HuYuDataInsight LLC

Zhaowei Cai

2024-05-02

Loading packages

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

```
library(stringr)
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.2.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.2.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

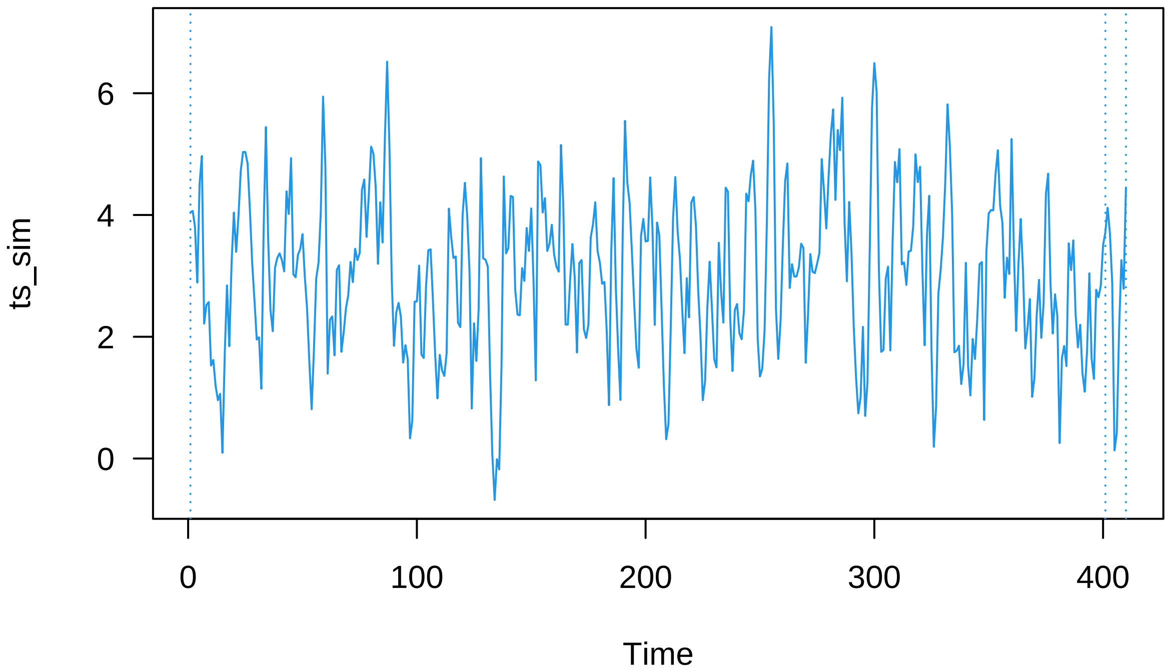
```
library(xts)
```

Question 1

```
set.seed(123)
#create a time series with right observations and first element is 0
ts_sim <- arima.sim(list(order = c(1,0,1), ar=0.5, ma=0.4), n = 410) + 3

left <- 401
right <- 410
it <- left:right
```

```
plot(ts_sim, col=4, las=1)
abline(v=c(1, left, right), lty="dotted", col=4)
```

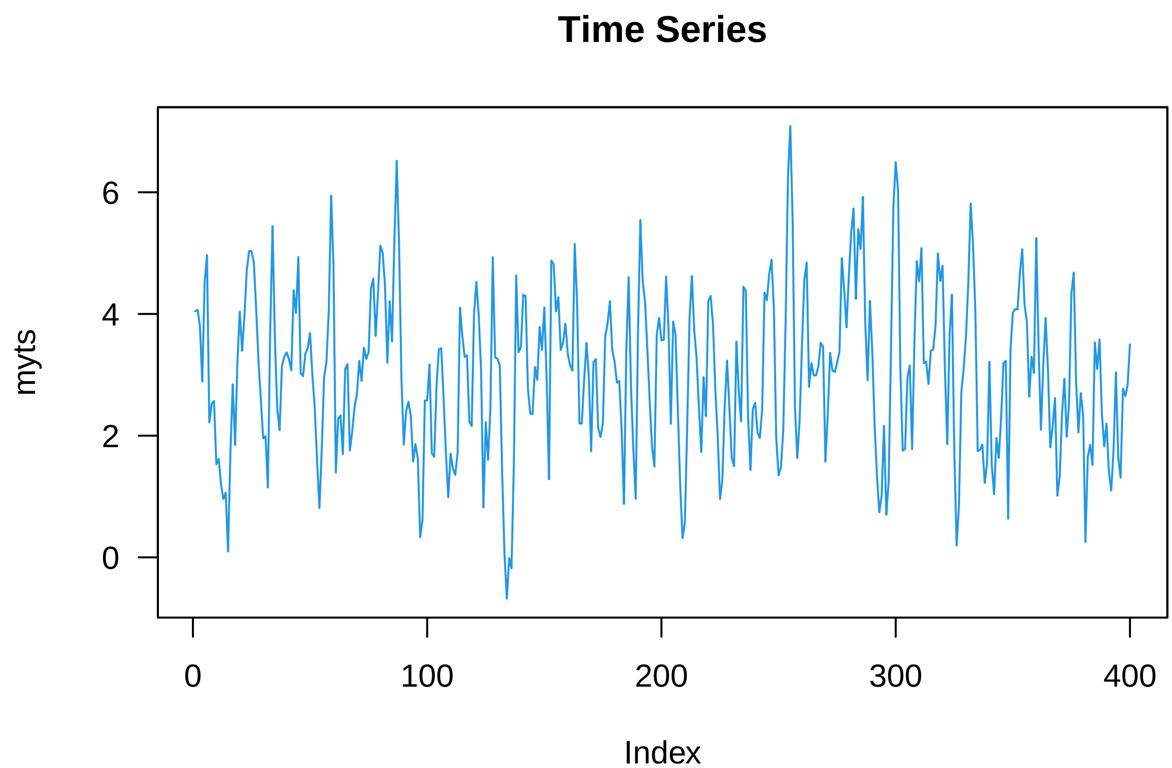


Question 2

```r
myts = subset(ts_sim, subset=rep(c(TRUE, FALSE), times=c(400, 10)))
```

Step 1: visualize myts

```r
plot.zoo(myts, col=4, las=1, main="Time Series")
```



Step 2: unit root test (augmented Dickey-Fuller) of myts

```r
adf.test(myts, alternative = 'stationary')
```

```
## Warning in adf.test(myts, alternative = "stationary"): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  myts
## Dickey-Fuller = -6.4015, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(myts)
```

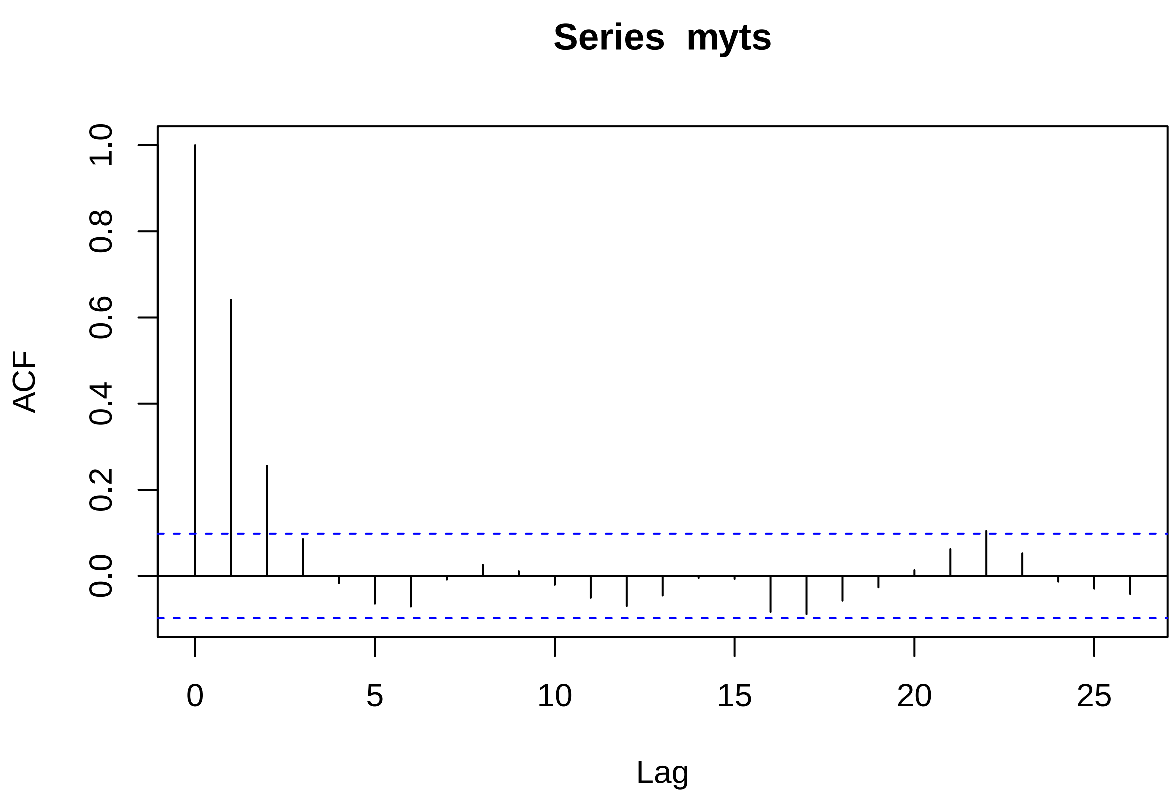
```
## Warning in kpss.test(myts): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  myts
## KPSS Level = 0.079482, Truncation lag parameter = 5, p-value = 0.1
```

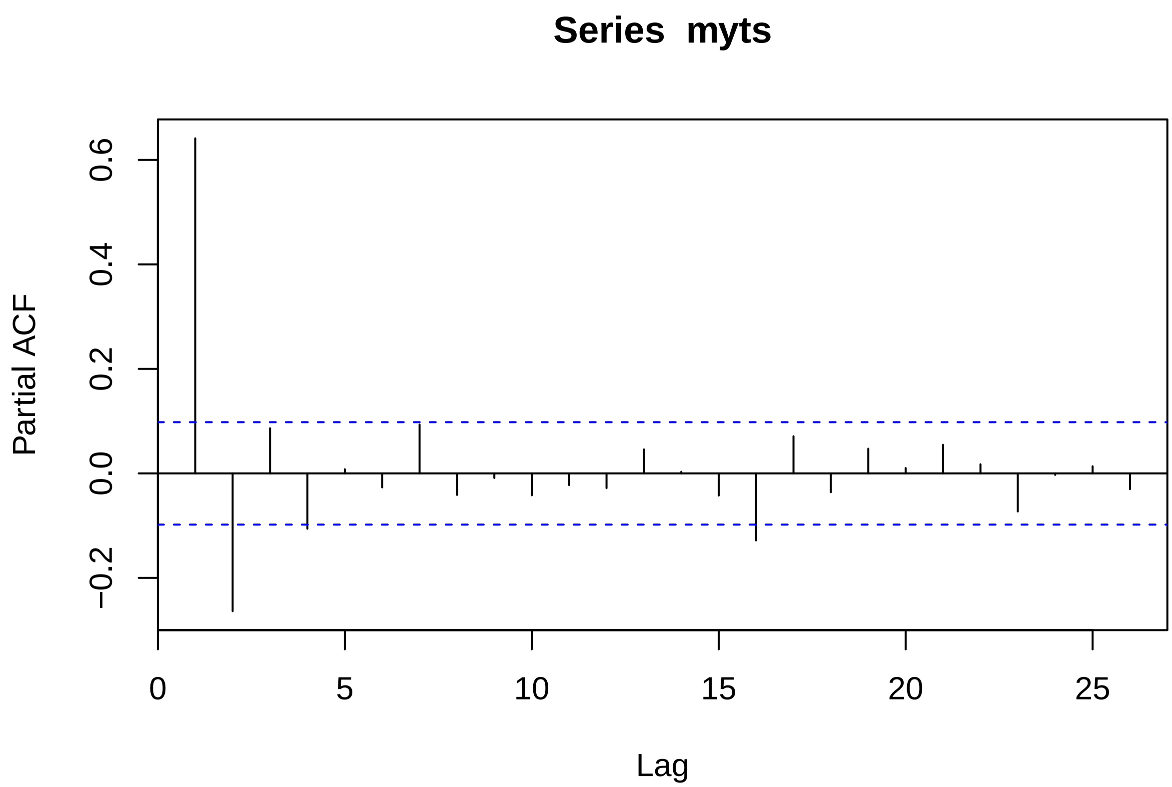
We have used both the KPSS test and the ADF Unit root test to check the stationary. The null hypothesis of the Unit root test (Adf.test) is rejected while the null hypothesis of KPSS test is accepted. Both indicate the data are stationary.

Step 3: Identifying lags

```
acf(myts)
```




```
pacf(myts)
```



The Partial ACF cut off after first lag, it shows the MA(1) part of ARIMA model (q=1).

Step 4: train the model with auto.arima

```
fit_myts = auto.arima(
  myts,
  max.p = 3,
  max.q = 3,
  ic = "aicc",
  seasonal = FALSE,
  stationary = TRUE,
  lambda = NULL,
  stepwise = FALSE,
  approximation = FALSE,
  trace = T
)
```

```
##
##  ARIMA(0,0,0) with zero mean     : 2089.856
##  ARIMA(0,0,0) with non-zero mean : 1354.638
##  ARIMA(0,0,1) with zero mean     : 1662.442
##  ARIMA(0,0,1) with non-zero mean : 1136.867
##  ARIMA(0,0,2) with zero mean     : 1493.718
##  ARIMA(0,0,2) with non-zero mean : 1118.227
##  ARIMA(0,0,3) with zero mean     : 1387.276
```

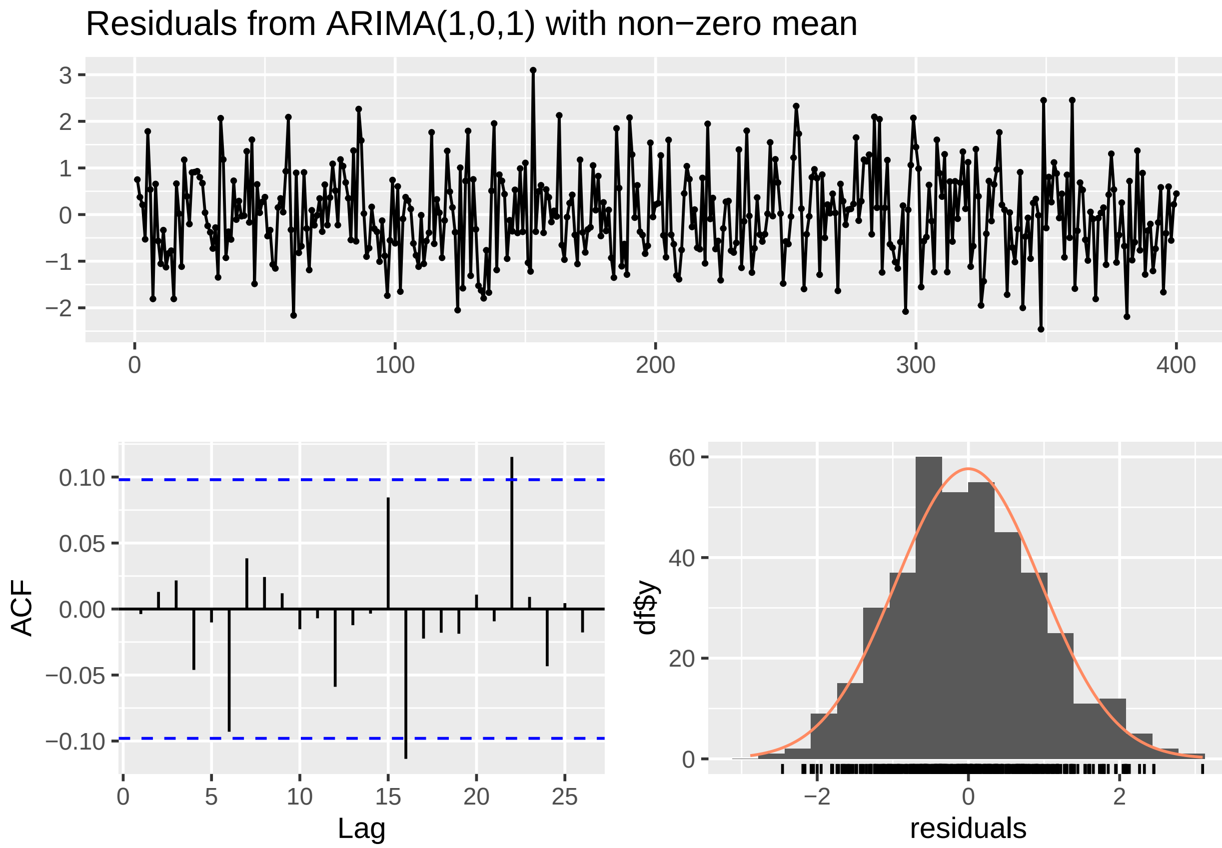
```
##  ARIMA(0,0,3) with non-zero mean : 1111.823
##  ARIMA(1,0,0) with zero mean     : 1212.843
##  ARIMA(1,0,0) with non-zero mean : 1144.845
##  ARIMA(1,0,1) with zero mean     : 1210.675
##  ARIMA(1,0,1) with non-zero mean : 1111.424
##  ARIMA(1,0,2) with zero mean     : Inf
##  ARIMA(1,0,2) with non-zero mean : 1113.232
##  ARIMA(1,0,3) with zero mean     : Inf
##  ARIMA(1,0,3) with non-zero mean : 1113.798
##  ARIMA(2,0,0) with zero mean     : 1213.027
##  ARIMA(2,0,0) with non-zero mean : 1118.063
##  ARIMA(2,0,1) with zero mean     : 1196.142
##  ARIMA(2,0,1) with non-zero mean : 1112.909
##  ARIMA(2,0,2) with zero mean     : Inf
##  ARIMA(2,0,2) with non-zero mean : 1111.717
##  ARIMA(2,0,3) with zero mean     : Inf
##  ARIMA(2,0,3) with non-zero mean : 1114.775
##  ARIMA(3,0,0) with zero mean     : 1182.846
##  ARIMA(3,0,0) with non-zero mean : 1117.109
##  ARIMA(3,0,1) with zero mean     : Inf
##  ARIMA(3,0,1) with non-zero mean : 1111.47
##  ARIMA(3,0,2) with zero mean     : Inf
##  ARIMA(3,0,2) with non-zero mean : 1113.451
##
##
##
##  Best model: ARIMA(1,0,1) with non-zero mean
```

```
best.fit = arima(myts, c(1,0,1))
summary(best.fit)
```

```
##
## Call:
## arima(x = myts, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.3864  0.4668     3.0225
## s.e.  0.0669  0.0675     0.1145
##
## sigma^2 estimated as 0.9218:  log likelihood = -551.66,  aic = 1111.32
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.001400413 0.9601088 0.7674834 -9.745727 69.07652 0.8868352
##                      ACF1
## Training set -0.003787956
```

There is an intercept equal to 3.0225.

```
checkresiduals(best.fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 5.6987, df = 8, p-value = 0.6809
##
## Model df: 2.   Total lags used: 10
```

```
Box.test(best.fit$residuals)
```

```
##
##  Box-Pierce test
##
## data:  best.fit$residuals
## X-squared = 0.0057394, df = 1, p-value = 0.9396
```

The p-value is higher than 0.05, so there is little evidence of non-zero autocorrelations in the forecast errors.

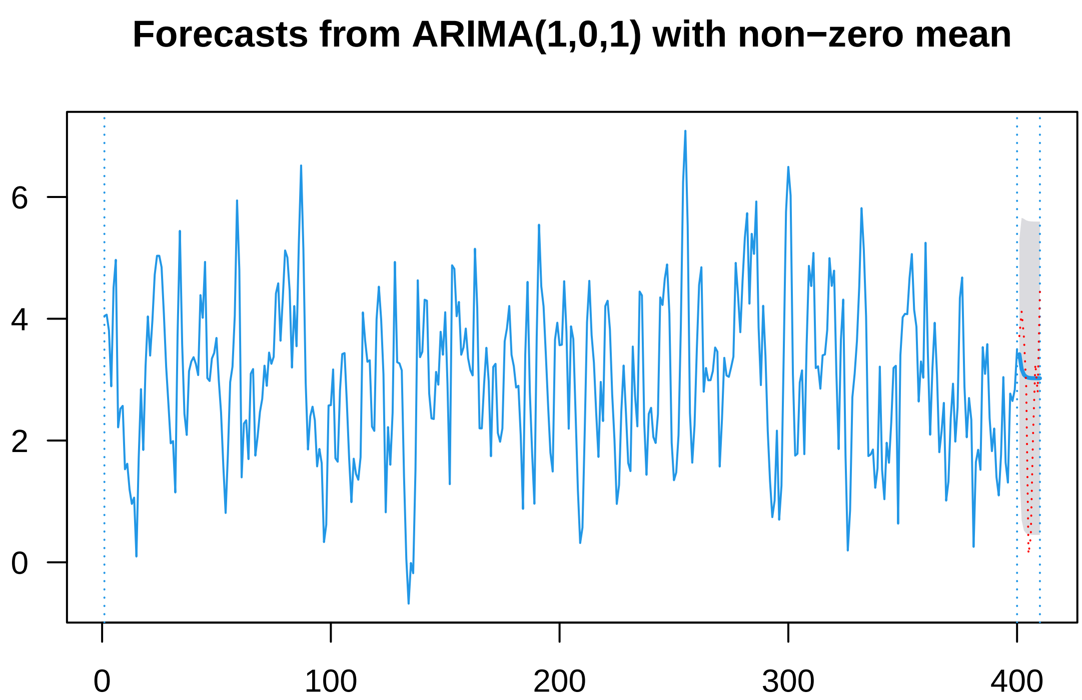
```
shapiro.test(best.fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  best.fit$residuals
## W = 0.99557, p-value = 0.3188
```

Shapiro-Wilk test confirms the normally distributed residuals as well.

Question 3

```
forecast_myts = forecast(fit_myts, h=10, level=0.95)
plot(forecast_myts, col=4, las=1)
abline(v=c(1, 400, 410), lty="dotted", col=4)
lines(401:410, ts_sim[401:410], lty="dotted", col="red")
```



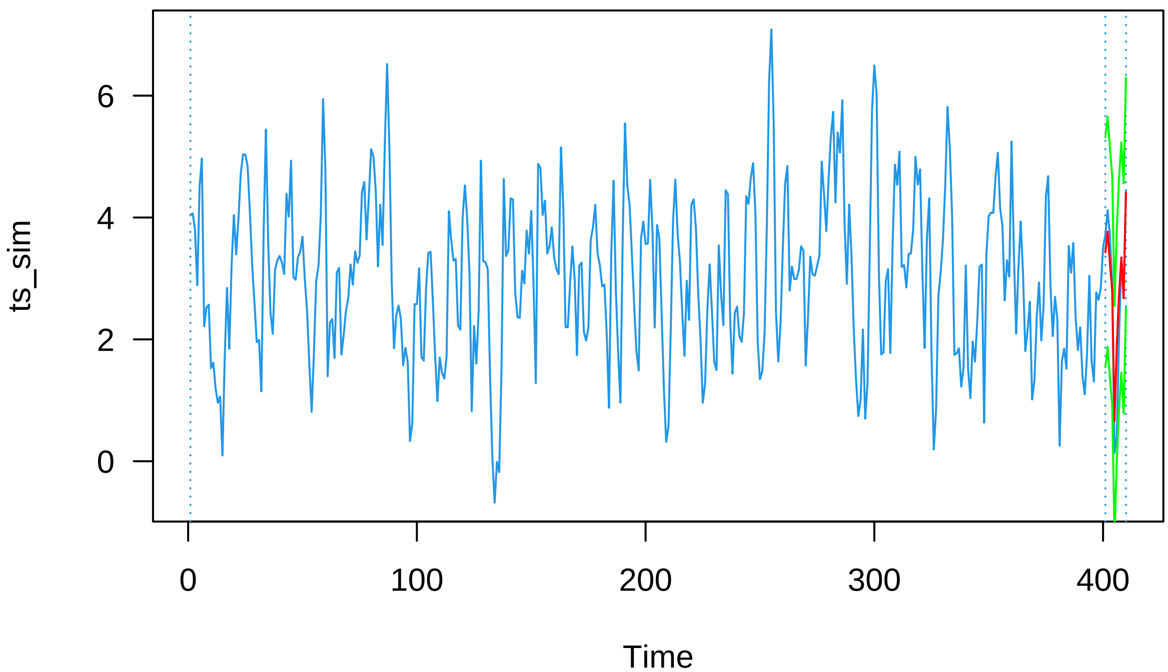

```
# red is observation and blue is prediction
```

```
# since it is one step ahead predictin, so we need use for loop
pred_df <- data.frame(NULL)
for(t in 401:410){
  pred_onestep <- forecast(ts_sim[1:t], h=1, level=0.95, model = fit_myts)
  pred_df <- rbind(pred_df, data.frame(mean = pred_onestep$mean[1], lower = pred_onestep$lower[1], uppe
}
```

```
plot(ts_sim, col=4, las=1)
abline(v=c(1, left, right), lty="dotted", col=4)

lines(it, pred_df$mean, col = 'red')
```

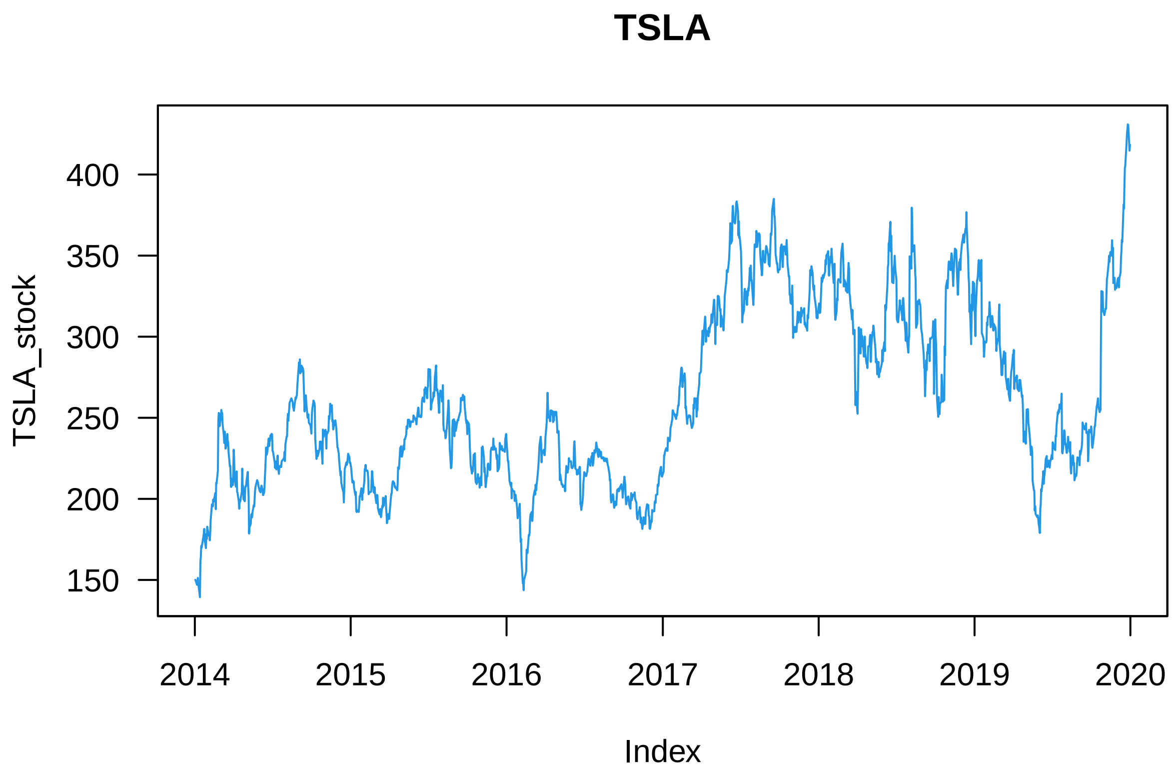
```r
lines(it, pred_df$lower, col = 'green')
lines(it, pred_df$upper, col = 'green')
legend(40, 40, legend=c("Observations", "Prediction", "Bounds of CI"),col=c("blue", "red", "green"),lty
```



Question 4

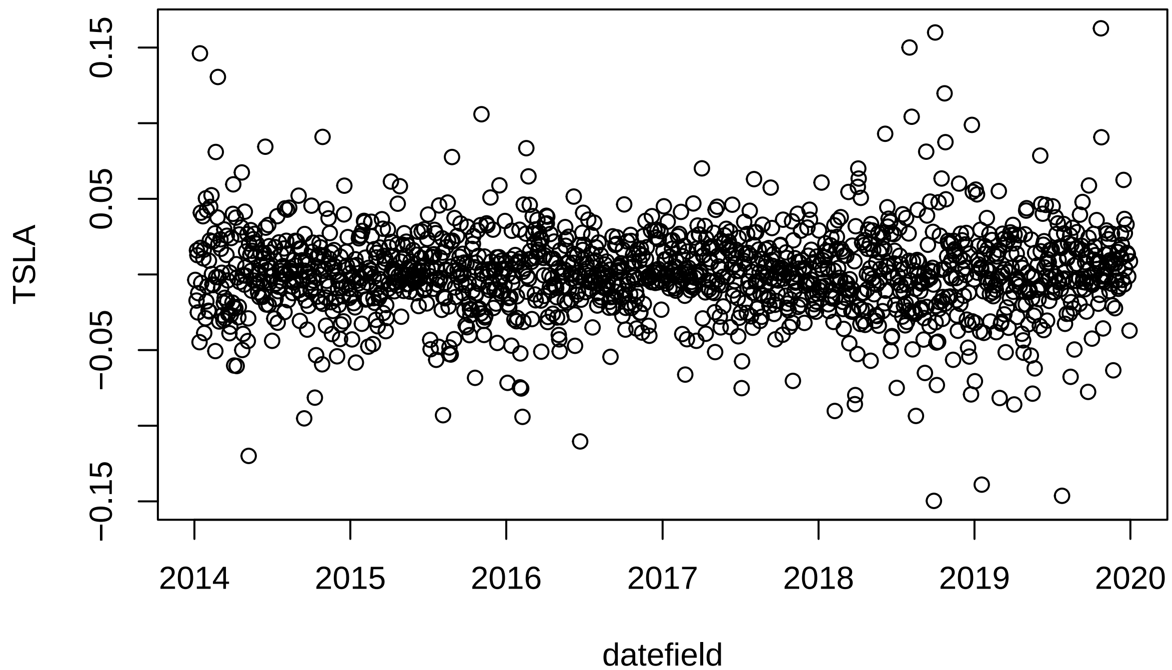
```r
#(a)
data = read.csv('TSLA1.csv')

library(forecast)
library(zoo)
library(tseries)
TSLA = data$Close
time = as.Date(data$Date, format = '%m/%d/%y')
df = data.frame(datefield = time, TSLA = TSLA)
TSLA_stock = with(df, zoo(TSLA, order.by = time))
plot.zoo(TSLA_stock, col=4, las=1, main="TSLA")
```

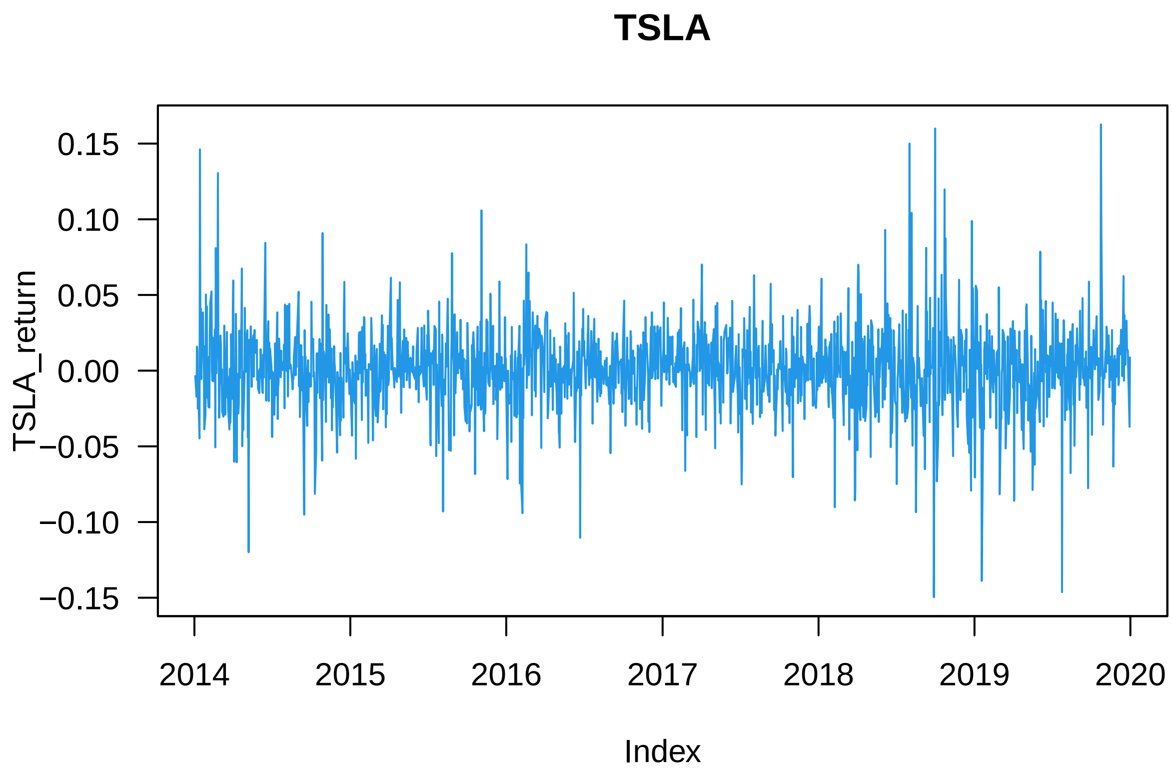



```r
# Use the closing price to get log return
log_return = na.omit(diff(log(data$Close))) # log return
time = as.Date(data$Date, format = '%m/%d/%y')[-1]
df = data.frame(datefield = time, TSLA = log_return)
TSLA_return = with(df, zoo(TSLA, order.by = time))
plot(df, main = "TSLA log returns")
```

TSLA log returns




```
plot.zoo(TSLA_return, col=4, las=1, main="TSLA")
```




```r
adf.test(log_return)
```

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

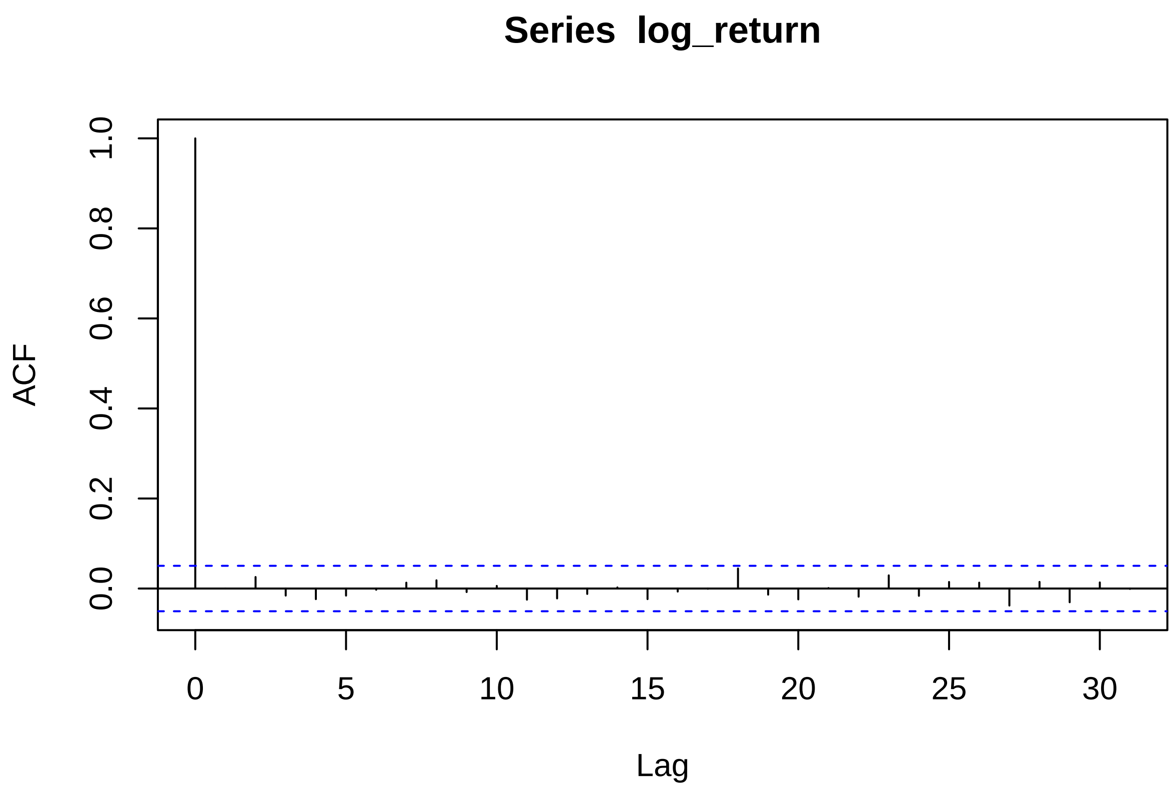
```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_return
## Dickey-Fuller = -11.651, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(log_return)
```

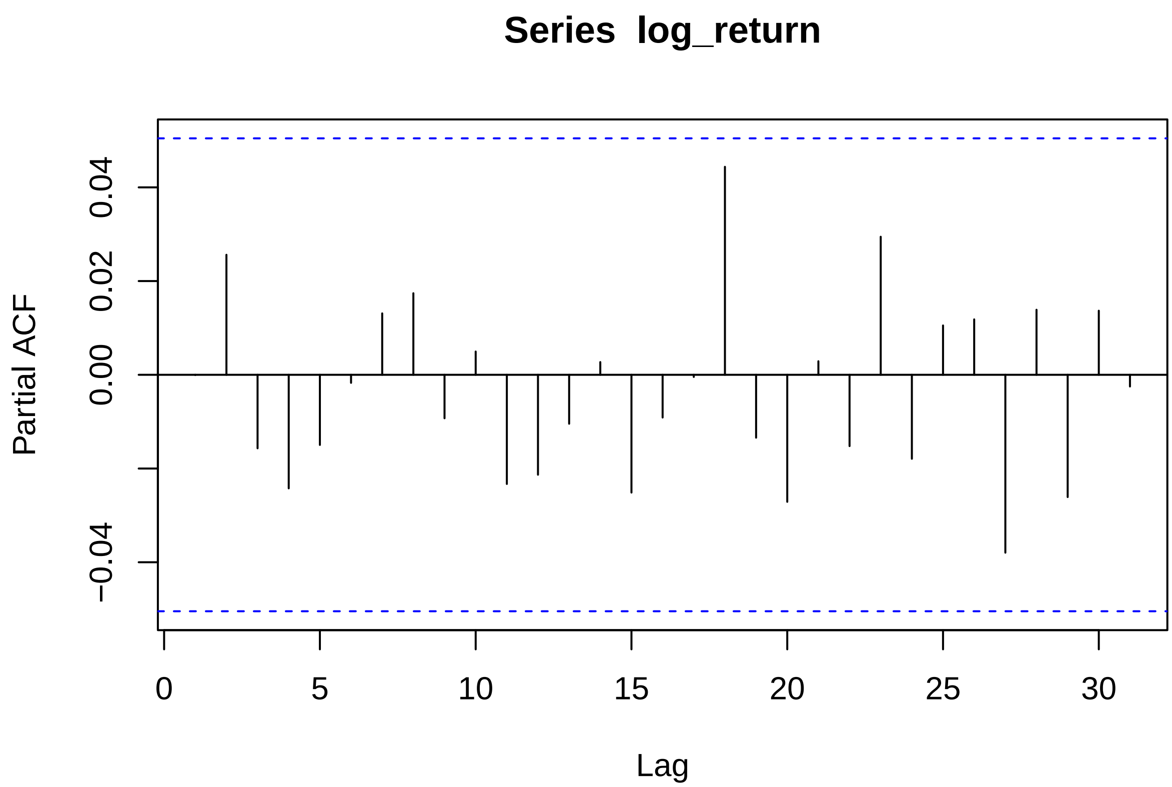
```
## Warning in kpss.test(log_return): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  log_return
## KPSS Level = 0.0507, Truncation lag parameter = 7, p-value = 0.1
```

```r
# Stationary
acf(log_return)
```




```
pacf(log_return)
```




```
model <- auto.arima(log_return, trace = T)
```

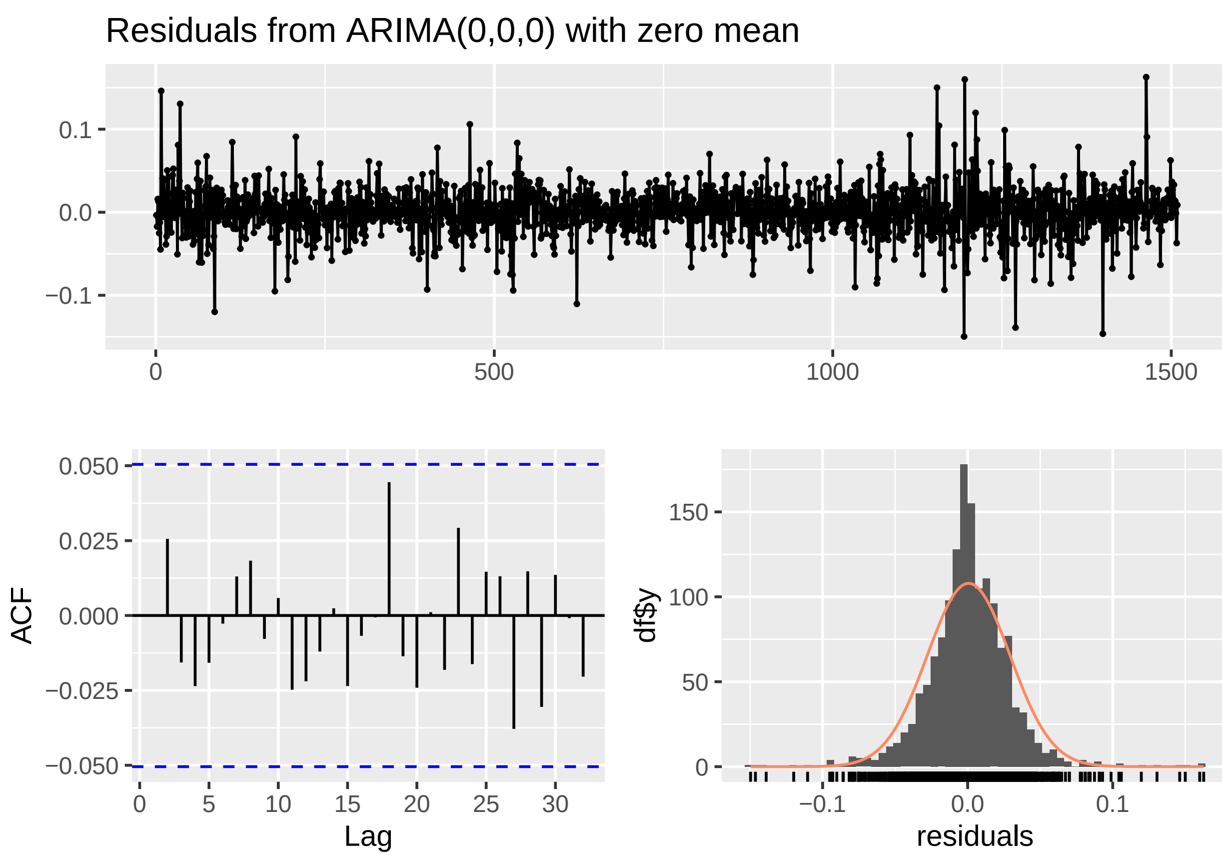
```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -6446.837
##  ARIMA(0,0,0) with non-zero mean : -6445.49
##  ARIMA(1,0,0) with non-zero mean : -6442.504
##  ARIMA(0,0,1) with non-zero mean : -6443.482
##  ARIMA(0,0,0) with zero mean     : -6446.642
##  ARIMA(1,0,2) with non-zero mean : -6439.586
##  ARIMA(2,0,1) with non-zero mean : -6439.043
##  ARIMA(3,0,2) with non-zero mean : -6447.353
##  ARIMA(3,0,1) with non-zero mean : Inf
##  ARIMA(4,0,2) with non-zero mean : Inf
##  ARIMA(3,0,3) with non-zero mean : Inf
##  ARIMA(2,0,3) with non-zero mean : -6445.079
##  ARIMA(4,0,1) with non-zero mean : -6434.57
##  ARIMA(4,0,3) with non-zero mean : Inf
##  ARIMA(3,0,2) with zero mean     : -6448.533
##  ARIMA(2,0,2) with zero mean     : -6448.268
##  ARIMA(3,0,1) with zero mean     : -6438.219
##  ARIMA(4,0,2) with zero mean     : Inf
##  ARIMA(3,0,3) with zero mean     : Inf
##  ARIMA(2,0,1) with zero mean     : -6440.253
```

```
##  ARIMA(2,0,3) with zero mean     : -6446.481
##  ARIMA(4,0,1) with zero mean     : -6435.724
##  ARIMA(4,0,3) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,2) with zero mean     : Inf
##  ARIMA(2,0,2) with zero mean     : Inf
##  ARIMA(3,0,2) with non-zero mean : Inf
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with zero mean     : -6446.642
##
##  Best model: ARIMA(0,0,0) with zero mean
```

```
model
```

```
## Series: log_return
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0008158:  log likelihood = 3224.32
## AIC=-6446.64   AICc=-6446.64   BIC=-6441.33
```

```
checkresiduals(model)
```




```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,0) with zero mean
## Q* = 3.5043, df = 10, p-value = 0.967
##
## Model df: 0.   Total lags used: 10
```

```r
Box.test(model$residuals)
```

```
##
##  Box-Pierce test
##
## data:  model$residuals
## X-squared = 4.5622e-06, df = 1, p-value = 0.9983
```

```r
shapiro.test(model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals
## W = 0.94548, p-value < 2.2e-16
```

```r
# (b)
# Log return is the first order diff of logged closing price.
log_cp = log(data$Close)
model <- auto.arima(log_cp, trace = T)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : -6439.726
##  ARIMA(0,1,0) with drift         : -6438.379
##  ARIMA(1,1,0) with drift         : -6435.393
##  ARIMA(0,1,1) with drift         : -6436.371
##  ARIMA(0,1,0)                    : -6439.53
##  ARIMA(1,1,2) with drift         : -6432.475
##  ARIMA(2,1,1) with drift         : -6431.932
##  ARIMA(3,1,2) with drift         : -6440.242
##  ARIMA(3,1,1) with drift         : Inf
##  ARIMA(4,1,2) with drift         : Inf
##  ARIMA(3,1,3) with drift         : Inf
##  ARIMA(2,1,3) with drift         : -6437.968
##  ARIMA(4,1,1) with drift         : -6427.459
##  ARIMA(4,1,3) with drift         : Inf
##  ARIMA(3,1,2)                    : -6441.422
##  ARIMA(2,1,2)                    : -6441.157
##  ARIMA(3,1,1)                    : -6431.108
##  ARIMA(4,1,2)                    : Inf
##  ARIMA(3,1,3)                    : Inf
##  ARIMA(2,1,1)                    : -6433.142
##  ARIMA(2,1,3)                    : -6439.37
```

```
##  ARIMA(4,1,1)                    : -6428.612
##  ARIMA(4,1,3)                    : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,1,2)                    : Inf
##  ARIMA(2,1,2)                    : Inf
##  ARIMA(3,1,2) with drift         : Inf
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(0,1,0)                    : -6446.642
##
##  Best model: ARIMA(0,1,0)
```

```
# ARIMA(0,1,0)
```