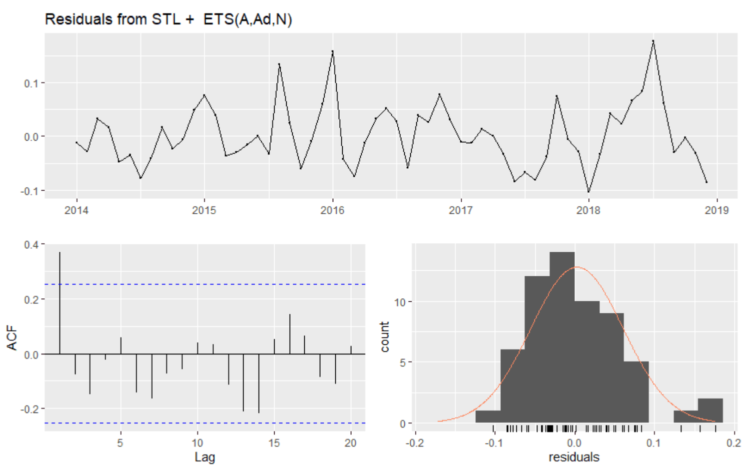

```
accuracy(fc10,test)[2,c(2,3,5,6)]
```

RMSE	MAE	MAPE	MASE
0.1781585	0.1581536	2.2609882	0.5715896

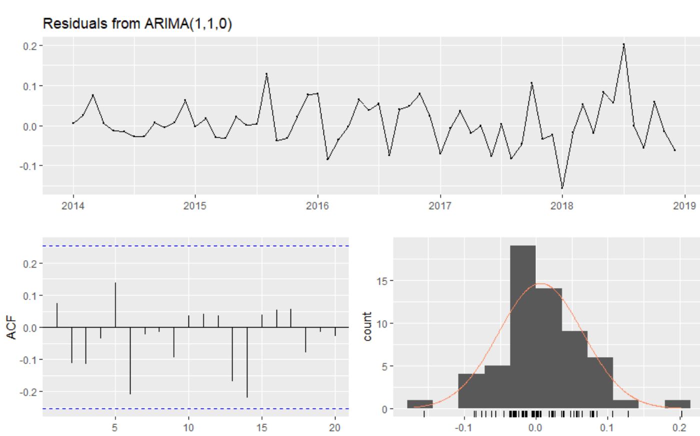
```
checkresiduals(fc4)
```




```
      Ljung-Box test
data:  Residuals from Linear regression model
Q* = 219.21, df = 3, p-value < 2.2e-16
Model df: 13.   Total lags used: 16
checkresiduals(fc6)
```




```
      Ljung-Box test
data:  Residuals from STL +  ETS(A,Ad,N)
Q* = 15.875, df = 7, p-value = 0.02629
Model df: 5.   Total lags used: 12
checkresiduals(fc10)
```




```
      Ljung-Box test
data:  Residuals from ARIMA(1,1,0)
Q* = 7.4119, df = 11, p-value = 0.7648
Model df: 1.   Total lags used: 12
```