Time-series Analysis

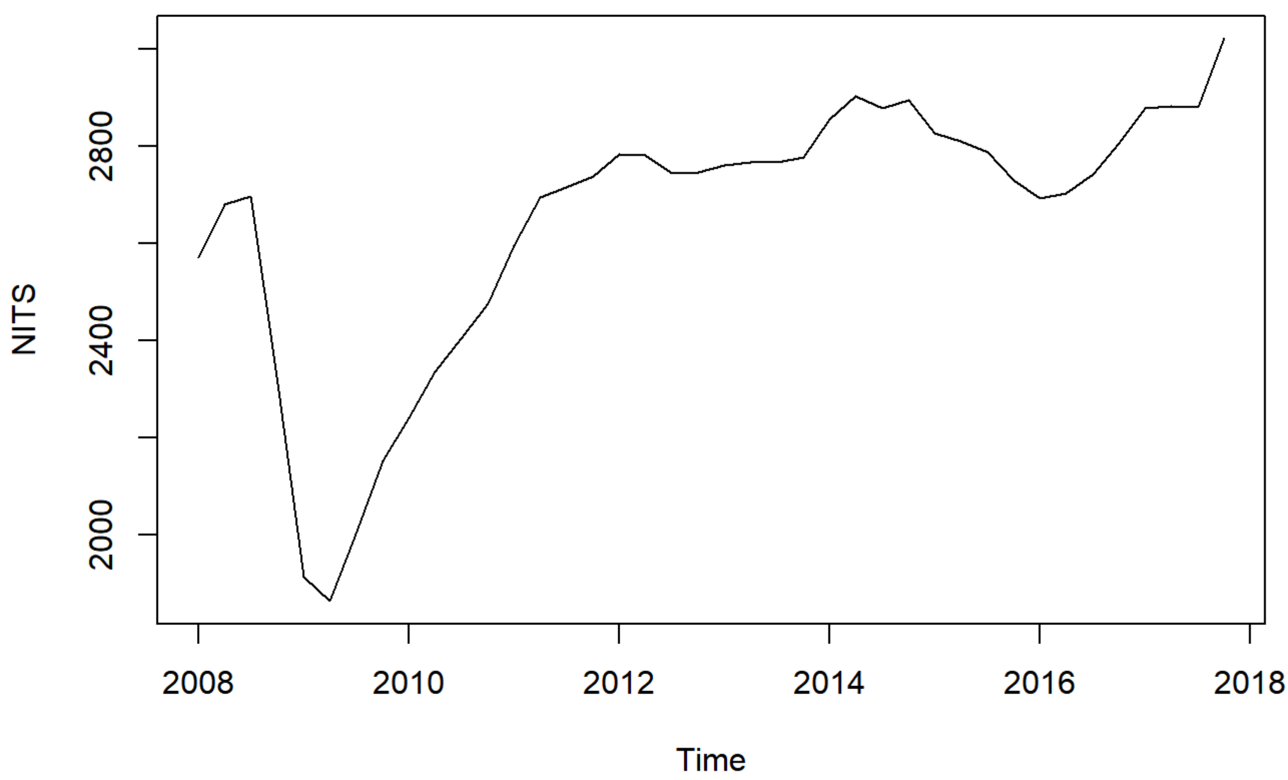
```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.4.4
```

```
Data_Spring_2018_NetImports <- read_csv("D:/Notes/Sem 2/Business Forecasting/Final/Data_Sprin
g_2018_NetImports.csv")
```

```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   Quarter = col_character(),
##   Imports = col_double()
## )
```

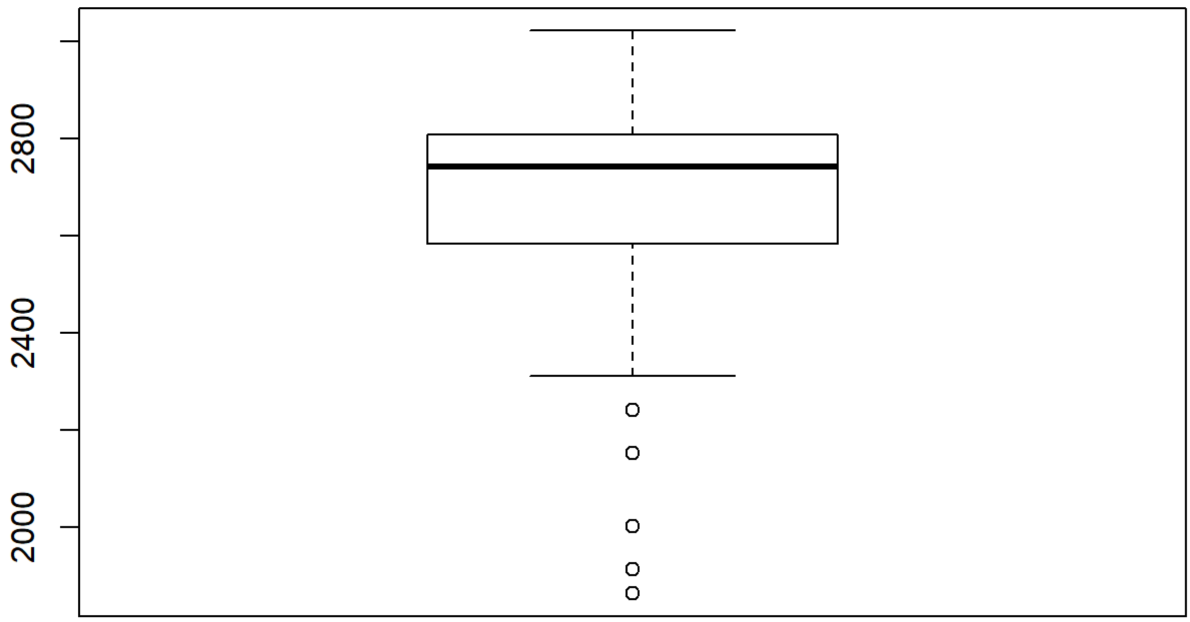
```
netImport <- Data_Spring_2018_NetImports
NITS <- ts(netImport$Imports,start=c(2008,01),frequency = 4)
plot(NITS)
```




```
summary(NITS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1864    2590    2743    2646    2807    3022
```

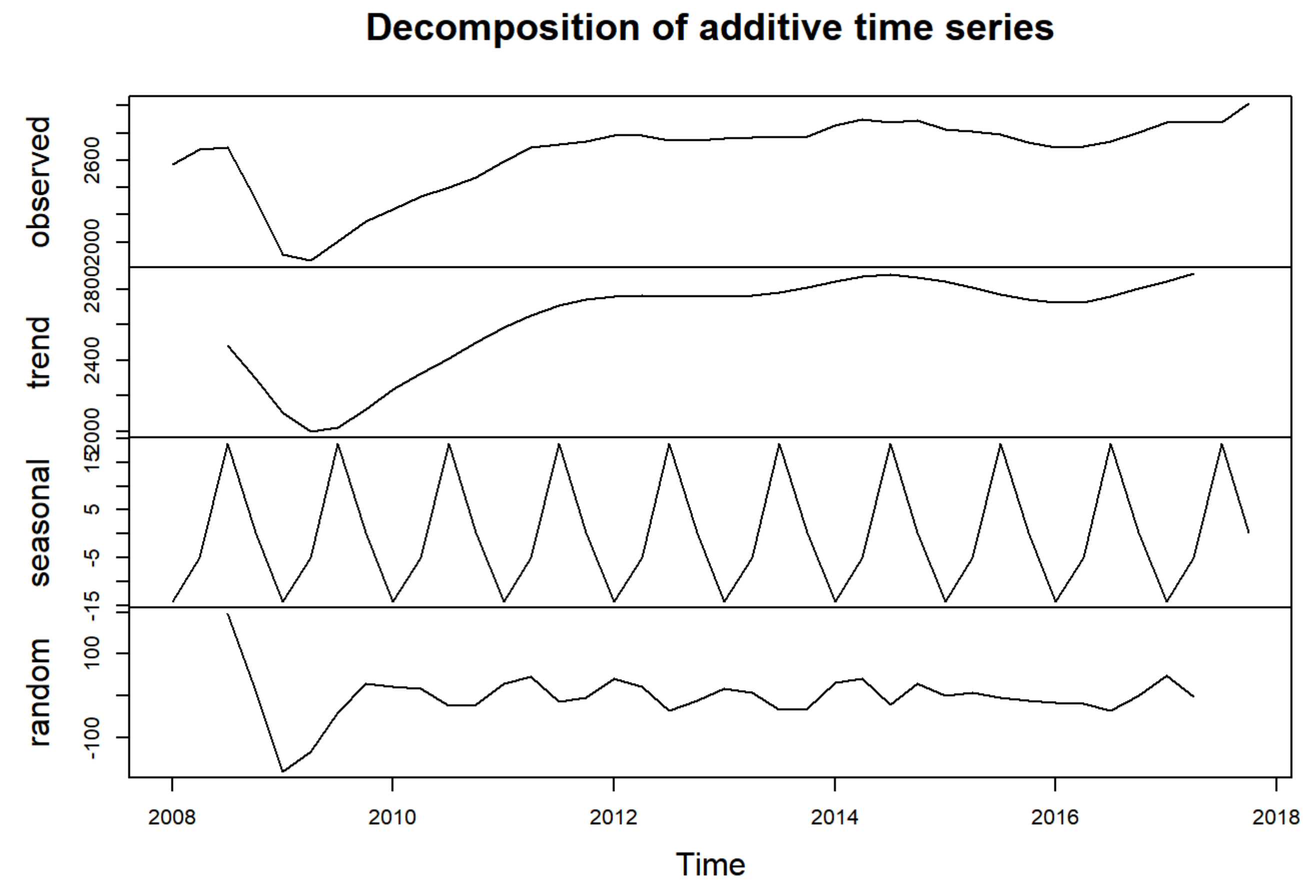
```
boxplot(NITS)
```



From the Boxplot there are 5 outliers. The data is skewed and there exists a spread between the 1st and the 3rd Quartile

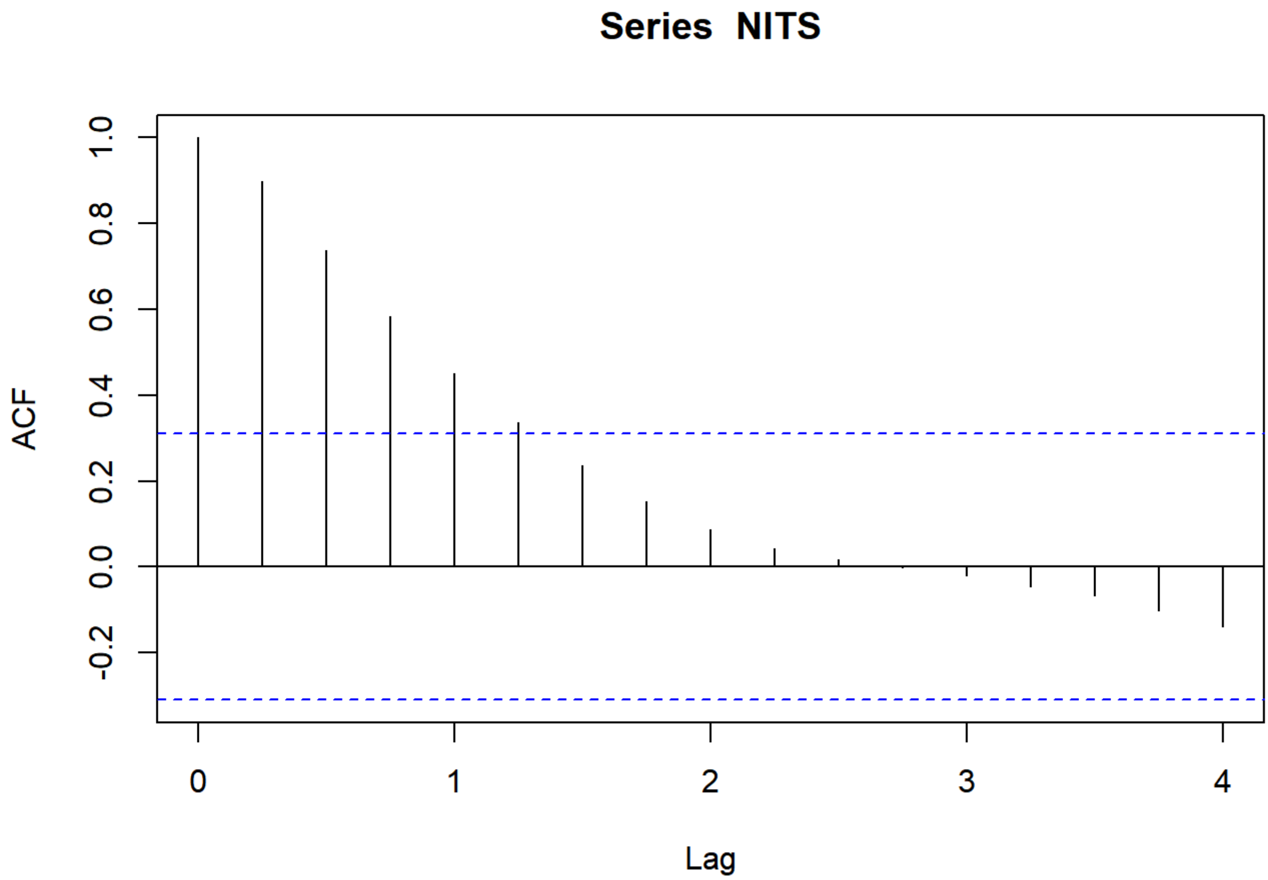
Decomposition

```
decomp <- decompose(NITS)
plot(decomp)
```



There is a significant trend compared to the seasonality in the time-series.

```
acf(NITS)
```



```
decomp$type
```

```
## [1] "additive"
```

```
decomp$seasonal
```

```
##            Qtr1        Qtr2        Qtr3        Qtr4
## 2008 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2009 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2010 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2011 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2012 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2013 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2014 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2015 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2016 -14.1503472  -4.9961806  18.9038194   0.2427083
## 2017 -14.1503472  -4.9961806  18.9038194   0.2427083
```

```
decomp$trend
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2008       NA       NA 2482.775 2298.600
## 2009 2109.775 2003.038 2024.263 2124.425
## 2010 2233.887 2324.625 2409.238 2498.225
## 2011 2581.775 2653.475 2709.938 2744.175
## 2012 2758.500 2762.987 2761.000 2756.563
## 2013 2757.762 2764.588 2780.438 2809.287
## 2014 2840.188 2868.613 2879.513 2864.263
## 2015 2841.312 2809.475 2772.162 2741.900
## 2016 2722.688 2726.375 2759.025 2804.562
## 2017 2844.250 2888.575       NA       NA
```

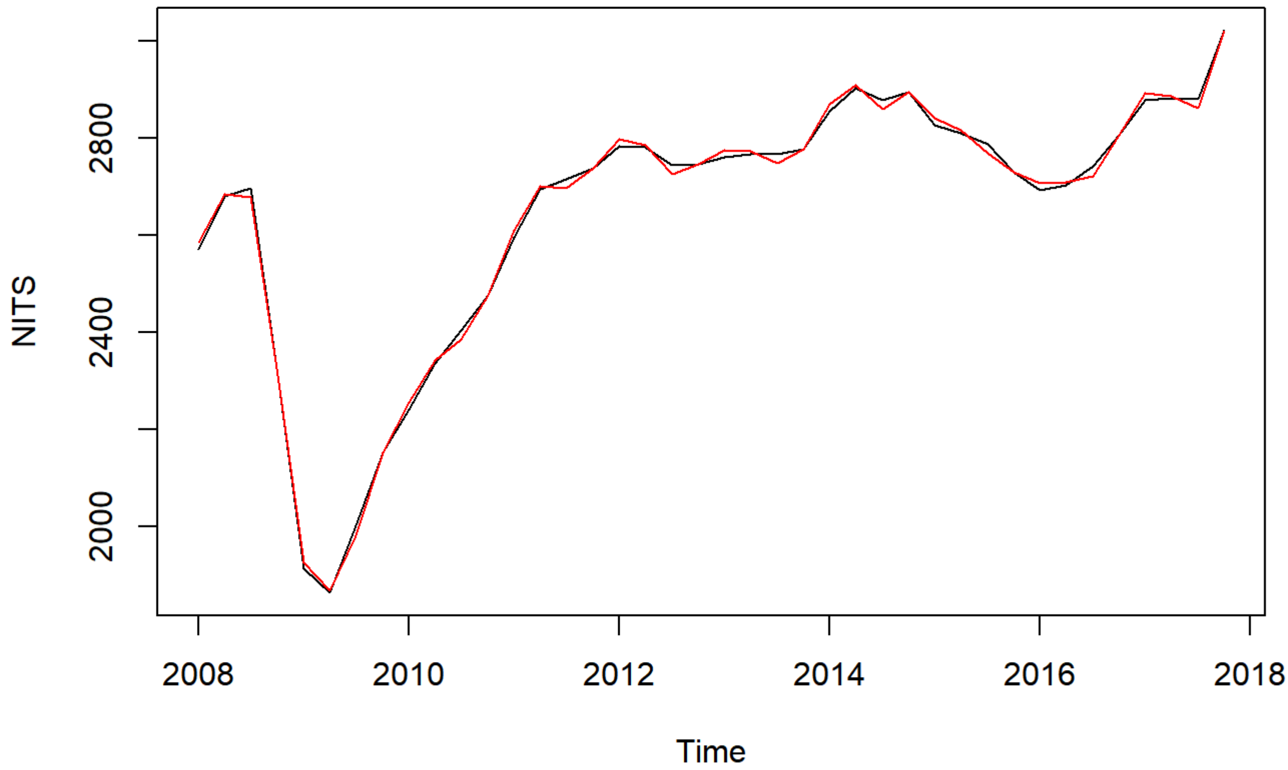
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.4
```

```
temp_seasadj <-seasadj(decomp)
seasadj(decomp)
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2008 2585.550 2684.896 2678.296 2311.357
## 2009 1927.550 1869.496 1983.096 2152.657
## 2010 2256.050 2342.296 2385.996 2475.657
## 2011 2609.950 2700.296 2696.396 2738.857
## 2012 2798.450 2785.696 2725.596 2745.557
## 2013 2775.850 2772.796 2748.096 2777.657
## 2014 2870.550 2908.896 2859.196 2893.957
## 2015 2841.450 2815.996 2768.496 2729.957
## 2016 2706.950 2708.396 2722.396 2805.557
## 2017 2892.550 2887.096 2861.196 3021.357
```

```
plot(NITS)
lines(temp_seasadj, col = 'red')
```

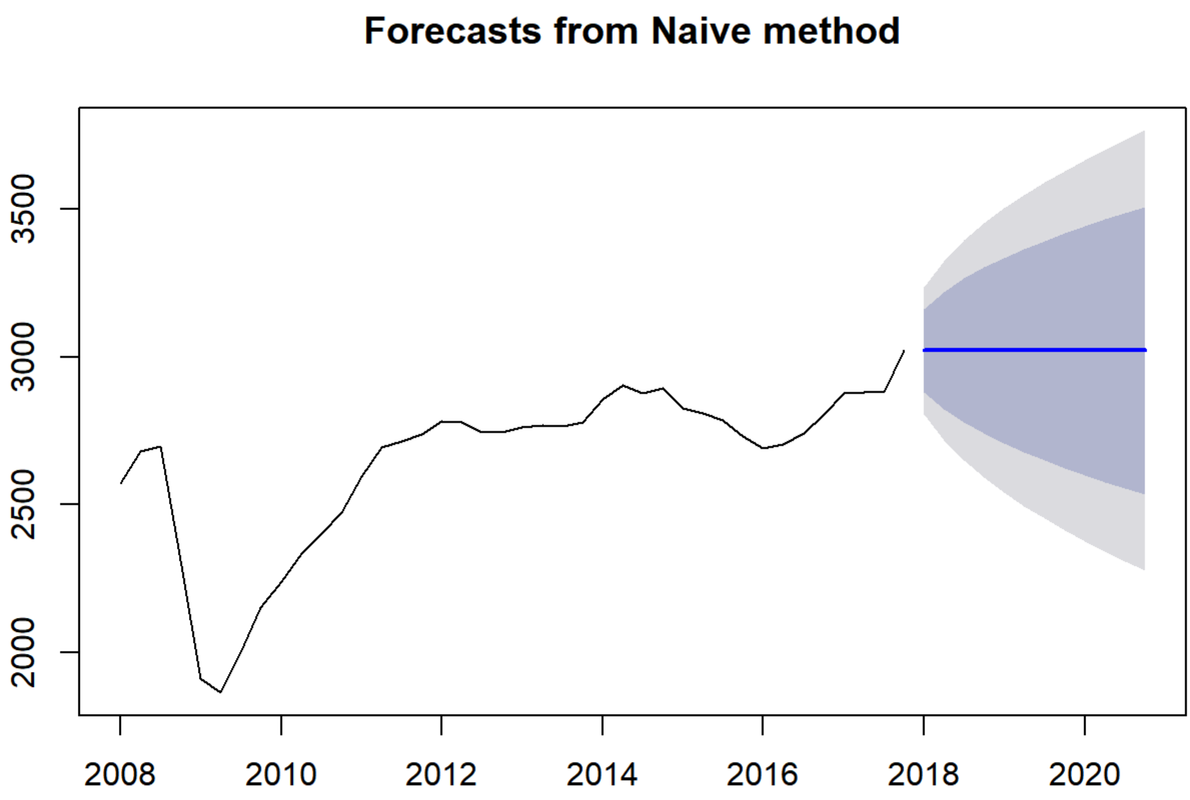


From the seasonal adjustment we see that there is no seasonal fluctuation. This also shows that the data is not seasonal.

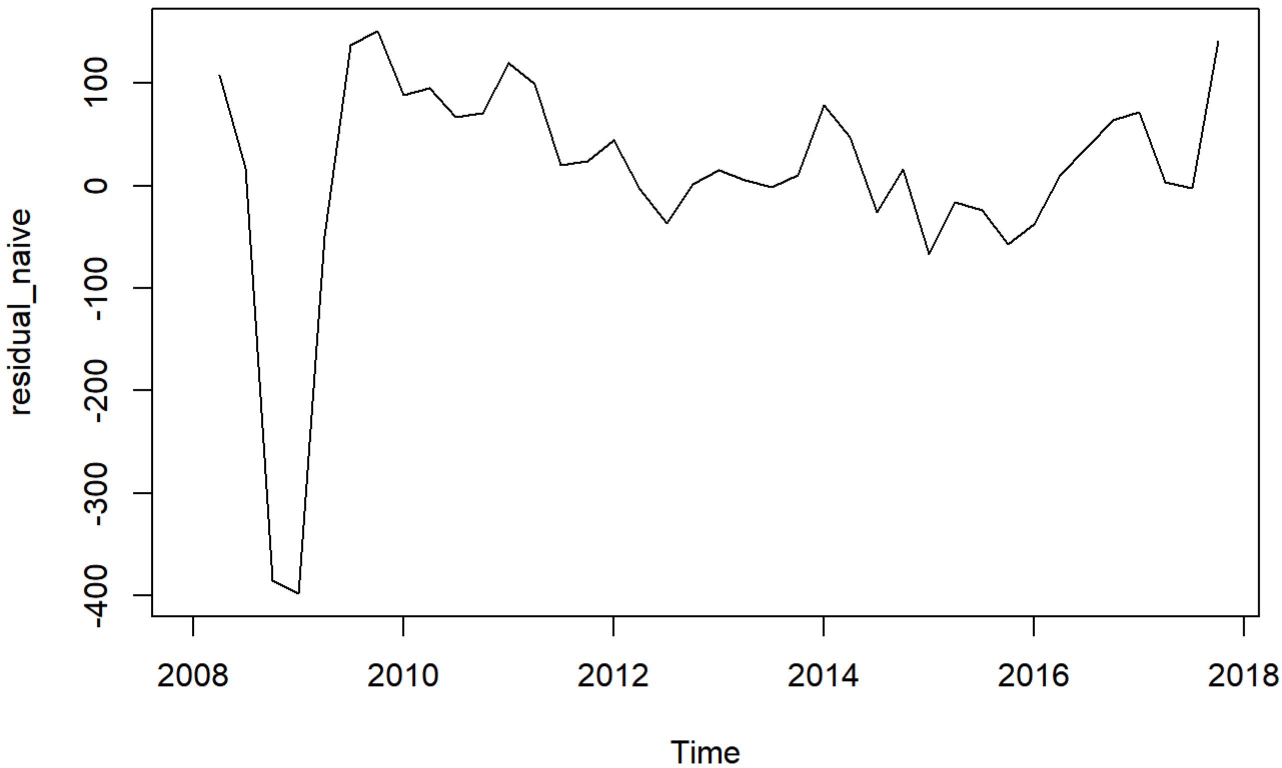
Forecasting US Net Imports

Naive Method

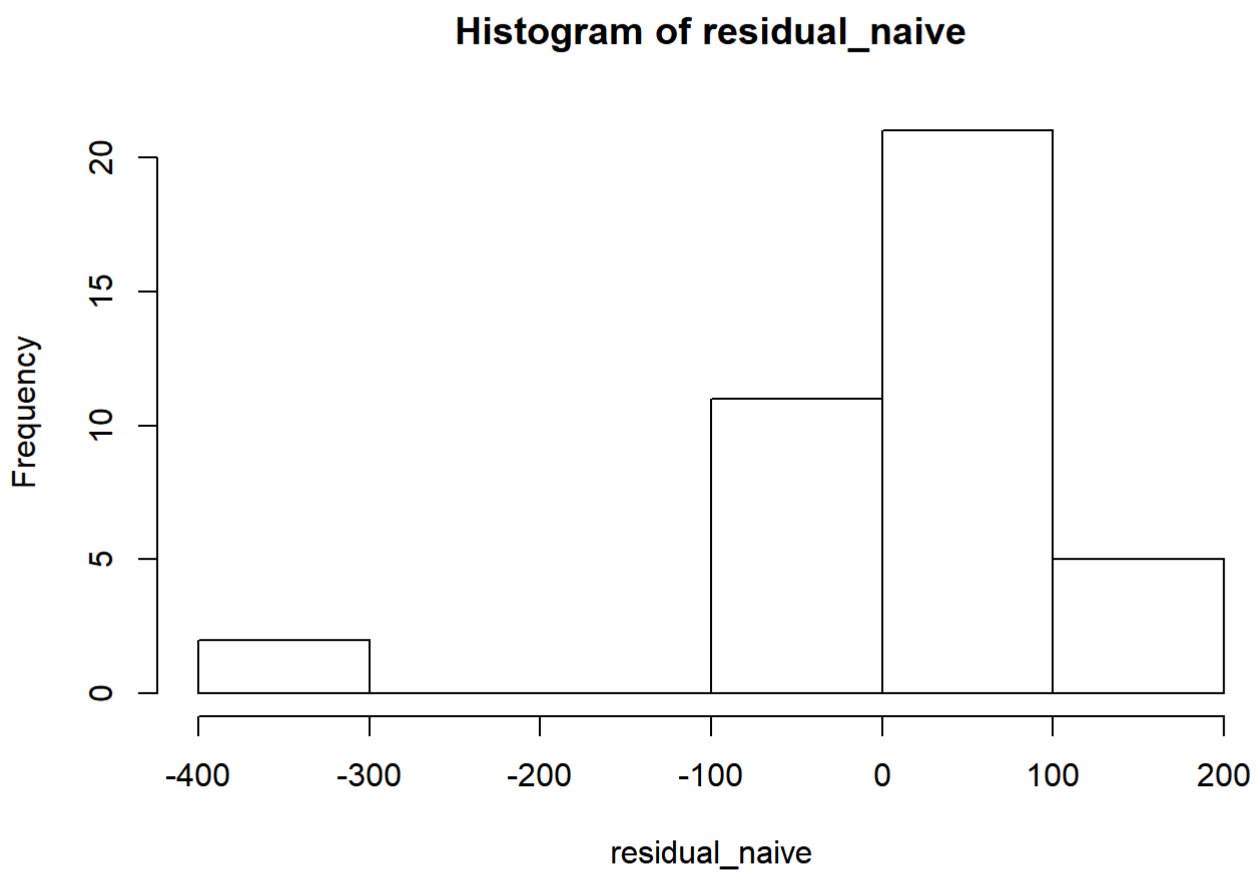
```
naive_forecast <- naive(NITS,12)
plot(naive_forecast)
```



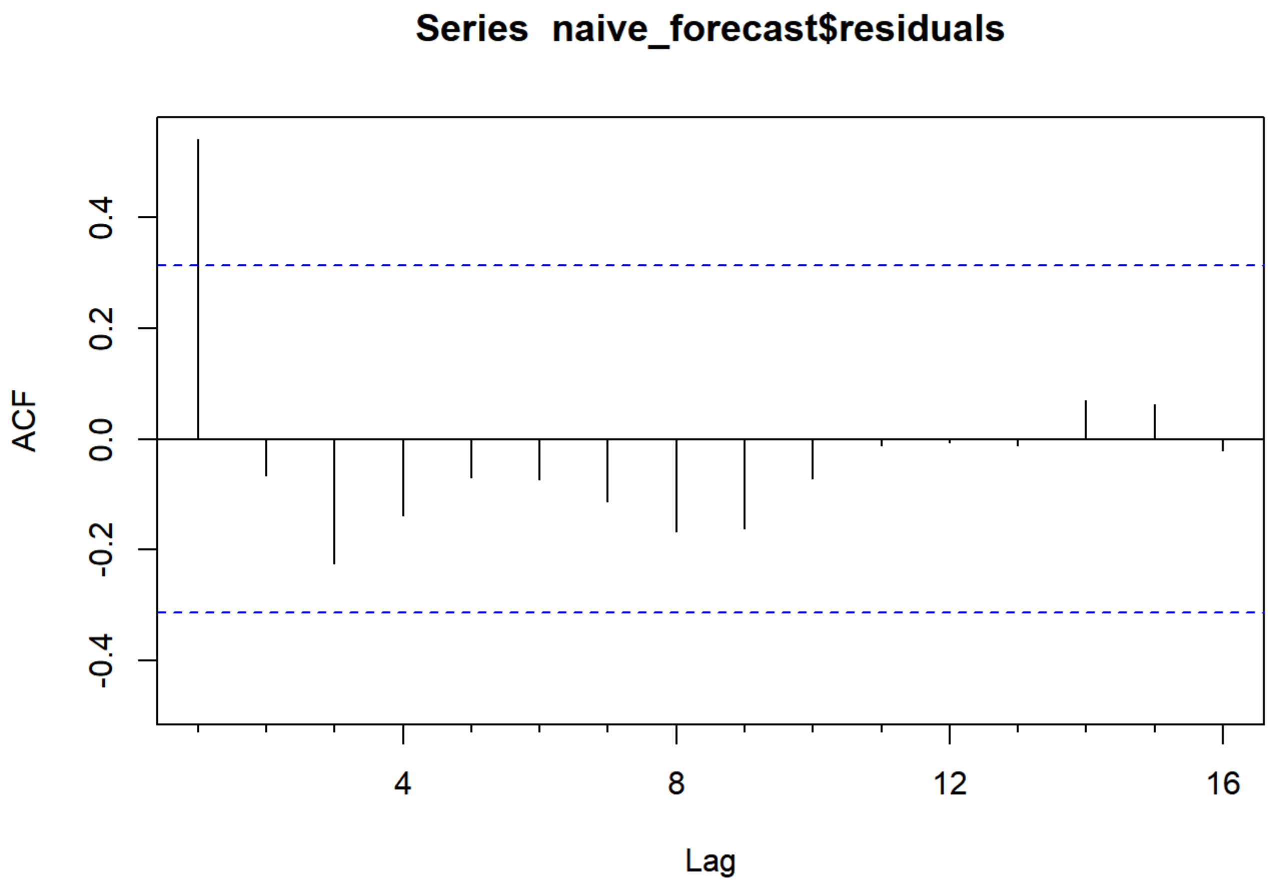

```
residual_naive <- residuals(naive_forecast)
plot(residual_naive)
```



```
hist(residual_naive)
```




```
Acf(naive_forecast$residuals)
```



These graphs show that the naïve method produces forecasts that appear to account for all available information. The mean of the residuals is close to zero and there is a little correlation in the residuals series. The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data and therefore the residual variance can be treated as constant. This can also be seen on the histogram of the residuals. The histogram suggests that the residuals are not be normal.

```
accuracy(naive_forecast)
```

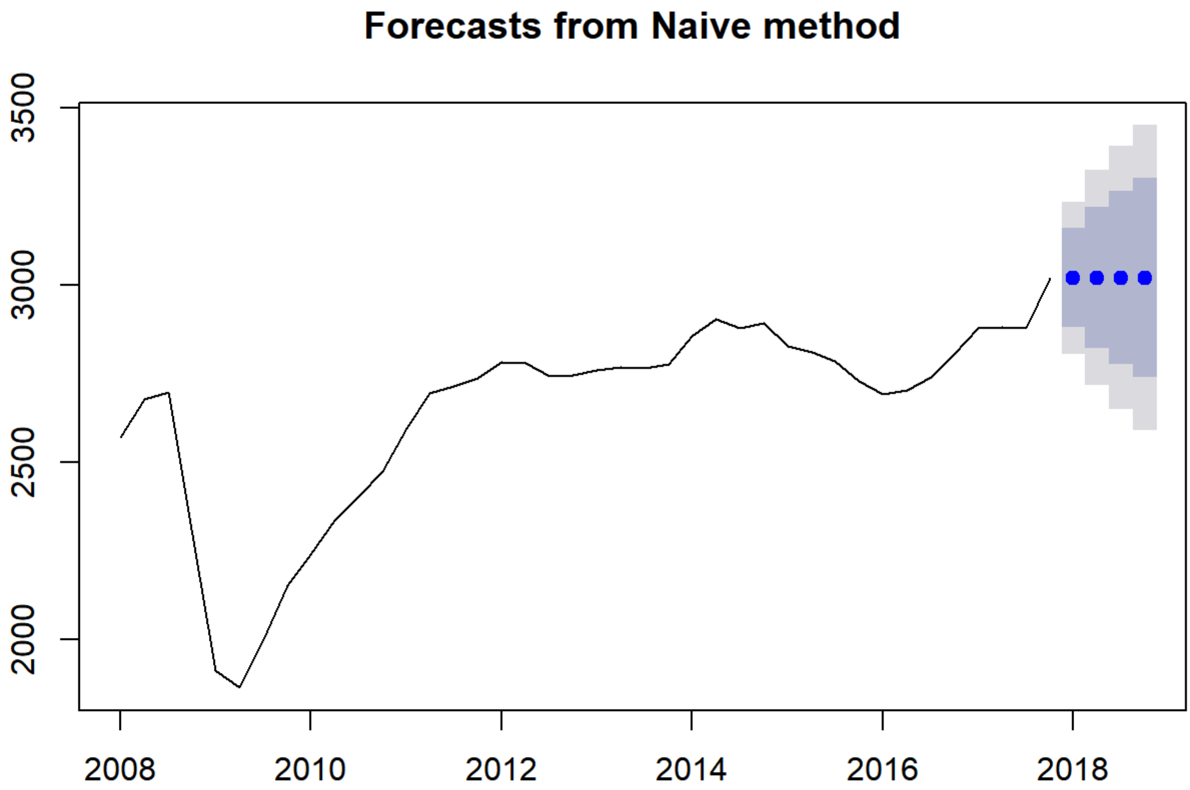
```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 11.54359 109.7054 68.08205 0.2970542 2.852688 0.3286562
##                   ACF1
## Training set 0.5405785
```

Since RMSE value which is very high i.e., 109.7054 indicates that the forecasting method doesn't capture all the information and is not the best method for forecasting.

```
forecast(naive_forecast, h=4)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018 Q1         3021.6 2881.006 3162.194 2806.580 3236.620
## 2018 Q2         3021.6 2822.770 3220.430 2717.516 3325.684
## 2018 Q3         3021.6 2778.084 3265.116 2649.174 3394.026
## 2018 Q4         3021.6 2740.412 3302.788 2591.560 3451.640
```

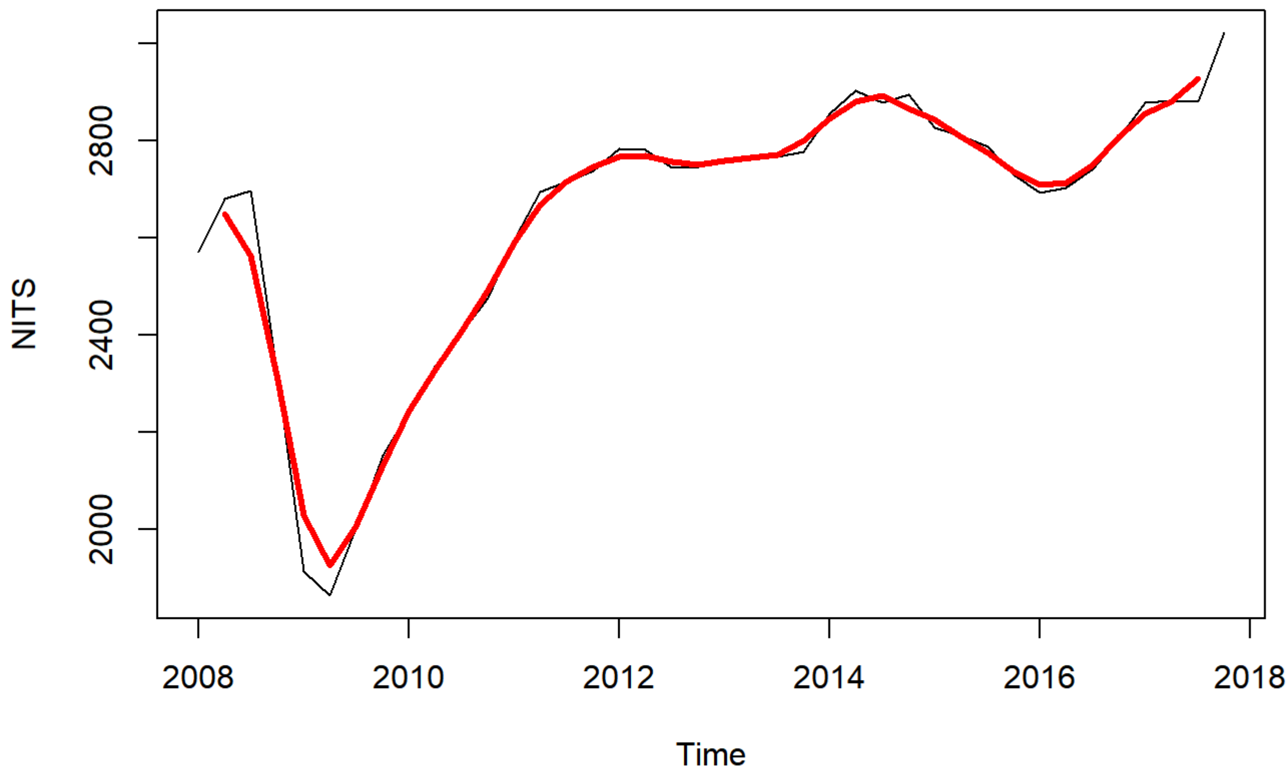
```
plot(forecast(naive_forecast, h=4))
```



Simple Exponential Smoothing

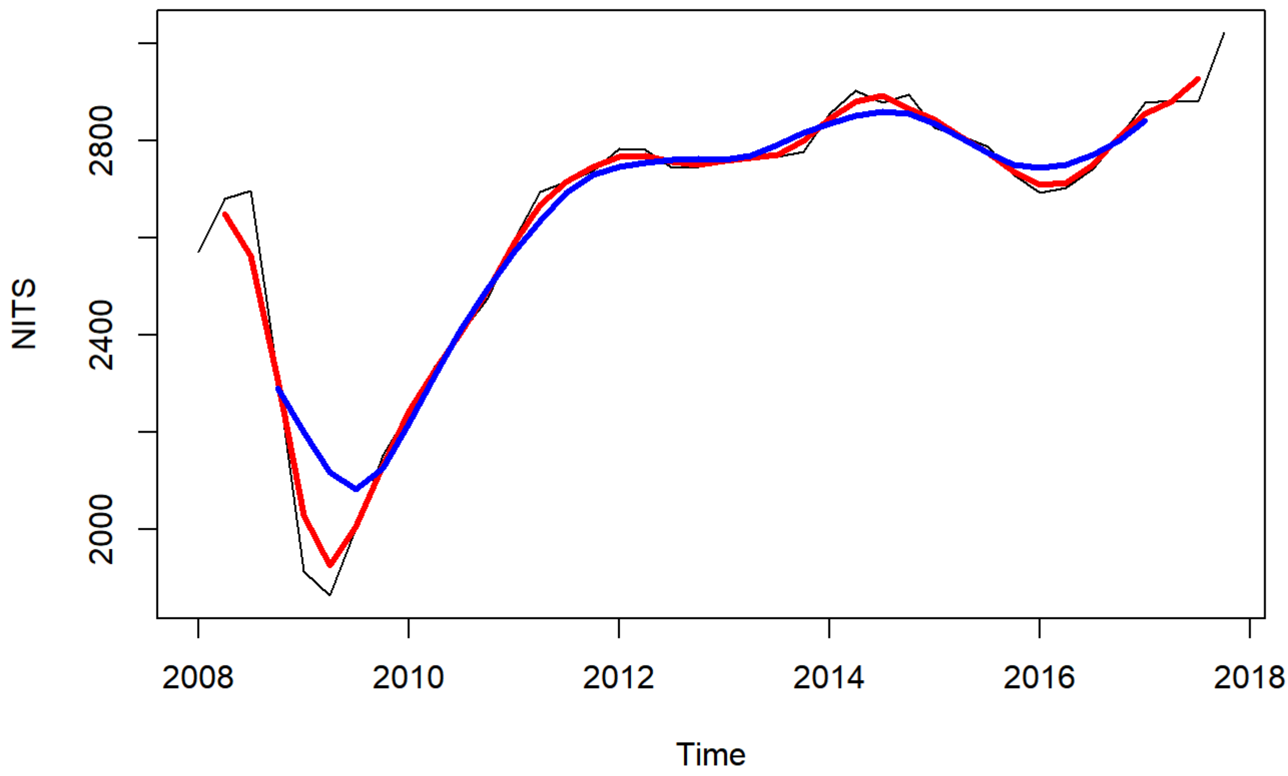
Simple Moving Averages

```
plot(NITS)
ma3 <- ma(NITS, order = 3)
lines(ma3, col= 'Red', lwd =3)
```



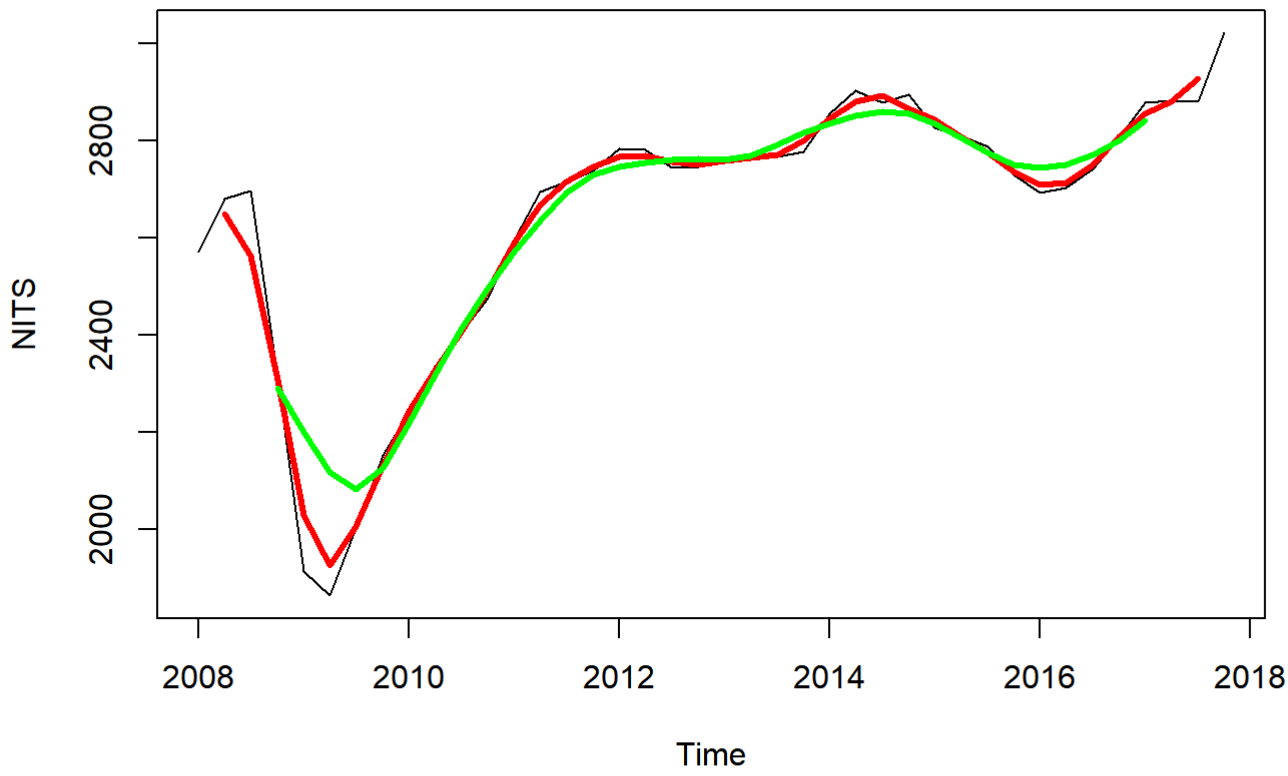
Simple Moving average of order 3 in Red

```
plot(NITS)
ma6 <- ma(NITS, order = 6)
lines(ma3, col= 'Red', lwd =3)
lines(ma6, col= 'Blue', lwd = 3)
```



Simple Moving average of order 6 in Blue

```
plot(NITS)
ma9 <- ma(NITS, order = 9)
lines(ma3, col= 'Red', lwd =3)
lines(ma6, col= 'Blue', lwd = 3)
lines(ma6, col= 'Green', lwd = 3)
```

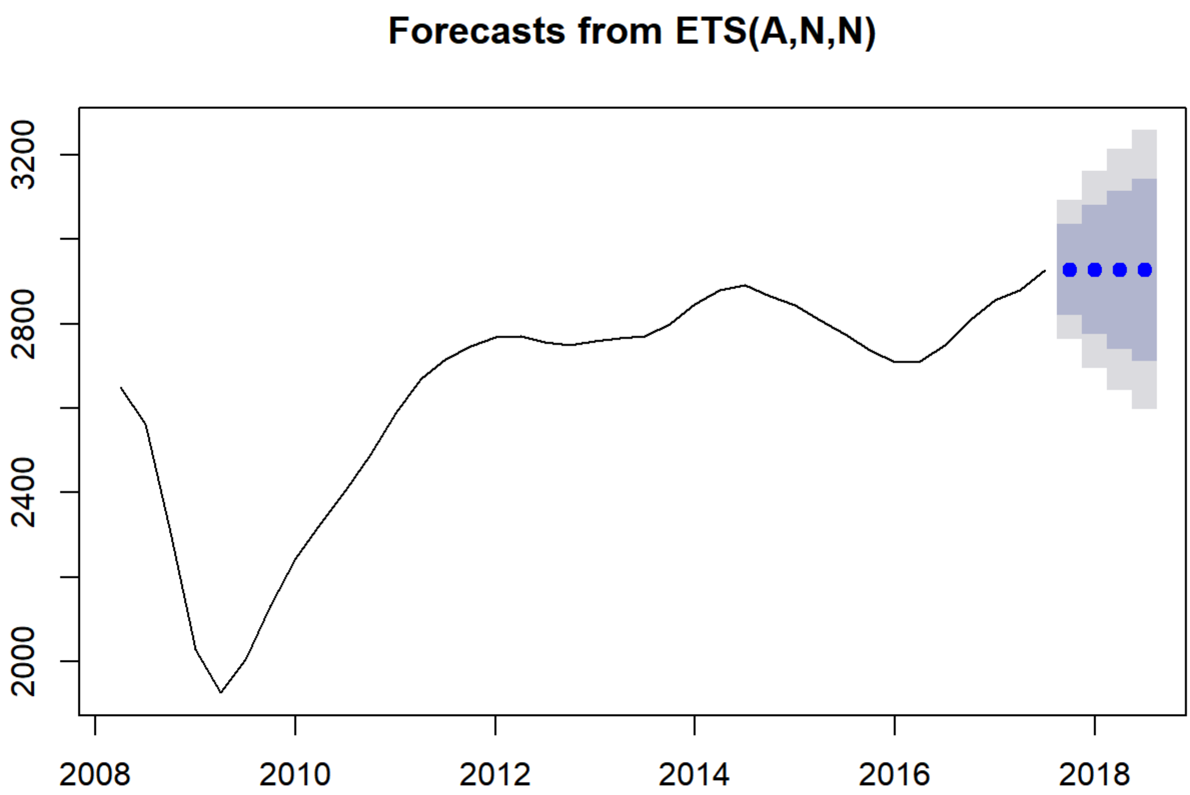


Simple Moving average of order 9 in Green

```
forecast_ma <- forecast(object = ma3, h = 4)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

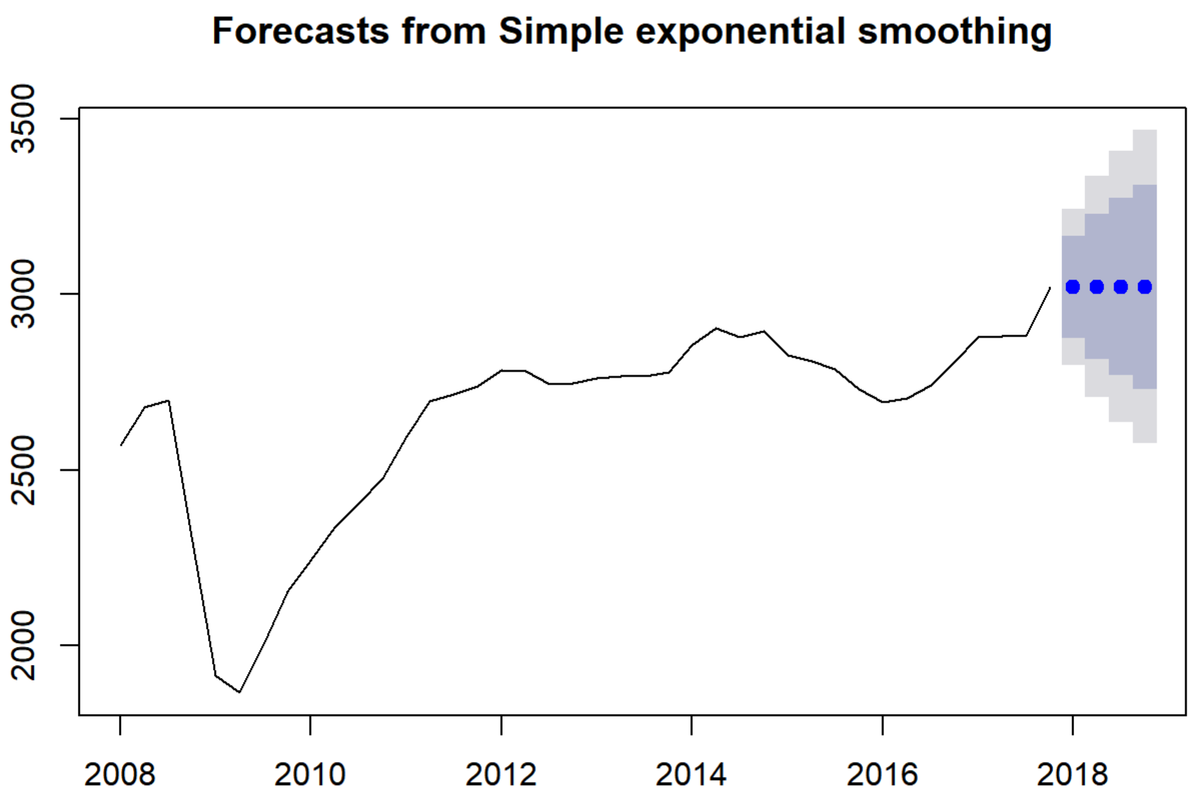
```
plot(forecast_ma)
```



Moving averages of order 3 is choosen because it overlaps with all the other orders and hence it makes better predictions.

Smoothing

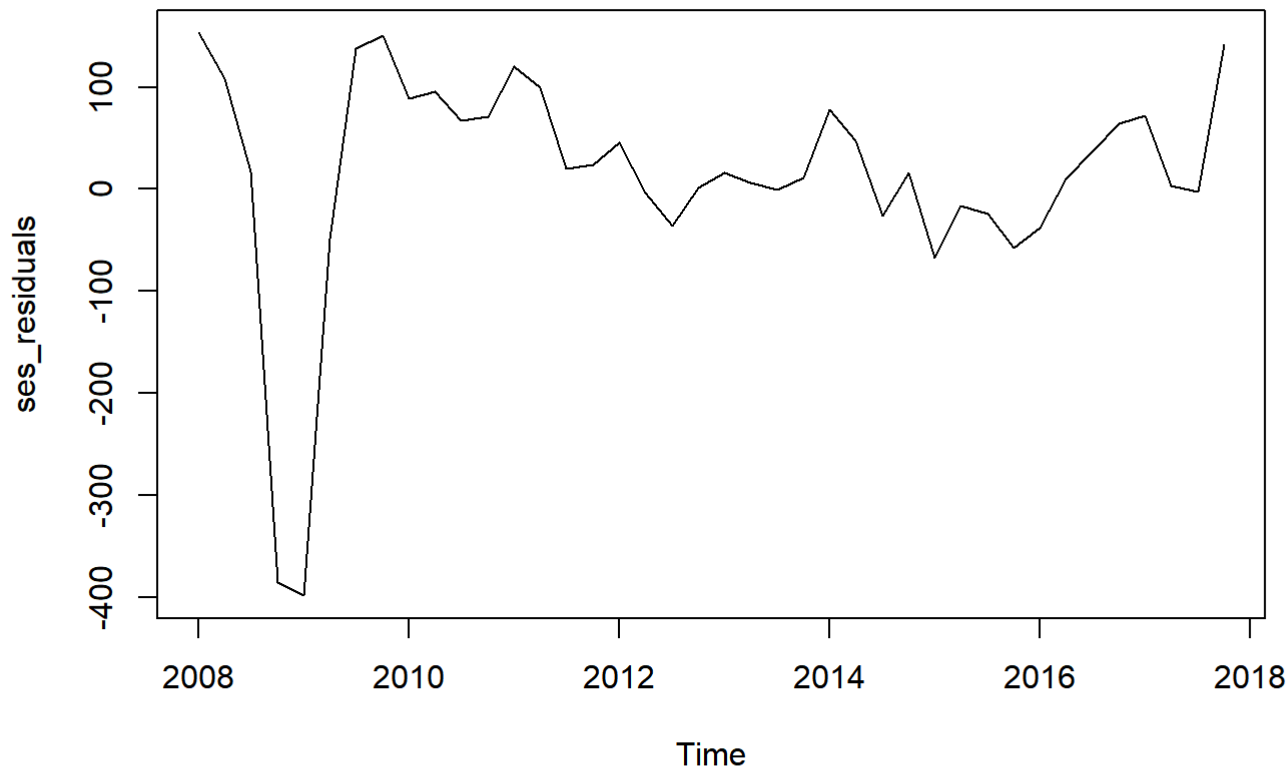
```
ses_forecast <- ses(NITS , h = 4)
plot(ses_forecast)
```



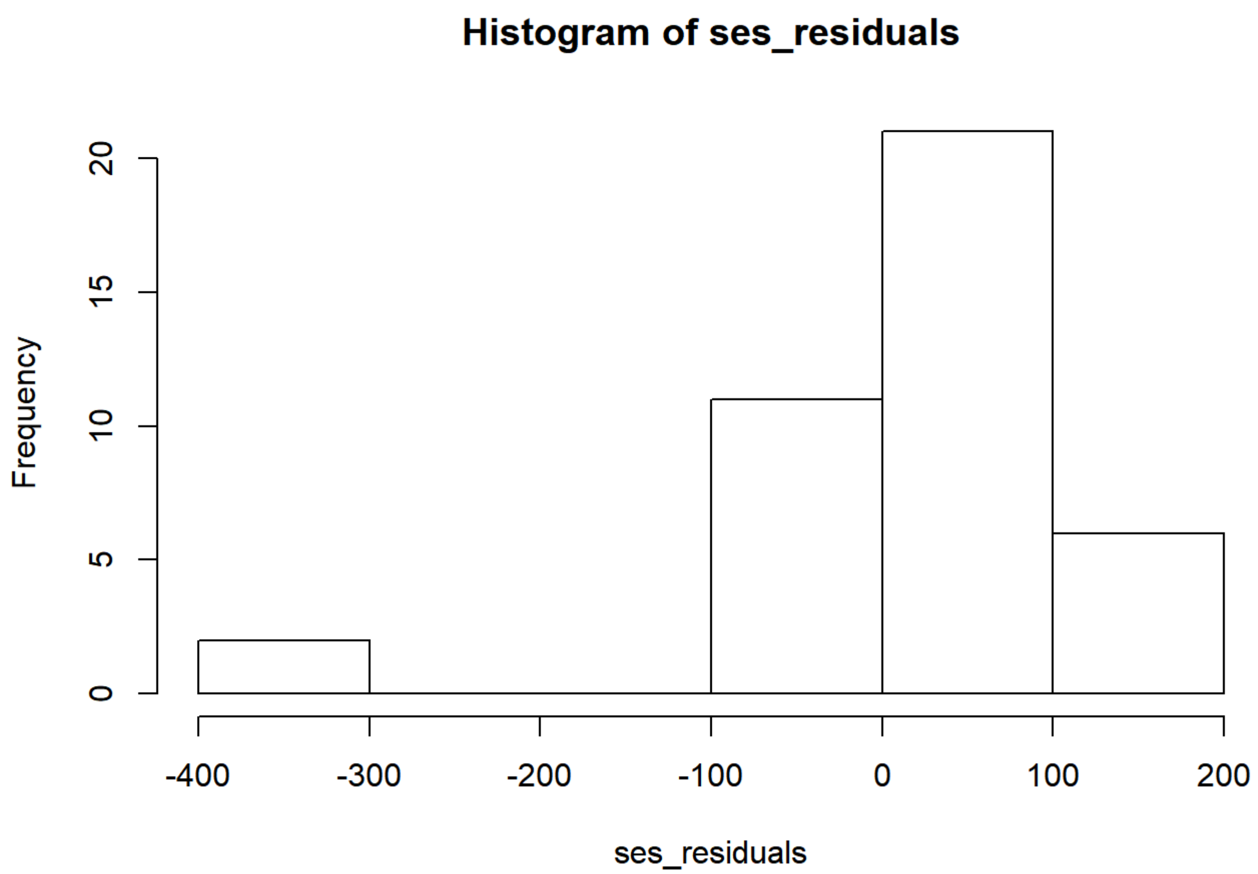

```
summary(ses_forecast)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = NITS, h = 4)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 2418.1821
##
##   sigma:  113.8911
##
##      AIC     AICc      BIC
## 530.3229 530.9896 535.3895
##
## Error measures:
##                   ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 15.0866 111.0074 70.21568 0.4386094 2.930566 0.338956
##                   ACF1
## Training set 0.5480613
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018 Q1       3021.586 2875.628 3167.543 2798.363 3244.808
## 2018 Q2       3021.586 2815.181 3227.990 2705.917 3337.254
## 2018 Q3       3021.586 2768.797 3274.375 2634.979 3408.193
## 2018 Q4       3021.586 2729.693 3313.479 2575.174 3467.997
```

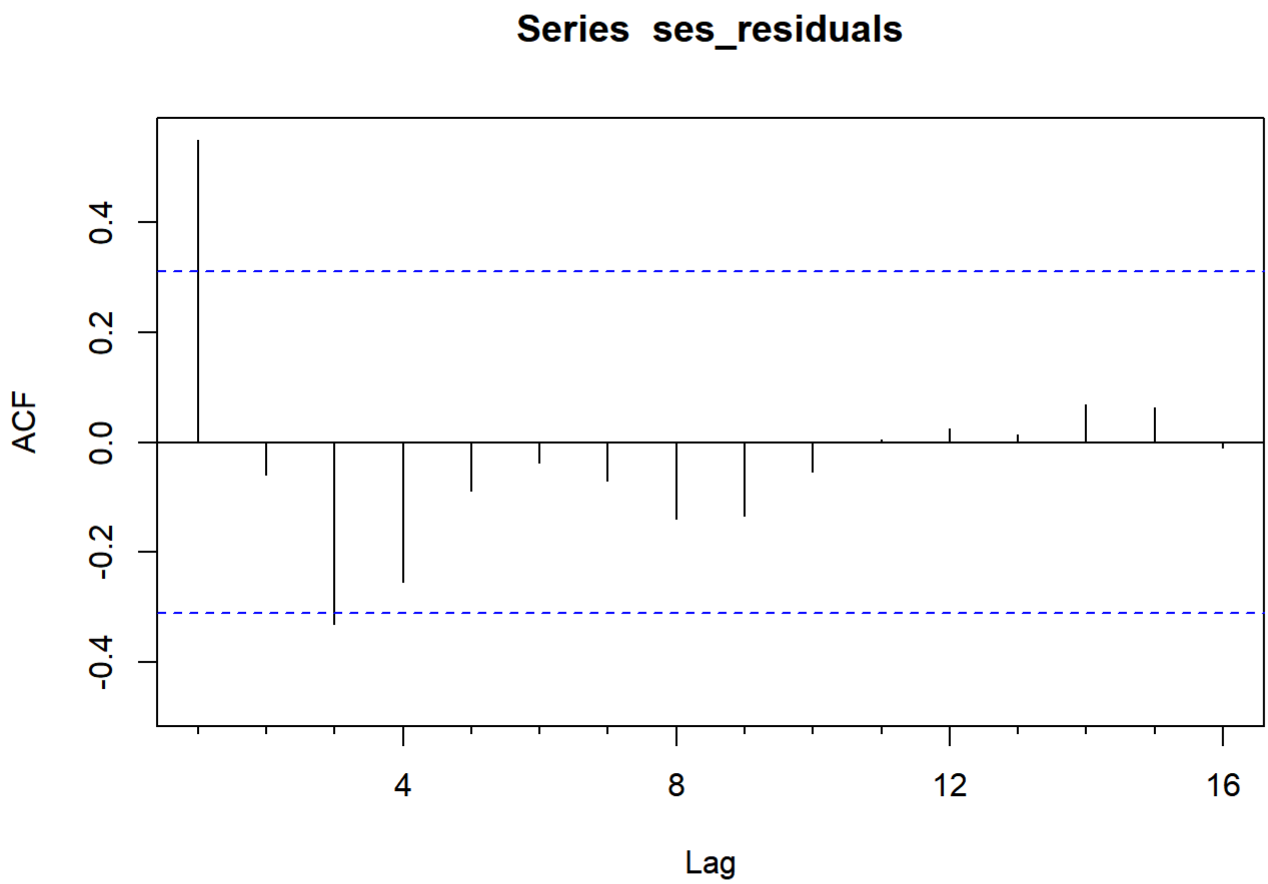
```
ses_residuals <- residuals(ses_forecast)
plot(ses_residuals)
```




```
hist(ses_residuals)
```



```
Acf(ses_residuals)
```



These graphs show that the simple exponential smoothing method produces forecasts that appear to account for all available information. The mean of the residuals is close to zero and there is a little correlation in the residuals series. This can also be seen on the histogram of the residuals. The histogram suggests that the residuals are not be normal.

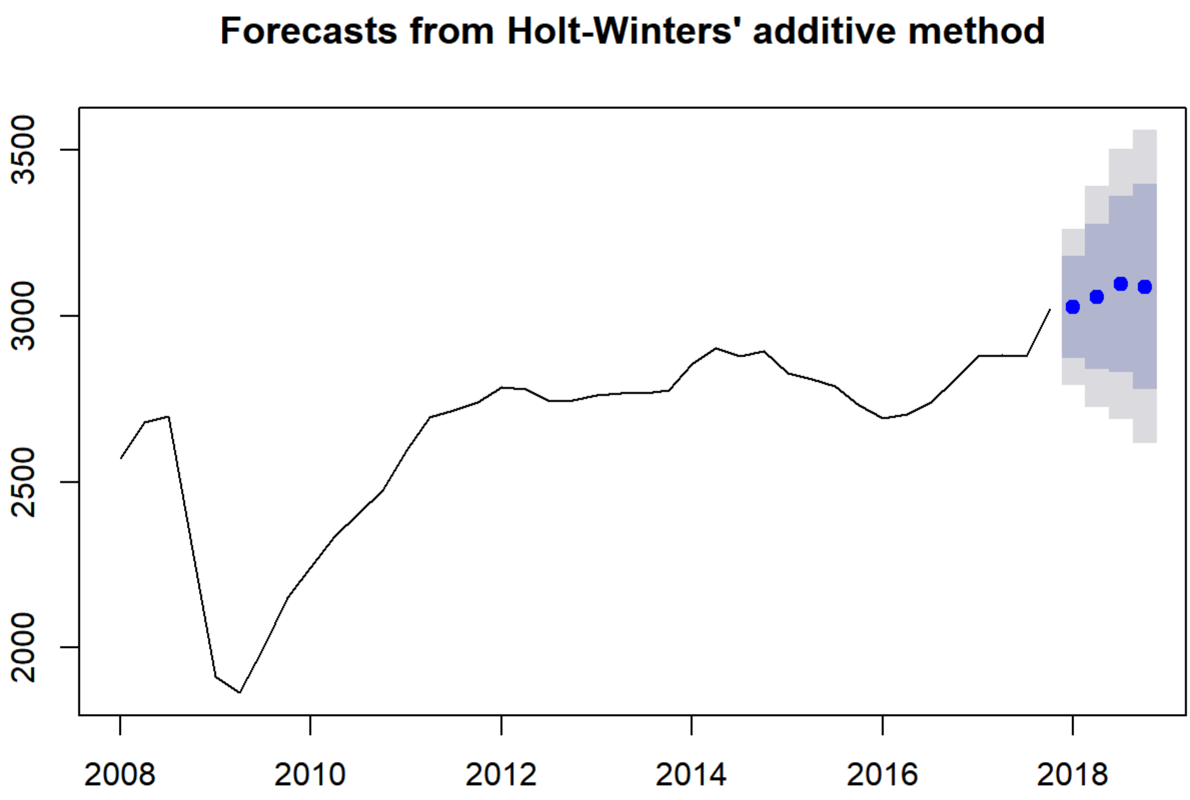
```
accuracy(ses_forecast)
```

```
##                   ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 15.0866 111.0074 70.21568 0.4386094 2.930566 0.338956
##                   ACF1
## Training set 0.5480613
```

Since RMSE value which is very high i.e., 111.0094 indicates that the forecasting method isn't performing well.

Holt-Winters

```
holt <- hw(NITS, h = 4)
plot(holt)
```

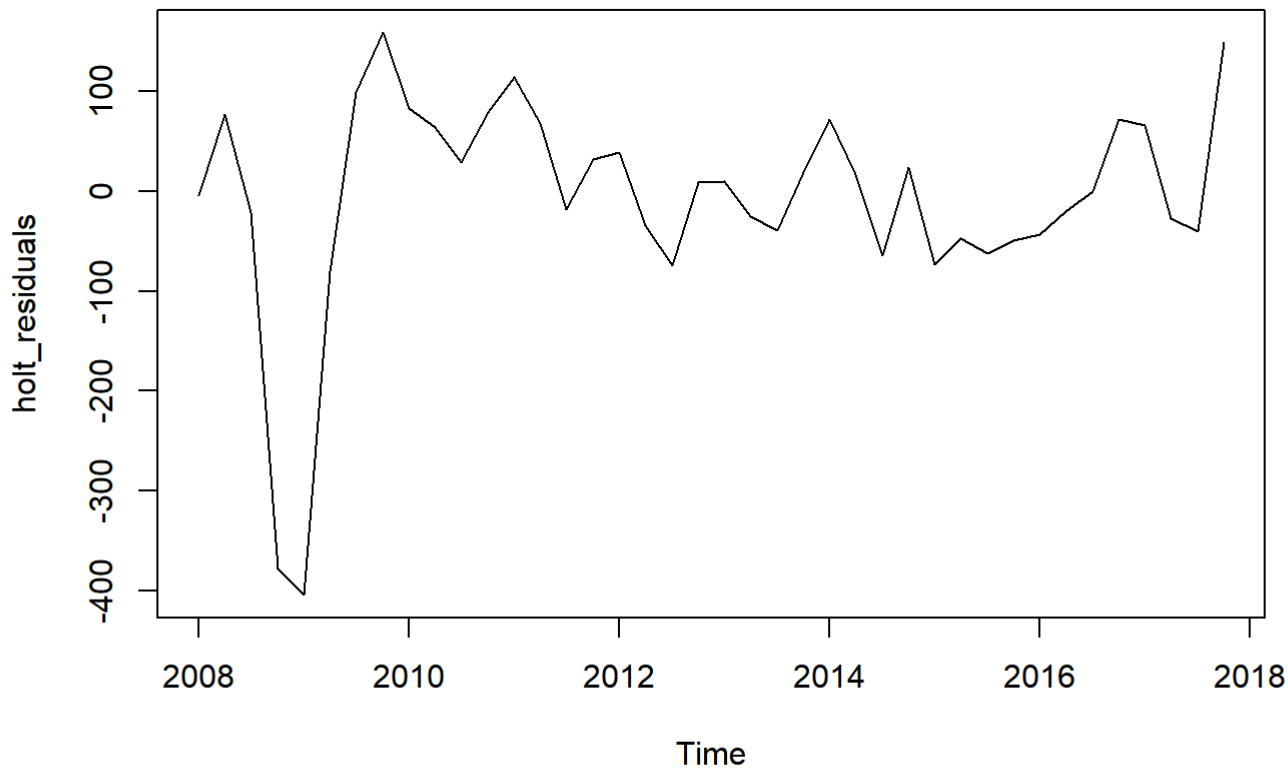



```
summary(holt)
```

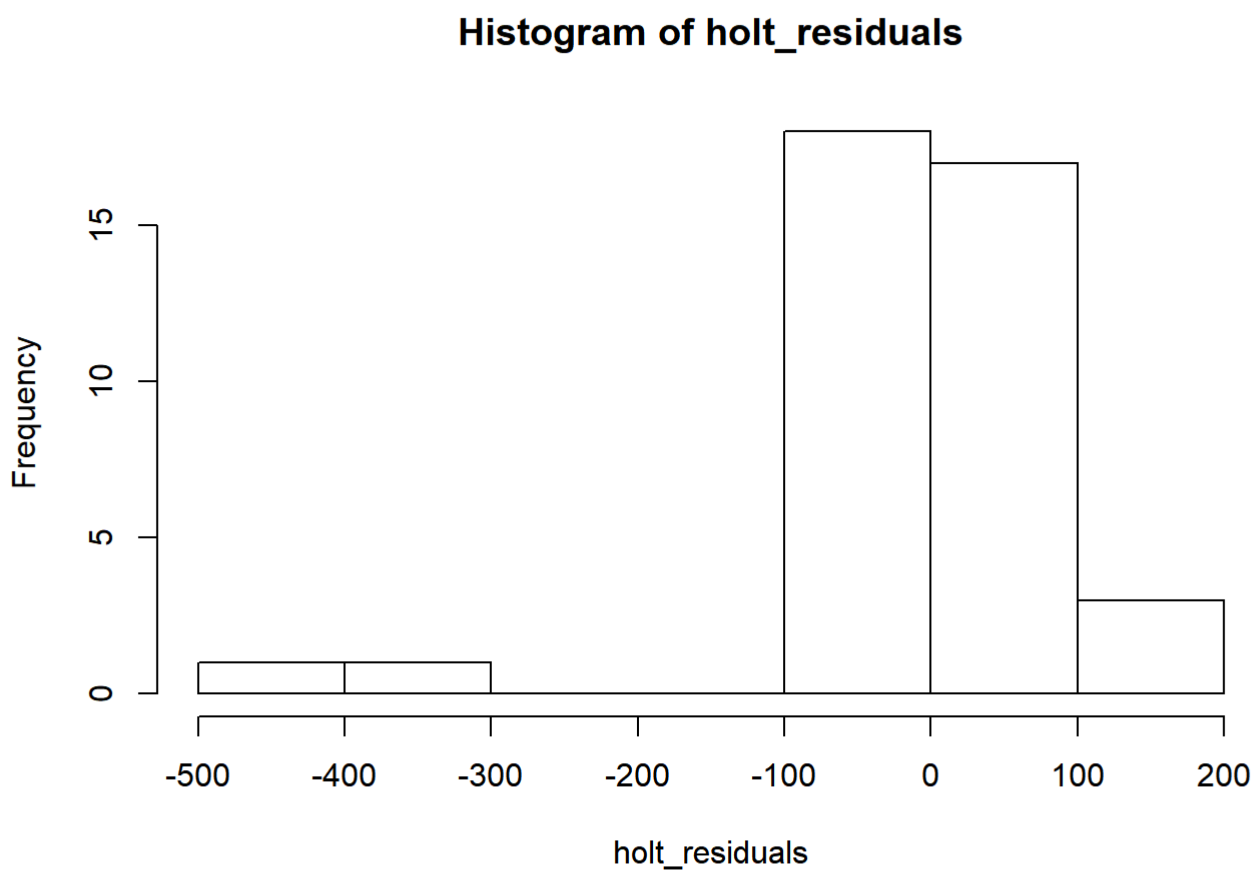
```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = NITS, h = 4)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 2573.5921
##     b = 16.7667
##     s = -4.3838 20.3996 -0.9418 -15.074
##
##   sigma:  120.2626
##
##      AIC     AICc      BIC
## 539.8037 545.8037 555.0036
##
## Error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -5.430435 107.5662 69.71705 -0.3492683 2.889605 0.3365489
##                   ACF1
## Training set 0.5309393
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018 Q1       3027.623 2873.500 3181.746 2791.912 3263.333
## 2018 Q2       3058.515 2840.555 3276.476 2725.173 3391.857
## 2018 Q3       3096.579 2829.624 3363.533 2688.307 3504.851
## 2018 Q4       3088.577 2780.312 3396.842 2617.126 3560.028
```

The value of alpha signifies that the variations are not smoothed and the predictions are unstable The value of beta signifies that trend completely depends on the previous period value. The value of gamma signifies that there is no seasonality in the predictions.

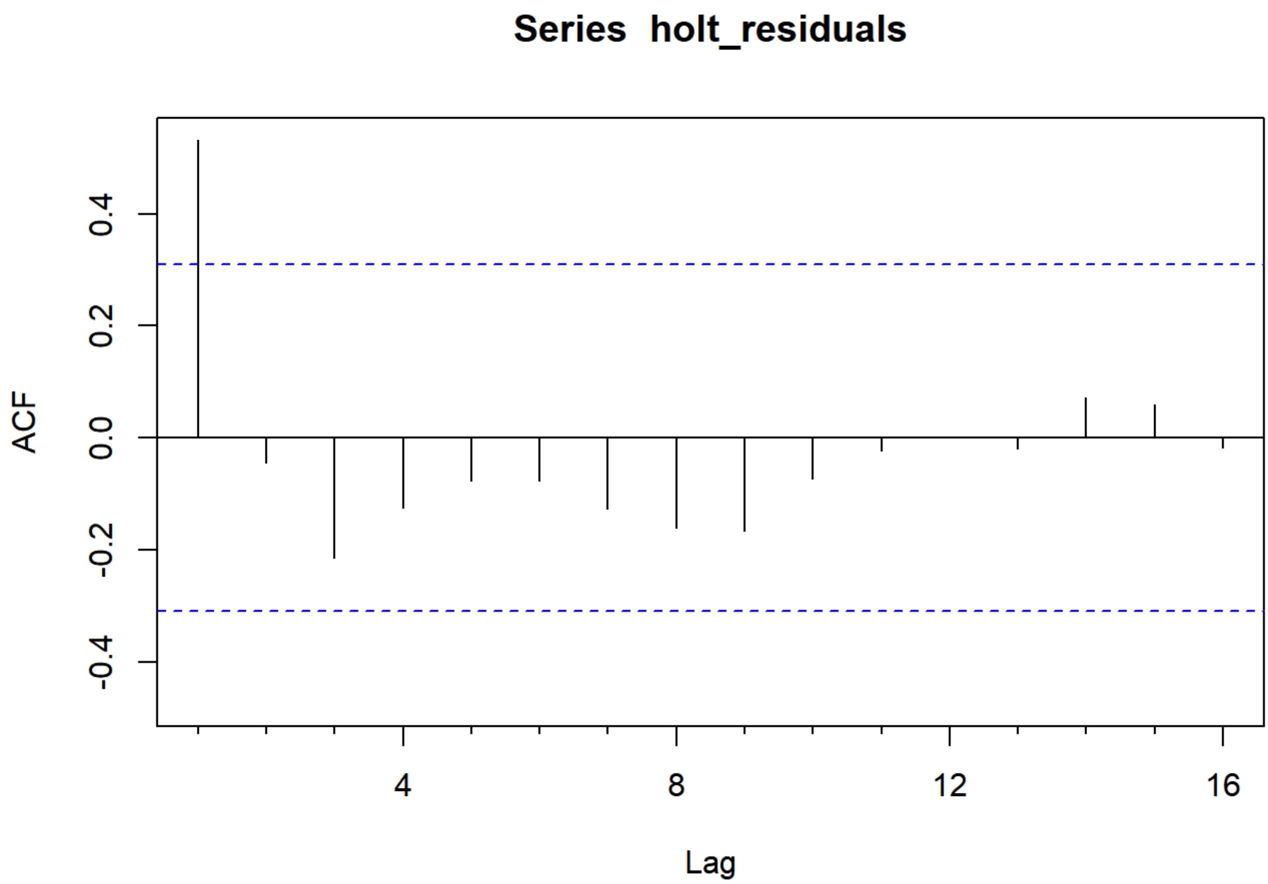
```
holt_residuals <- resid(holt)
plot(holt_residuals)
```




```
hist(holt_residuals)
```



```
Acf(holt_residuals)
```



These graphs show that the Holt Winters method produces forecasts that appear to account for all available information. The mean of the residuals is close to zero and there is a little correlation in the residuals series. This can also be seen on the histogram of the residuals. The histogram suggests that the residuals are not be normal.

```
forecast(holt)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018 Q1       3027.623 2873.500 3181.746 2791.912 3263.333
## 2018 Q2       3058.515 2840.555 3276.476 2725.173 3391.857
## 2018 Q3       3096.579 2829.624 3363.533 2688.307 3504.851
## 2018 Q4       3088.577 2780.312 3396.842 2617.126 3560.028
```

```
accuracy(holt)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -5.430435 107.5662 69.71705 -0.3492683 2.889605 0.3365489
##                   ACF1
## Training set 0.5309393
```

Since RMSE value which is very high i.e., 107.5662 indicates that the forecasting method isn't performing well.

ARIMA or Box-Jenkins

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.4.4
```

```
adf.test(NITS)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  NITS
## Dickey-Fuller = -2.907, Lag order = 3, p-value = 0.2175
## alternative hypothesis: stationary
```

```
kpss.test(NITS)
```

```
## Warning in kpss.test(NITS): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  NITS
## KPSS Level = 1.1832, Truncation lag parameter = 1, p-value = 0.01
```

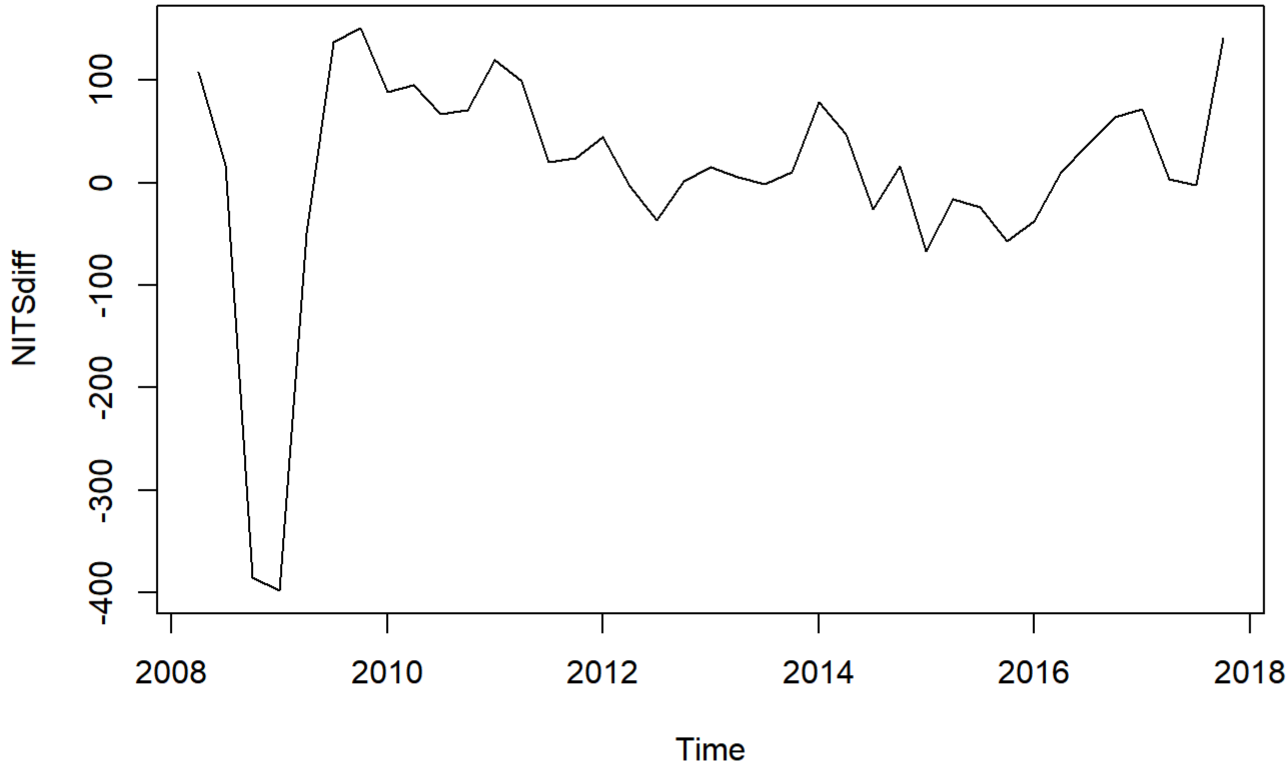
From both ADF and KPSS test it is evident that time-seriesis not stationary

```
ndiffs(NITS)
```

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

1 difference is needed to make the time-series stationary

```
NITSdiff <- diff(NITS, differences = 1)
plot(NITSdiff)
```




```
adf.test(NITSdiff)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  NITSdiff
## Dickey-Fuller = -4.9257, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

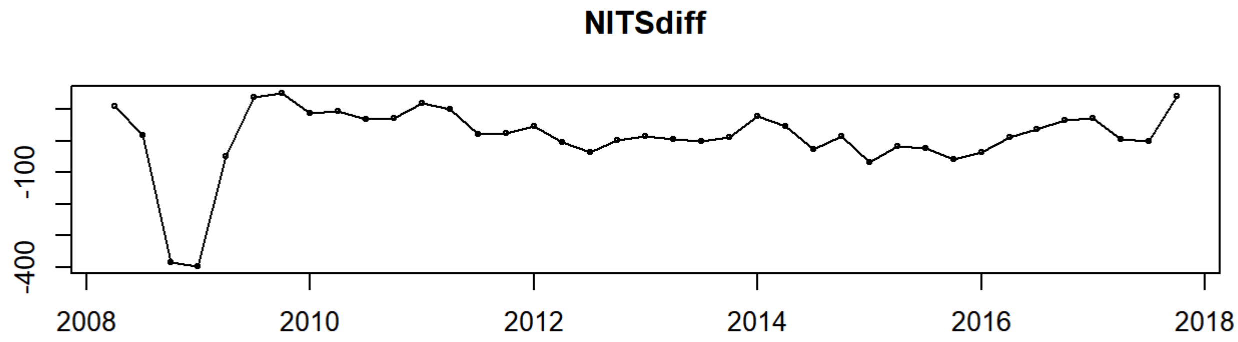
```
kpss.test(NITSdiff)
```

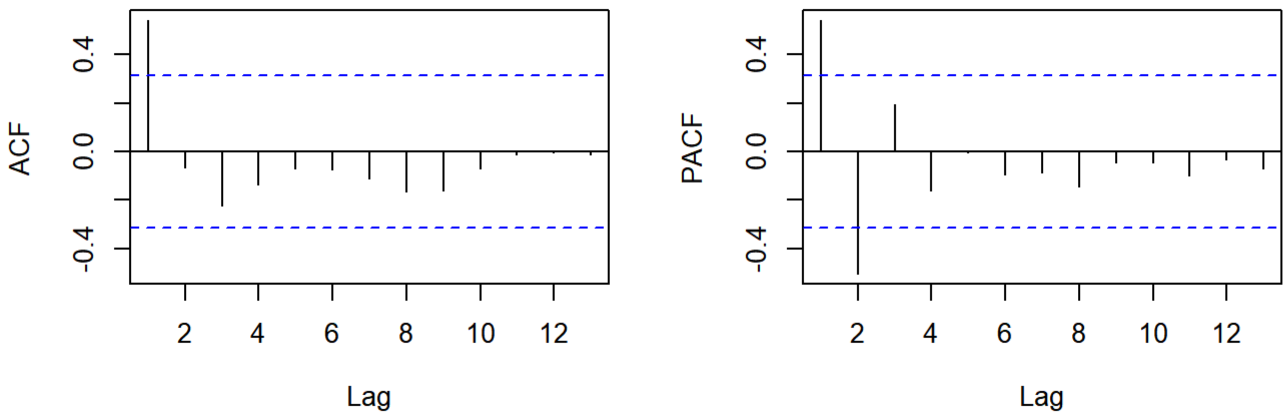
```
## Warning in kpss.test(NITSdiff): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  NITSdiff
## KPSS Level = 0.093613, Truncation lag parameter = 1, p-value = 0.1
```

The adf and kpss retests indicates that time-series is now statinary

```
tsdisplay(NITSdiff)
```





Based on ACF and PACF plots the possible ARIMA models are ARIMA(1,1,1) ARIMA(2,1,1)

```
arima(NITS, order = c(1,1,1))
```

```
##
## Call:
## arima(x = NITS, order = c(1, 1, 1))
##
## Coefficients:
##          ar1     ma1
##       0.2973  0.7816
## s.e.  0.1682  0.0924
##
## sigma^2 estimated as 5372:  log likelihood = -223.55,  aic = 453.1
```

```
arima(NITS, order = c(2,1,1))
```

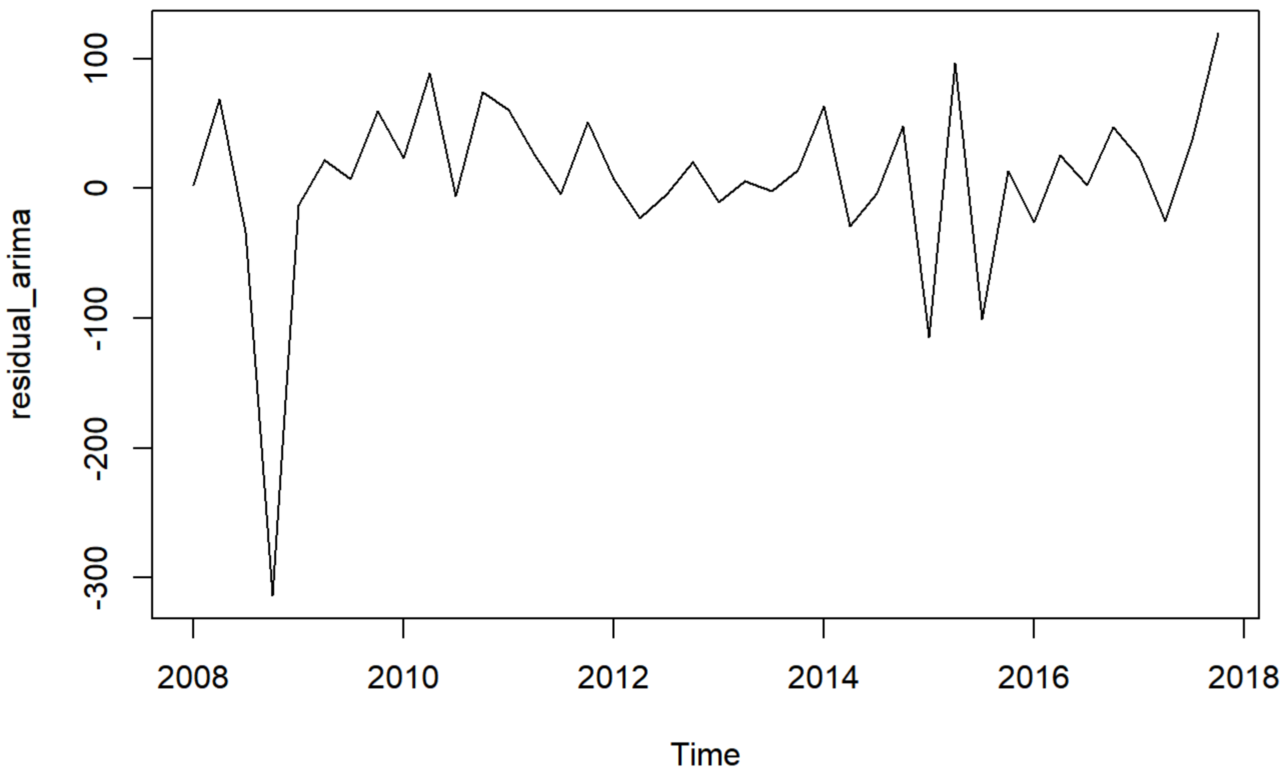
```
##
## Call:
## arima(x = NITS, order = c(2, 1, 1))
##
## Coefficients:
##          ar1      ar2     ma1
##       0.5013  -0.3525  0.6409
## s.e.  0.1958   0.1824  0.1657
##
## sigma^2 estimated as 4912:  log likelihood = -221.93,  aic = 451.87
```

Based on the AIC, BIC Sigma^2 values ARIMA(2,1,1) model performs well. We can validate this by running auto.arima()

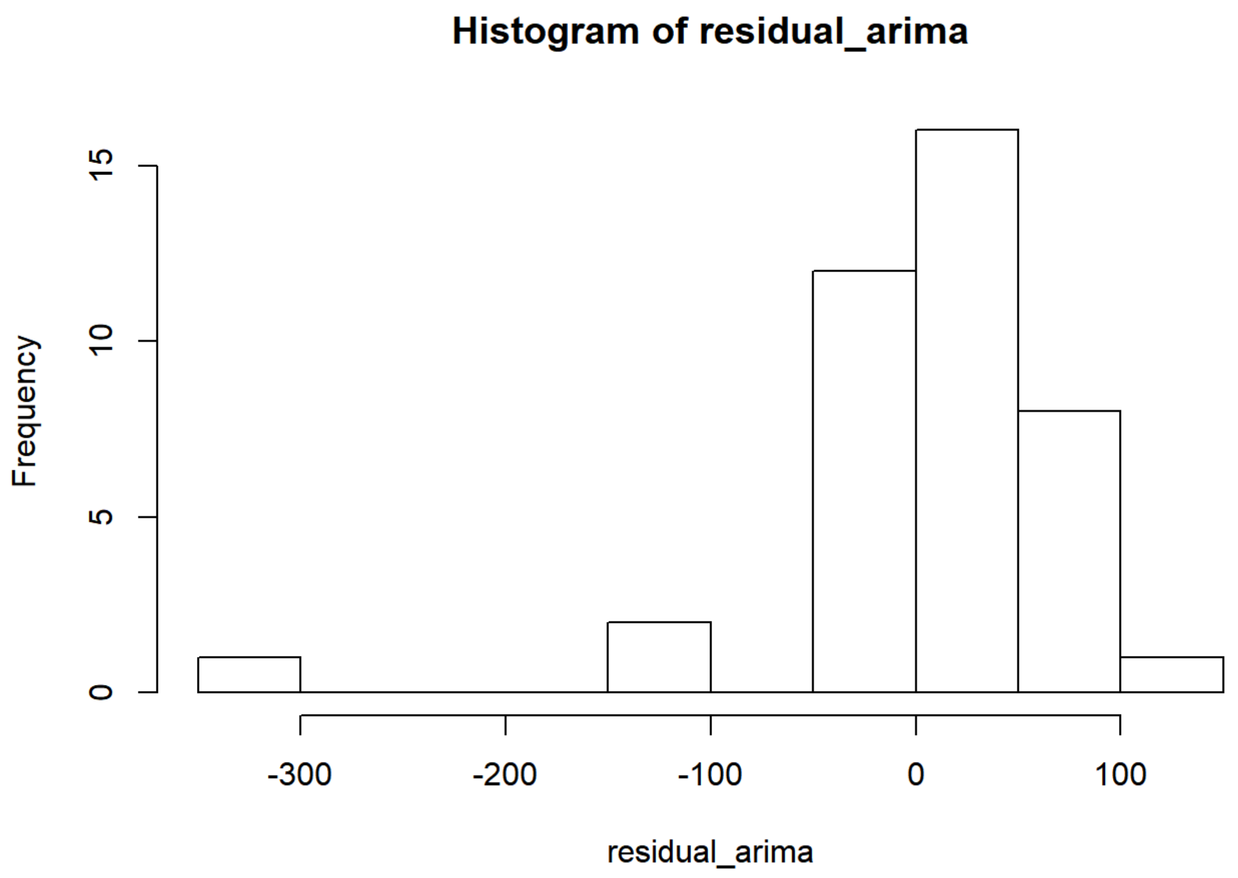
```
arima_NITS <- auto.arima(NITS)
arima_NITS
```

```
## Series: NITS
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2     ma1
##       0.5013  -0.3525  0.6409
## s.e.  0.1958   0.1824  0.1657
##
## sigma^2 estimated as 5322:  log likelihood=-221.93
## AIC=451.87   AICc=453.04   BIC=458.52
```

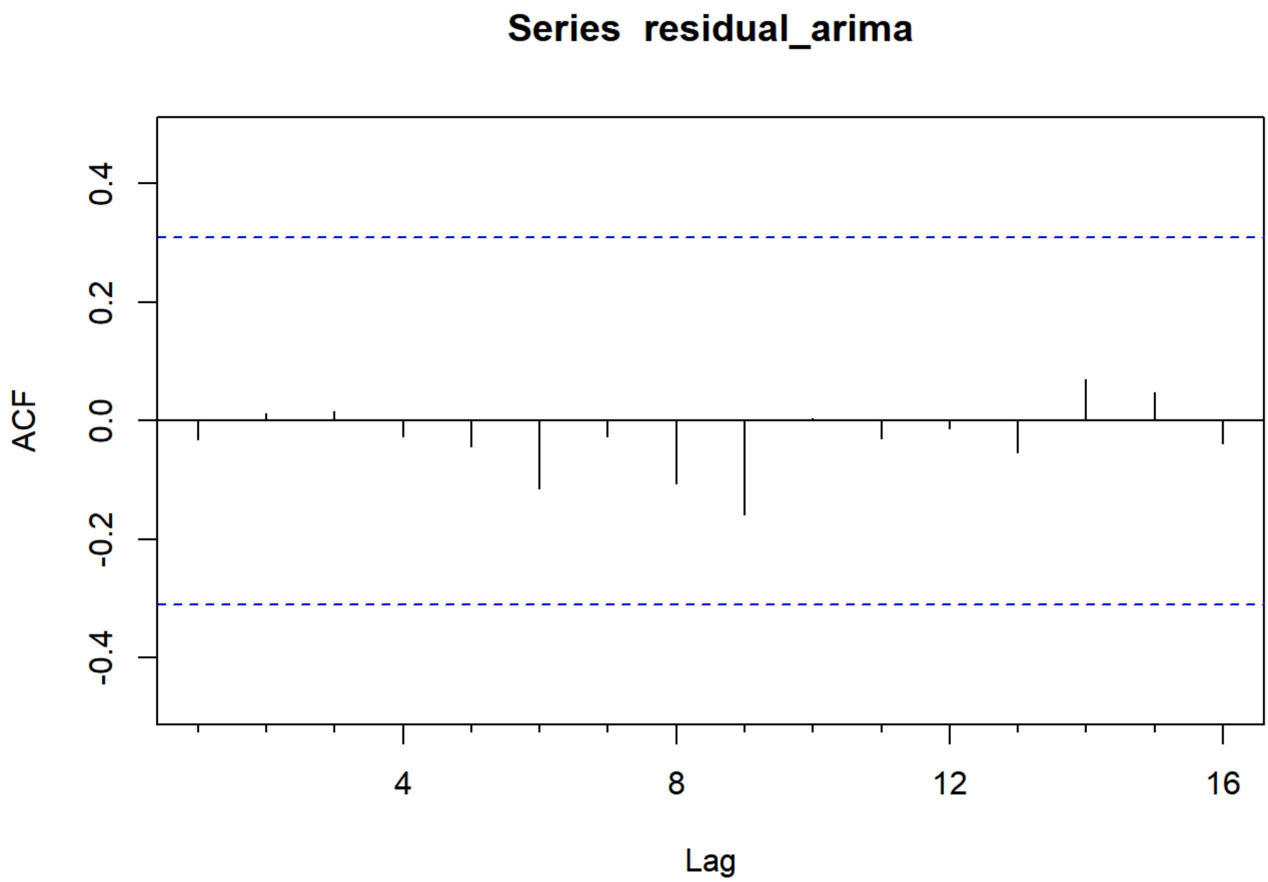
```
residual_arima <- residuals(arima_NITS)
plot(residual_arima)
```




```
hist(residual_arima)
```




```
Acf(residual_arima)
```

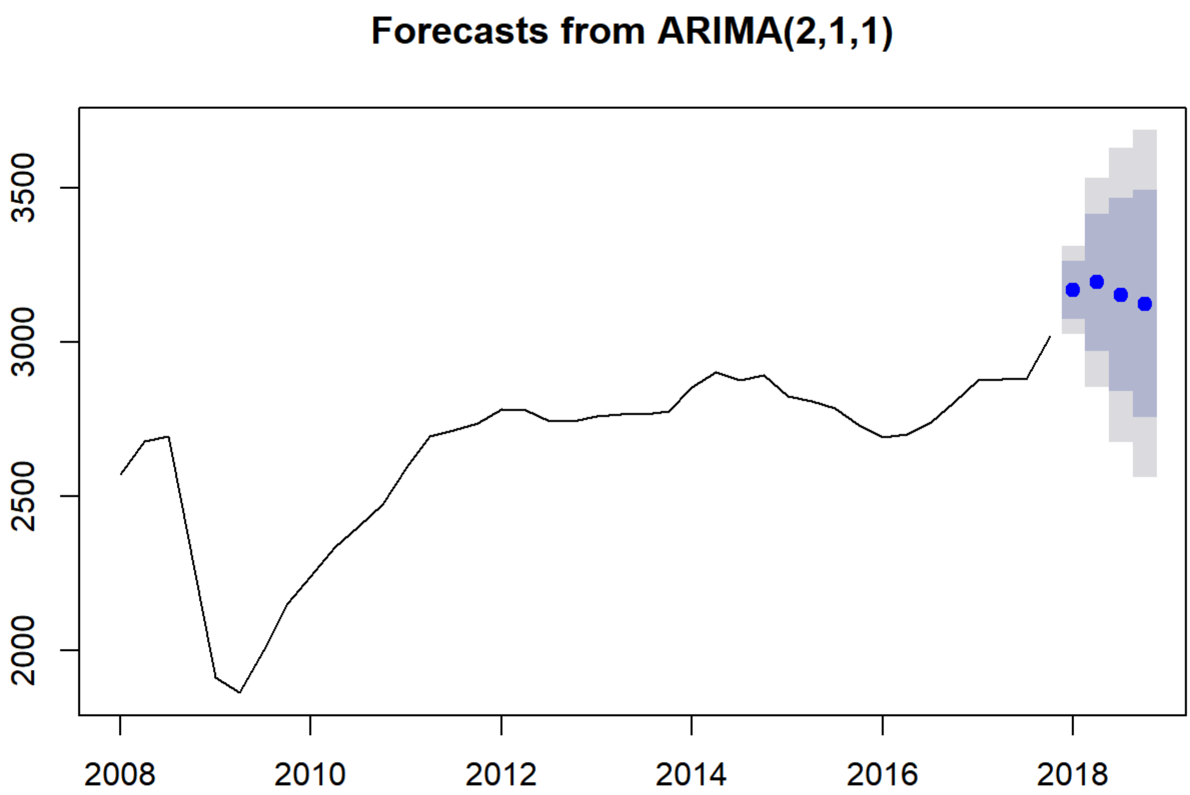


These graphs show that the ARIMA model produces forecasts that appear to account for all available information. The mean of the residuals is close to zero and there is no significant correlation in the residuals series. This can also be seen on the histogram of the residuals. The histogram suggests that the residuals are not be normal.

```
forecast(arima_NITS, h =4)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018 Q1       3169.929 3076.438 3263.419 3026.947 3312.910
## 2018 Q2       3194.406 2973.382 3415.430 2856.379 3532.433
## 2018 Q3       3154.390 2841.935 3466.844 2676.532 3632.247
## 2018 Q4       3125.701 2758.169 3493.233 2563.609 3687.793
```

```
plot(forecast(arima_NITS, h=4))
```




```
summary(arima_NITS)
```

```
## Series: NITS
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2     ma1
##       0.5013  -0.3525  0.6409
## s.e.  0.1958   0.1824  0.1657
##
## sigma^2 estimated as 5322:  log likelihood=-221.93
## AIC=451.87   AICc=453.04   BIC=458.52
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 7.607852 69.20748 42.97381 0.263098 1.657254 0.2074498
##                     ACF1
## Training set -0.03197029
```

This model performs best as the RMSE value i.e 69.20748 is low.

Naive: This model uses last period’s data to forecast without adjusting them for any conditions. More useful when we want to compare this model’s forecast with other forecasts generated by better models.

Simple Moving average: Simple Moving Average depends on the order, the smoothness increases with the order.

Simple Exponential Smoothing: This model makes the data smooth using the exponential window function and are used to assign exponentially decreasing weights over time. More useful when recent observations need to be given more weight than past observations.

Holt-Winters: This model is used to capture seasonality and comprises of three major components i.e. error, trend and seasonality. More useful when we want the model to be fast as it is incremental and saves time and it’s three components helps us analyze how data is split.

ARIMA: ARIMA model produces forecasts based upon prior values in the time series and the errors made by previous predictions. This allows the model to rapidly adjust for sudden changes in trend, resulting in more accurate forecasts.