HW5-Avery

Avery Loftin

10/20/2018

```
library(forecast)
library(quantmod)
library(TSA)
```

1. Use arima.sim to simulate 100 observations for the following MA(2) models with parameters as specifies and sketch their corresponding autocorrelation functions. Use par(mfrow=c(2,2)) to display the acf as a matrix form.

(a) $\theta_1 = .5$, $\theta_2 = 0.4$

```
MA2_.5_.4 <- arima.sim(model = list(ma=c(.5,.4)), n = 100)
```

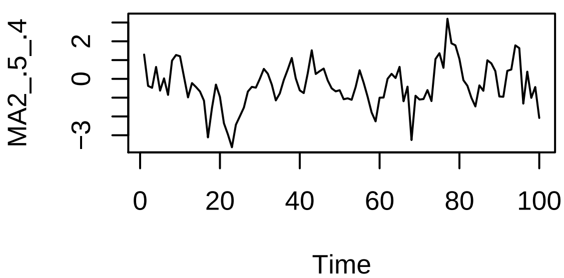
(b) $\theta_1 = 1.2$, $\theta_2 =$ -.7

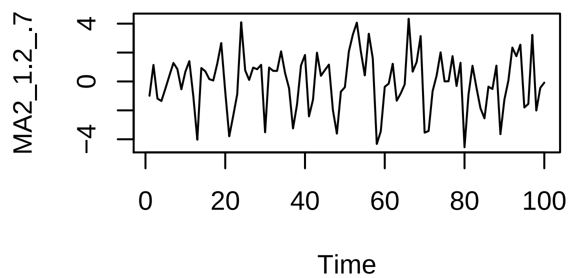
```
MA2_1.2_.7 <- arima.sim(model = list(ma=c(1.2,-.7)), n = 100)
```

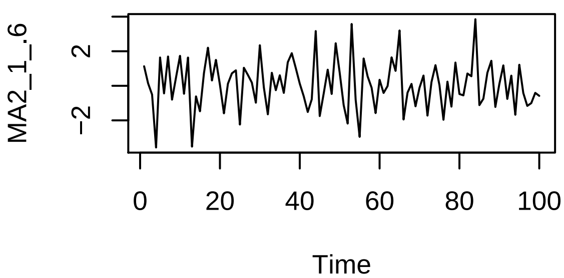
(c) $\theta_1 =$ -1, $\theta_2 =$ -0.6

```
MA2_1_.6 <- arima.sim(model = list(ma=c(-1,-.6)), n = 100)
```

```
par(mfrow=c(2,2))
plot.ts(MA2_.5_.4)
plot.ts(MA2_1.2_.7)
plot.ts(MA2_1_.6)
```

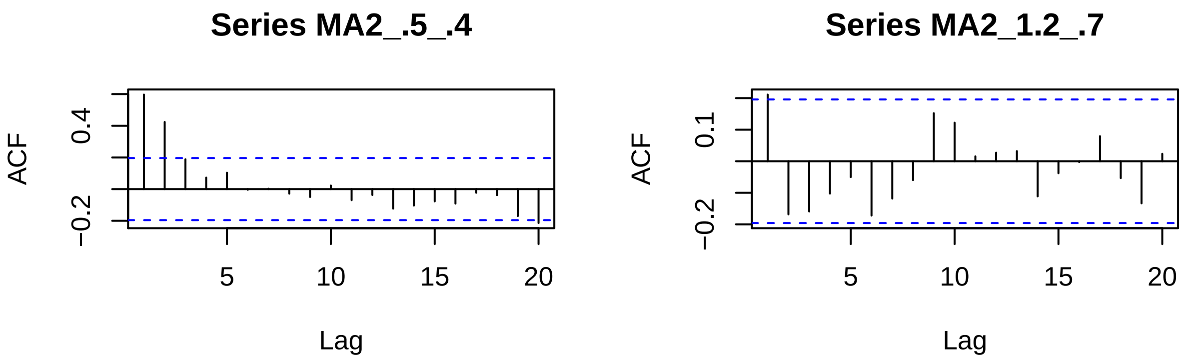


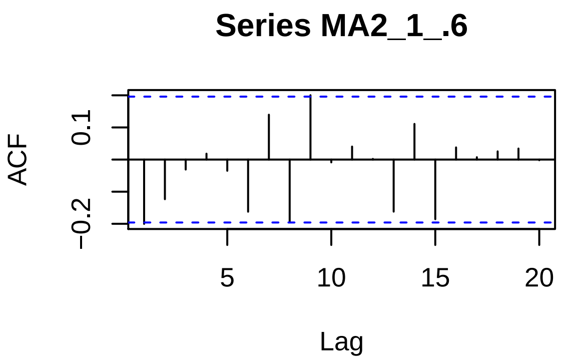





```
par(mfrow=c(2,2))
acf(MA2_.5_.4)
```

```r
acf(MA2_1.2_.7)
acf(MA2_1_.6)
```





2. Use arima.sim to simulate 100 observations for the following AR(p) models with parameters as specifies and sketch their corresponding autocorrelation functions. Use par(mfrow=c(2,2)) to display the acf as a matrix form.

(a) $\phi_1 = .6$

```r
AR1_.6 <- arima.sim(model = list(ar=c(.6)), n = 100)
```

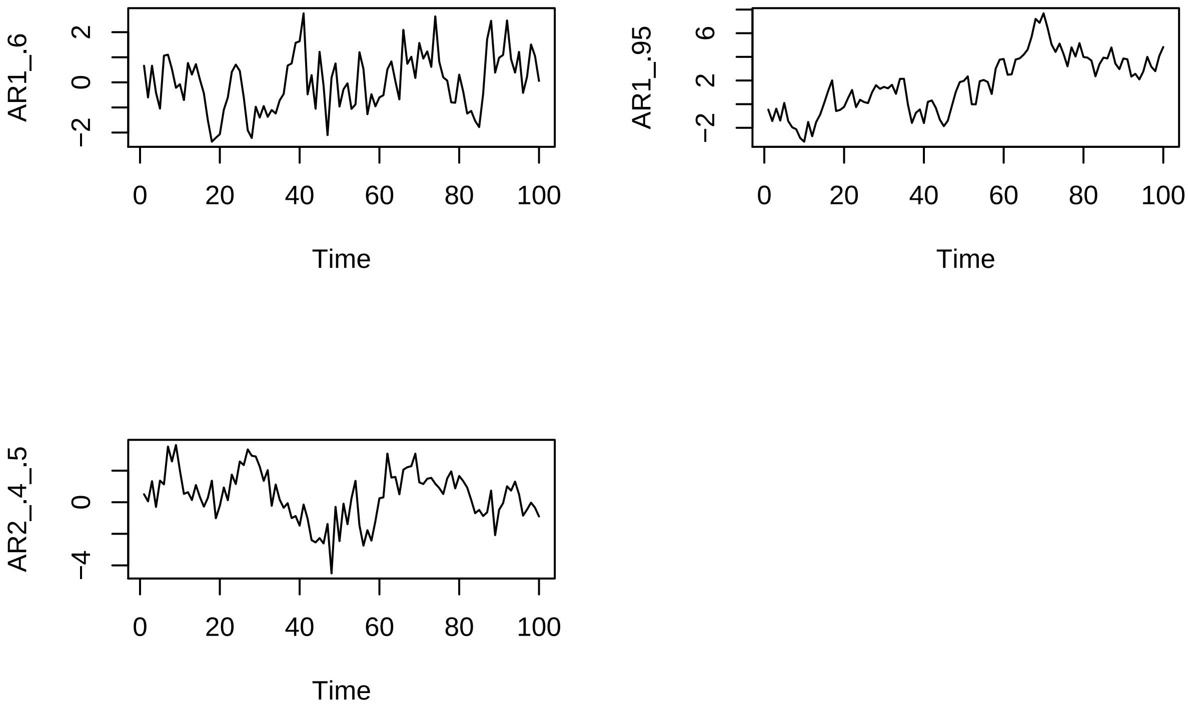
(b) $\phi_1 = 0.95$

```r
AR1_.95 <- arima.sim(model = list(ar=c(.95)), n = 100)
```

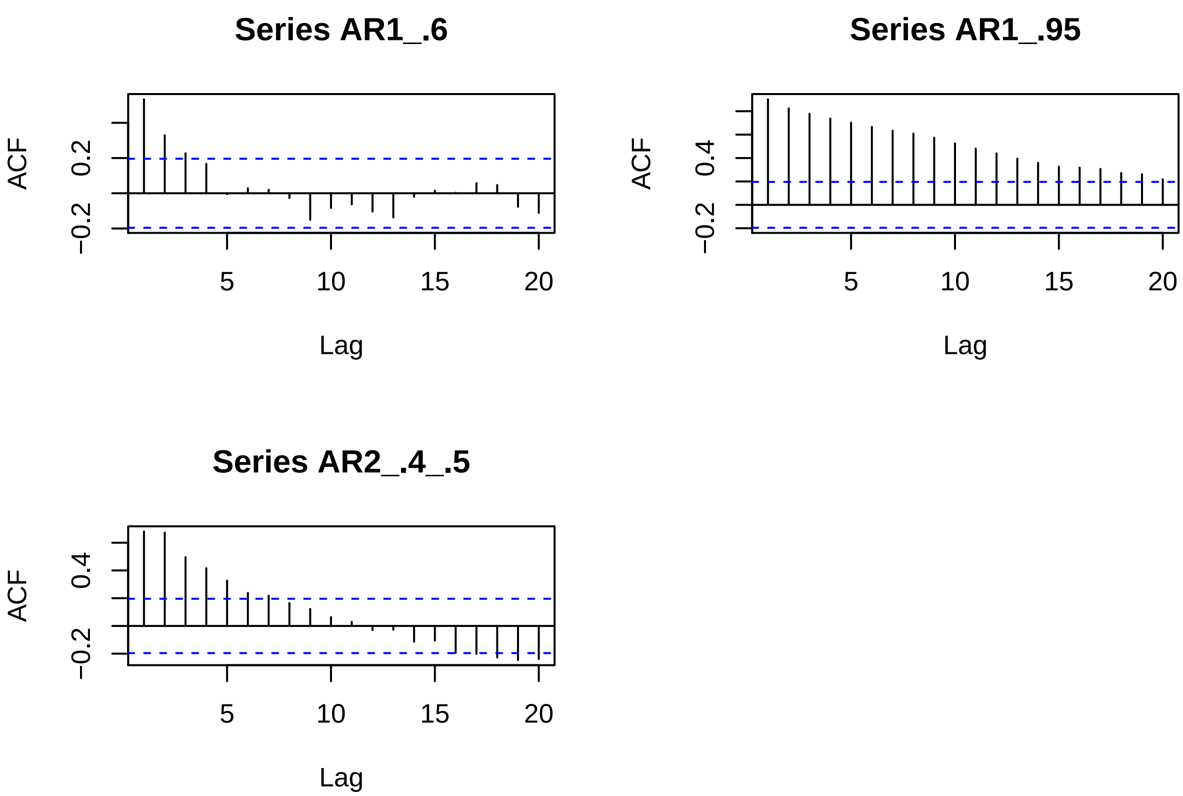
(c) $\phi_1 = .4$, $\phi_2 = 0.5$

```r
AR2_.4_.5 <- arima.sim(model = list(ar=c(.4,.5)), n = 100)
```

```r
par(mfrow=c(2,2))
plot.ts(AR1_.6)
plot.ts(AR1_.95)
plot.ts(AR2_.4_.5)
```

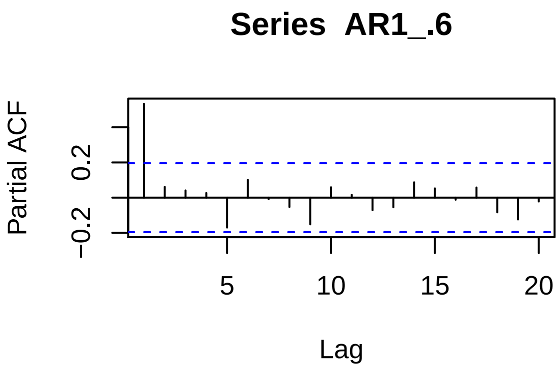


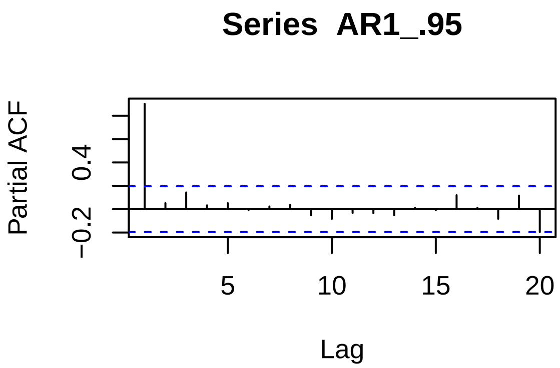

```
par(mfrow=c(2,2))
acf(AR1_.6)
acf(AR1_.95)
acf(AR2_.4_.5)
```

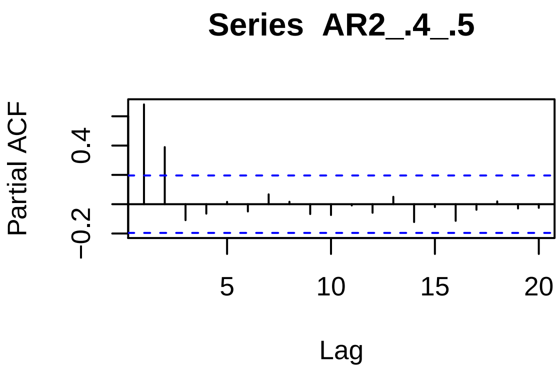



```
par(mfrow=c(2,2))
pacf(AR1_.6)
```

```r
pacf(AR1_.95)
pacf(AR2_.4_.5)
```







3. Use your own data set to answer the following questions.

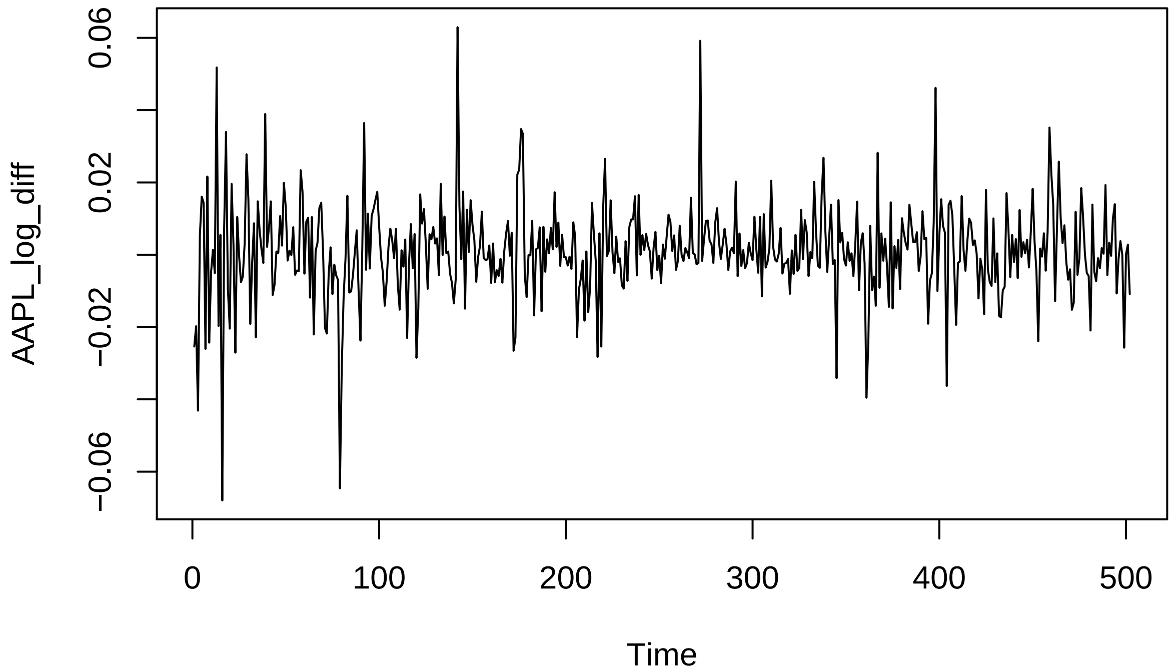
```r
getSymbols("AAPL", from="2016-01-01", to="2018-01-01")
```

```
## [1] "AAPL"
```

```r
AAPL_log_diff <- diff(log(AAPL$AAPL.Close))
AAPL_log_diff <- AAPL_log_diff[-1]
```

(a) Plot the data set as a time series and comment if there is any trend and/or seasonality.

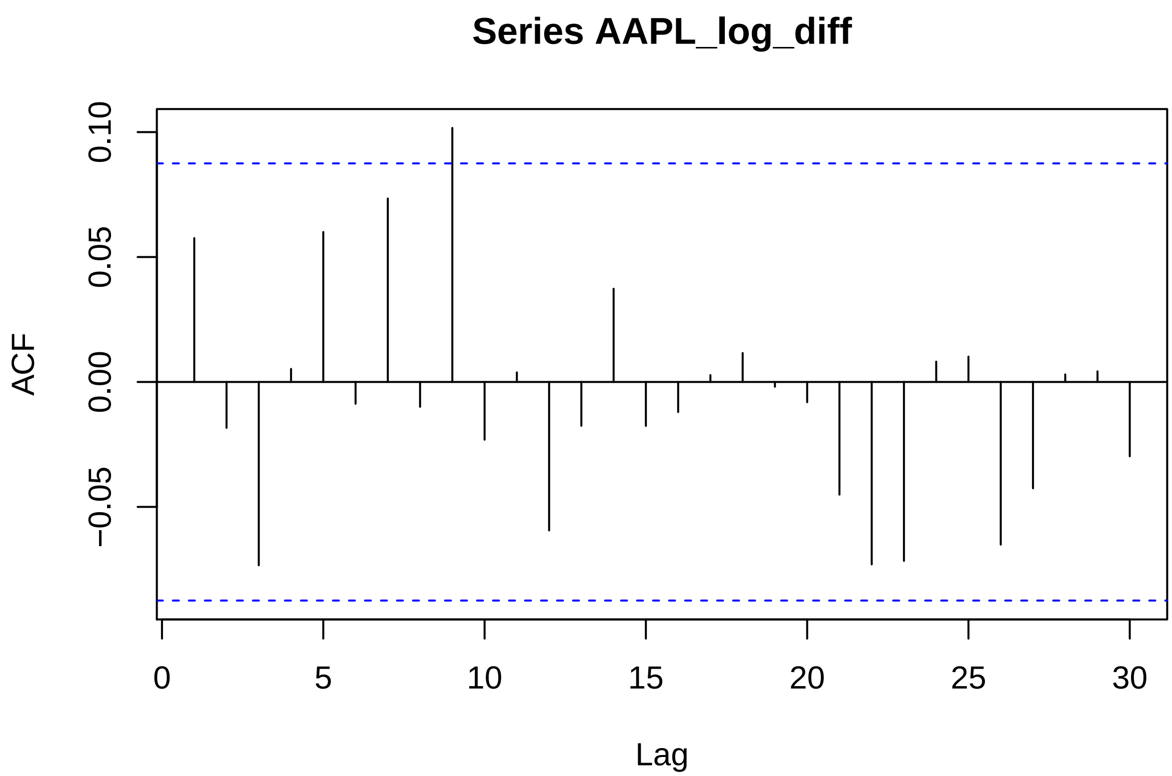
```r
plot.ts(AAPL_log_diff)
```



The log difference of Apple's closing stock prices from 2016 to 2018 shows no seasonality or trend.

(b) Sketch the autocorrelation function up to lag of 30 and decide if MA, AR, or ARIMA would be an appropriate model to use.

```
acf(AAPL_log_diff, lag.max = 30)
```



The acf doesn't seem to support the use of either model, as there are no significant correlations. When a MA is a good fit, there would be at least one significant lag at the start with all the rest zero. When an AR model is appropriate, the lags exponentially decay toward zero.

(c) Use the auto.arima(data) to determine the number of lags you would need for AR and/or MA model and write the model using parameter estimates.

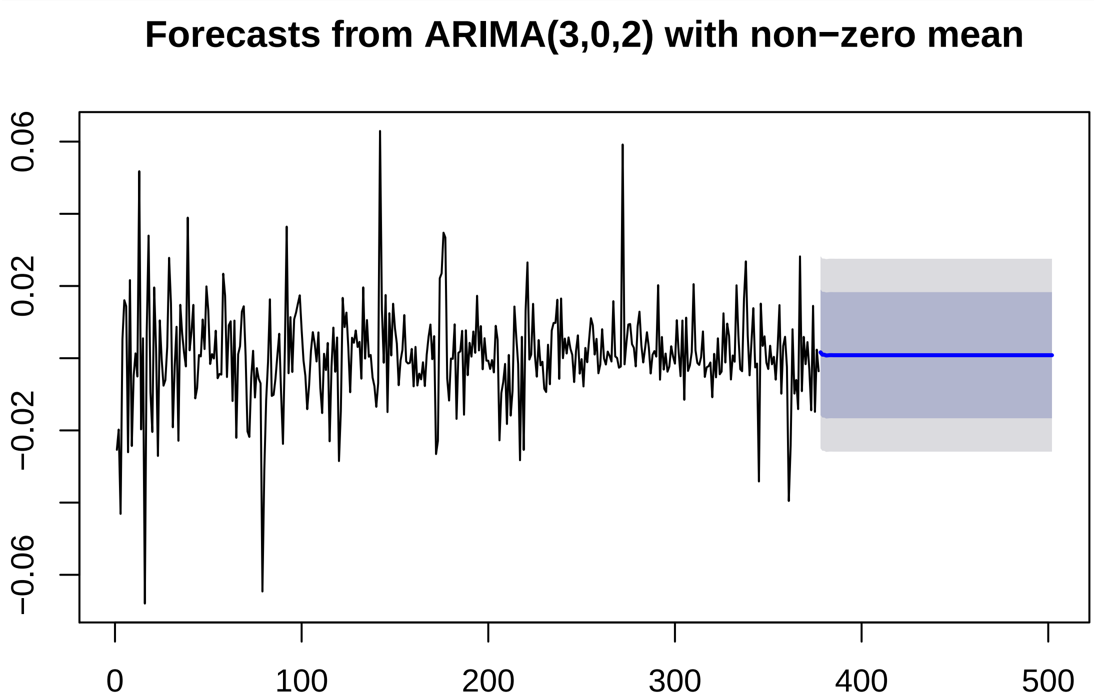
```
auto.arima(AAPL_log_diff)
```

```
## Series: AAPL_log_diff
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3    ma1     ma2   mean
##       0.0301  -0.4528  -0.0502  0.027  0.4374  1e-03
## s.e.  0.8161   0.9042   0.1143  0.818  0.8502  6e-04
##
## sigma^2 estimated as 0.0001704:  log likelihood=1468.64
## AIC=-2923.28   AICc=-2923.06   BIC=-2893.75
```

auto.arima suggests using an ARMA(3,2) model. $Y_t = .0301 * Y_{t-1} - .4528 * Y_{t-2} - .0502 * Y_{t-3} + .027 * e_{t-1} + .4374 * e_{t-2} + e_t$

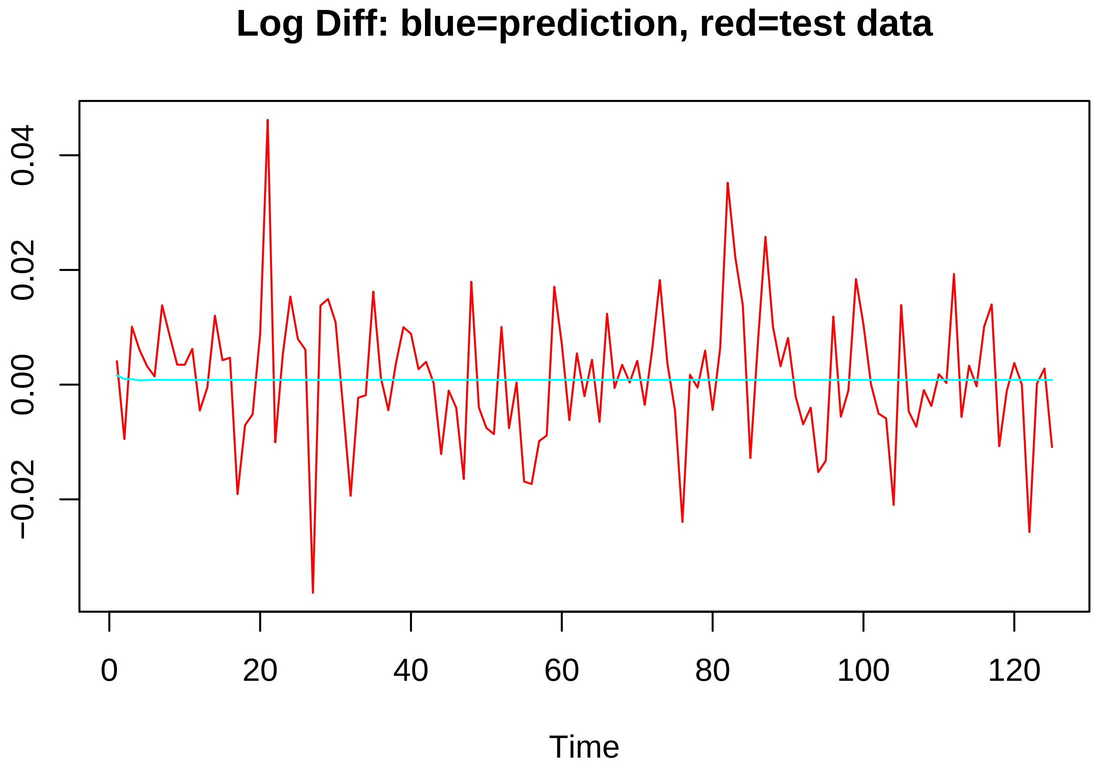
(d) Now, use 75% of the data to train the model and plot both your original data and the predicted data. Use different colors

```
AAPL_log_diff_Train <- AAPL_log_diff[1:377]
AAPL_log_diff_Test <- AAPL_log_diff[378:length(AAPL_log_diff)]

Model <- auto.arima(AAPL_log_diff_Train)

plot(forecast(Model, 125))
```

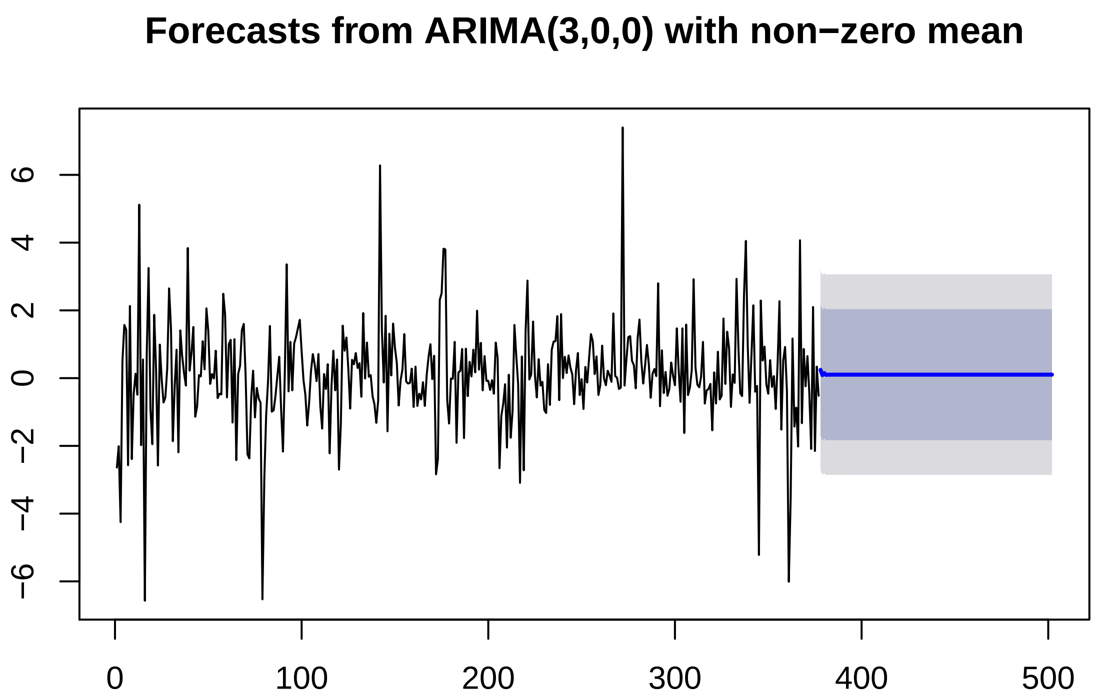



```
prediction <- forecast(Model, 125)

testandpredict <- data.frame(AAPL=AAPL_log_diff_Test, Predict=prediction$mean)
```

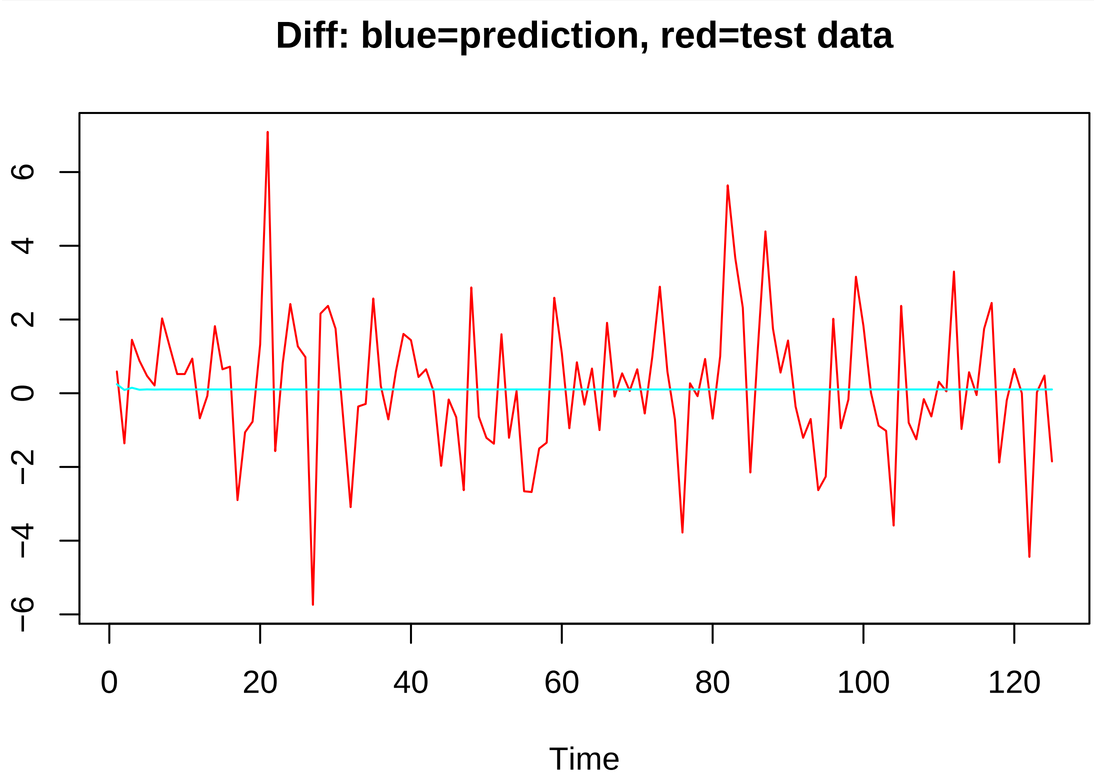
```r
ts.plot(testandpredict, col=rainbow(2), main="Log Diff: blue=prediction, red=test data")
```




```r
AAPL_diff <- diff(AAPL$AAPL.Close)[-1]

AAPL_diff_Train <- AAPL_diff[1:377]
AAPL_diff_Test <- AAPL_diff[378:length(AAPL_diff)]

Model <- auto.arima(AAPL_diff_Train)

plot(forecast(Model, 125))
```




```r
prediction <- forecast(Model, 125)

testandpredict <- data.frame(AAPL=AAPL_diff_Test, Predict=prediction$mean)

ts.plot(testandpredict, col=rainbow(2), main="Diff: blue=prediction, red=test data")
```



I tried predicting for both the logged and untrasformed differences and found nearly the exact same results.

e) Calculate MSE for the predicted value and decide if it is a reasonable model to use.

```
MSE <- mean((ts(AAPL_log_diff_Test, start = c(378,1))-testandpredict$Predict)^2)
cat(MSE)
```

```
## 0.01090345
```