Extra Credit Assignment

Yousra Belgaid

2023-04-25

Read and Display the Data

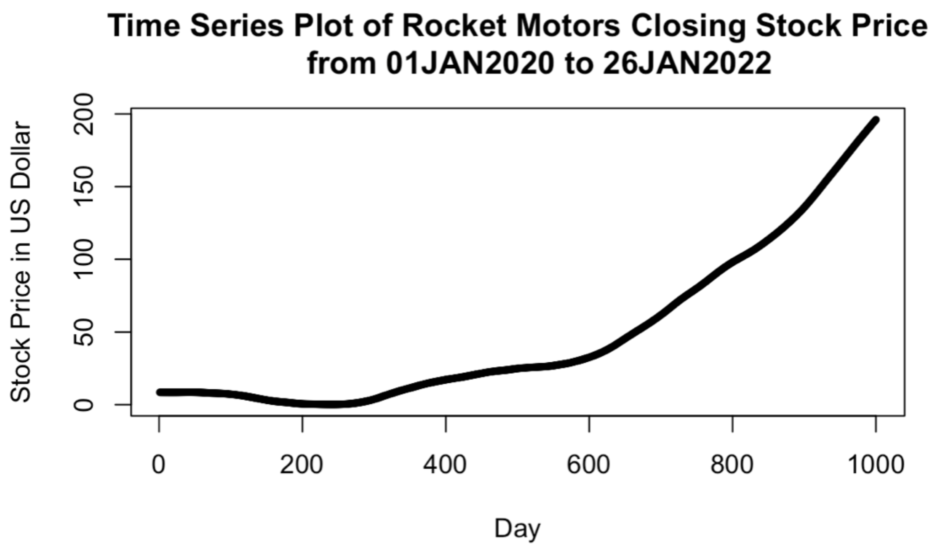
```
rocket<-read.table("~/Desktop/Stat 463/Rocket_Motors-1.txt", header = TRUE)
names(rocket)

## [1] "Date"  "Price"

attach(rocket)
length(Price)

## [1] 1000
```

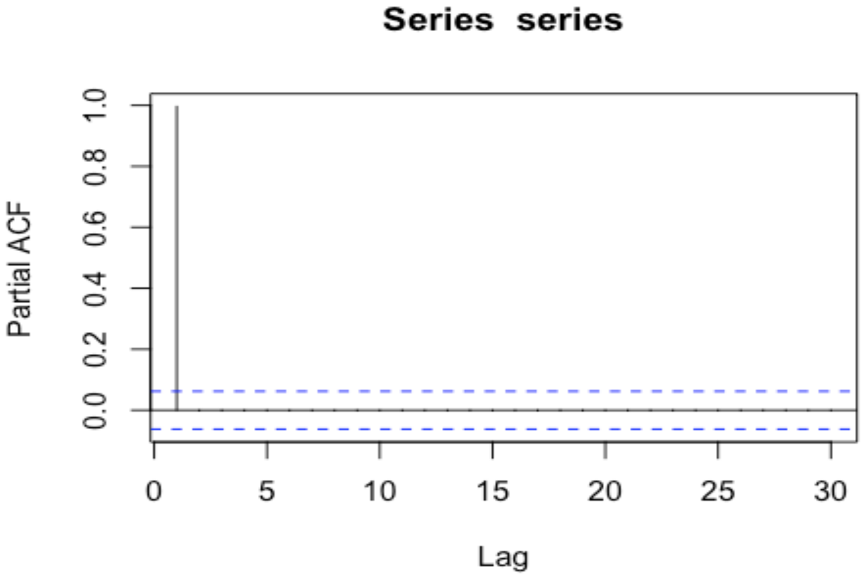
1. The following time series plot shows a clear upward trend of the data. There is clearly a trend element throughout the plot. The ACF plot shows that there are significant autocorrelations throughout. Therefore the data should be differenced in order to remove autocorrelation.

   The ARIMA model's two major assumptions are that the dataset is **(1)** stationary and **(2)** univariate. The beginning step is running a unit root test to examine how a time series dataset being determined by a trend. Such a test has a null hypothesis that the time series is nonstationary. If a unit root does exist, then the time series is nonstationary. Moreover, based on the Augmented Dickey-Fuller Test which is a stationary Unit Root test considers the null hypothesis that the data is non-stationary. In the case of Rocket Motors plot, it is clearly not stationary because the p-value of 0.99 > the alpha value of 0.05.

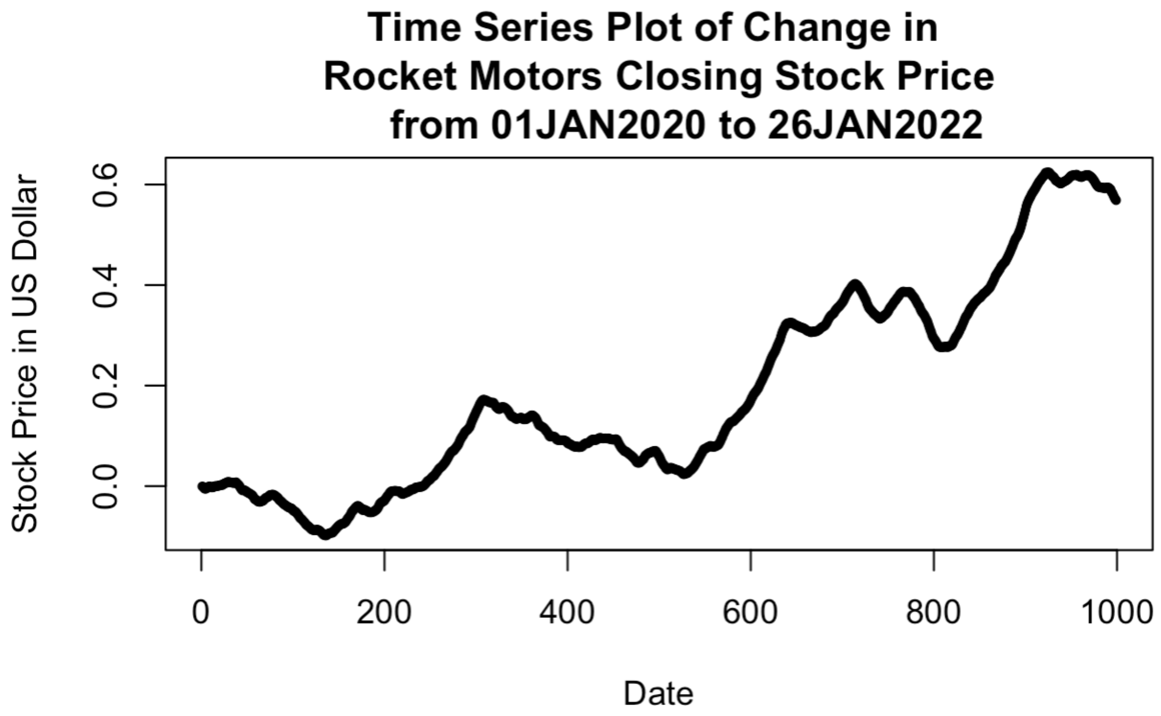
   Another Unit test used is the KPSS Test for Level Stationarity where the null hypothesis consideres the time series is trend stationary. If the p-value of the test is less than the significance level ($\alpha = .05$) then we reject the null hypothesis and conclude that the time series is not stationary. In this case, our p-value is 0.01 < 0.05, therefore, we reject the null and conclude that our data is not stationary.

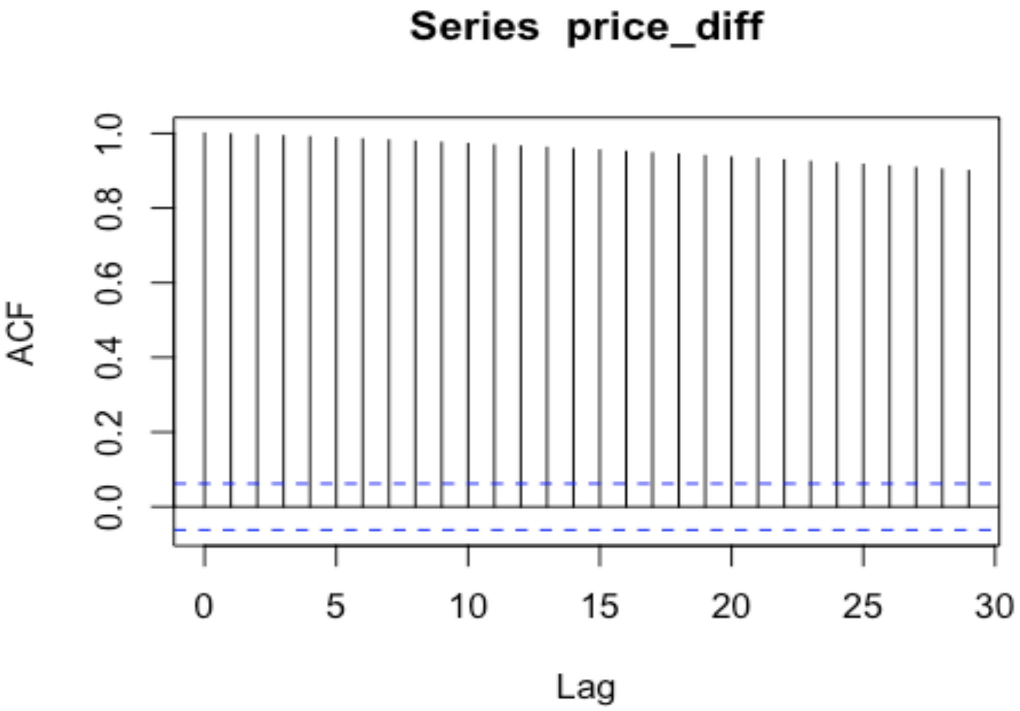
```
plot(Price, pch = 19, cex = 0.5, type = "o", xlab = "Day", ylab = "Stock
Price in US Dollar", main = "Time Series Plot of Rocket Motors Closing Stock
Price
     from 01JAN2020 to 26JAN2022")

series <- ts(rocket$Price,frequency=1)
acf(series, type = "correlation")

acf(series, type = "partial")
```

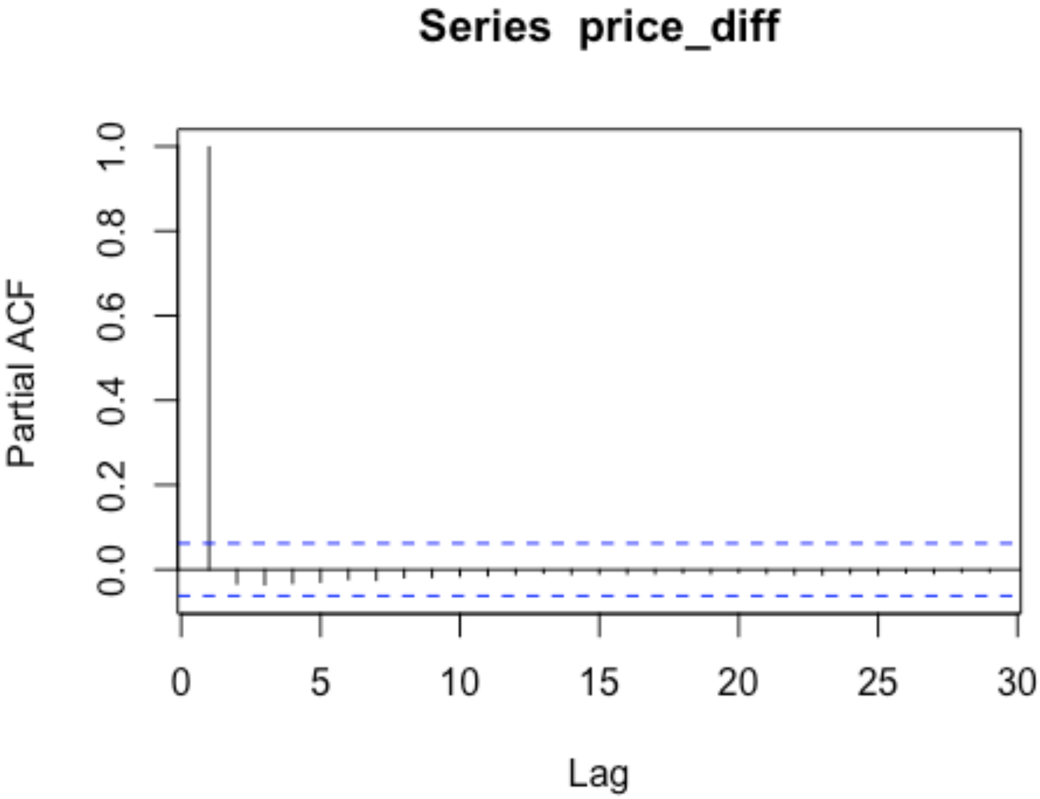




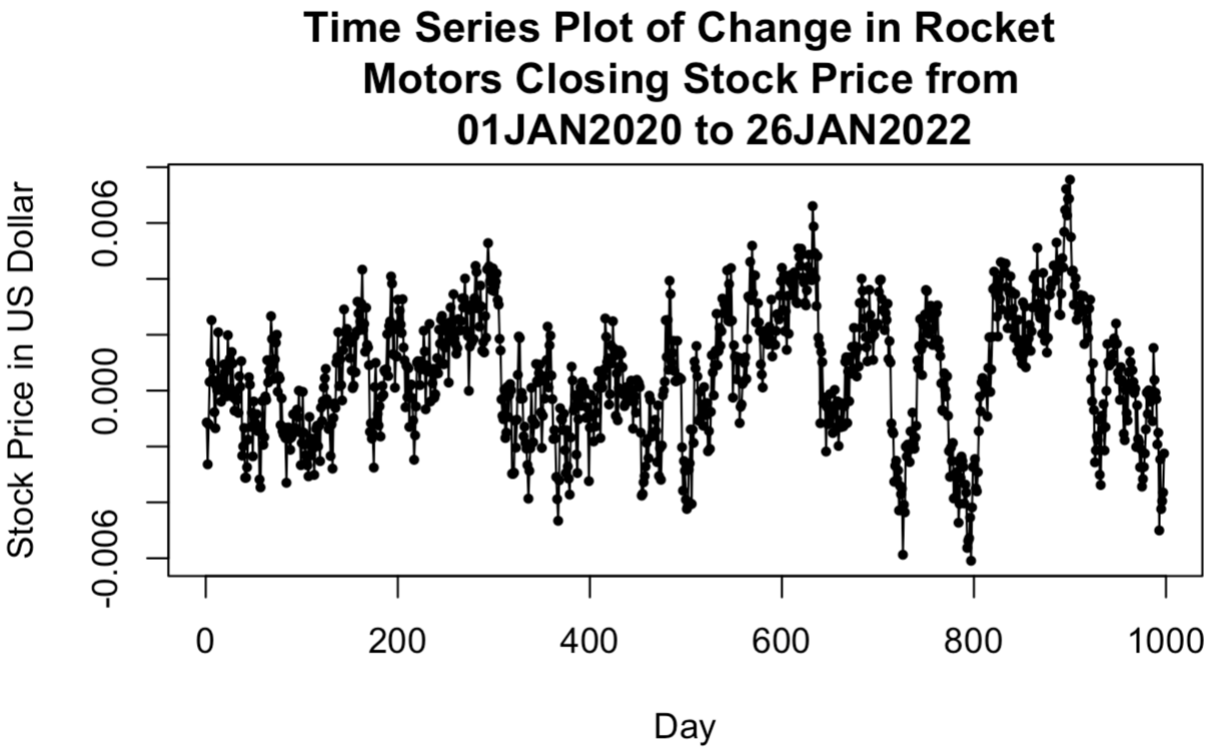
```
adf.test(series)
##  Augmented Dickey-Fuller Test
##
## data:  series
## Dickey-Fuller = 0.65879, Lag order = 9, p-value = 0.99
## alternative hypothesis: stationary

kpss.test(series)
##  KPSS Test for Level Stationarity
## data:  series
## KPSS Level = 10.42, Truncation lag parameter = 7, p-value = 0.01

price_diff = rocket$Price[2:length(rocket$Price)] -
rocket$Price[1:length(rocket$Price)-1]
plot(price_diff, pch = 19, cex = 0.5, type = "o", xlab = "Date", ylab =
"Stock Price in US Dollar", main = "Time Series Plot of Change in Rocket
Motors Closing Stock Price
     from 01JAN2020 to 26JAN2022")
```



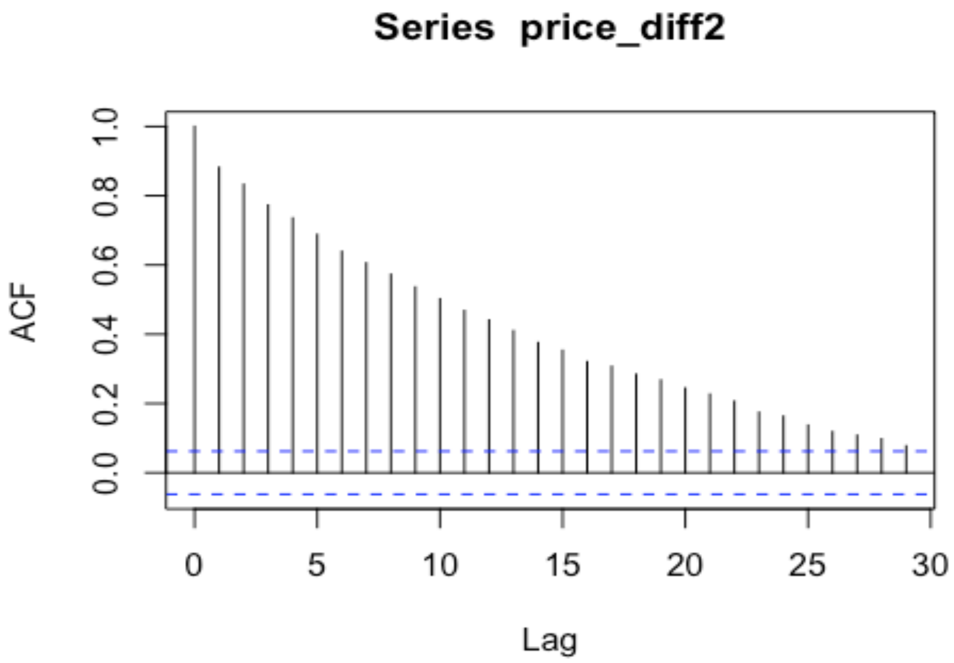

Series price_diff
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Lag

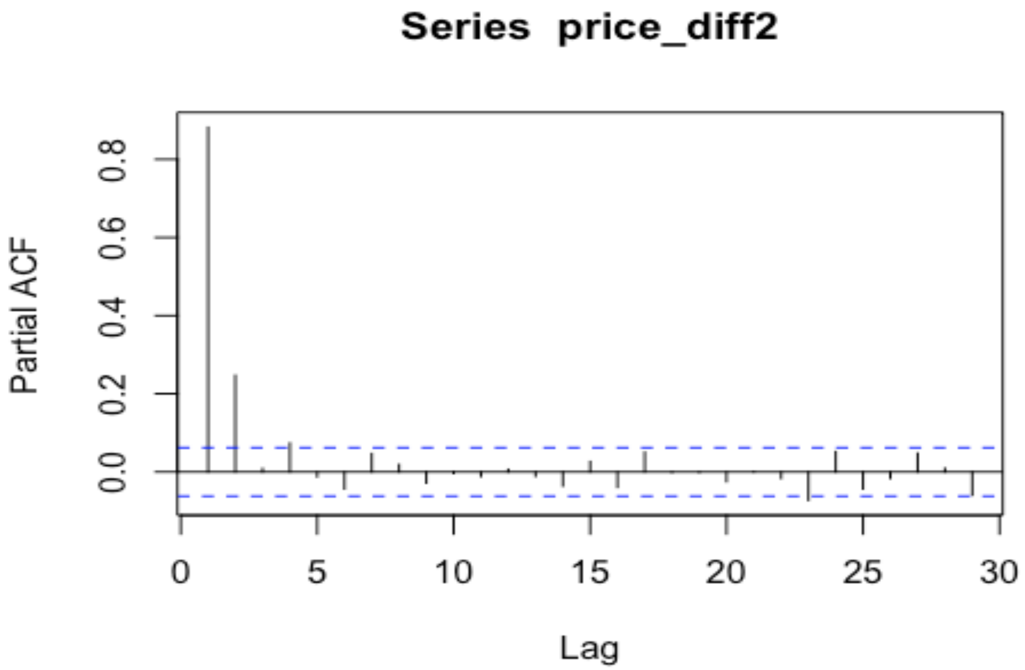

Series price_diff
Partial ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Lag

```
acf(price_diff, type = "correlation")

acf(price_diff, type = "partial")

price_diff2 = price_diff[2:length(price_diff)] -
price_diff[1:length(price_diff)-1]
plot(price_diff2, pch = 19, cex = 0.5, type = "o", xlab = "Day", ylab =
"Stock Price in US Dollar", main = "Time Series Plot of Change in Rocket
Motors Closing Stock Price
     from 01JAN2020 to 26JAN2022")
```


Time Series Plot of Change in Rocket Motors Closing Stock Price from 01JAN2020 to 26JAN2022

```
acf(price_diff2, type = "partial")

acf(price_diff2, type = "correlation")
```





```
change_diff2<-ts(price_diff2, frequency = 1)
adf.test(change_diff2)

## Warning in adf.test(change_diff2): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  change_diff2
## Dickey-Fuller = -4.8479, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary

kpss.test(change_diff2)

##
##  KPSS Test for Level Stationarity
##
## data:  change_diff2
## KPSS Level = 0.49716, Truncation lag parameter = 7, p-value = 0.04231
```

2. First, in order to fit an ARIMA(p,q,d) model, the time series plot needs to satisfy stationarity assumptions. In this case, we used Differencing: Take the first-order difference (subtracting each observation from the previous observation) and second-order difference (subtracting each first-order difference from the previous first-order difference) until the series becomes stationary.

   By then conducting the adf test, we were able to find that p-value is 0.01 which is less than the alpha value of 0.05, therefore, we can assume that the time series is stationary. ARIMA models have three components: the order of autoregression, degree of differencing, and the order of the moving average. In this case, we used an order differencing of 2 to assume stationarity so we will use a (0,2,0) model. To find the AR (**Autoregressive terms are lagged values of the variable being predicted**) and MA (**Moving average terms are lagged forecast errors, which represent the difference between the predicted value and the actual value**) components, I used the 2nd ordered difference Price with the lowest AIC value to determine the best fit.

   The log-likelihood values of the third model is higher than that of the first model, which indicates that it fits the data better. However, the AIC (Akaike Information Criterion) values of the third model is lower than that of the first model, which indicates that it provides a better trade-off between goodness of fit and model complexity. Therefore, based on the provided output, it seems that the third model with an AR order of 3 is the best model among the three, as it has the lowest AIC value. Similarly, when fitting the MA model, the third model with an MA order of 3 is the best model among the three, as it has the lowest AIC value. The ACF plot of the residuals from the ARIMA(3,2,3) model shows that all auto-correlations are within the threshold limits, indicating that the residuals are behaving like white noise. A portmanteau test returns a large p-value, also suggesting that the residuals are white noise.Therefore, the best ARIMA model is the ARIMA(3,2,3).

The characteristic polynomial of the AR part is denoted by ϕ(B), where B is the backward shift operator. The roots of this polynomial are the values of B that solve the equation ϕ(B) = 0. These roots are also known as the autoregressive roots, and they provide information about the dynamics of the AR process. Similarly, the characteristic polynomial of the MA part is denoted by θ(B), and its roots are the values of B that solve the equation θ(B) = 0. These roots are also known as the moving average roots, and they provide information about the dynamics of the MA process. When fitting an ARIMA model, it is important to ensure that the roots of both ϕ(B) and θ(B) lie inside the unit circle. This is because a stationary ARIMA model must have all of its roots lie inside the unit circle, and an invertible ARIMA model must have all of its MA roots lie inside the unit circle. In the provided output, all of the red dots representing the roots of ϕ(B) and θ(B) are inside the unit circle, which indicates that the fitted ARIMA model is both stationary and invertible.

I then used the auto.arima() function to justify that the model I chose is the appropriate model. By using this function, I was also able to get that the model is best fitted as the ARIMA(3,2,3) model.

```
AR1b = arima(x = price_diff2, order = c(1, 0, 0), method = "ML", include.mean
= FALSE)
print(AR1b)

##
## Call:
## arima(x = price_diff2, order = c(1, 0, 0), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ar1
##       0.8895
## s.e.  0.0144
##
## sigma^2 estimated as 1.142e-06:  log likelihood = 5410.98,  aic = -
10817.97

AR2 = arima(x = price_diff2, order = c(2, 0, 0), method = "ML", include.mean
= FALSE)
print(AR2)

##
## Call:
## arima(x = price_diff2, order = c(2, 0, 0), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ar1     ar2
##       0.6592  0.2590
## s.e.  0.0306  0.0306
##
```

```
## sigma^2 estimated as 1.065e-06:  log likelihood = 5445.49,  aic = -
10884.99
```

```
AR3 = arima(x = price_diff2, order = c(3, 0, 0), method = "ML", include.mean
= FALSE)
print(AR3)
```

```
##
## Call:
## arima(x = price_diff2, order = c(3, 0, 0), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ar1     ar2     ar3
##       0.6553  0.2491  0.0151
## s.e.  0.0316  0.0371  0.0317
##
## sigma^2 estimated as 1.065e-06:  log likelihood = 5445.6,  aic = -10883.21
```

```
MA1 = arima(x = price_diff2, order = c(0, 0, 1), method = "ML", include.mean
= FALSE)
print(MA1)
```

```
##
## Call:
## arima(x = price_diff2, order = c(0, 0, 1), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ma1
##       0.6522
## s.e.  0.0177
##
## sigma^2 estimated as 2.758e-06:  log likelihood = 4971.33,  aic = -9938.67
```

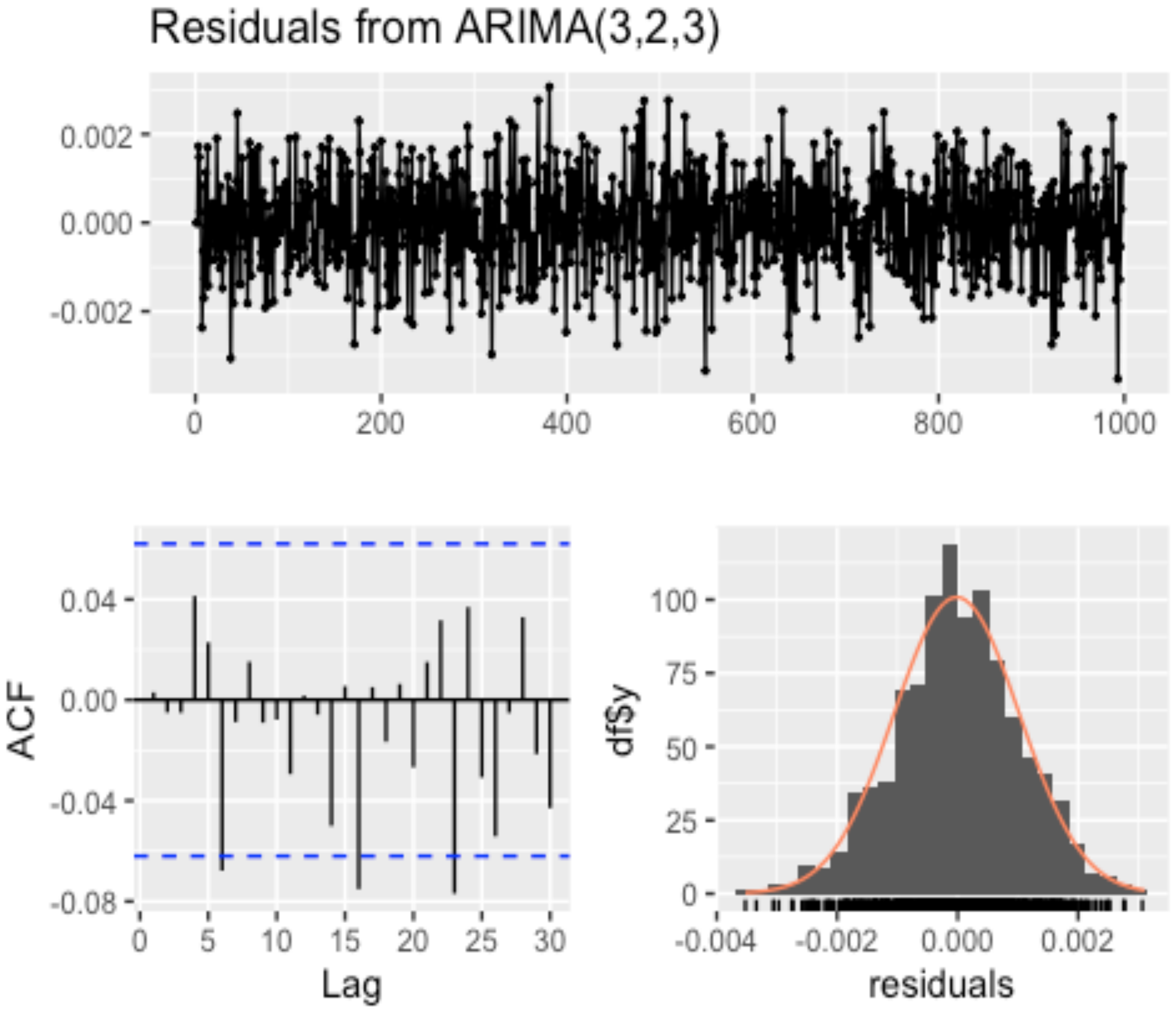
```
MA2 = arima(x = price_diff2, order = c(0, 0, 2), method = "ML", include.mean
= FALSE)
print(MA2)
```

```
##
## Call:
## arima(x = price_diff2, order = c(0, 0, 2), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ma1     ma2
##       0.7976  0.5612
## s.e.  0.0287  0.0219
##
## sigma^2 estimated as 1.832e-06:  log likelihood = 5175.13,  aic = -
10344.26
```

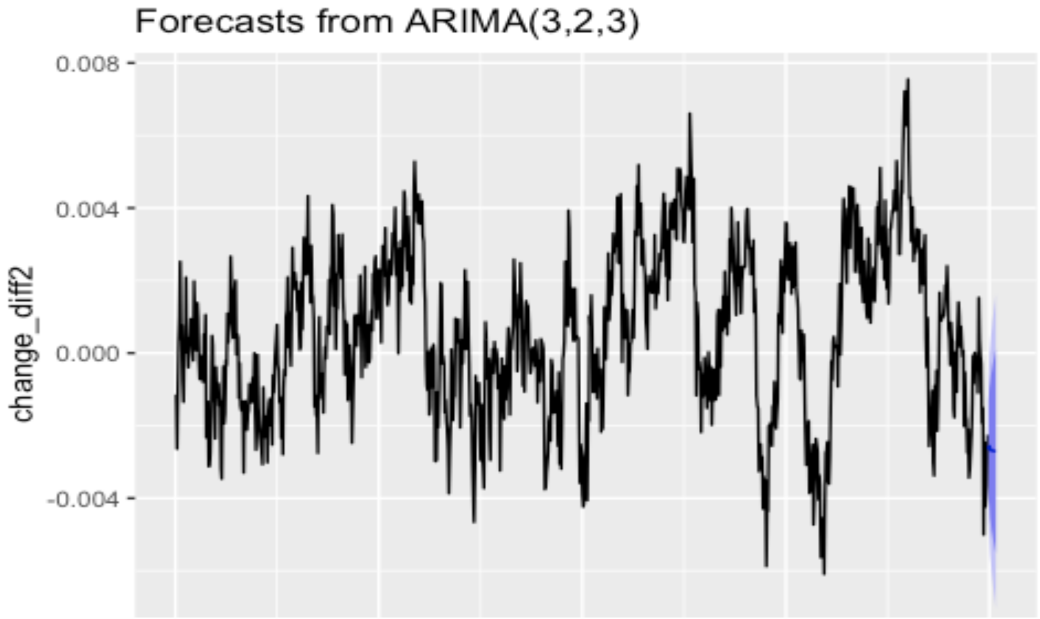
```
MA3 = arima(x = price_diff2, order = c(0, 0, 3), method = "ML", include.mean
= FALSE)
print(MA3)

##
## Call:
## arima(x = price_diff2, order = c(0, 0, 3), include.mean = FALSE, method =
"ML")
##
## Coefficients:
##          ma1     ma2     ma3
##       0.8566  0.7081  0.3224
## s.e.  0.0322  0.0292  0.0257
##
## sigma^2 estimated as 1.588e-06:  log likelihood = 5246.56,  aic = -
10485.12

(fit <- Arima(change_diff2, order=c(3,2,3)))

## Series: change_diff2
## ARIMA(3,2,3)
##
## Coefficients:
##           ar1     ar2     ar3      ma1      ma2     ma3
##       -0.0616  0.3576  0.0258  -1.2623  -0.0881  0.3505
## s.e.   0.3660  0.4561  0.1691   0.3625   0.8115  0.5217
##
## sigma^2 = 1.092e-06:  log likelihood = 5421.93
## AIC=-10829.86   AICc=-10829.75   BIC=-10795.54

checkresiduals(fit)
```

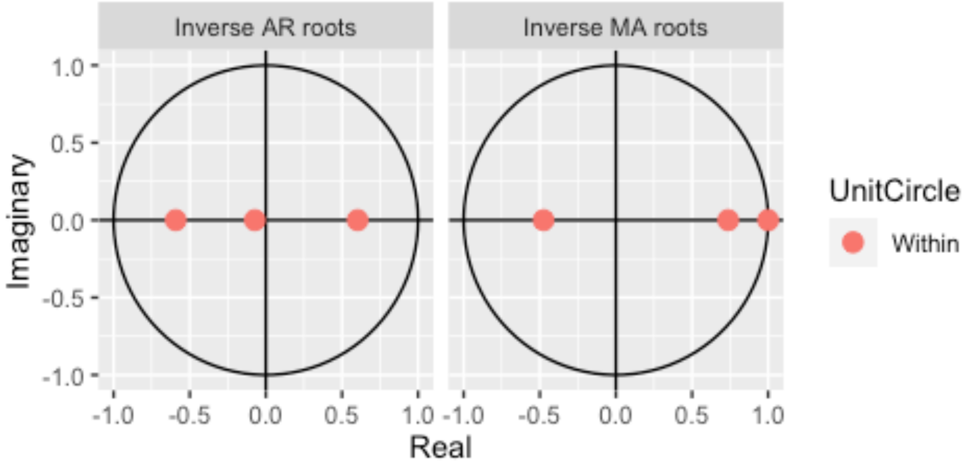
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,2,3)
## Q* = 7.3881, df = 4, p-value = 0.1167
##
## Model df: 6.   Total lags used: 10

autoplot(forecast(fit))
```

```
autoplot(fit)
```



```
auto.arima(series)

## Series: series
## ARIMA(3,2,3)
##
## Coefficients:
##           ar1     ar2     ar3     ma1     ma2      ma3
##       -0.0505  0.3119  0.5983  0.7074  0.3866  -0.2218
## s.e.   0.2754  0.1366  0.2778  0.2753  0.2245   0.0703
##
## sigma^2 = 1.208e-06:  log likelihood = 5450.05
## AIC=-10886.1   AICc

=-10885.99   BIC=-10851.76
```





3. **Yt = –0.0505Yt-1 + 0.3119Yt-2 + 0.5983Yt-2 +et + 0.7074et-1 + 0.3866et-2 - 0.2218et-3**