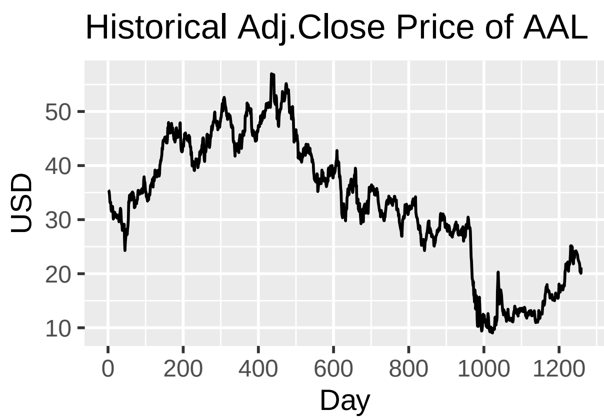
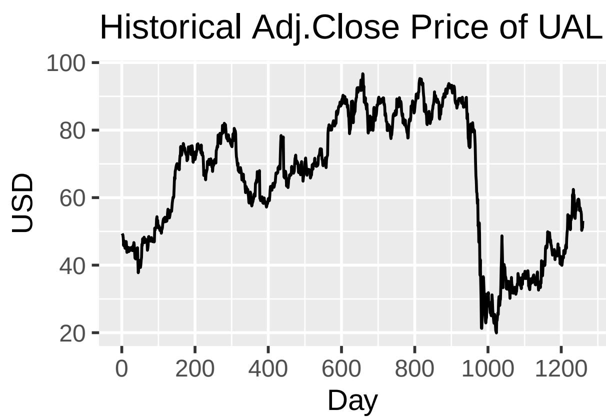
Airline Stocks Time Series

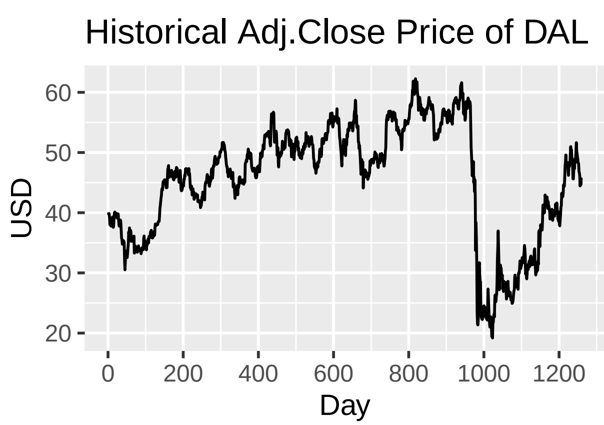
Ava Zhang, Betty Chen, Roopali Negi, Steven Wang, Xi Yang

```
##        Date      AAL   UAL      DAL
## 1 2016/4/25 35.45536 49.37 40.05110
## 2 2016/4/26 35.37871 49.14 39.74113
## 3 2016/4/27 35.17748 48.94 39.83230
## 4 2016/4/28 34.46836 48.06 39.07558
## 5 2016/4/29 33.24179 45.81 37.99065
## 6  2016/5/2 33.08803 46.66 38.44650
```

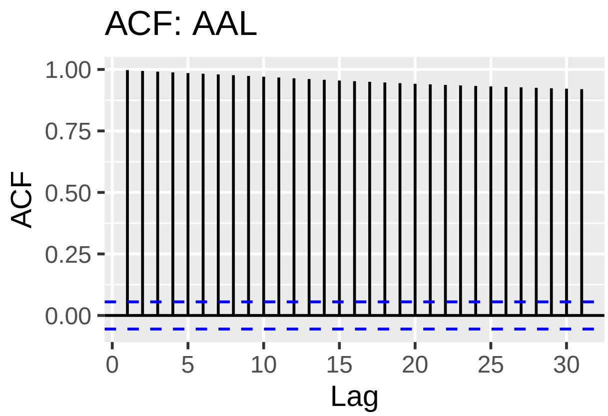
We represent below the closing prices for the 3 different stocks:

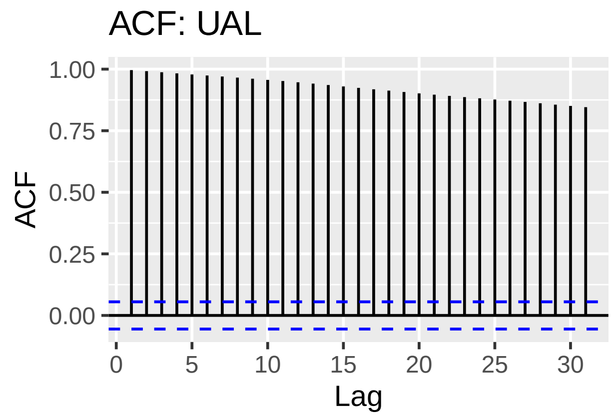


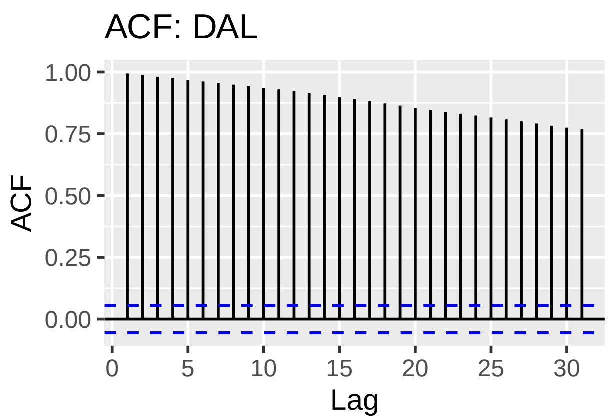




We run for every stock the autocorrelation function, to take a first look at the presence of stationarity. The ACF of stationary data drops to zero relatively quickly while the ACF of non-stationary data decreases slowly.



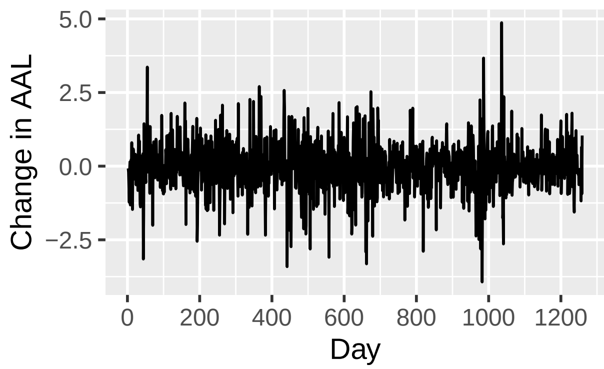


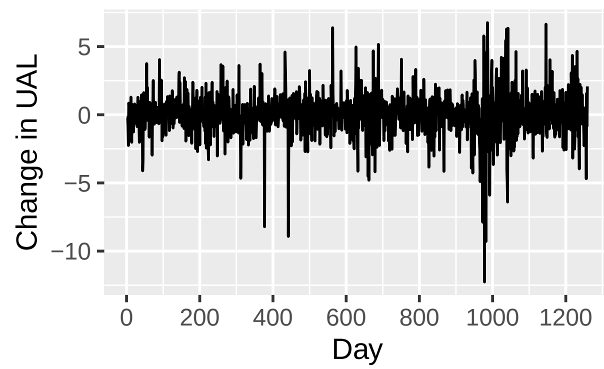


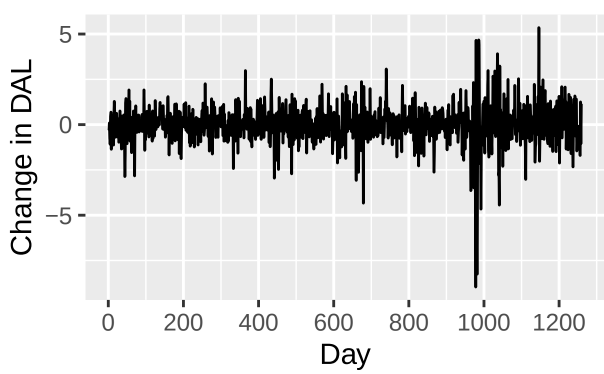
trend data, non-stationary. non-seasonal. auto-correlation in the data.

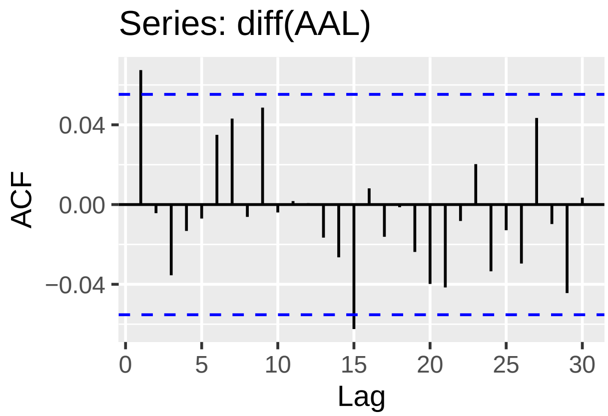
In this case, we can see that the autocorrelation function tends to 0 very slowly. All 3 variables are non-stationary.

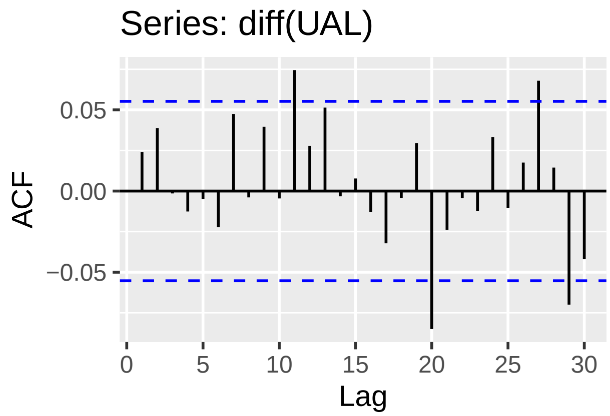
We can try to stabilize the data by differentiating it.

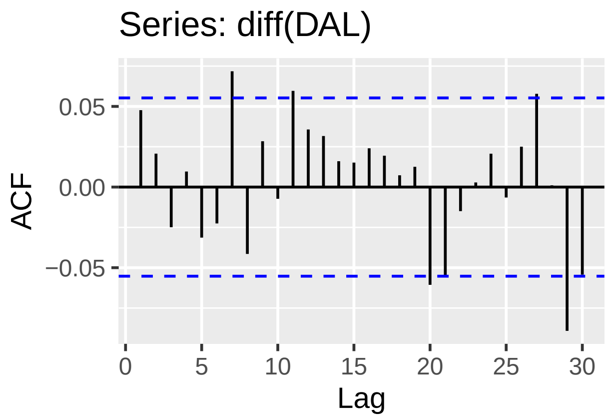

Change in AAL
5.0
2.5
0.0
−2.5
0
200
400
600
800
1000
1200
Day


Change in UAL
5
0
−5
−10
0
200
400
600
800
1000
1200
Day


Change in DAL
5
0
−5
0
200
400
600
800
1000
1200
Day







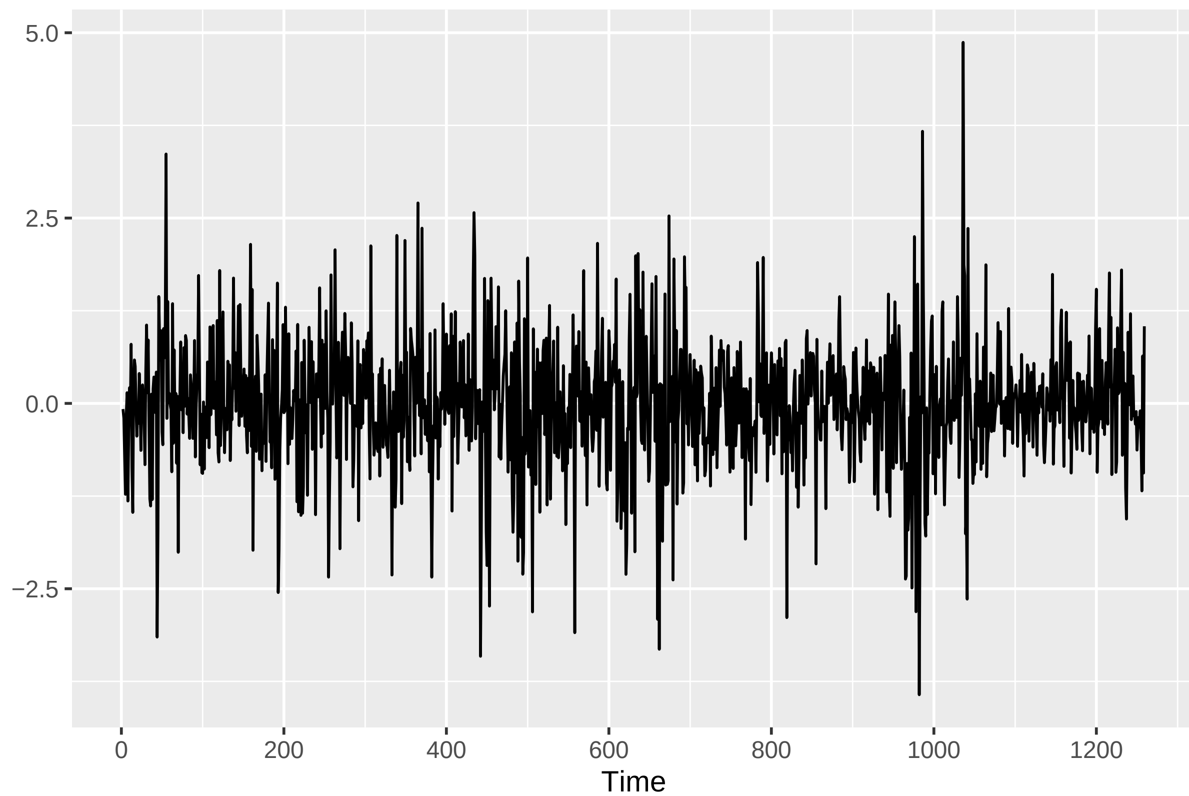
By differencing the AAL stock, we were able to make the data stationary.

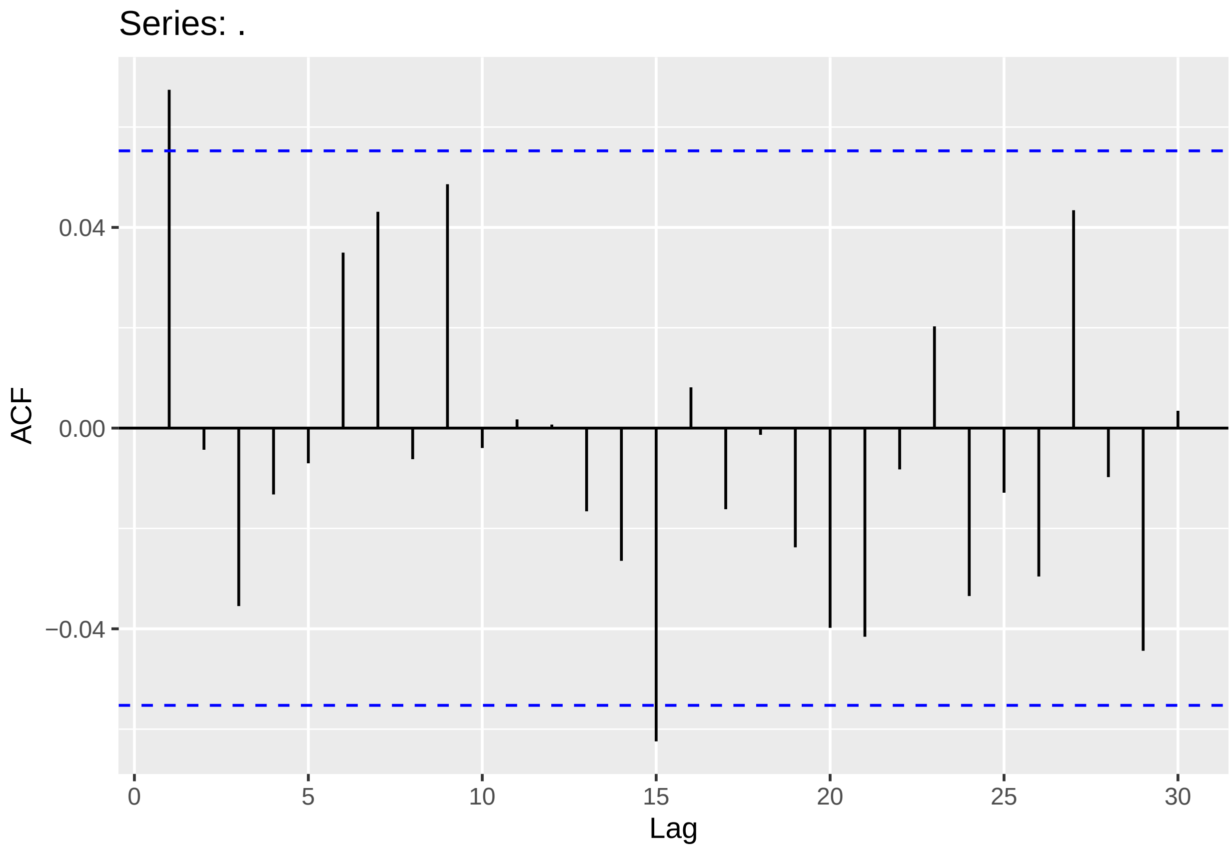
The autocorrelation function shows values close to zero for every lag.

The ndiffs function uses a unit root test to determine the number of differences required for time series x to be made stationary. The result shows that one difference was enough to make the data stationary.

In KPSS test, the null hypothesis is that the data is stationary and non-seasonal. The low value of the test-statistic confirms that the data is stationary.

```
## [1] 0
```


5.0
2.5
0.0
−2.5
0
200
400
600
800
1000
1200
Time




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.1153
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

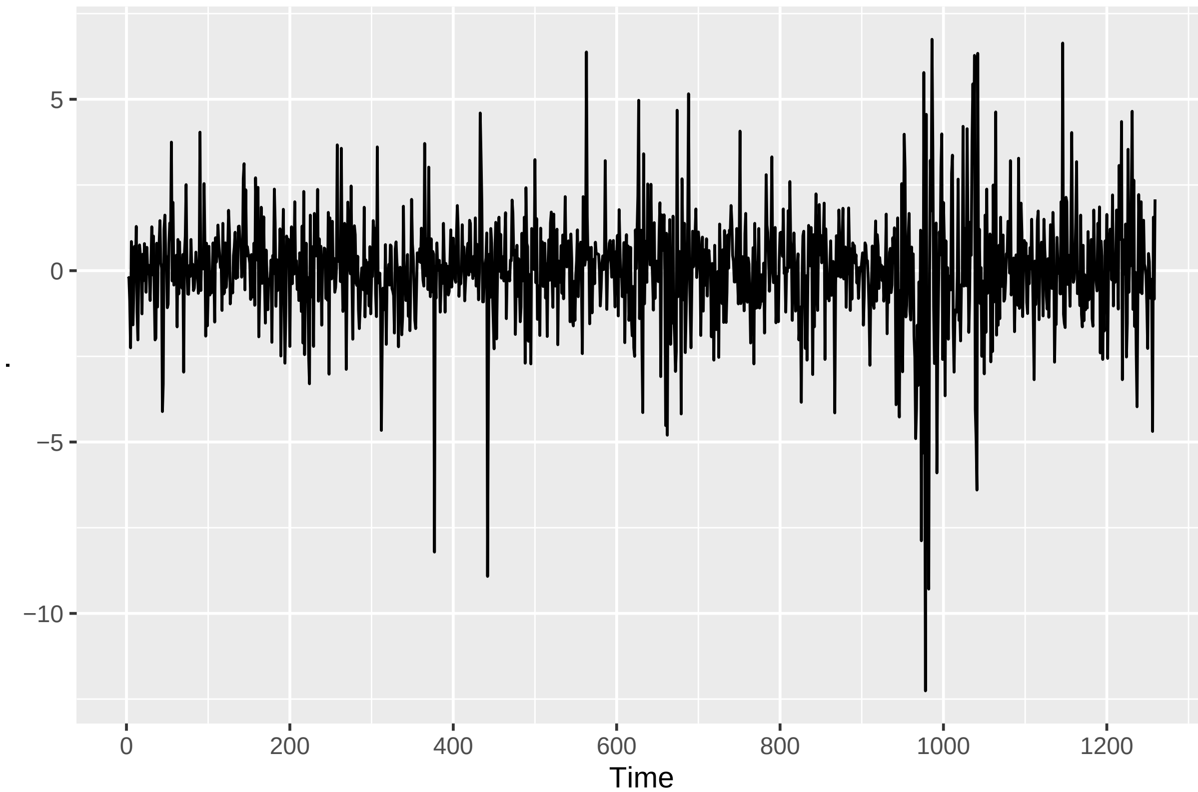
By differencing the UAL stock, we were able to make the data stationary.

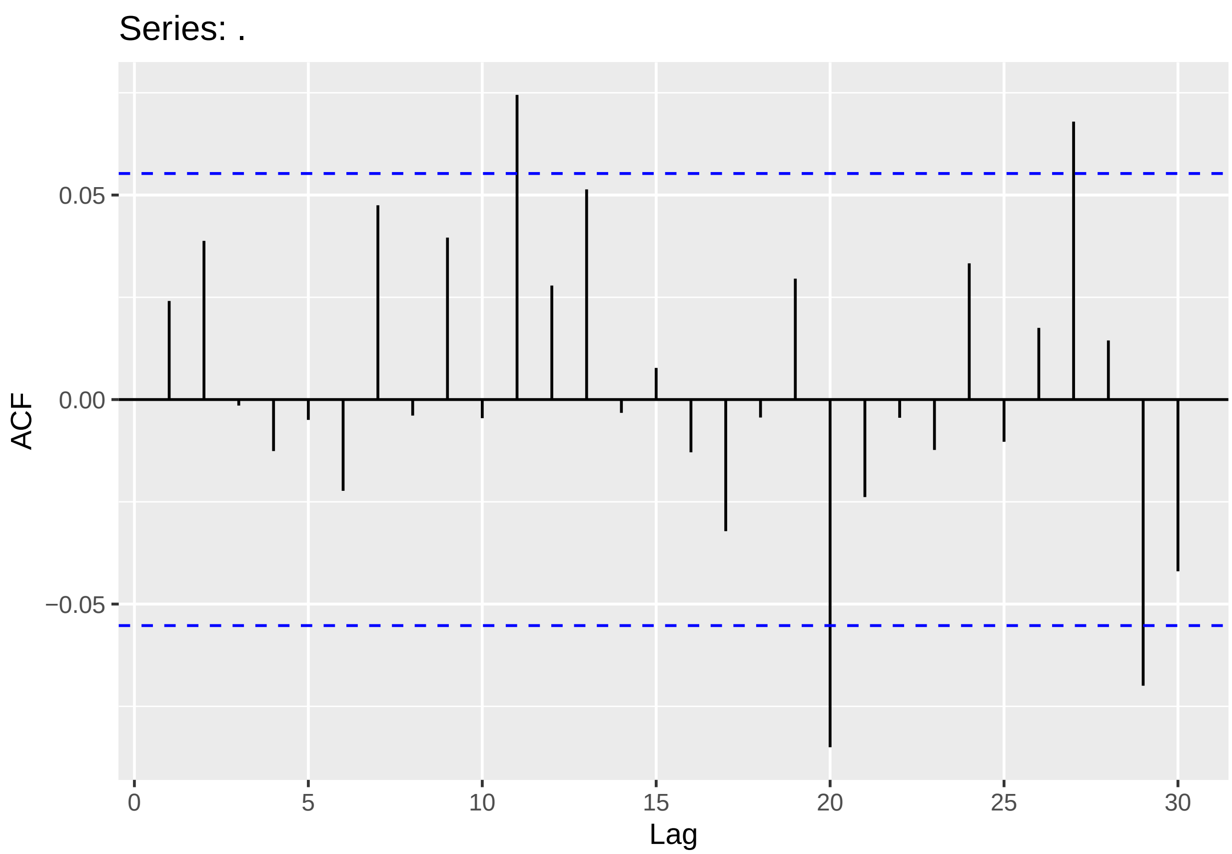
The autocorrelation function shows values close to zero for every lag.

The ndiffs function shows that one difference was enough to make the data stationary.

The low value of the test-statistic, in KPSS test, confirms that the data is stationary.

```
## [1] 0
```


5
0
−5
−10
0
200
400
600
800
1000
1200
Time
.




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.1638
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

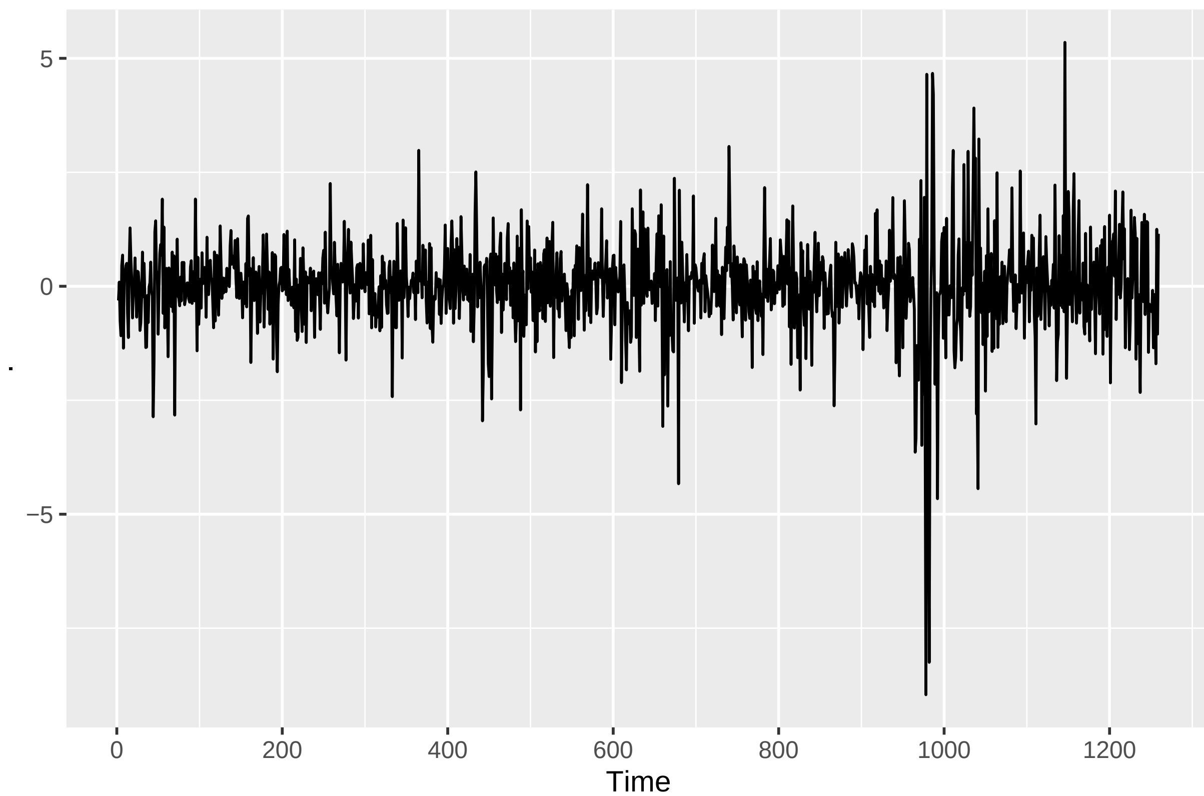
By differencing the DAL stock, we were able to make the data stationary.

The autocorrelation function shows values close to zero for every lag.

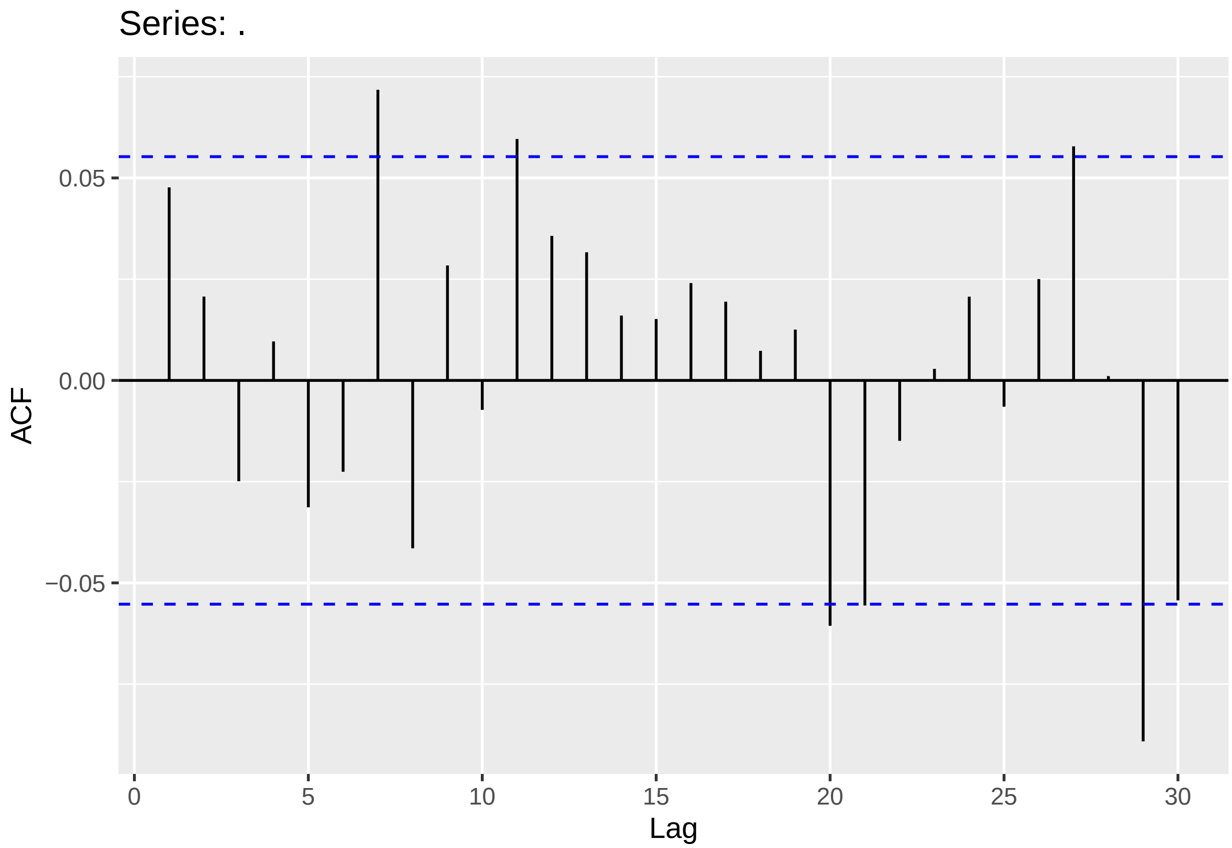
The ndiffs function shows that one difference was enough to make the data stationary.

The low value of the test-statistic, in KPSS test, confirms that the data is stationary.

```
## [1] 0
```



5
0
−5
.
0
200
400
600
800
1000
1200
Time




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.0728
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Log transformation can be used to stabilize the variance of a series with non-constant variance.

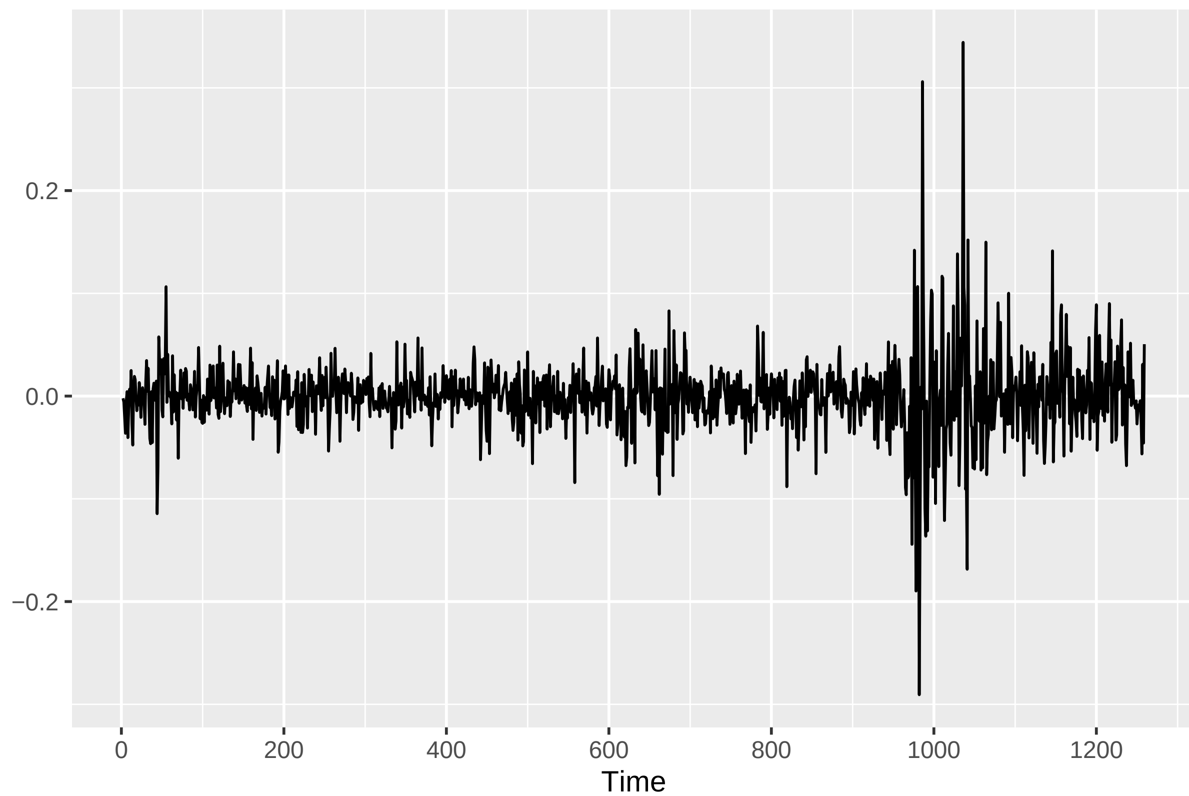
The ndiffs function shows that using a log transformation is not enough to make the data stationary. Instead, by using a log transformation and differencing, the ndiffs functions give us a result of 0.

The autocorrelation function shows values close to zero for every lag.

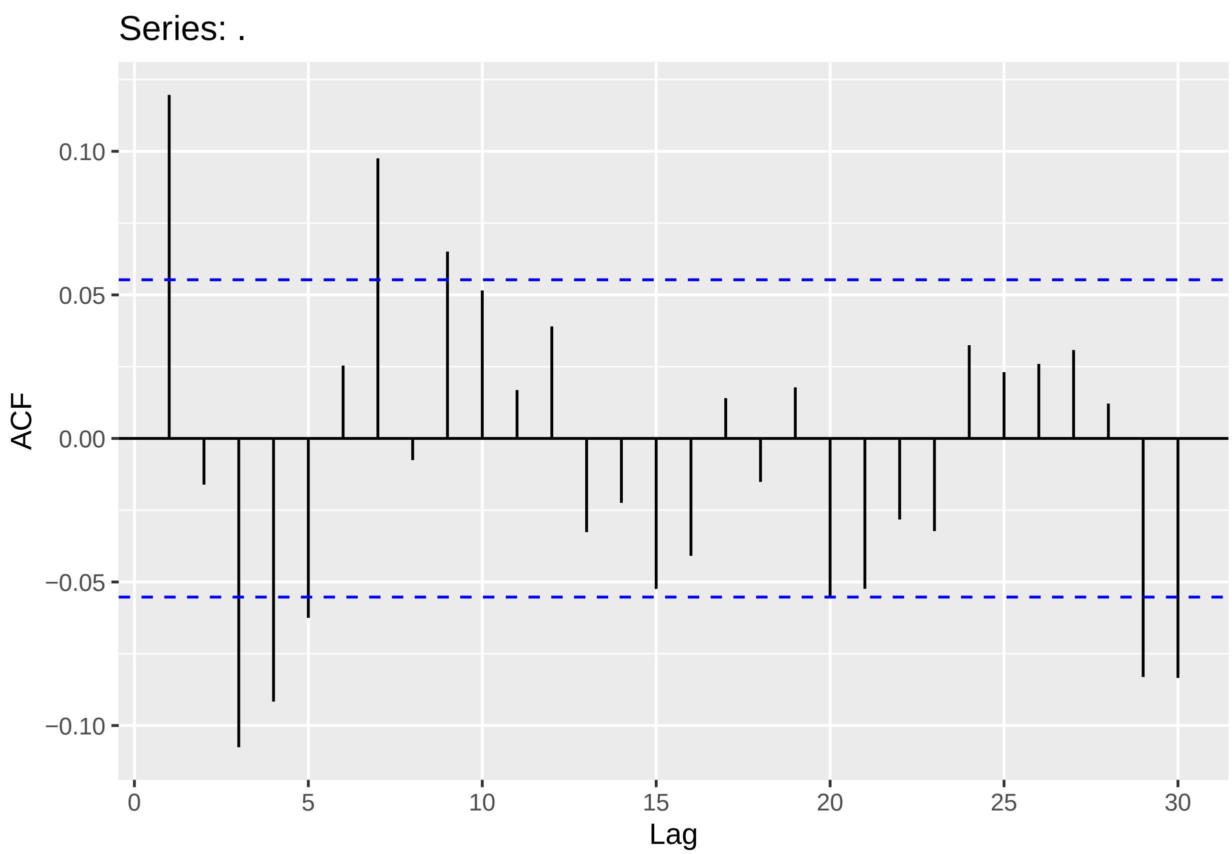
The KPSS test using a log transformation and differencing, in KPSS test, gives us a lower value of test-statistic, compared to only differentiating.

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

```
## [1] 0
```




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.093
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```



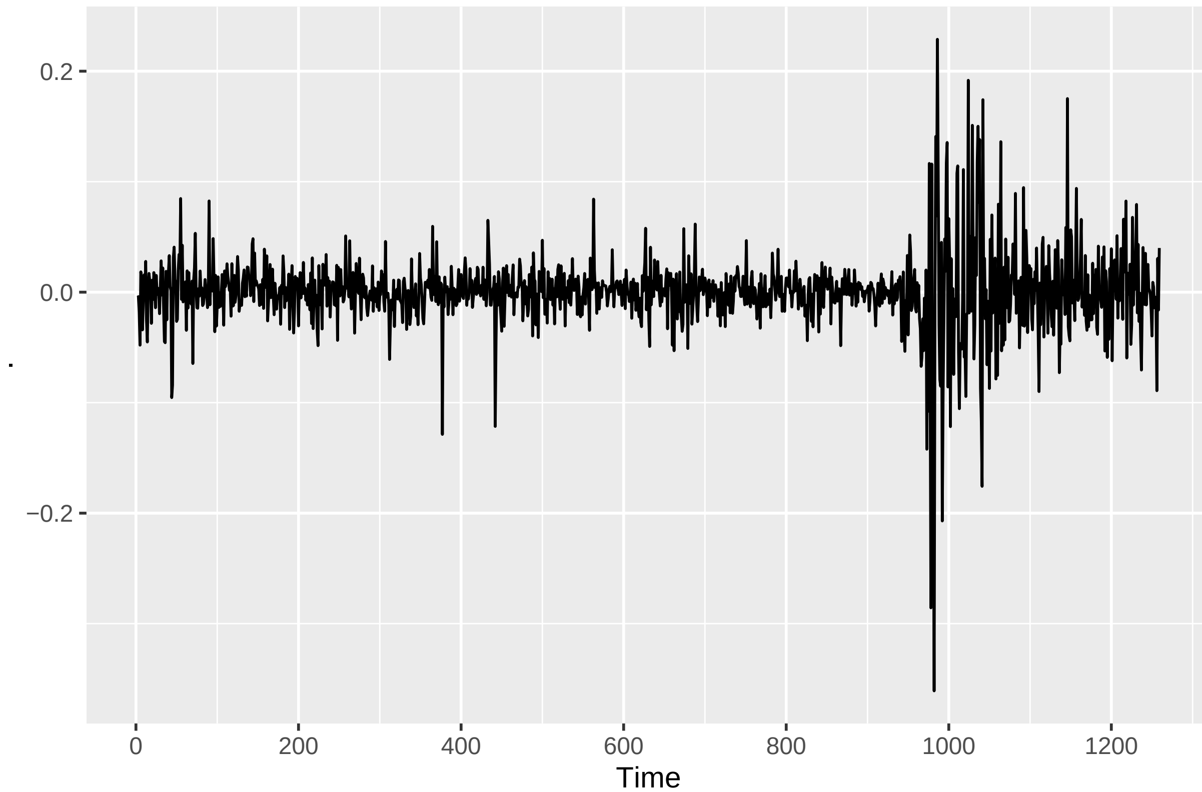
The ndiffs function shows that using a log transformation is not enough to make the data stationary. Instead, by using a log transformation and differencing, the ndiffs functions give us a result of 0.

The autocorrelation function shows values close to zero for every lag.

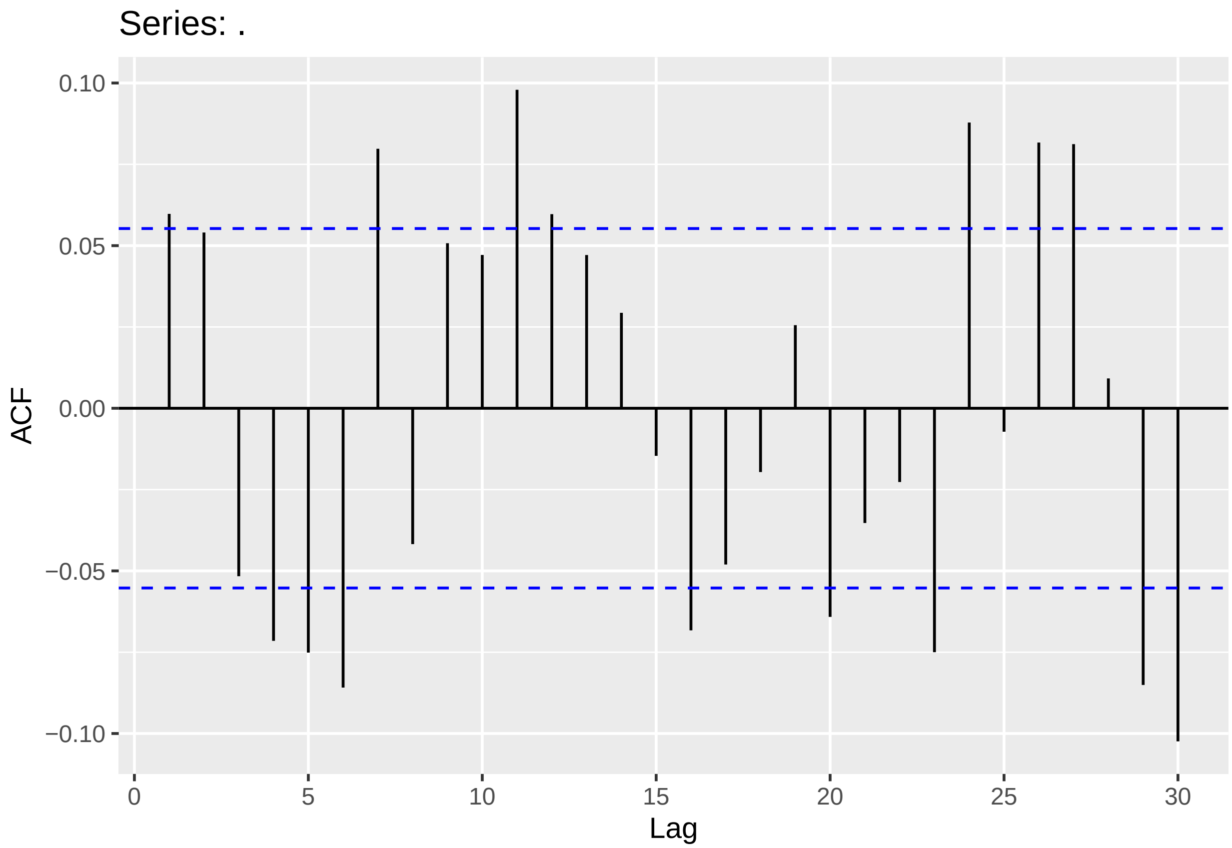
The KPSS test using a log transformation and differencing, in KPSS test, gives us a lower value of test-statistic, compared to only differentiating.

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

```
## [1] 0
```




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.1085
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```



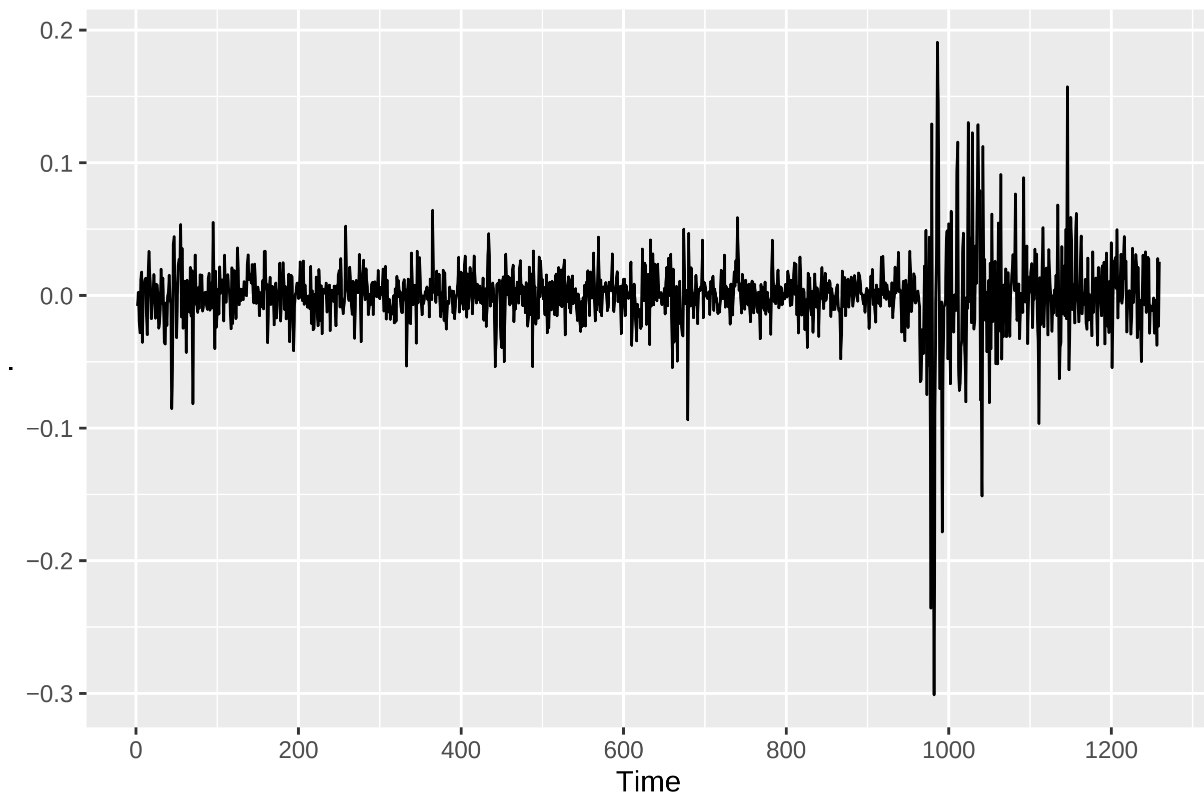
The ndiffs function shows that using a log transformation is not enough to make the data stationary. Instead, by using a log transformation and differencing, the ndiffs functions give us a result of 0.

The autocorrelation function shows values close to zero for every lag.

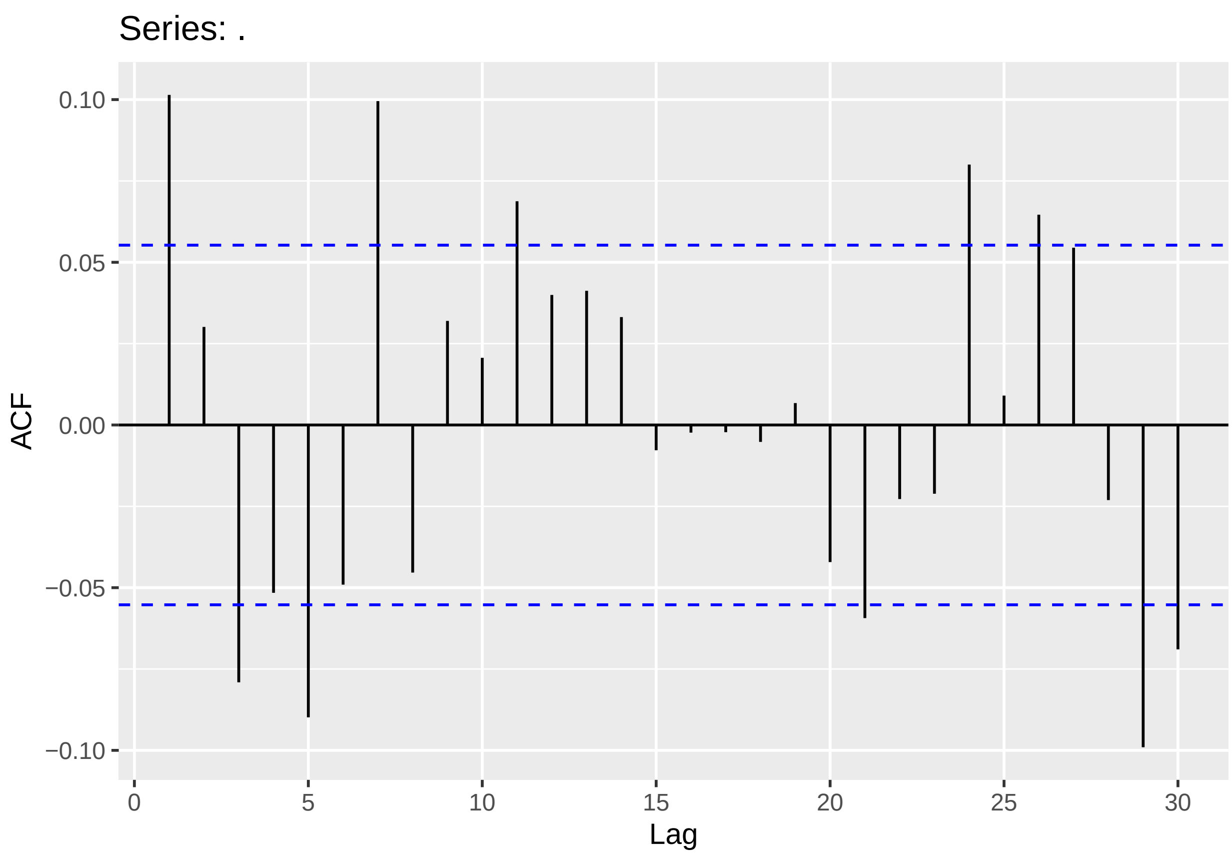
The KPSS test using a log transformation and differencing, in KPSS test, gives us a lower value of test-statistic, compared to only differentiating.

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

```
## [1] 0
```




```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 0.0633
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

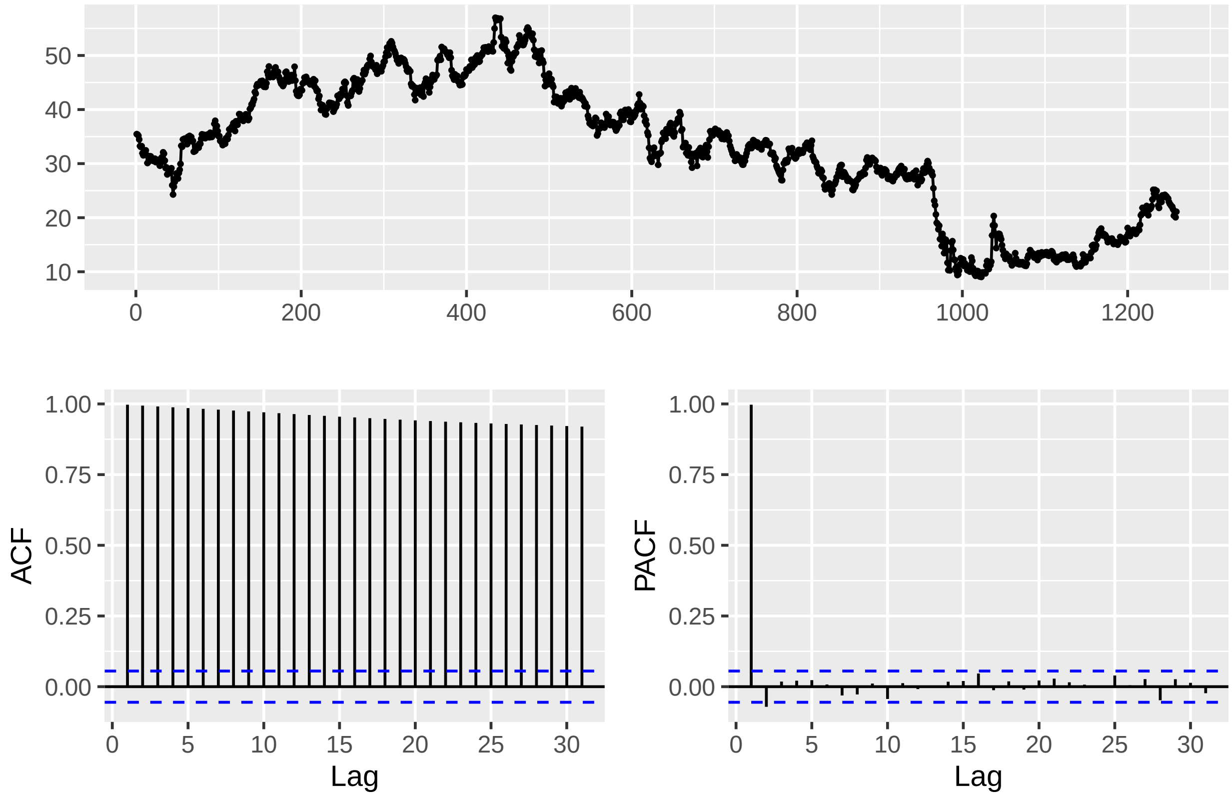


To forecast the future valuesof these stocks, we will use an arima(p,i,q) model.

We have to choose the correct value of the different parameters:

- i: we've already seen that we need to differentiate the data one time, so we can set i equal to 1.
- p: PACF has all zero spikes beyond the pth spike.
- q: ACF has all zero spikes beyond the qth spike

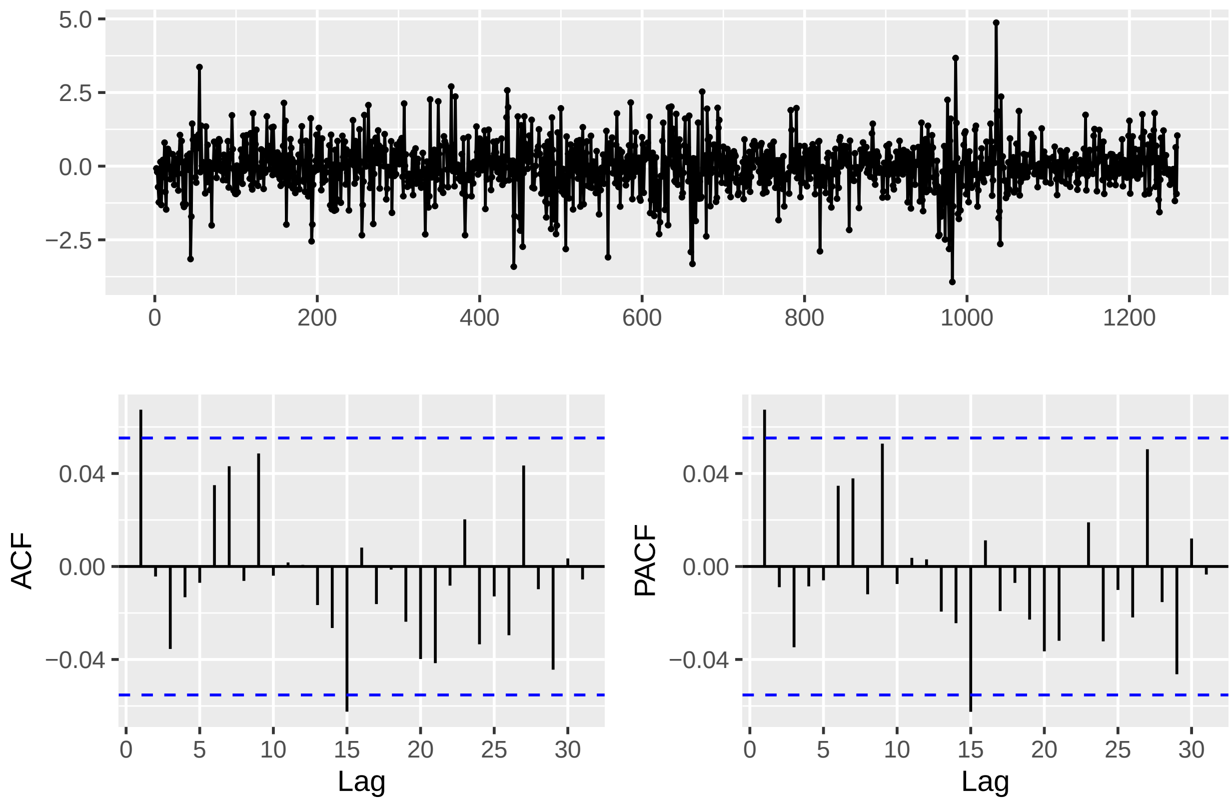
```
#AAL ARIMA
ggtsdisplay(AAL)
```




```r
ndiffs(AAL)
```

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

```r
ggtsdisplay(diff(AAL))
```



By looking at the autocorrelation funtion and partial autocorrelation function of the AAL stock, we could use a value of p=1 and q=1.

```r
#AAL ARIMA
fit_AAL <- Arima(AAL,order=c(0,1,1))
summary(fit_AAL)
```

```
## Series: AAL
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.0681
## s.e.  0.0281
##
## sigma^2 estimated as 0.7182:  log likelihood=-1576.33
## AIC=3156.66   AICc=3156.67   BIC=3166.94
##
## Training set error measures:
##                       ME      RMSE      MAE         MPE     MAPE      MASE
## Training set -0.01058389 0.8468002 0.612775 -0.09566096 2.221756 0.9973682
##                       ACF1
## Training set -9.017117e-05
```

```r
fit_AAL1 <- Arima(AAL,order=c(1,1,0))
summary(fit_AAL1)
```

```
## Series: AAL
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.0676
## s.e.  0.0281
##
## sigma^2 estimated as 0.7182:  log likelihood=-1576.36
## AIC=3156.71   AICc=3156.72   BIC=3166.99
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.0105394 0.8468166 0.6128064 -0.0952034 2.221897 0.9974193
##                        ACF1
## Training set 0.0004785698
```

The auto.arima functions indicates to use an ARIMA(0,1,1).

```r
#AAL ARIMA
fitauto_AAL <- auto.arima(AAL)
summary(fitauto_AAL)
```

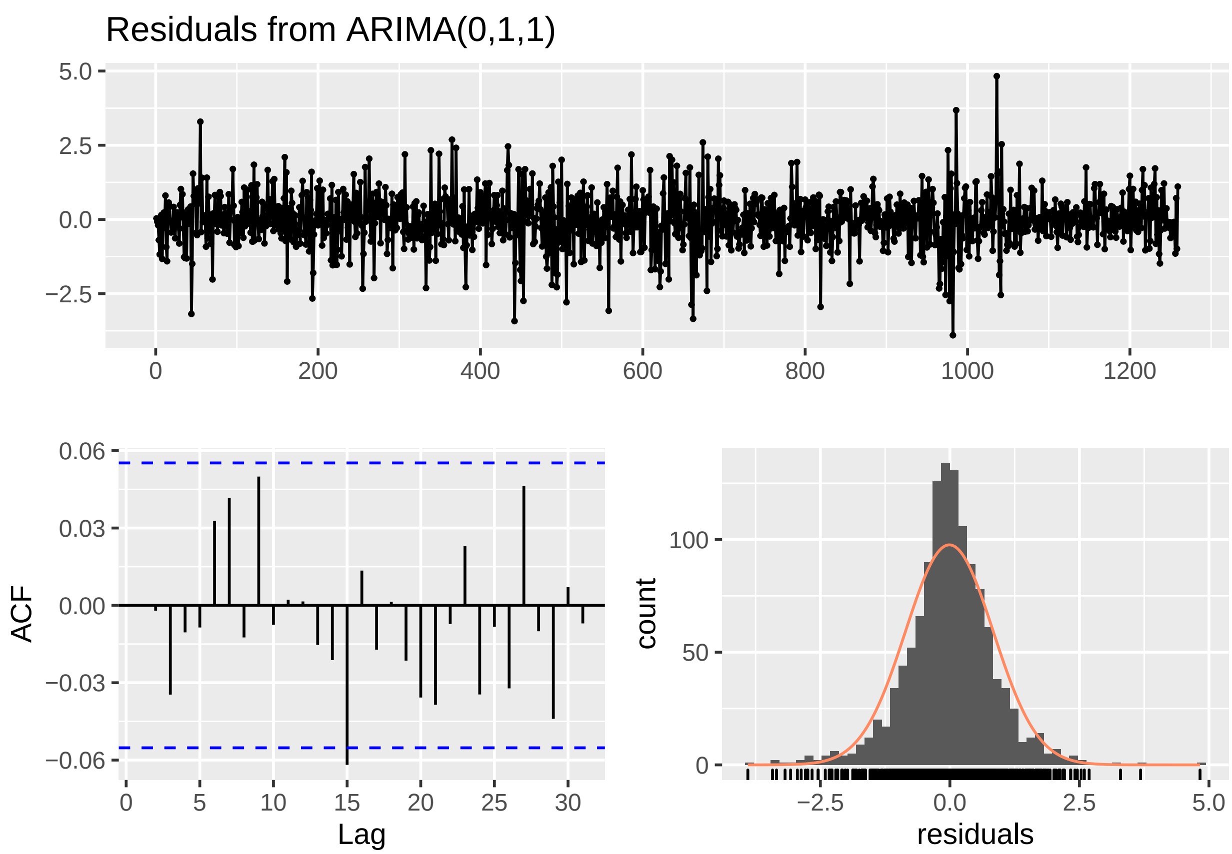
```
## Series: AAL
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.0681
## s.e.  0.0281
##
## sigma^2 estimated as 0.7182:  log likelihood=-1576.33
## AIC=3156.66   AICc=3156.67   BIC=3166.94
##
## Training set error measures:
##                         ME      RMSE      MAE         MPE     MAPE      MASE
## Training set -0.01058389 0.8468002 0.612775 -0.09566096 2.221756 0.9973682
##                         ACF1
## Training set -9.017117e-05
```

```r
#AAL ARIMA
auto.arima(AAL, stepwise=FALSE,
           approximation=FALSE)
```

```
## Series: AAL
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
```

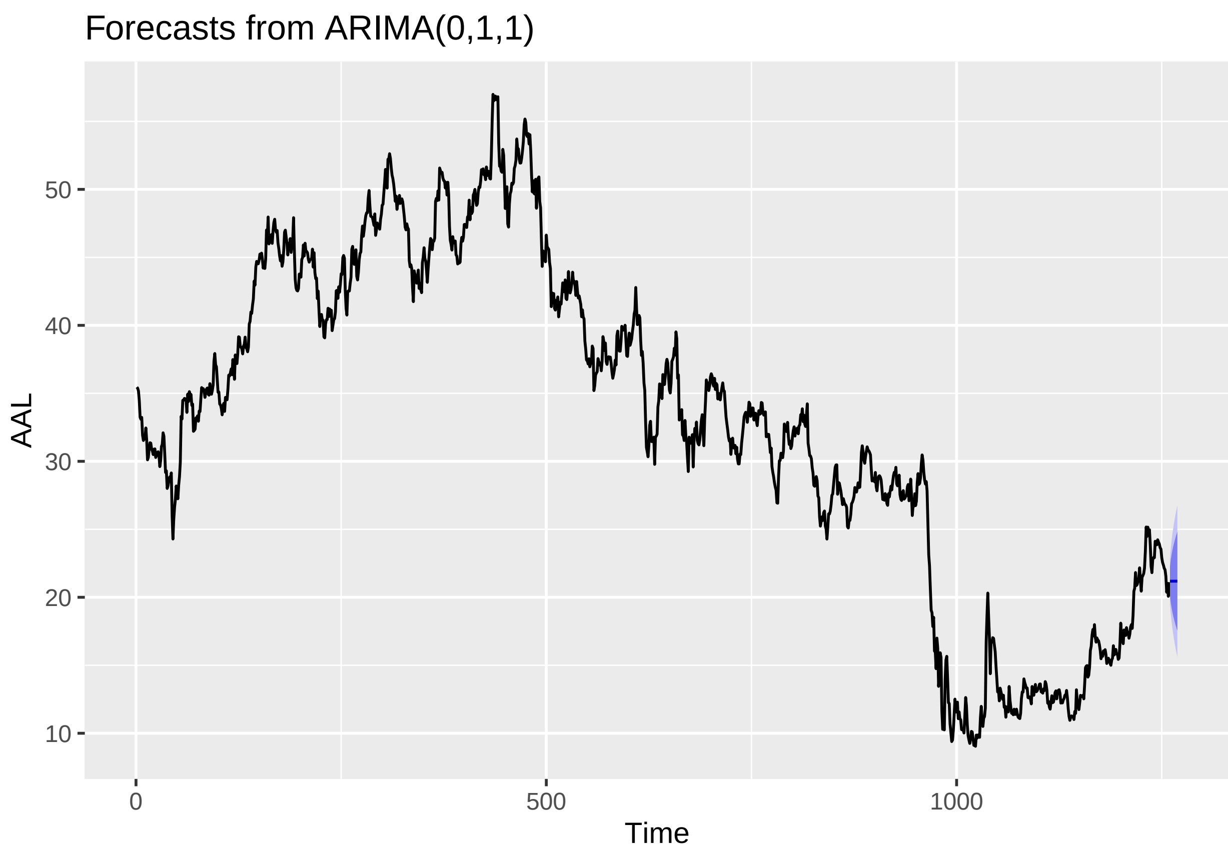
```
##       0.0681
## s.e.  0.0281
##
## sigma^2 estimated as 0.7182:  log likelihood=-1576.33
## AIC=3156.66   AICc=3156.67   BIC=3166.94
```

```r
#AAL ARIMA Forecast

checkresiduals(fit_AAL)
```

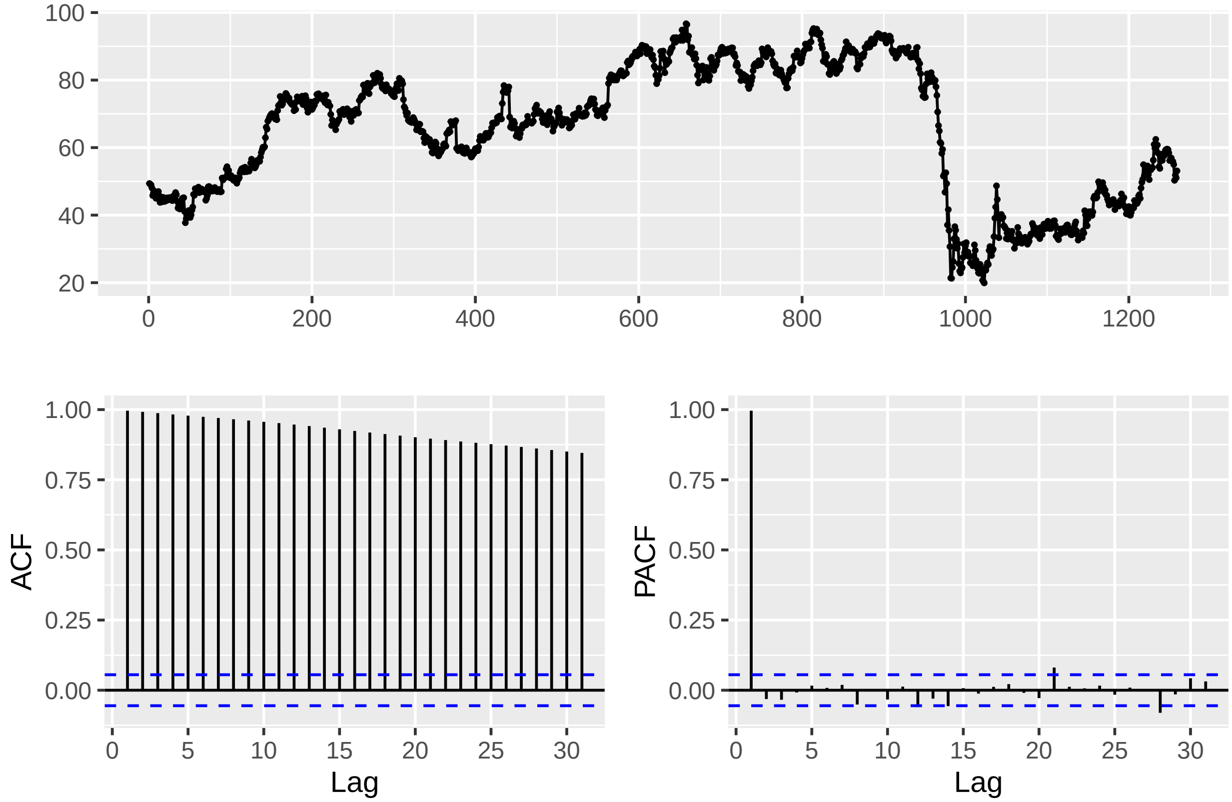



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)
## Q* = 8.7446, df = 9, p-value = 0.4612
##
## Model df: 1.   Total lags used: 10
```

```r
fit_AAL %>% forecast %>% autoplot
```



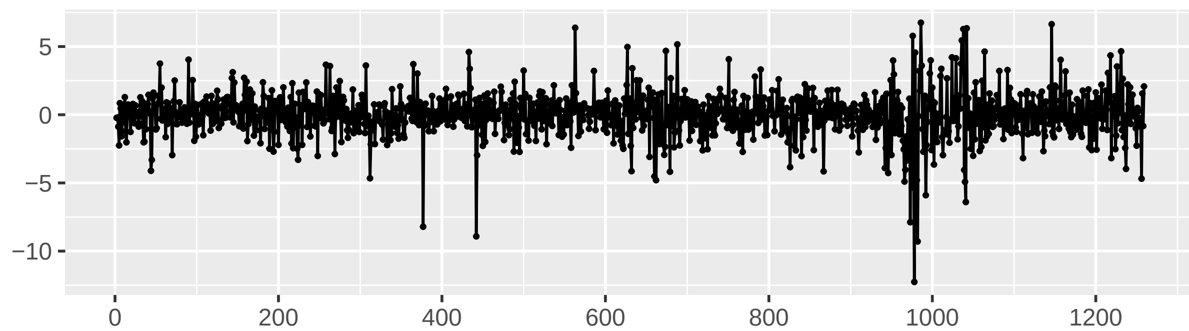

```r
#UAL ARIMA
ggtsdisplay(UAL)
```

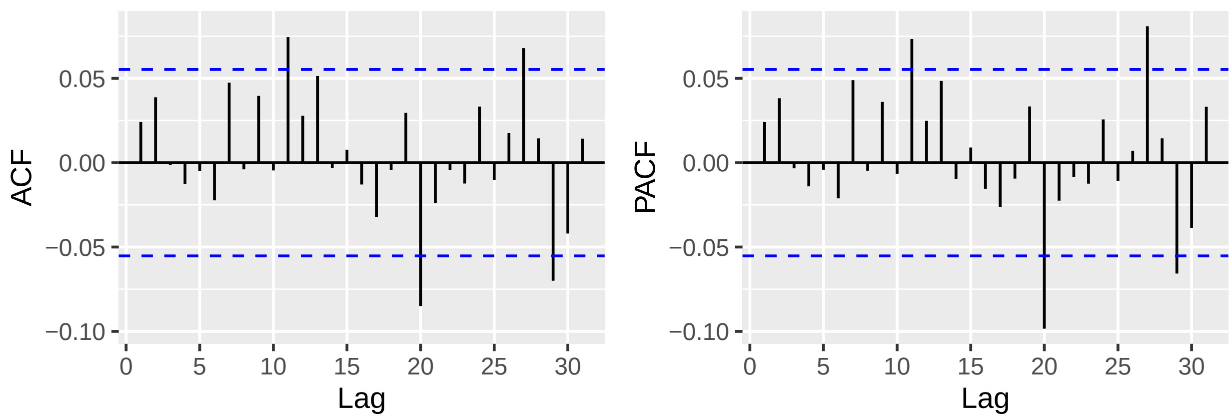



```r
ndiffs(UAL)
```

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

```r
ggtsdisplay(diff(UAL))
```





By looking at the autocorrelation funtion and partial autocorrelation function of the UAL stock, in the first 10 lags we don't see any spike. In this case, p = 0 and q = 0 could be the best choice.

```
#UAL ARIMA
fit_UAL <- Arima(UAL,order=c(0,1,0))
summary(fit_UAL)
```

```
## Series: UAL
## ARIMA(0,1,0)
##
## sigma^2 estimated as 2.636:  log likelihood=-2394.6
## AIC=4791.2   AICc=4791.21   BIC=4796.34
##
## Training set error measures:
##                       ME     RMSE     MAE         MPE     MAPE     MASE
## Training set 0.003009827 1.622817 1.11127 -0.05535287 2.033455 0.999241
##                   ACF1
## Training set 0.0240974
```

The auto.arima functions indicates to use an ARIMA(0,1,0).

```
#UAL ARIMA
fitauto_UAL <- auto.arima(UAL)
summary(fitauto_UAL)
```

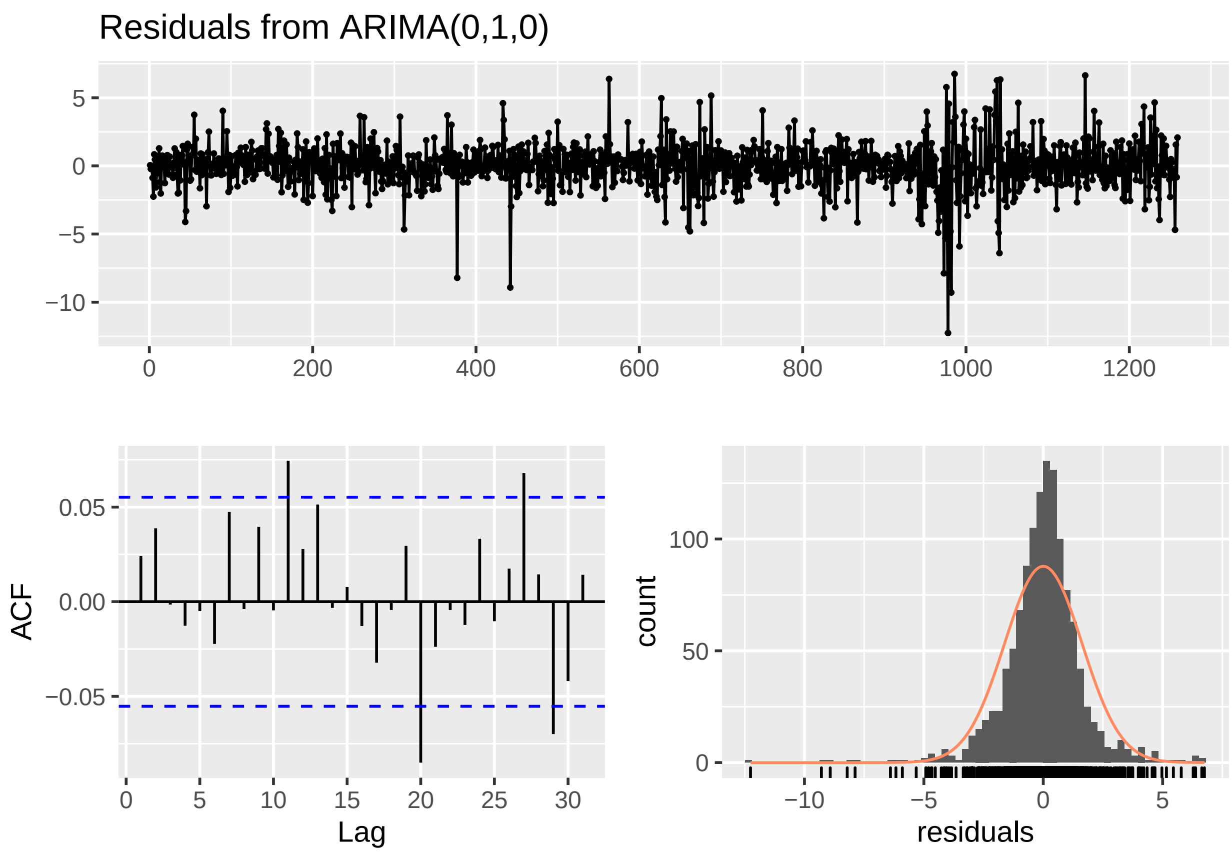
```
## Series: UAL
```

```
## ARIMA(0,1,0)
##
## sigma^2 estimated as 2.636:  log likelihood=-2394.6
## AIC=4791.2   AICc=4791.21   BIC=4796.34
##
## Training set error measures:
##                       ME     RMSE     MAE         MPE     MAPE     MASE
## Training set 0.003009827 1.622817 1.11127 -0.05535287 2.033455 0.999241
##                   ACF1
## Training set 0.0240974
```

```
auto.arima(UAL, stepwise=FALSE,
           approximation=FALSE)
```

```
## Series: UAL
## ARIMA(0,1,0)
##
## sigma^2 estimated as 2.636:  log likelihood=-2394.6
## AIC=4791.2   AICc=4791.21   BIC=4796.34
```

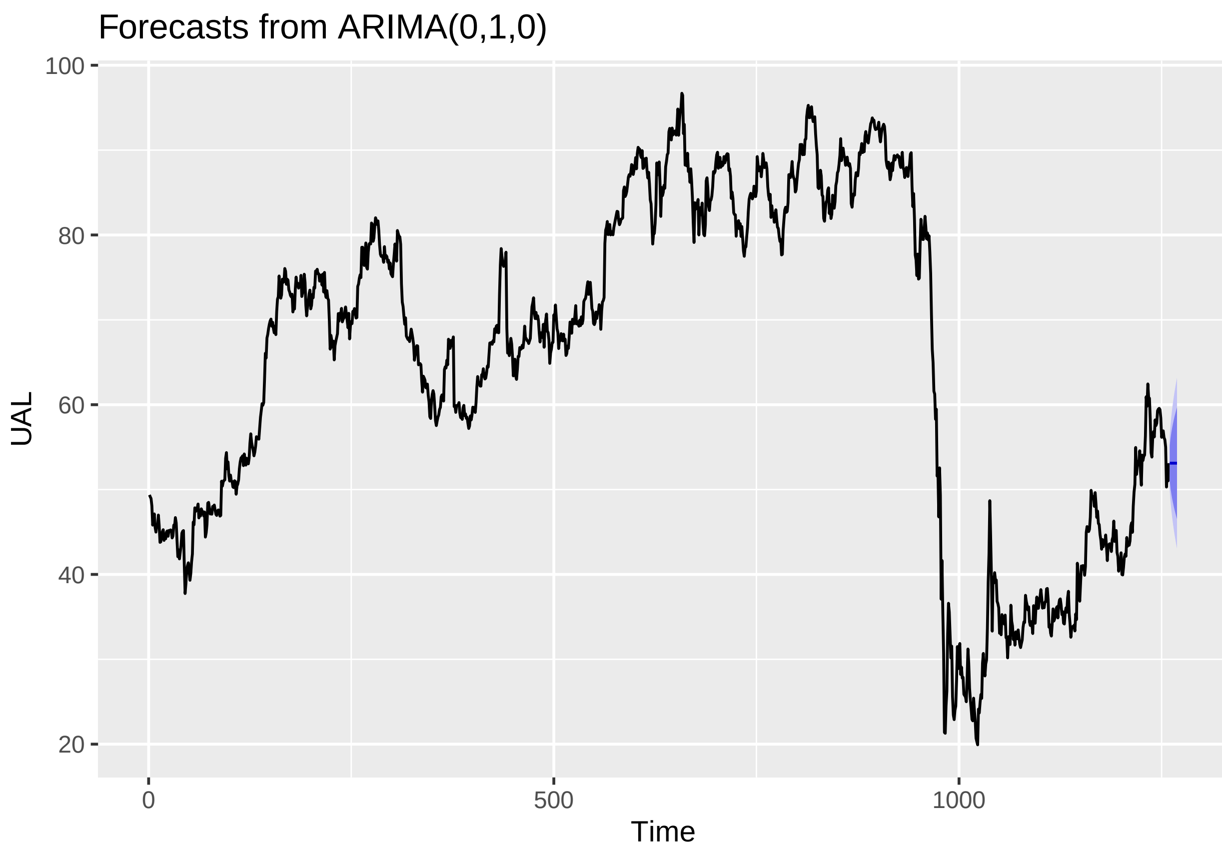
```
checkresiduals(fit_UAL)
```



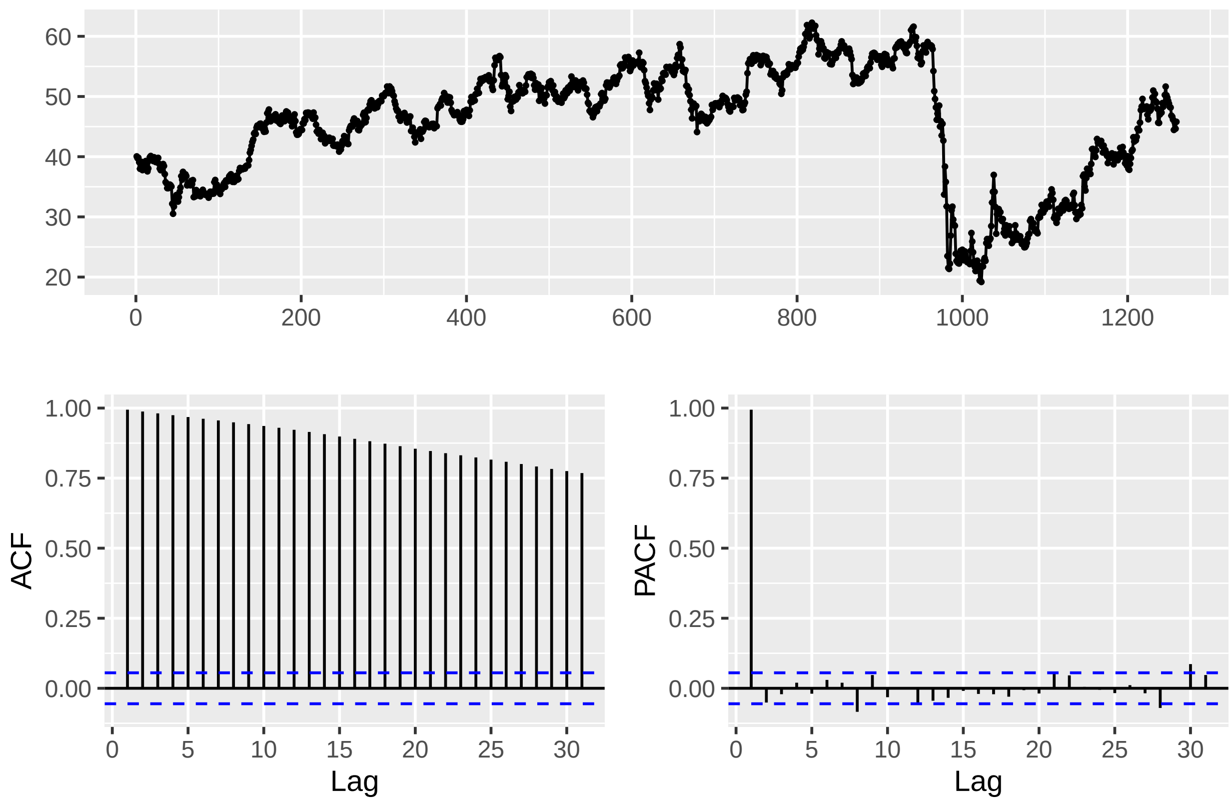

```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 8.3943, df = 10, p-value = 0.5904
##
## Model df: 0.   Total lags used: 10
```

```r
fit_UAL %>% forecast %>% autoplot
```



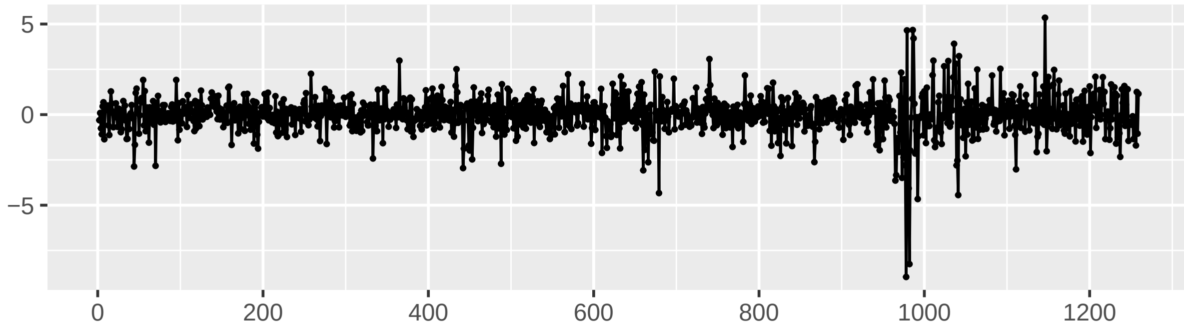

```r
#DAL ARIMA
ggtsdisplay(DAL)
```

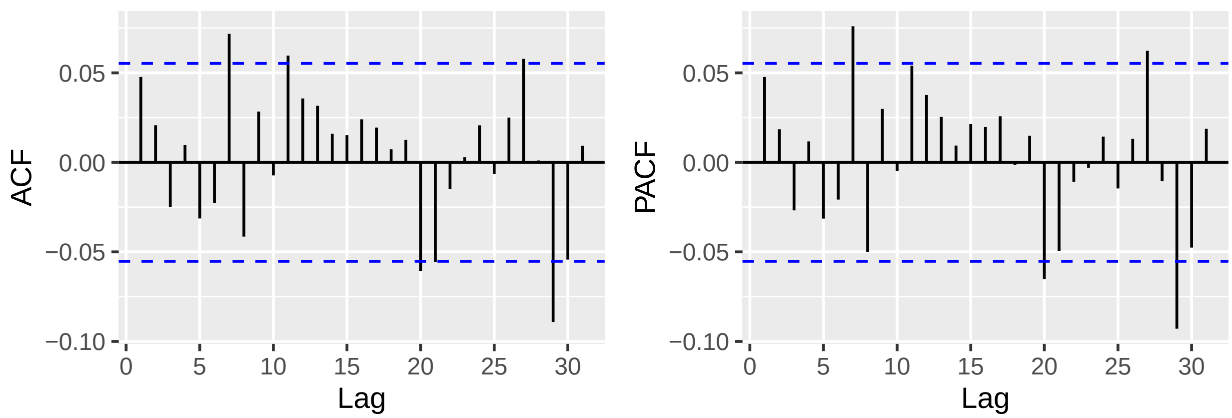



```r
ndiffs(DAL)
```

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

```r
ggtsdisplay(diff(DAL))
```





By looking at the autocorrelation funtion and partial autocorrelation function of the DAL stock, in the first 6 lags we don't see any spike. In this case, p = 0 and q = 0 could be the best choice.

```r
fit_DAL <- Arima(DAL,order=c(0,1,1))
summary(fit_DAL)
```

```
## Series: DAL
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.0457
## s.e.  0.0275
##
## sigma^2 estimated as 1.032:  log likelihood=-1804.34
## AIC=3612.68   AICc=3612.69   BIC=3622.95
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE    MAPE      MASE
## Training set 0.004440247 1.015068 0.6854359 -0.02620381 1.66315 0.9999237
##                     ACF1
## Training set 0.0010639
```

```r
fit_DAL1 <- Arima(DAL,order=c(1,1,0))
summary(fit_DAL1)
```

```
## Series: DAL
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.0477
## s.e.  0.0282
##
## sigma^2 estimated as 1.032:  log likelihood=-1804.28
## AIC=3612.56   AICc=3612.57   BIC=3622.84
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE     MAPE      MASE
## Training set 0.004424091 1.015021 0.6854486 -0.02604784 1.663048 0.9999422
##                       ACF1
## Training set -0.0008699448
```

The auto.arima functions indicates to use an ARIMA(0,1,0).

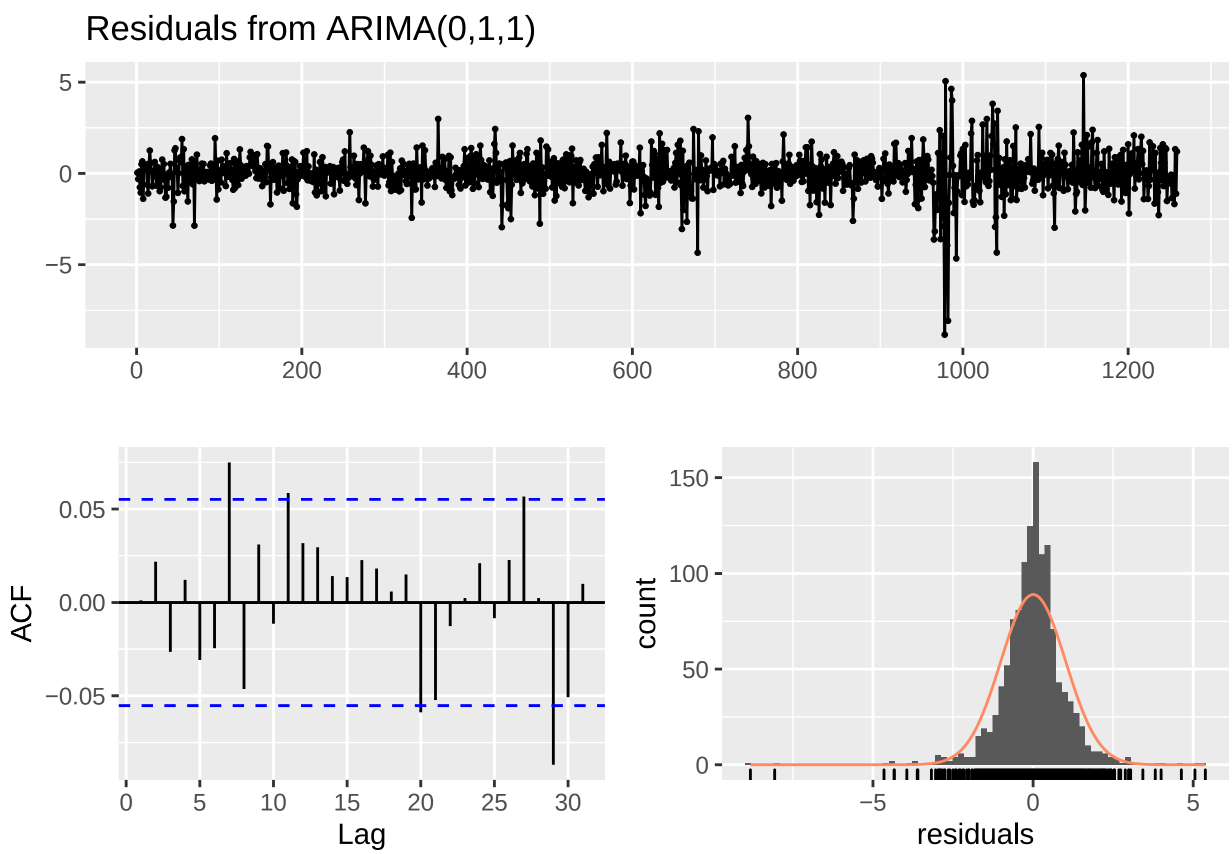
```
fitauto_DAL <- auto.arima(DAL)
summary(fitauto_DAL)
```

```
## Series: DAL
## ARIMA(0,1,0)
##
## sigma^2 estimated as 1.033:  log likelihood=-1805.71
## AIC=3613.43   AICc=3613.43   BIC=3618.56
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set 0.00459805 1.016179 0.6849756 -0.0278958 1.663635 0.9992521
##                    ACF1
## Training set 0.04766066
```

```
auto.arima(DAL, stepwise=FALSE,
           approximation=FALSE)
```

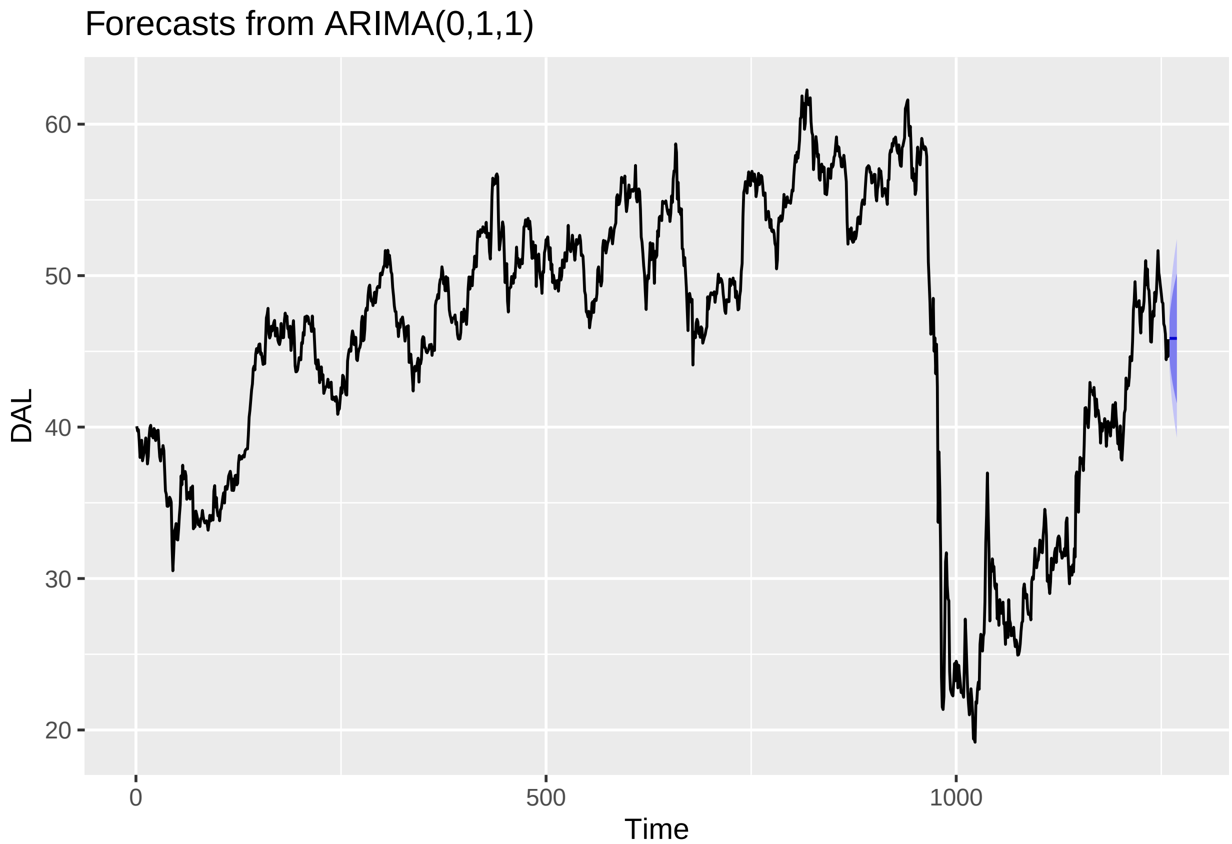
```
## Series: DAL
## ARIMA(2,1,2)
##
## Coefficients:
##           ar1      ar2     ma1     ma2
##       -1.4286  -0.8170  1.4522  0.8643
## s.e.   0.1129   0.0993  0.0987  0.0844
##
## sigma^2 estimated as 1.028:  log likelihood=-1800.64
## AIC=3611.29   AICc=3611.33   BIC=3636.97
```

```
checkresiduals(fit_DAL)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)
## Q* = 14.86, df = 9, p-value = 0.09484
##
## Model df: 1.   Total lags used: 10
```

```
fit_DAL %>% forecast %>% autoplot
```



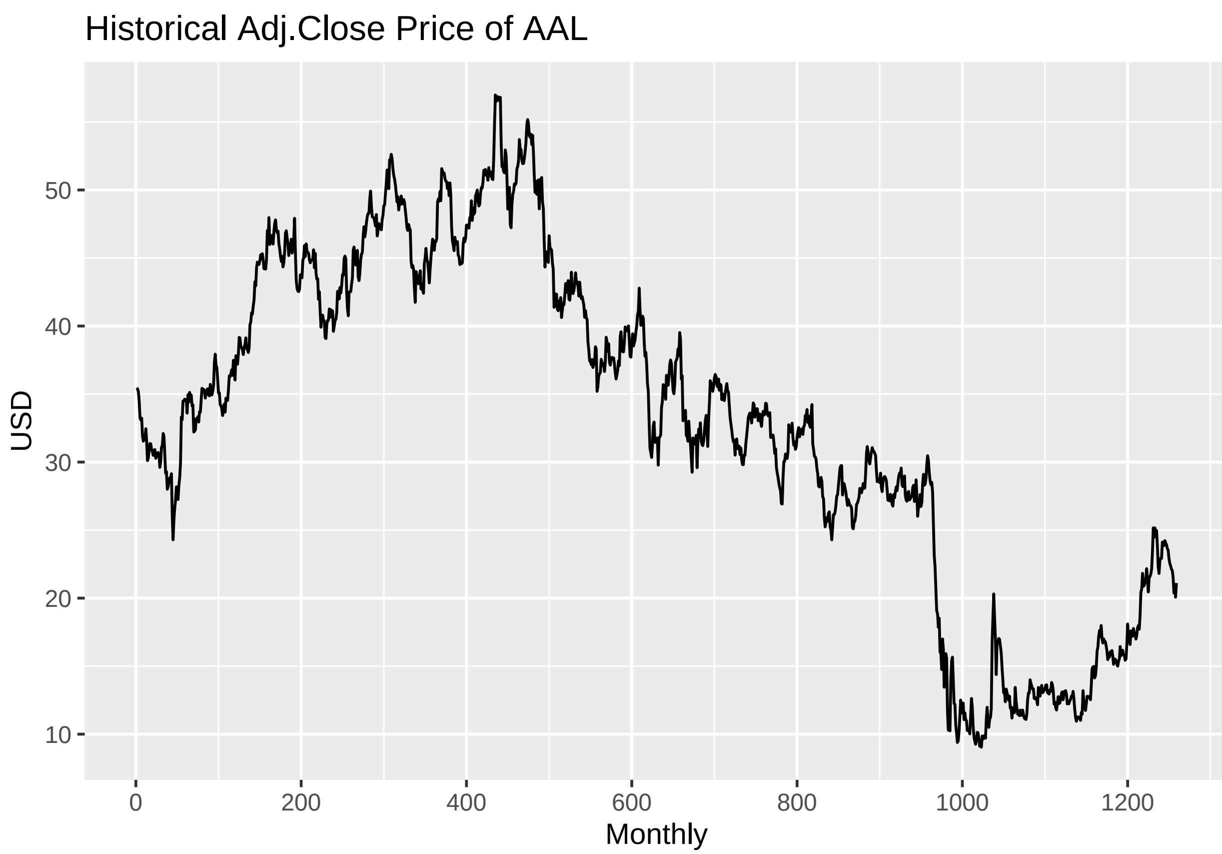
Now we try the 5 years' monthly closing data of these 3 airlines:

```
airline.month <- read.csv(file = 'C:/Users/Steve/Documents/Santa Clara University Classes/Spring 2021/T
head(airline.month)
```

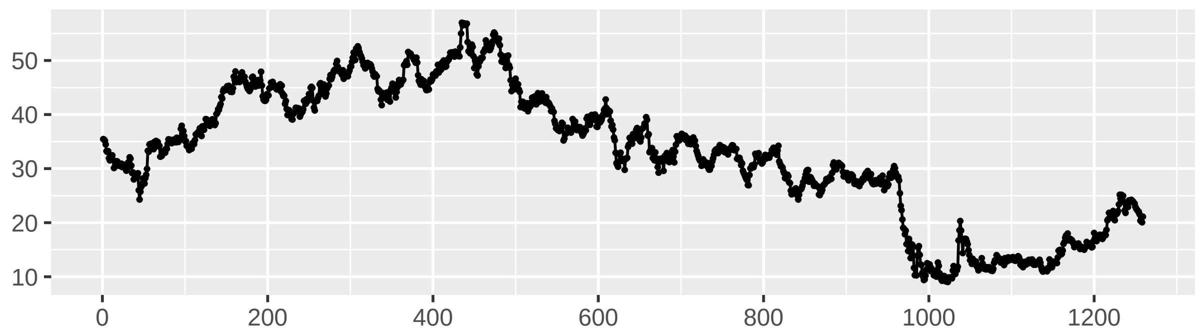
```
##        Date      AAL   UAL      DAL
## 1 2016/4/25 35.45536 49.37 40.05110
## 2 2016/4/26 35.37871 49.14 39.74113
## 3 2016/4/27 35.17748 48.94 39.83230
## 4 2016/4/28 34.46836 48.06 39.07558
## 5 2016/4/29 33.24179 45.81 37.99065
## 6  2016/5/2 33.08803 46.66 38.44650
```

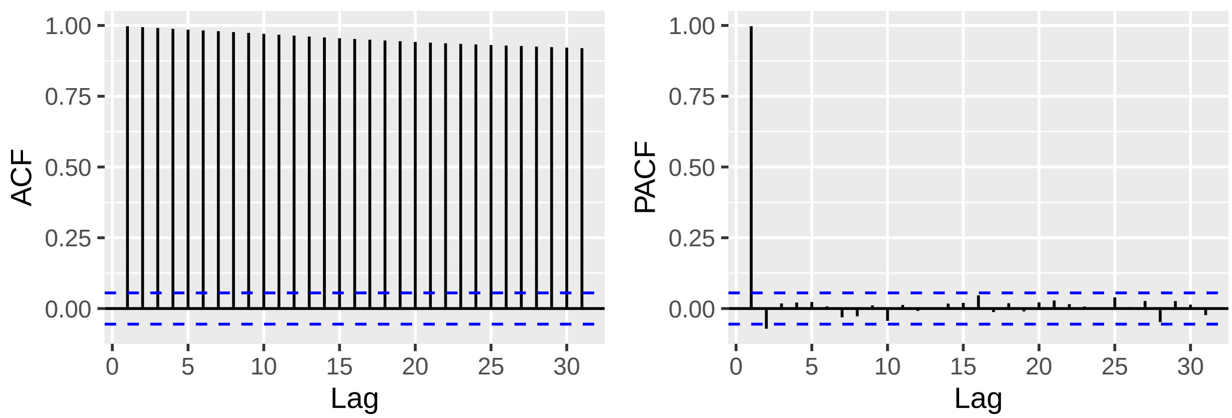
The plot shows that there's no trend, non-stationary, non-seasonal. Similiar to daily results.

```
# Monthly data, 5 years,

airline.month <- ts(airline.month)

AAL.m <- airline.month[,2]

autoplot(AAL.m) +
  ggtitle("Historical Adj.Close Price of AAL")+
  xlab("Monthly")+ylab("USD")
```




```r
ggtsdisplay(AAL.m)
```





Since P/E ratio (https://www.investopedia.com/terms/p/price-earningsratio.asp) is an important indicator of stock price.

We create dynamic regression model to find correlation to quarterly price with EPS.

Basic EPS: how much of a firm's net income was allotted to each share of common stock. From AAL 10K and 10Q reports.

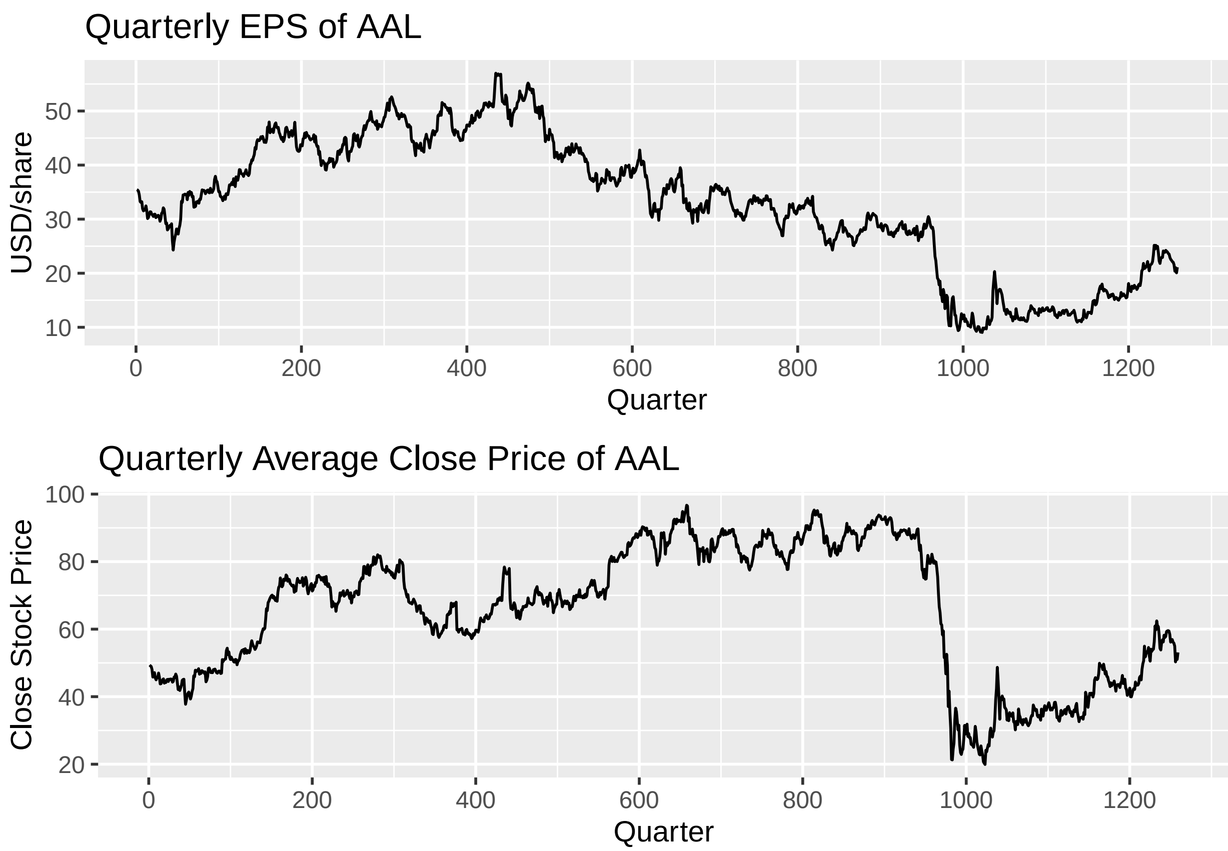
Quarterly average price: calculated base on monthly data from Yahoo Finance. (Q1=Jan,Feb,Mar….)

```
# Dynamic regression models, based on quarterly price & EPS

airline.eps <- read.csv(file = 'C:/Users/Steve/Documents/Santa Clara University Classes/Spring 2021/Tim

head(airline.eps)
```

```
##        Date      AAL   UAL      DAL
## 1 2016/4/25 35.45536 49.37 40.05110
## 2 2016/4/26 35.37871 49.14 39.74113
## 3 2016/4/27 35.17748 48.94 39.83230
## 4 2016/4/28 34.46836 48.06 39.07558
## 5 2016/4/29 33.24179 45.81 37.99065
## 6  2016/5/2 33.08803 46.66 38.44650
```

```
airline.eps <- ts(airline.eps)
```

```r
AAL.eps <- airline.eps[,2]
p.eps <- autoplot(AAL.eps)+
  ggtitle("Quarterly EPS of AAL")+
  xlab("Quarter")+ylab("USD/share")

AAL.q <- airline.eps[,3]
p.q <- autoplot(AAL.q) +
  ggtitle("Quarterly Average Close Price of AAL")+
  xlab("Quarter")+ylab("Close Stock Price")

gridExtra::grid.arrange(p.eps,p.q, nrow=2)
```



From the ACF plot, we can see that there's no evidence of serial correlation. And the Ljung-Box test stats' p-value is 0.09 suggesting that there's no autocorrelation remaining in the residual.

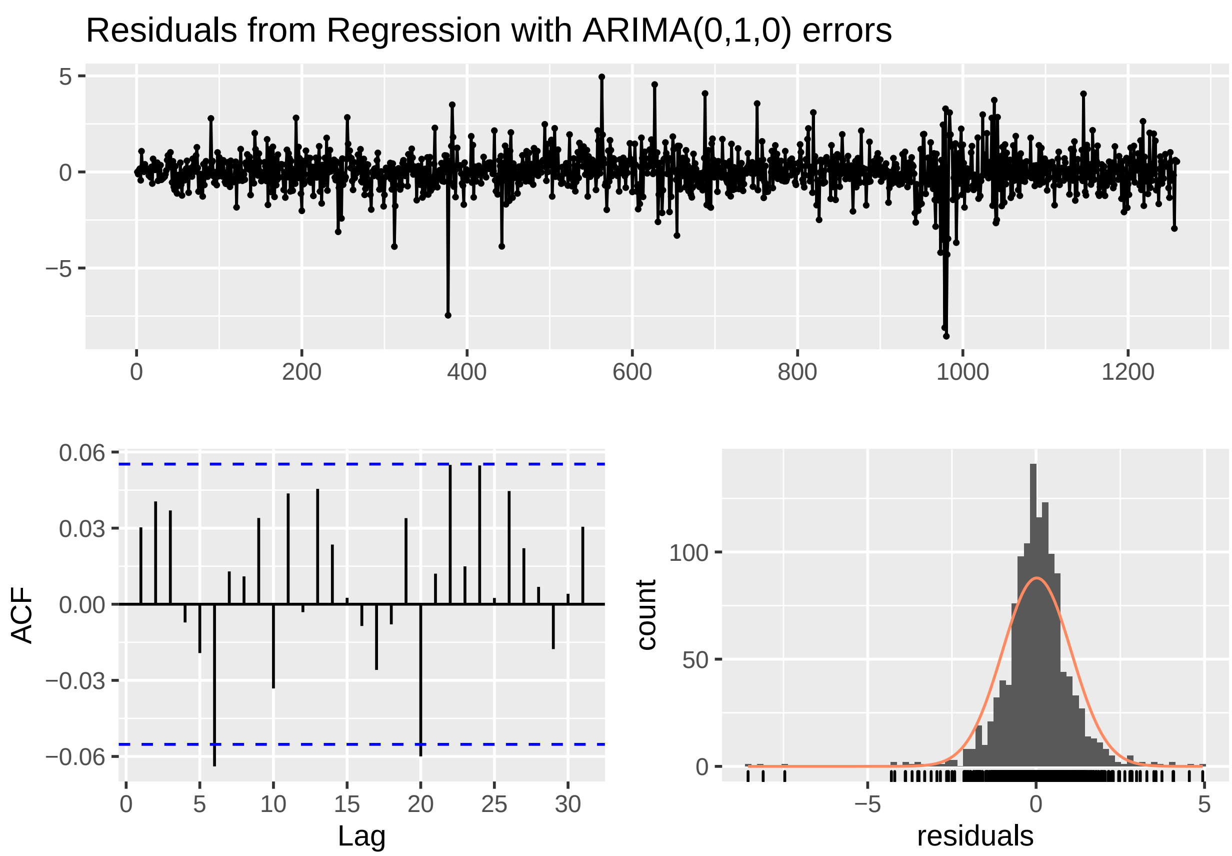
Using the ARIMA model (1, 0 0), we make the forecast for the next 20 quarters is like this - within the range of USD 20-30 in stock price.

```r
# create regression
fit.q <- auto.arima(AAL.q, xreg=AAL.eps)
fit.q
```

```
## Series: AAL.q
## Regression with ARIMA(0,1,0) errors
##
## Coefficients:
```

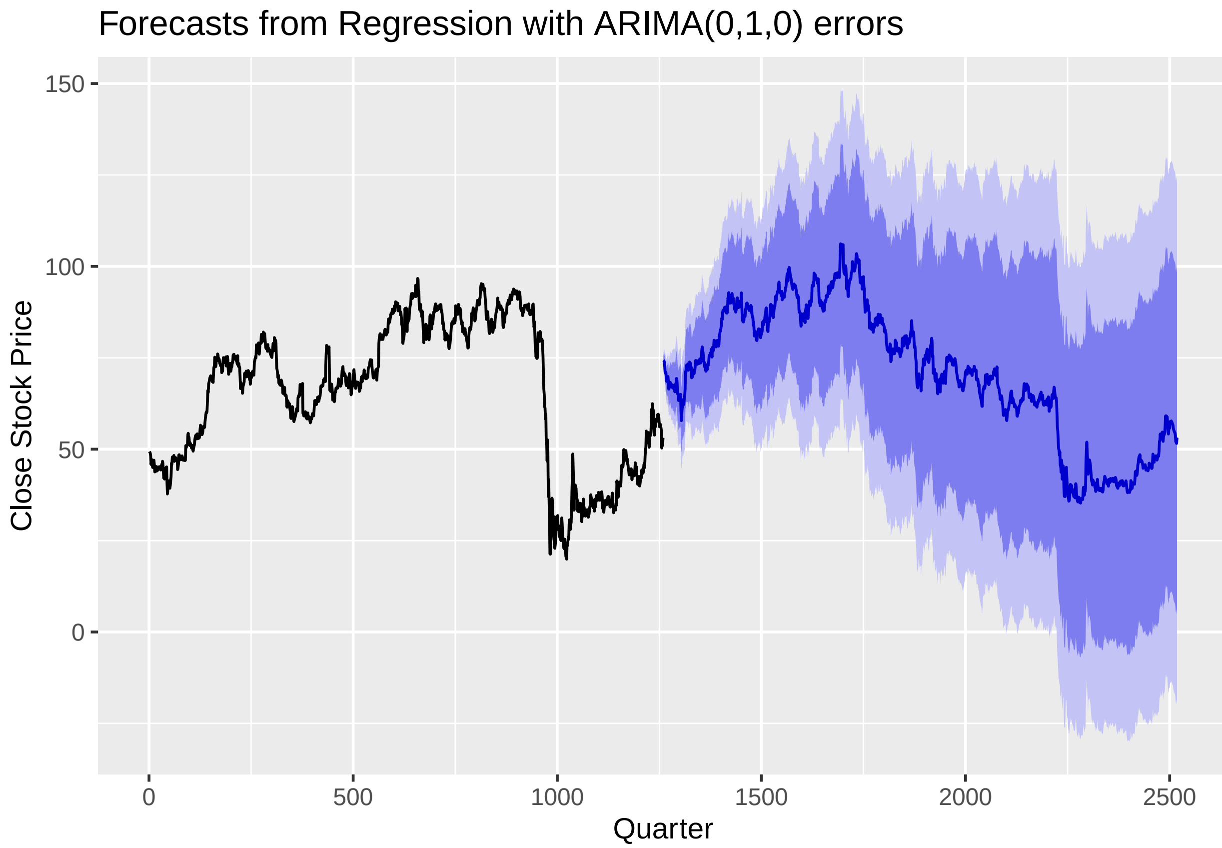
```
##         xreg
##       1.4791
## s.e.  0.0342
##
## sigma^2 estimated as 1.059:  log likelihood=-1820.7
## AIC=3645.41   AICc=3645.42   BIC=3655.68
```

```
checkresiduals(fit.q)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,0) errors
## Q* = 13.902, df = 9, p-value = 0.1258
##
## Model df: 1.   Total lags used: 10
```

```
# forecast
fc <- forecast(fit.q, xreg=AAL.eps)
autoplot(fc) + xlab("Quarter") + ylab("Close Stock Price")
```



We use the generic “predict” function to get our forecast. The column yhat contains our the forecast result. The shaded blue area shows the uncertainty intervals with seasonal components.