HW5

Oluwafemi Shobowale

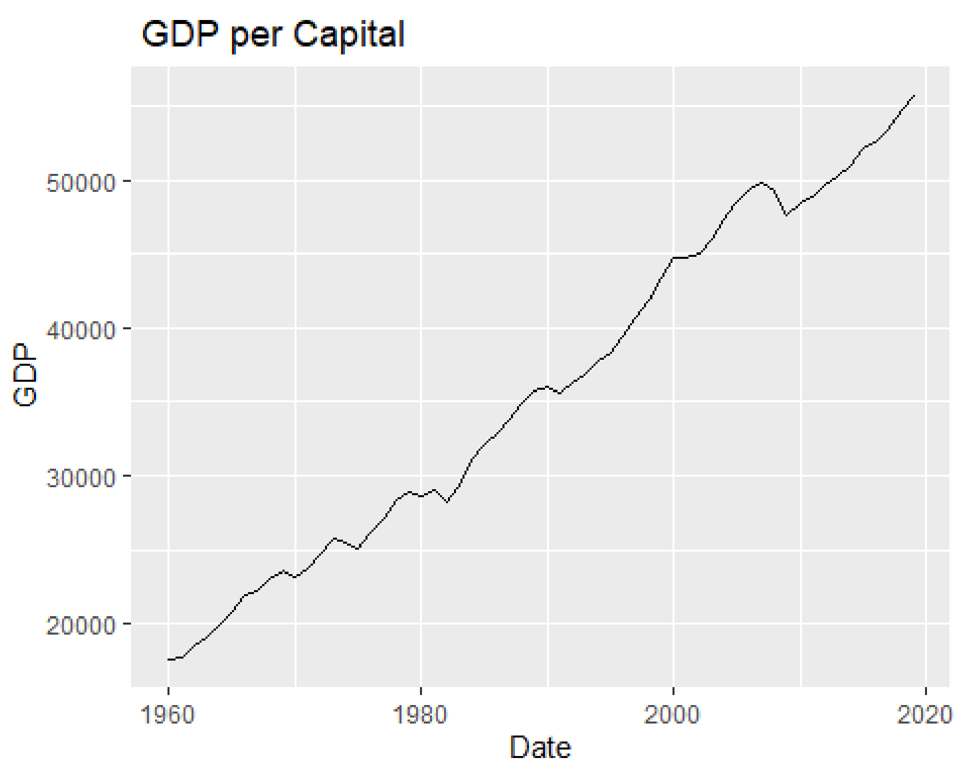
5/3/2022

1,

1a, Observation from the graph indicated a linear trend, increase in variance over time and trend are signs which indicated non-stationarity. There seem to be some downward deviation/pull toward the end (drift).

Hypotesis , an AR with some sort of drift , trend stationary would fit this seires. Series shows no seasonal pattern.

```
ggplot(GDP_ts,aes(x= time(GDP_ts) ,y= GDP_ts
,group=1))+geom_line()+labs(title = " GDP per Capital ")+xlab("Date")+ylab("
GDP")
```



1b, GDP series fail to reject null hypotesis for all 3 ADF test (‘nc’,‘c’,‘ct’). The KPSS test reject stationary for level and fail to reject for trend-stationary. They both agree for time-trend in ADP and trend stationary in KPSS.

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: 1.8467
##   P VALUE:
##     0.9819
##
## Description:
##  Mon May 23 12:49:07 2022 by user: soboa

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: 0.0349
##   P VALUE:
##     0.9554
##
## Description:
##  Mon May 23 12:49:07 2022 by user: soboa

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -2.0699
##   P VALUE:
##     0.5465
##
## Description:
##  Mon May 23 12:49:07 2022 by user: soboa

##
##  KPSS Test for Level Stationarity
```

```
##
## data:  GDP_ts
## KPSS Level = 1.601, Truncation lag parameter = 3, p-value = 0.01

##
##  KPSS Test for Trend Stationarity
##
## data:  GDP_ts
## KPSS Trend = 0.14504, Truncation lag parameter = 3, p-value = 0.05178
```

1c, OLS shows a significant R^2 value, all coefficeient are significant. Residual ACF decays slowly to zero MA(1) maybe, PACF suggest and AR(2) and eacf suggest ARIMA(1,1)

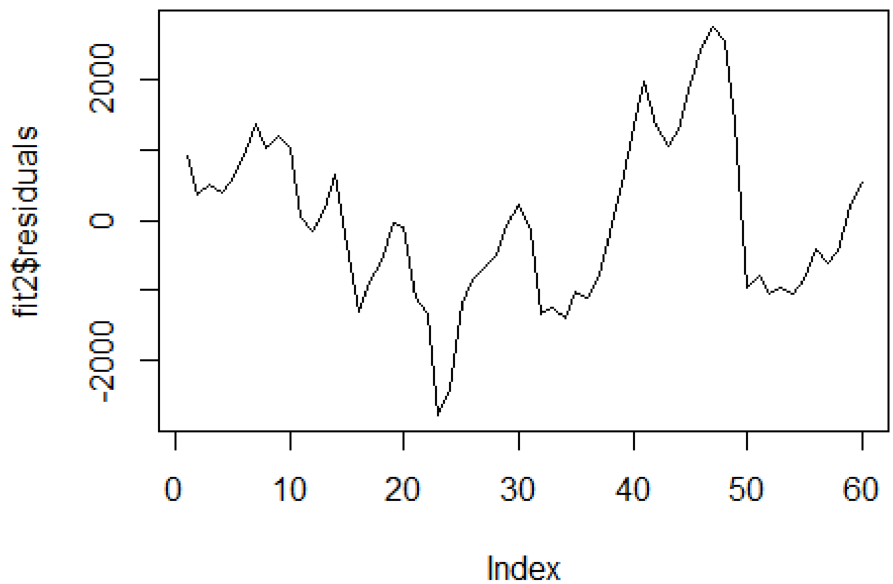
ACF show slow decay to zero indicates non-stationary, pacf doesn't show much either an expoinential fall or a slow fall hence PACF could not be used to draw a conclusion. EACF has no long trends of x"s on the top hence series is stationary

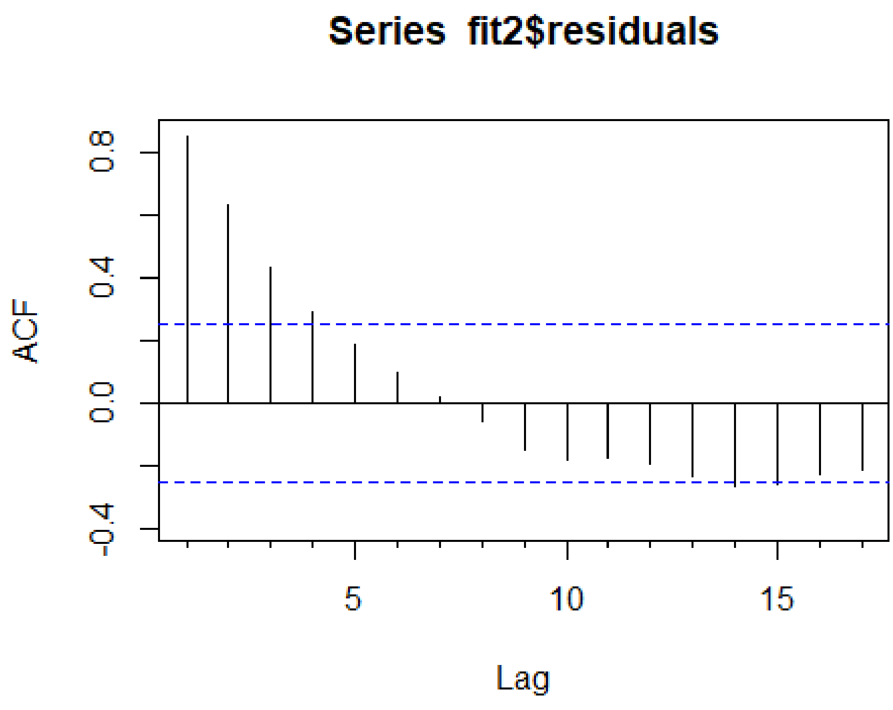
ADP rejects non-stationary and KPSS fail to reject stationarity.

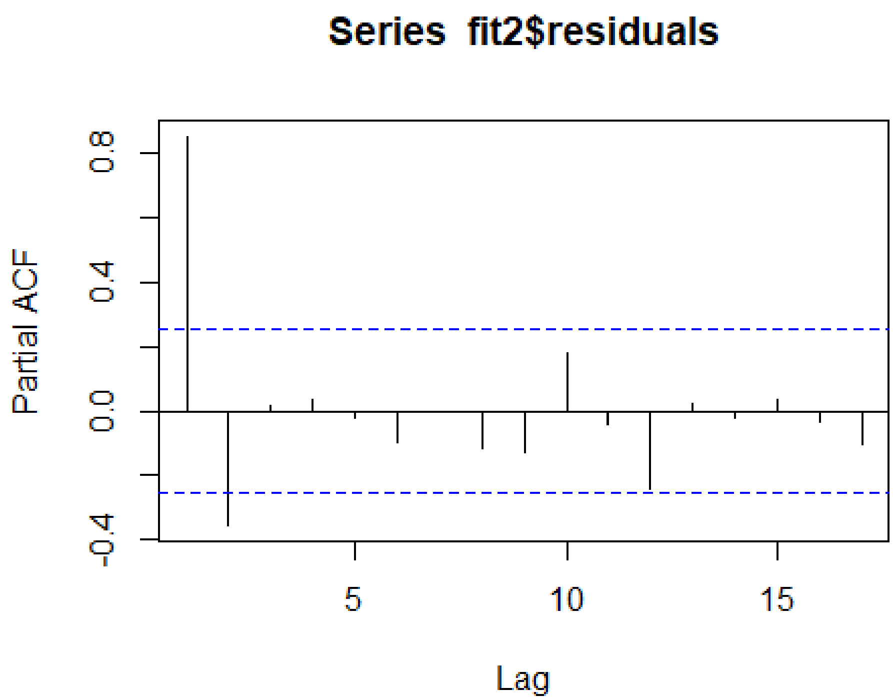
This test simply acknowledge the previous test which suggest the series is trend-stationarity, the OLS resisudual simply implies that if we de-trend(take out the time trend) the series it become stationary.

```
##
## Call:
## lm(formula = GDP_ts ~ time(GDP_ts))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2767.1  -890.0  -131.3   912.0  2749.3
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.262e+06  1.760e+04  -71.69   <2e-16 ***
## time(GDP_ts)  6.523e+02  8.846e+00   73.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1187 on 58 degrees of freedom
## Multiple R-squared:  0.9894, Adjusted R-squared:  0.9893
## F-statistic:  5437 on 1 and 58 DF,  p-value: < 2.2e-16

##
## t test of coefficients:
##
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  -1.2618e+06  1.7600e+04 -71.693 < 2.2e-16 ***
## time(GDP_ts)  6.5226e+02  8.8460e+00  73.736 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


fit2$residuals
2000
0
-2000
0
10
20
30
40
50
60
Index


Series fit2$residuals
ACF
0.8
0.4
0.0
-0.4
5
10
15
Lag



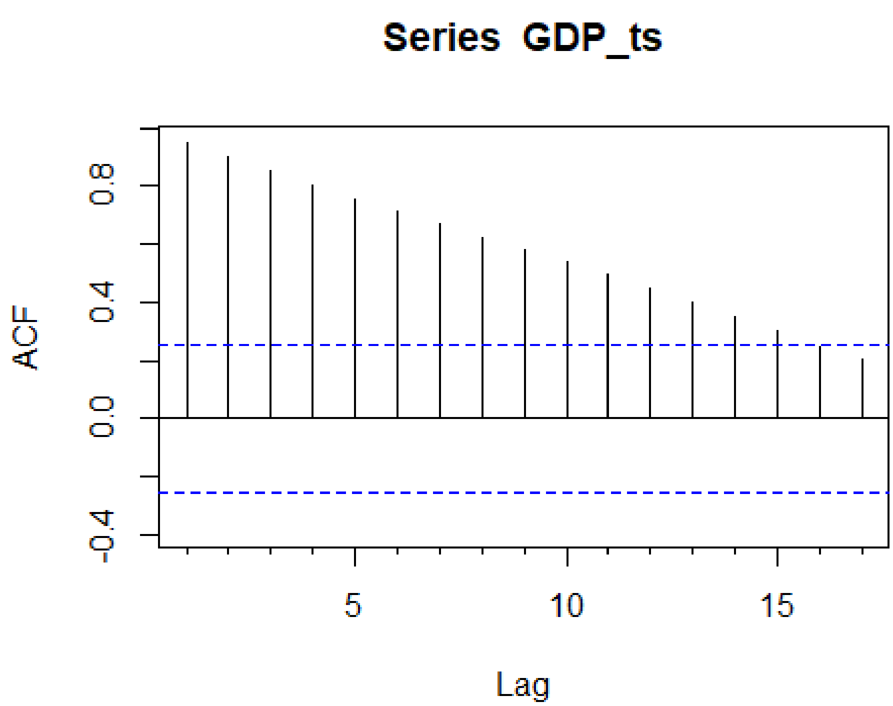

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 o o o o o o o o o o o  o  o  o
## 5 o o o o o o o o o o o  o  o  o
## 6 o x x o o o o o o o o  o  o  o
## 7 o x o o o o o o o o o  o  o  o
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  fit2$residuals
## Dickey-Fuller = -2.3317, Lag order = 3, p-value = 0.4407
## alternative hypothesis: stationary
```

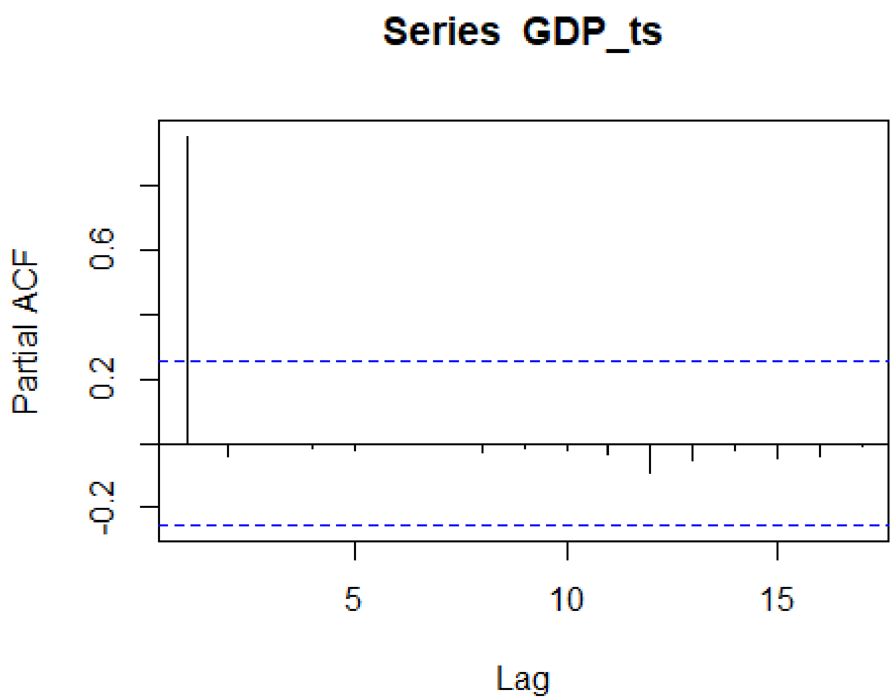
```
##
##  KPSS Test for Level Stationarity
##
## data:  fit2$residuals
## KPSS Level = 0.14504, Truncation lag parameter = 3, p-value = 0.1
```

1d, The original ACF/PACF/EACF of the GDP series suggest an Regression with ARIMA(1,0,1) errors . And the ACF and EACF for the residual also suggest Regression with ARIMA(1,0,1) errors .

```
Acf(GDP_ts)
```




```
pacf(GDP_ts)
```




```
eacf(GDP_ts)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o o o o o o o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o x o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 o x x o o o o o o o o  o  o  o
## 7 o x o o o o o o o o o  o  o  o

# So, let's fit an ARMA(1, 1) process to the residuals
fit4 = Arima(GDP_ts, xreg=time(GDP_ts), order=c(1, 0, 1))
```

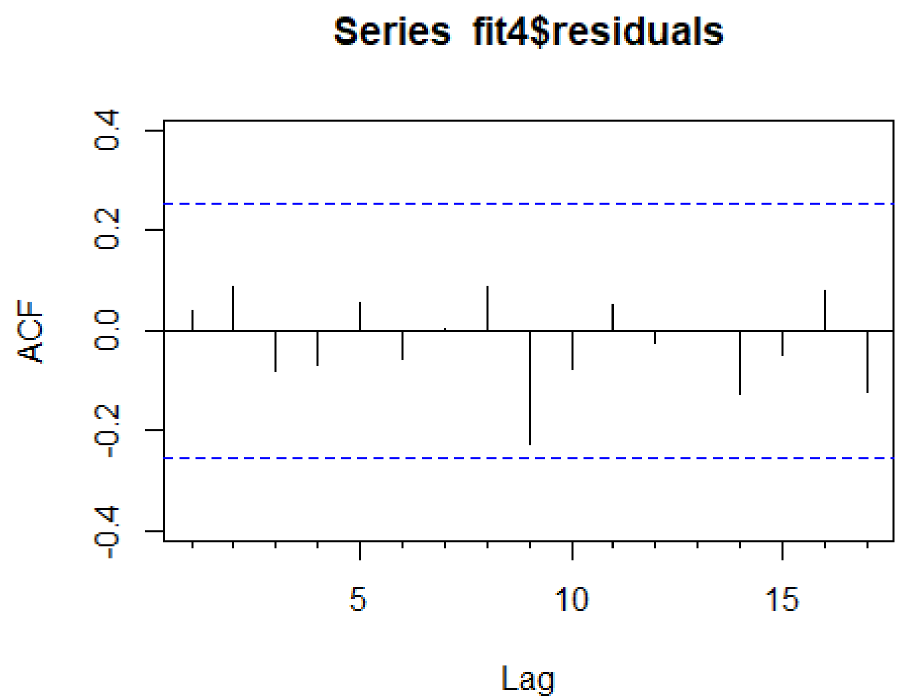
1e,

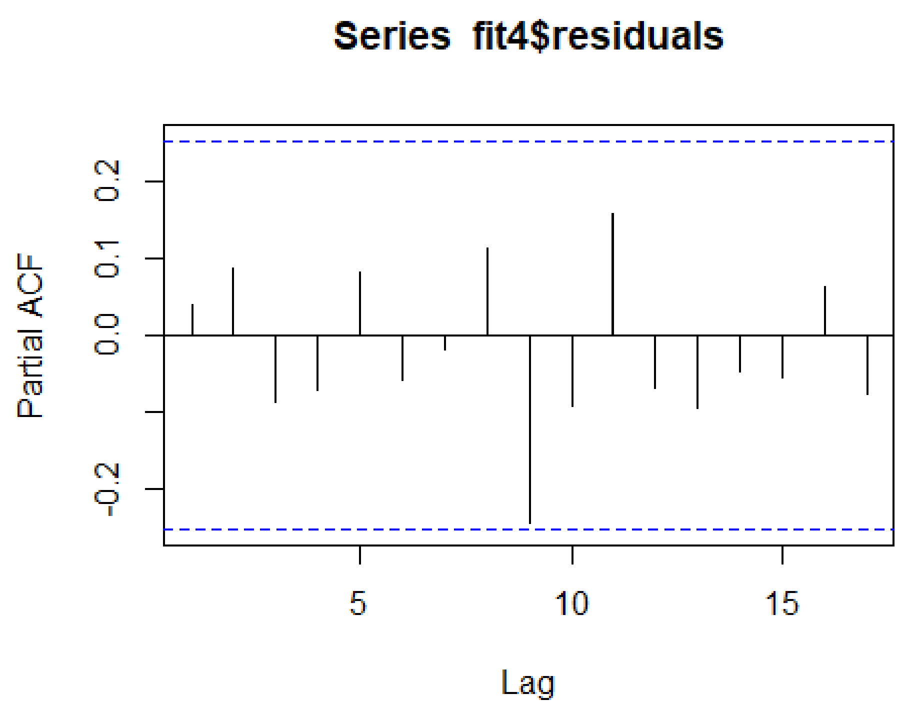
Using Regression with ARIMA(1,0,1) errors, all coeficient are higly significant. The residual ACF/PACF shows that all information have been captured. Box-Ljung test on the residual failed to reject white noise. At this stage model is somewhat a good fit since it passes all test, for optimum evaluatioin we need to make sure model can forcast well.

```
## Series: GDP_ts
## Regression with ARIMA(1,0,1) errors
```

```
##
## Coefficients:
##          ar1     ma1    intercept      xreg
##       0.7665  0.4055  -1256298.00  649.5531
## s.e.  0.0880  0.1231     41815.17   21.0162
##
## sigma^2 = 328240:  log likelihood = -464.91
## AIC=939.83   AICc=940.94   BIC=950.3
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1        7.6655e-01  8.8021e-02   8.7087 < 2.2e-16 ***
## ma1        4.0546e-01  1.2311e-01   3.2936 0.0009893 ***
## intercept -1.2563e+06  4.1815e+04 -30.0441 < 2.2e-16 ***
## xreg       6.4955e+02  2.1016e+01  30.9072 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```






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

```
## data:  fit4$residuals
## X-squared = 42.123, df = 56, p-value = 0.9153
```

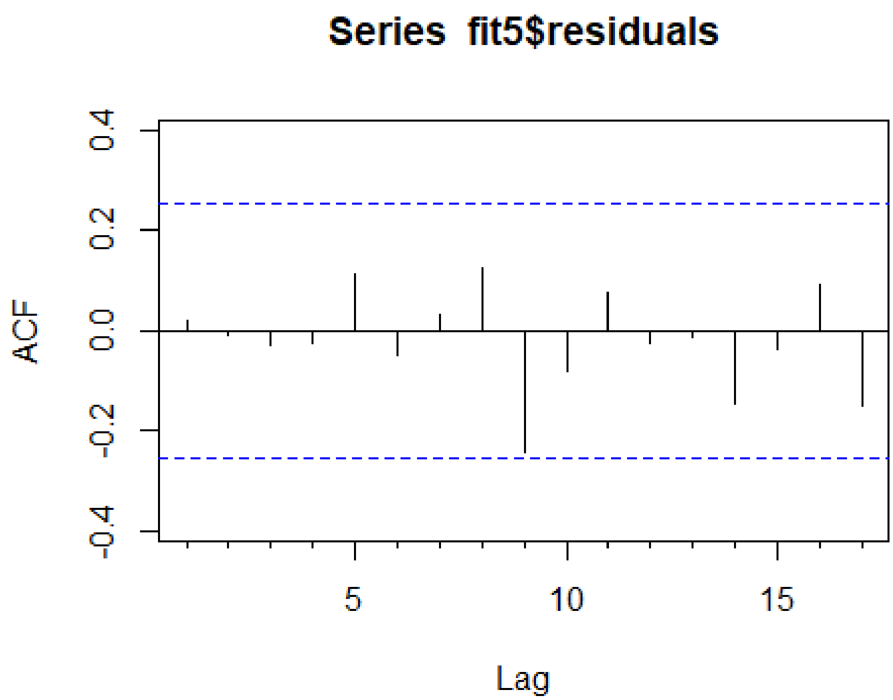
1f, Auto ARIMA modeled the series as ASeries: GDP_ts Regression with ARIMA(2,0,0) errors . This result is different from my model Regression with ARIMA(1,0,1) errors.

Regression with ARIMA(1,0,1) errors “Mean Absolute Percentage error” 0.01370412

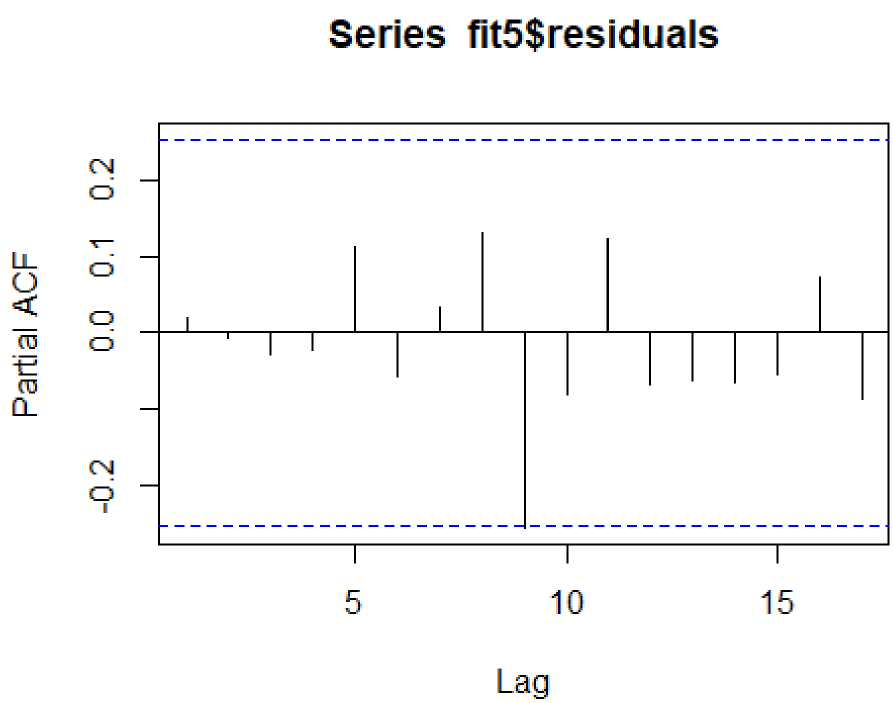
Series: GDP_ts Regression with ARIMA(2,0,0) errors “Mean Absolute Percentage error” 0.01134208

AUTO ARIMA has lower MAPE error, and a lower RSME.

```
# Check BIC auto.arima
fit5 = auto.arima(GDP_ts, xreg=time(GDP_ts), ic="bic")
fit5  # ARMA(1, 1), so a bit simpler, but with higher sigma^2

## Series: GDP_ts
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2   intercept      xreg
##       1.1871  -0.3846  -1258359.7  650.5800
## s.e.  0.1181   0.1176     37250.5   18.7222
##
## sigma^2 = 322956:  log likelihood = -464.45
## AIC=938.9   AICc=940.01   BIC=949.37

Acf(fit5$residuals)
```




```r
pacf(fit5$residuals)
```

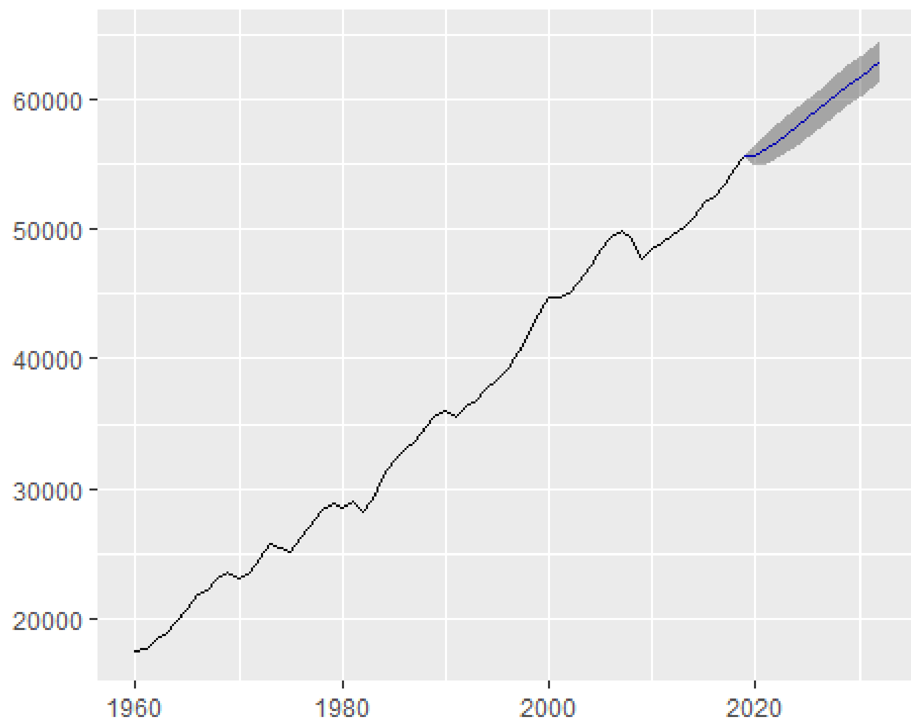



```r
Box.test(fit5$residuals, lag = 10, type = 'Ljung')
```

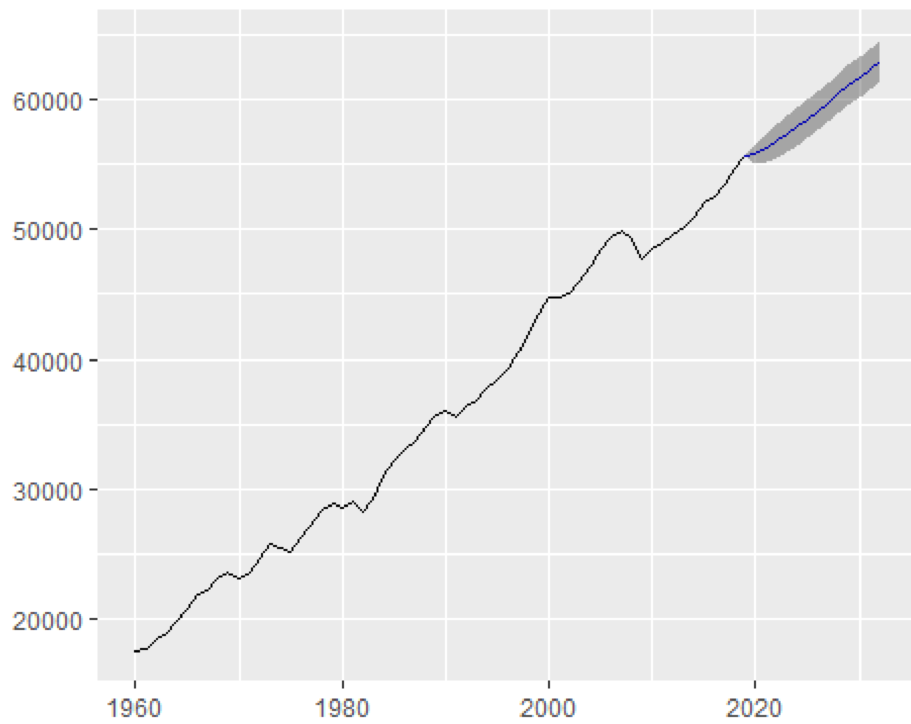
```
##
##  Box-Ljung test
##
## data:  fit5$residuals
## X-squared = 7.0822, df = 10, p-value = 0.7177

## [1] 60

## [1] "RMSE of out-of-sample forecasts"
## [1] 796.0467
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 690.2817
## [1] "Mean Absolute Percentage error"
## [1] 0.01370412
## [1] "Symmetric Mean Absolute Percentage error"
## [1] 0.01376522

## [1] 60

## [1] "RMSE of out-of-sample forecasts"
## [1] 770.516
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 568.3635
## [1] "Mean Absolute Percentage error"
## [1] 0.01134208
## [1] "Symmetric Mean Absolute Percentage error"
## [1] 0.01140233
```

1g, Here the forcast follows the trend line and always reverts back to it, since it is a trend stationary series. GDP over the next 10 year might deviate from the mean but will revert back to the mean.

```
#future= ((time(GDP_ts)+1):(time(GDP_ts)+5))
f= forecast(fit4, xreg = 2019:2031)
autoplot(f, title="Forecast Regression with ARIMA(1,0,1) errors")
```




```
#plot(forecast(fit4, xreg=2014:2029), xlim=c(1960, 2030), ylim=c(10000,
70000))
#abline(lm(GDP_ts ~ time(GDP_ts)))

#future= ((time(GDP_ts)+1):(time(GDP_ts)+5))
f= forecast(fit5, xreg = 2019:2031)
autoplot(f)
```

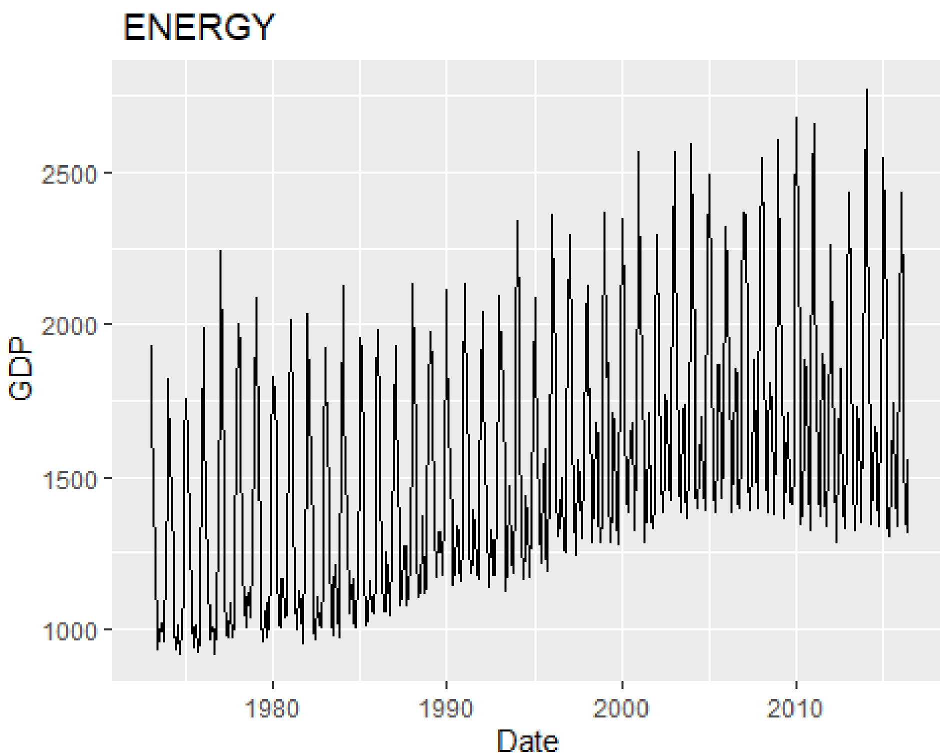



```
#plot(forecast(fit5, xreg=2014:2035),xlim=c(1960, 2035), ylim=c(10000,
70000))
#abline(lm(GDP_ts ~ time(GDP_ts)))
```

2,

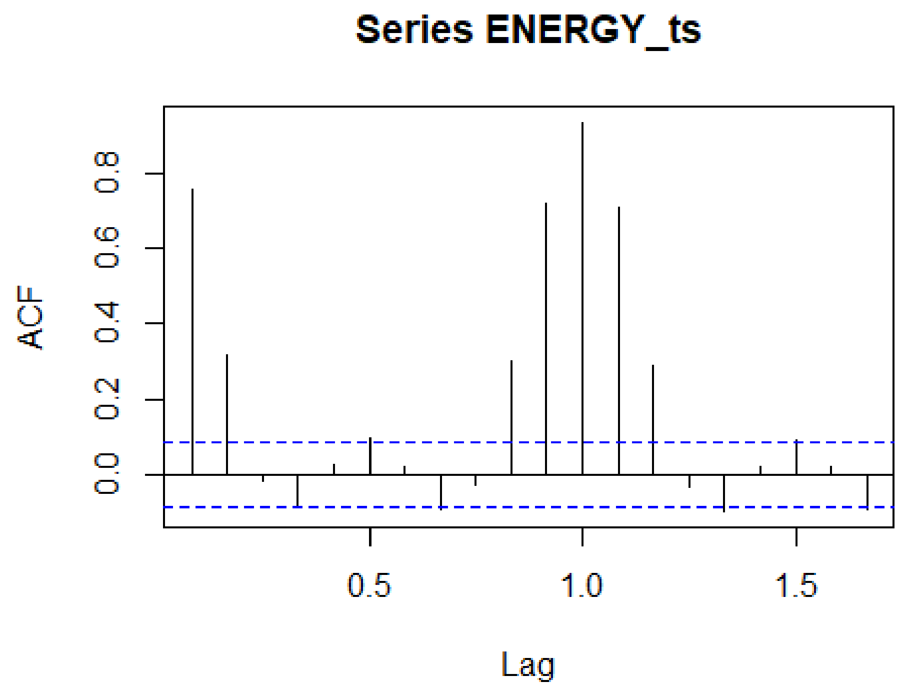
2a, Series exhibits a somewhat seasonality,this seasonality suggests series is non-stationary. No obious trends (linear , exponental) but some sort of cosine trend is present.

```
ggplot(ENERGY_ts,aes(x= time(ENERGY_ts) ,y= ENERGY_ts
,group=1))+geom_line()+labs(title = " ENERGY ")+xlab("Date")+ylab(" GDP")
```



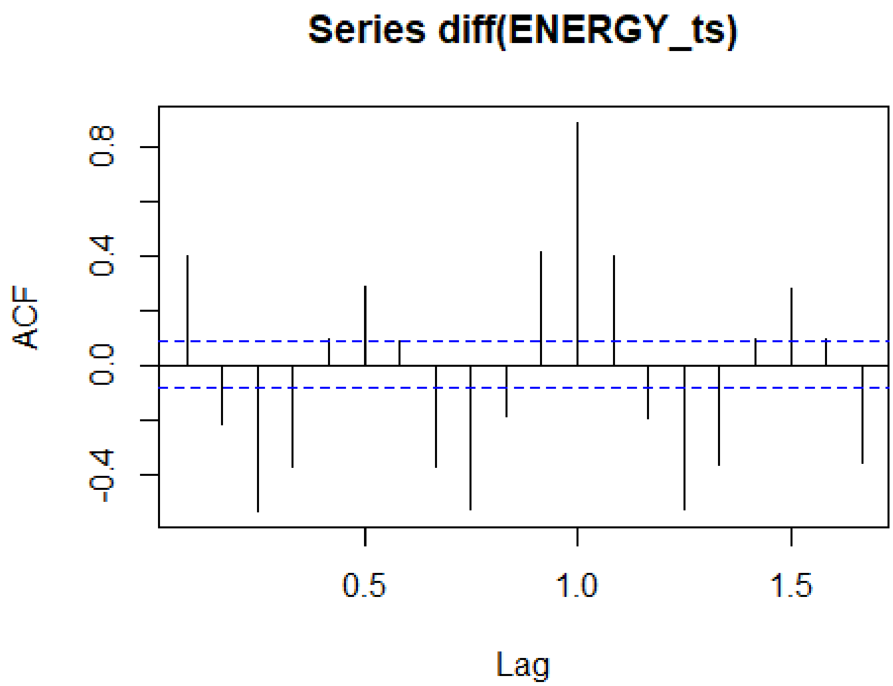
2b, ACF falls slowly to zero/ not exponential fall to zero (non-stationary), the ACF also show repeating patterns and also has peaks at other lags.(non-stationary and seasonality)

```
acf(ENERGY_ts , lag.max = 20)
```



2c, The difference of the energy shows signs of seasonality, it has a peak at the 12th order, line the “airline model”

```
acf(diff(ENERGY_ts) , lag.max = 20)
```



2d, ADF fails to reject non-stationarity with "c" and "nc" but rejects reject trend "ct". This implies no trend in the series but seires is non-stationary.

KPSS rejects level stationary but fail to reject trend stationary.

Series is non-stationary,

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: 0.6075
##   P VALUE:
##     0.81
##
## Description:
##  Mon May 23 12:49:09 2022 by user: soboa

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
```

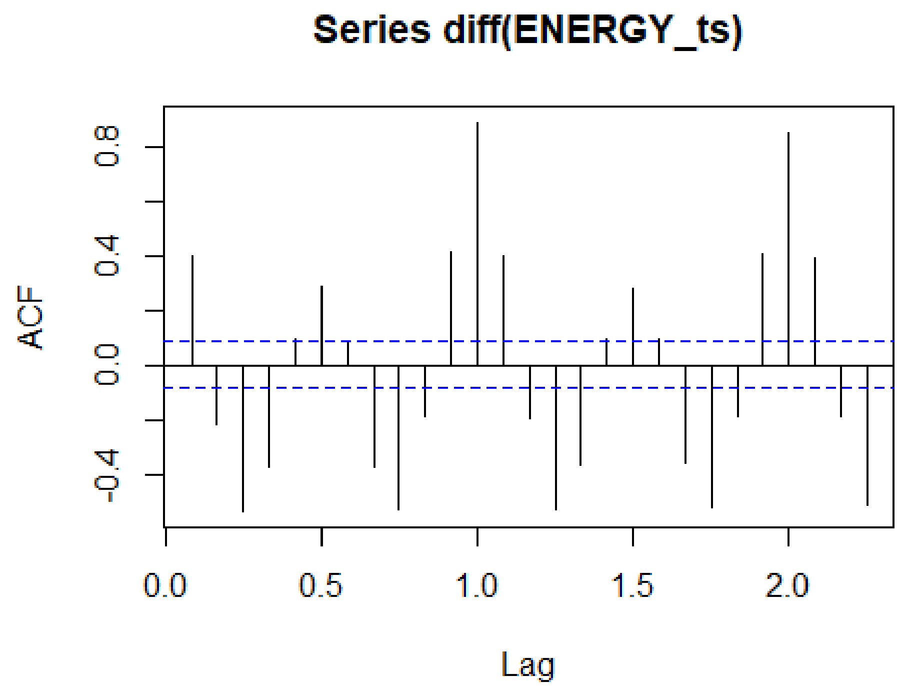
```
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -1.6035
##   P VALUE:
##     0.4613
##
## Description:
##  Mon May 23 12:49:09 2022 by user: soboa

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -3.6538
##   P VALUE:
##     0.02752
##
## Description:
##  Mon May 23 12:49:09 2022 by user: soboa

##
##  KPSS Test for Level Stationarity
##
## data:  ENERGY_ts
## KPSS Level = 4.7141, Truncation lag parameter = 6, p-value = 0.01

##
##  KPSS Test for Trend Stationarity
##
## data:  ENERGY_ts
## KPSS Trend = 0.10894, Truncation lag parameter = 6, p-value = 0.1
```

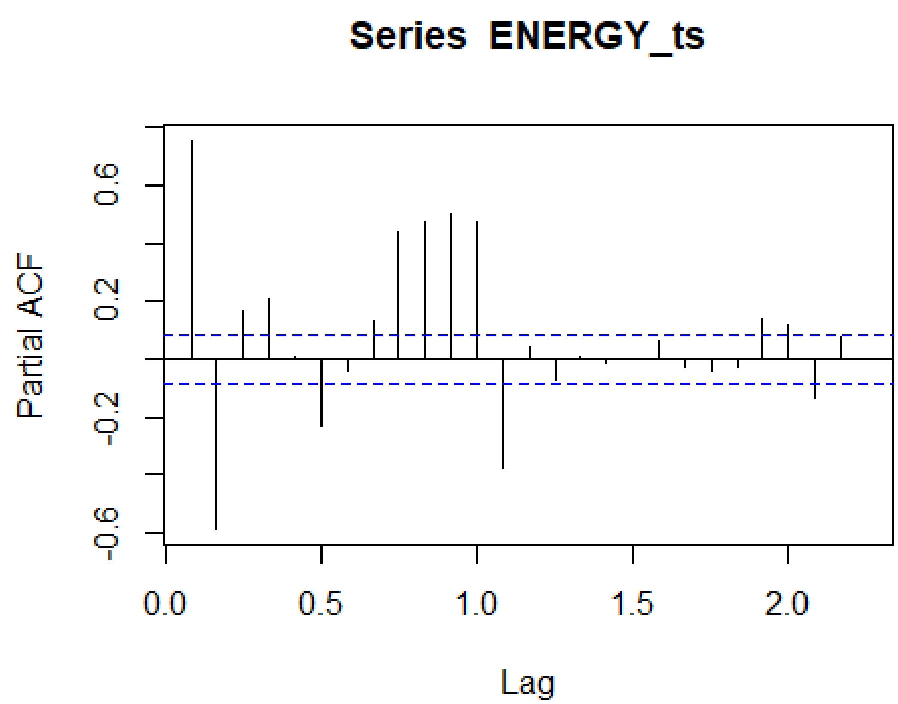
2e,

My model = ARIMA(1,1,2)(0,1,1)[12]

Auto Arima = ARIMA(1,0,0)(0,1,1)[12] with drift

My model would be fit1 it has a significant AIC , BIC and sigma^2 value. The residual analysis for my model fails to reject white noise with p-value 0.67 which is high than auto arima's 0.14. The acf of my residual also shows more information is been catured by my model than AUTO ARIMA. At lag-20 auto arima residual rejects white noise.



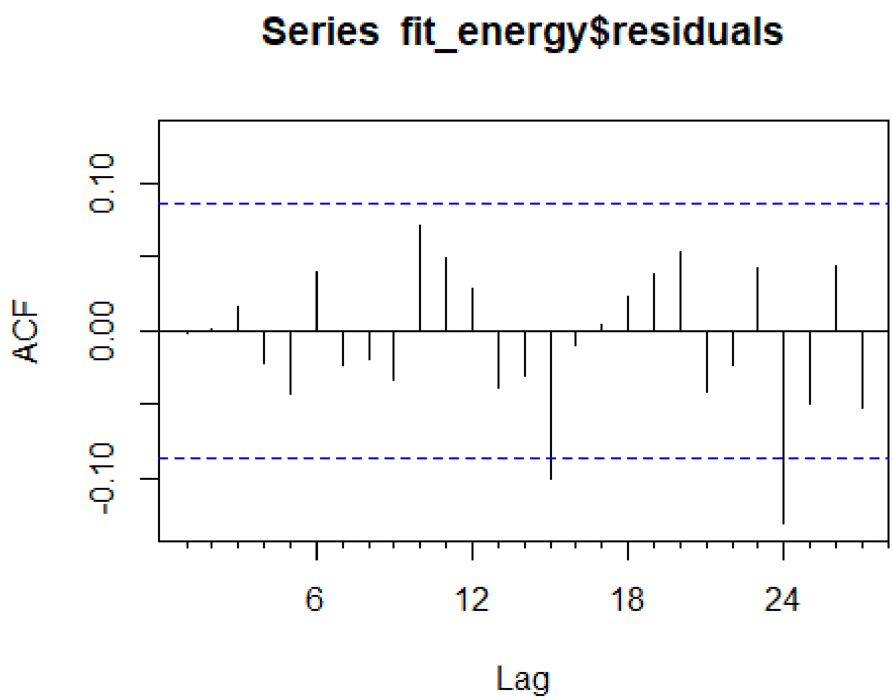


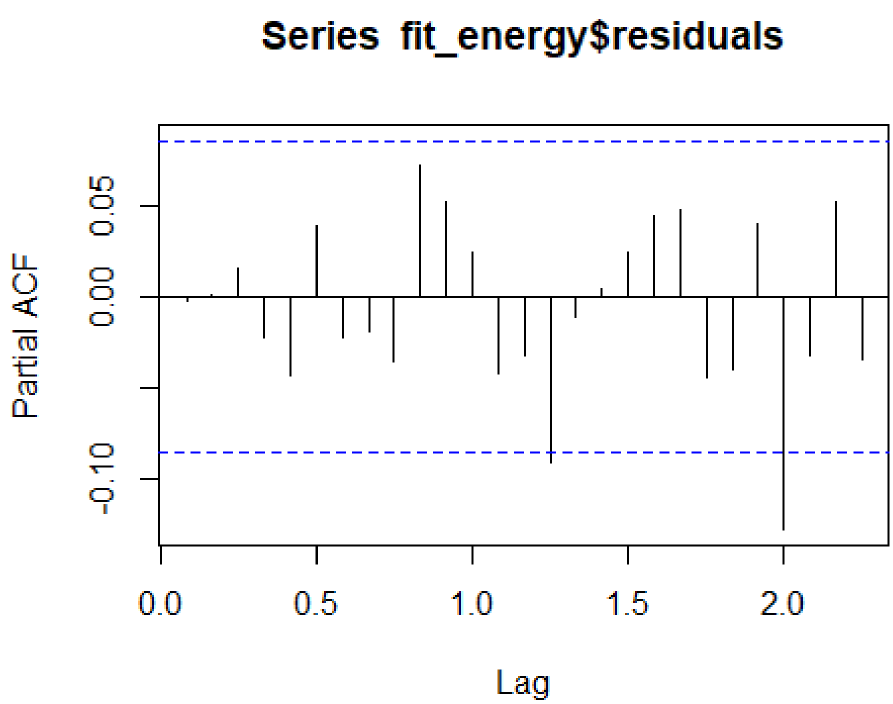

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x o o o x o x o x x  x  x  x
## 1 x x x o o x o x o x x  x  x  x
```

```
## 2 x x x o o o o o o x x  x  x  x
## 3 x x x o o o o o o x o  x  x  x
## 4 o x x x o o o o o o o  x  x  x
## 5 o x x o o o o o o o o  x  x  x
## 6 x x x x x o o o o o o  x  x  x
## 7 x x x x x o o o o o o  x  x  o
```

```
## Series: ENERGY_ts
## ARIMA(1,1,2)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1      ma2     sma1
##       0.2928  -0.7329  -0.2195  -0.7959
## s.e.  0.0941   0.0964   0.0887   0.0284
##
## sigma^2 = 6990:  log likelihood = -2980.74
## AIC=5971.47   AICc=5971.59   BIC=5992.64
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.292760   0.094106   3.1109  0.001865 **
## ma1  -0.732895   0.096445  -7.5991 2.982e-14 ***
## ma2  -0.219532   0.088688  -2.4753  0.013311 *
## sma1 -0.795915   0.028372 -28.0532 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

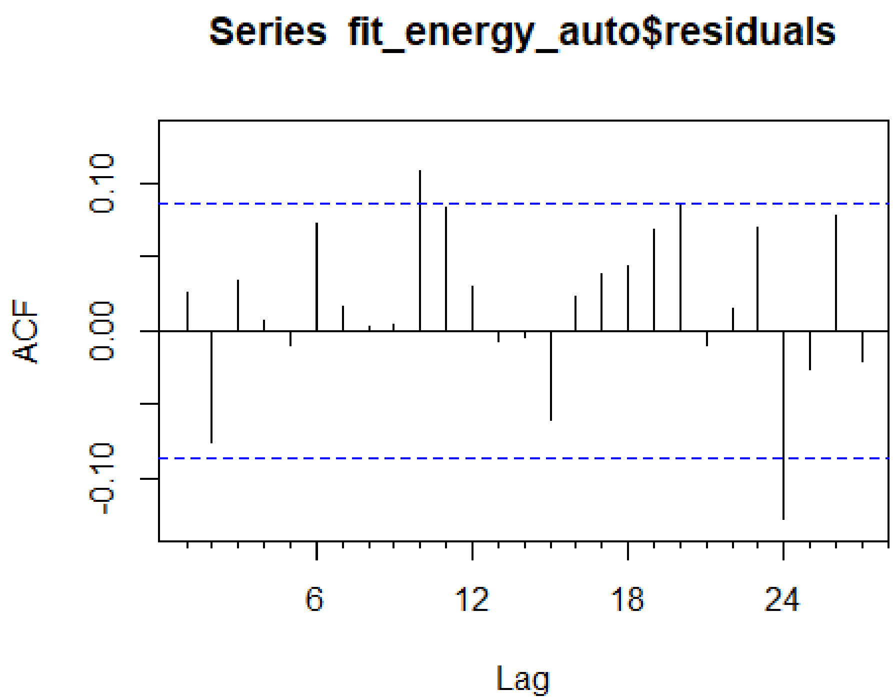


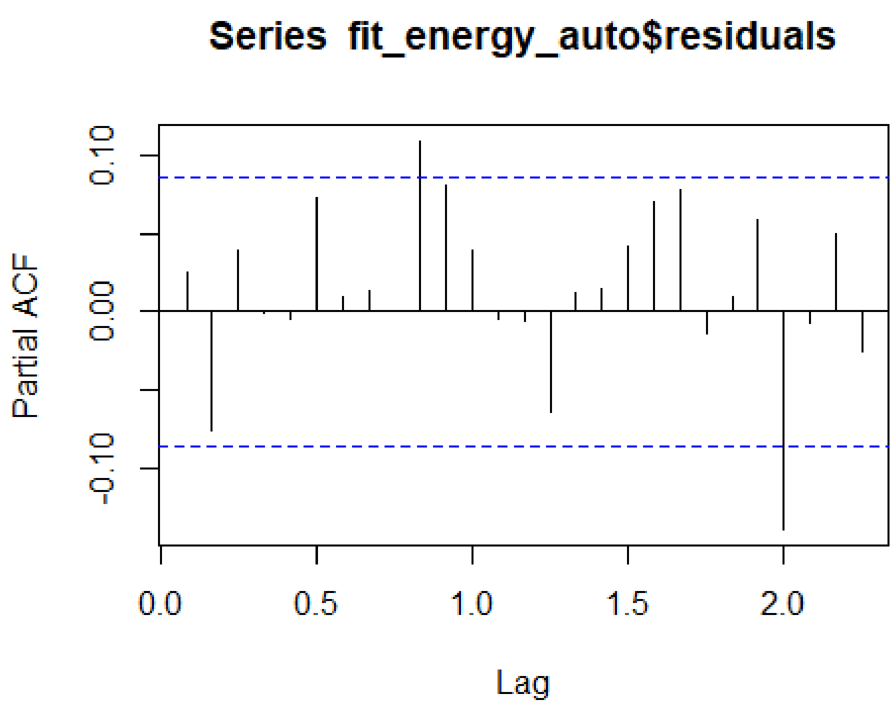



```
##
##  Box-Ljung test
##
```

```
## data:  fit_energy$residuals
## X-squared = 16.954, df = 20, p-value = 0.656
```

```
## Series: ENERGY_ts
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1   drift
##       0.5406  -0.7579  1.0695
## s.e.  0.0380   0.0320  0.1787
##
## sigma^2 = 7173:  log likelihood = -2991.38
## AIC=5990.75   AICc=5990.83   BIC=6007.69
```






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

```
## data:  fit_energy_auto$residuals
## X-squared = 27.818, df = 20, p-value = 0.1138
```

2f, After taling the lm of energy on time, observing the residual acf shows an airline seaonality at the 12th term also the diff in residual also shows the same pattern. The ACF also suggest a MA(1) process.

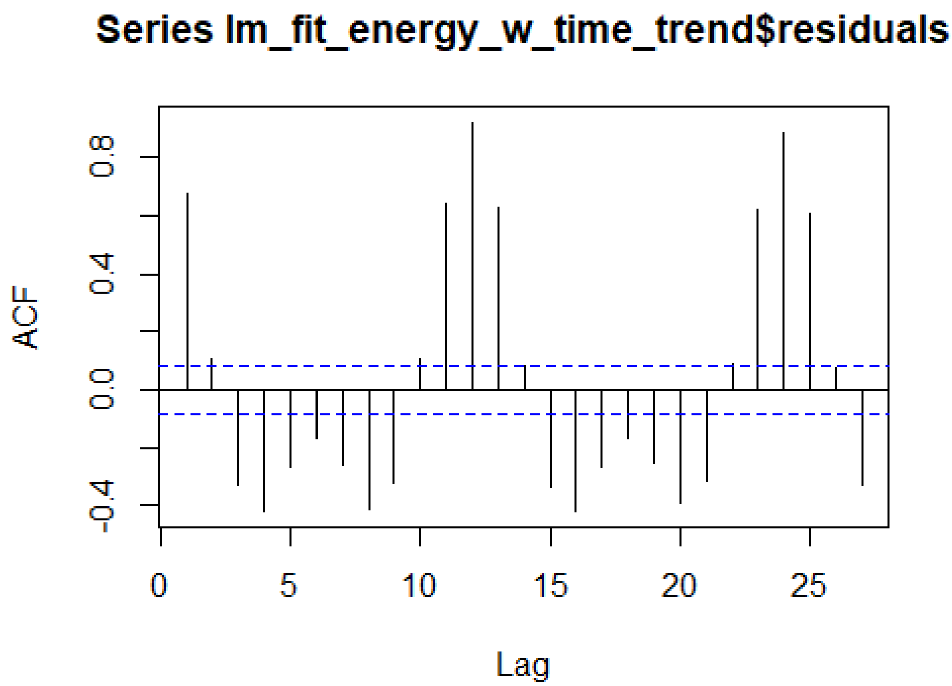
The PACF of the residual suggest an AR(2), there seem to be a peak at the 12th lag in the residual PACF.

My model = Regression with ARIMA(1,0,1)(0,1,1)[12] errors

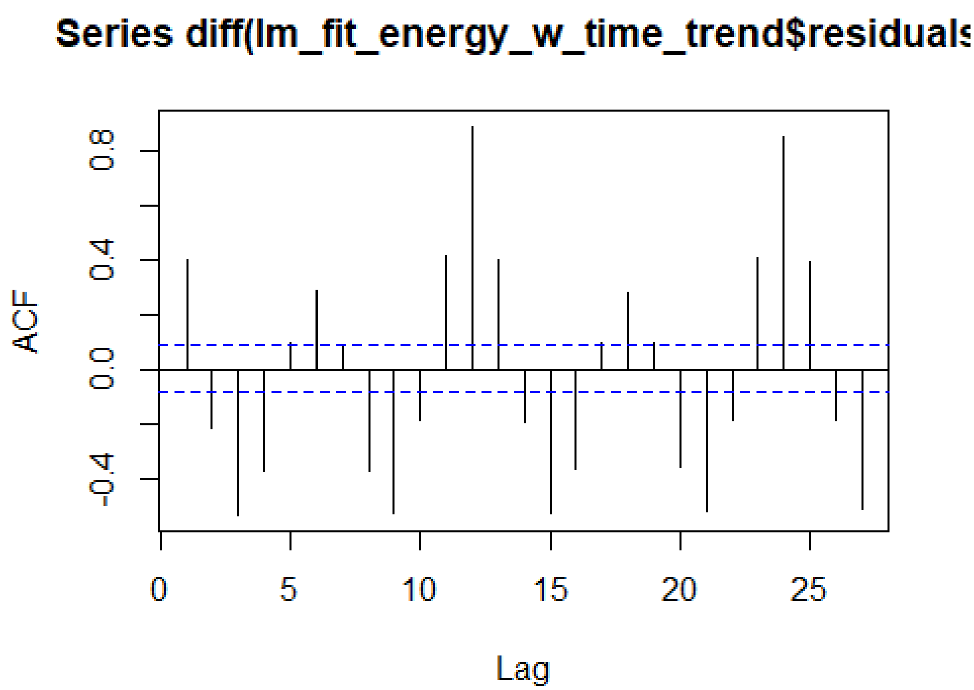
Auto ARIMA = Regression with ARIMA(2,0,2)(0,0,2)[12] errors

My fit would be fit2 , auto arima has an high AIC and BIC value but fails to reject rejects white noise in the residual.

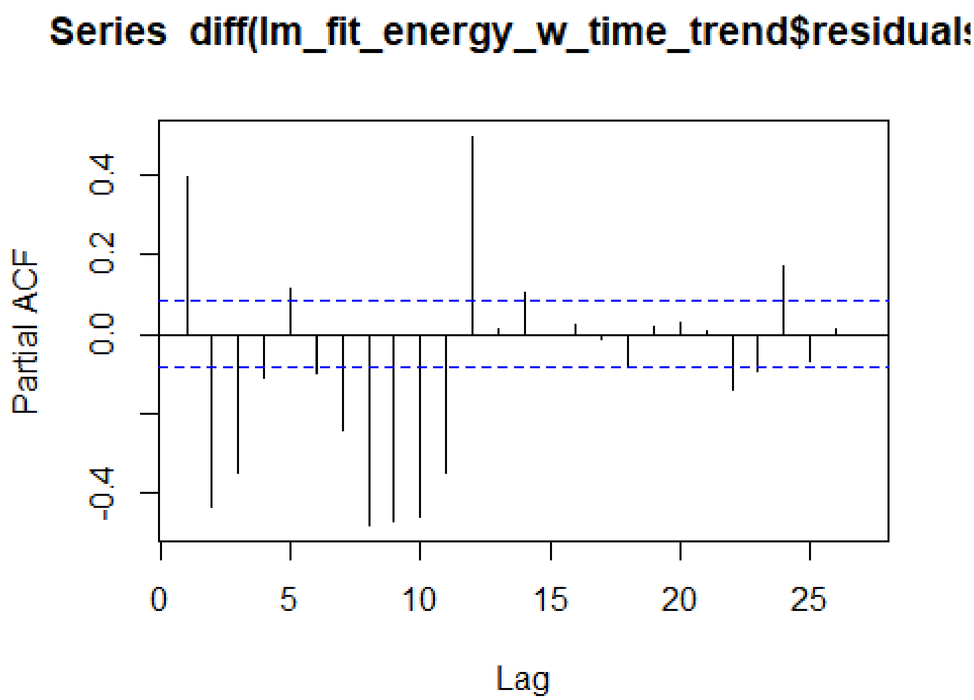
```
##
## Call:
## lm(formula = ENERGY_ts ~ time(ENERGY_ts))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -536.5 -273.6 -132.3  260.9  974.1
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -28198.115   2402.997  -11.73   <2e-16 ***
## time(ENERGY_ts)     14.903      1.205   12.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 345.6 on 520 degrees of freedom
## Multiple R-squared:  0.2274, Adjusted R-squared:  0.2259
## F-statistic:   153 on 1 and 520 DF,  p-value: < 2.2e-16
```

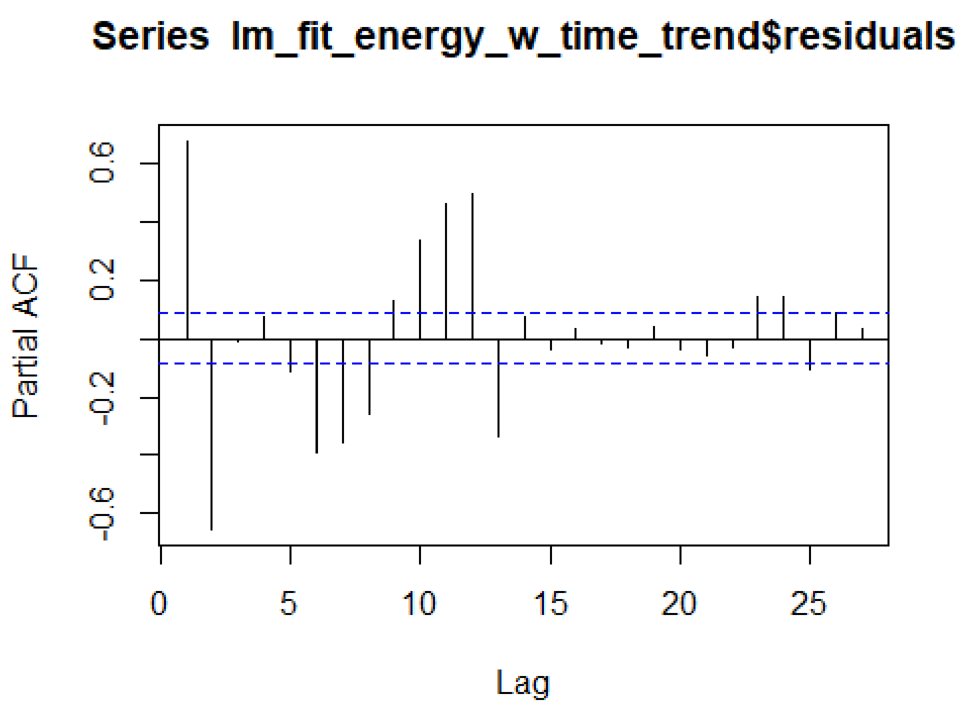


Series lm_fit_energy_w_time_trend$residuals
ACF
-0.4
0.0
0.4
0.8
0
5
10
15
20
25
Lag



Series diff(lm_fit_energy_w_time_trend$residuals
ACF
-0.4
0.0
0.4
0.8
0
5
10
15
20
25
Lag






```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  o
## 1 x x x x o x o x x o x  x  x  x
```

```
## 2 o x x x o x o x x o o  x  x  x
## 3 o o o x x x o x x x o  x  x  o
## 4 x x o x o x o x o o o  x  x  x
## 5 x x x x o o o x o o o  x  x  x
## 6 x x x x x o o o o o x  x  x  x
## 7 x x x x o o o o o o o  x  x  o
```

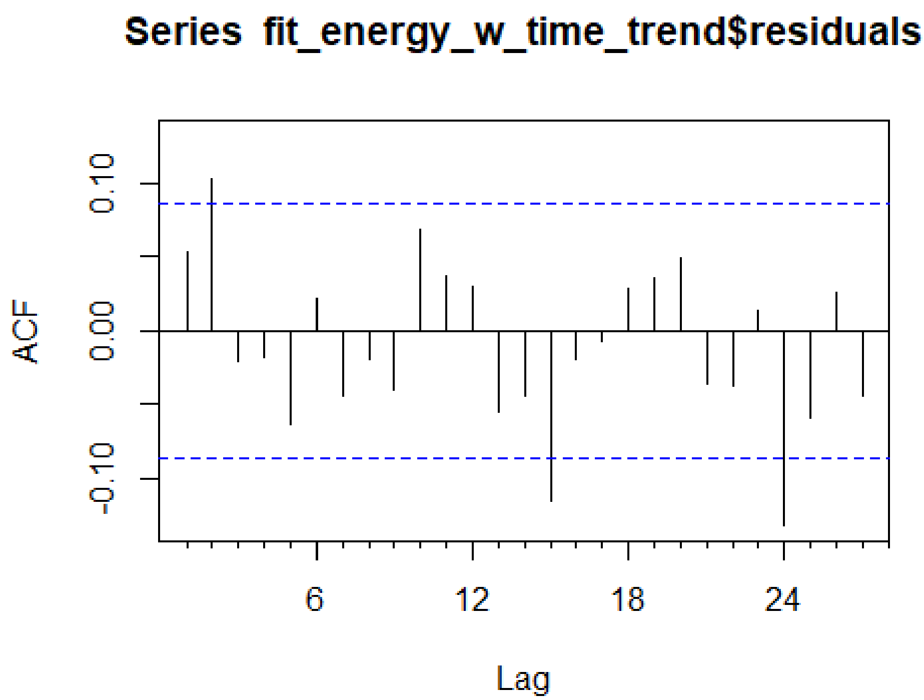
```
fit_energy_w_time_trend = Arima(ENERGY_ts, xreg=time(ENERGY_ts), order=c(1,
0, 2),  seasonal = list (order= c(0,1,1), period=12))
fit_energy_w_time_trend
```

```
## Series: ENERGY_ts
## Regression with ARIMA(1,0,2)(0,1,1)[12] errors
##
## Coefficients:
##          ar1      ma1      ma2     sma1     xreg
##       0.9861  -0.4773  -0.4208  -0.7972  10.5881
## s.e.  0.0165   0.0447   0.0429   0.0283   5.9195
##
## sigma^2 = 7101:  log likelihood = -2987.92
## AIC=5987.85   AICc=5988.02   BIC=6013.26
```

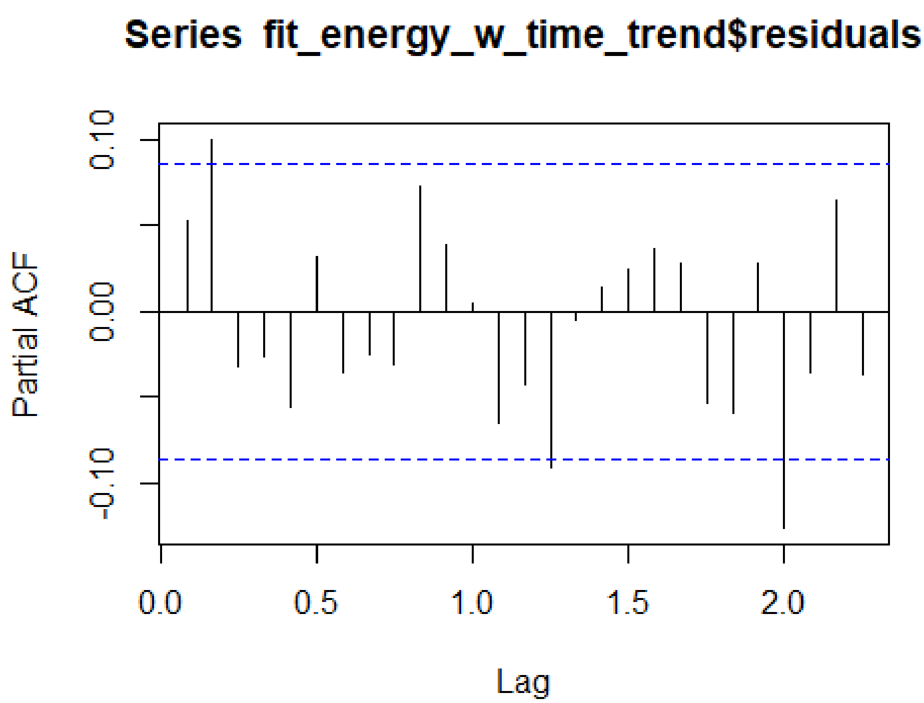
```
coeftest(fit_energy_w_time_trend)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1   0.986109   0.016516  59.7046  < 2e-16 ***
## ma1  -0.477262   0.044677 -10.6825  < 2e-16 ***
## ma2  -0.420751   0.042888  -9.8104  < 2e-16 ***
## sma1 -0.797221   0.028274 -28.1961  < 2e-16 ***
## xreg 10.588097   5.919542   1.7887  0.07367 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Acf(fit_energy_w_time_trend$residuals)    # Looks like descending behaviour
... maybe not much MA
```




```r
pacf(fit_energy_w_time_trend$residuals)    # AR(3) ?
```



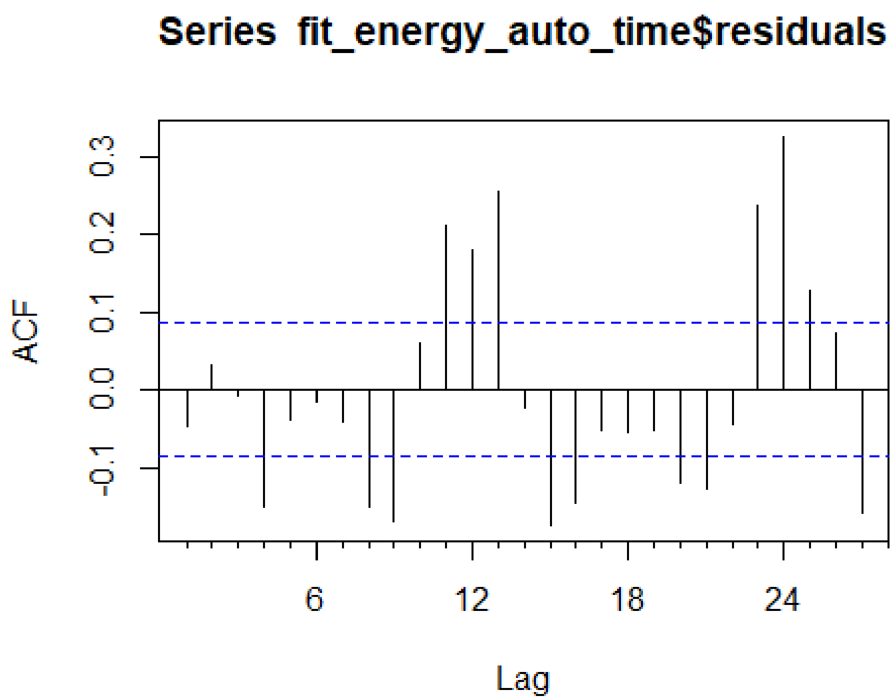

```r
Box.test(fit_energy_w_time_trend$residuals, lag = 12, type = 'Ljung')
```

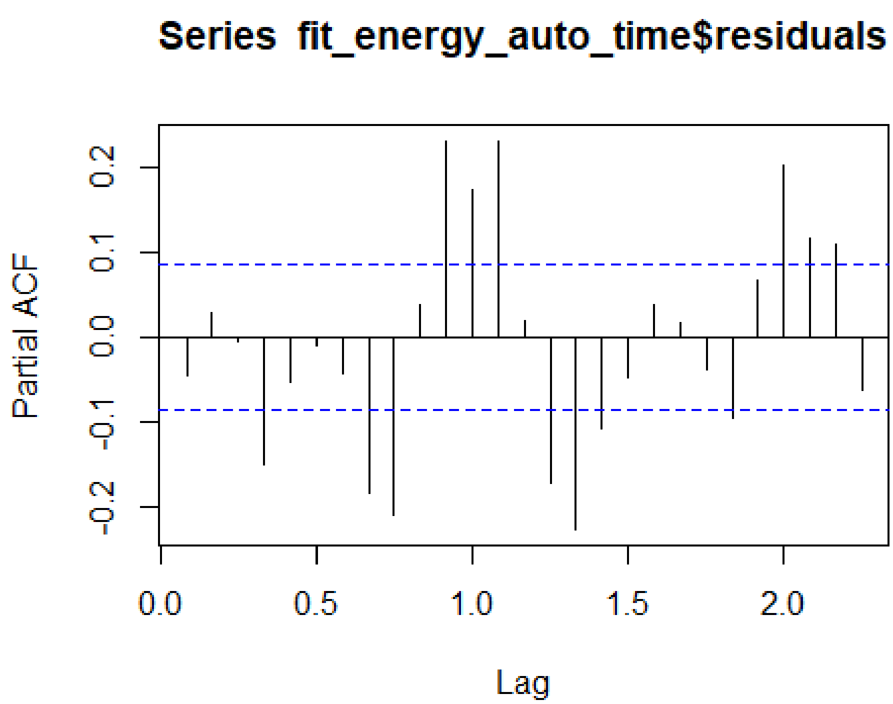
```
##
##  Box-Ljung test
##
## data:  fit_energy_w_time_trend$residuals
## X-squared = 15.415, df = 12, p-value = 0.2195
```

```
fit2_energy = fit_energy_w_time_trend
```

```
## Series: ENERGY_ts
## Regression with ARIMA(2,0,0)(0,0,2)[12] errors
##
## Coefficients:
##          ar1      ar2    sma1    sma2   intercept     xreg
##       0.9089  -0.4503  0.6923  0.3792  -26994.587  14.2987
## s.e.  0.0436   0.0429  0.0508  0.0445    3400.852   1.7049
##
## sigma^2 = 18161:  log likelihood = -3301.37
## AIC=6616.74   AICc=6616.96   BIC=6646.55
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1        9.0887e-01  4.3642e-02  20.8254 < 2.2e-16 ***
## ar2       -4.5026e-01  4.2942e-02 -10.4853 < 2.2e-16 ***
## sma1       6.9227e-01  5.0784e-02  13.6317 < 2.2e-16 ***
## sma2       3.7921e-01  4.4484e-02   8.5247 < 2.2e-16 ***
## intercept -2.6995e+04  3.4009e+03  -7.9376 2.061e-15 ***
## xreg       1.4299e+01  1.7049e+00   8.3868 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```






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

```
## data:  fit_energy_auto_time$residuals
## X-squared = 159.61, df = 20, p-value < 2.2e-16
```

```
rtest_= 0.10*length(ENERGY_ts)
rtrain_= 0.90*length(ENERGY_ts)

pm1_ = backtest(fit1_energy,ENERGY_ts, orig =rtest_,h=1)
```

```
## [1] "RMSE of out-of-sample forecasts"
## [1] 106.7
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 72.97478
## [1] "Mean Absolute Percentage error"
## [1] NaN
## [1] "Symmetric Mean Absolute Percentage error"
## [1] NaN
```

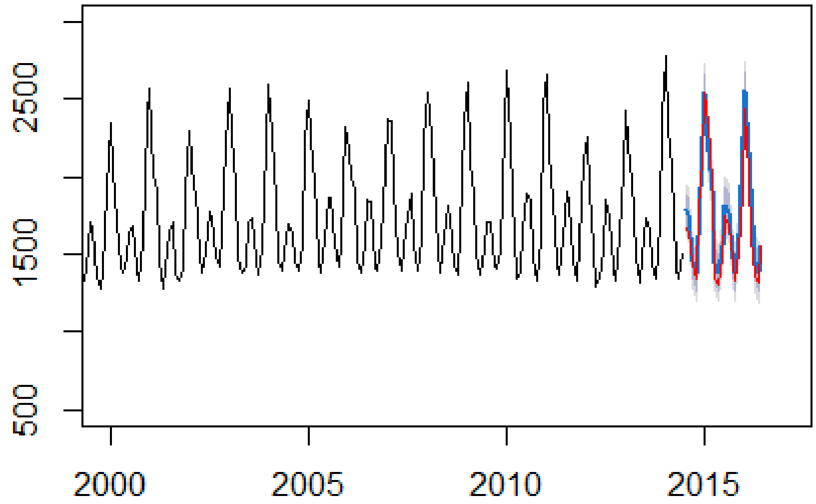
```
pm2_ = backtest(fit2_energy, ENERGY_ts,orig =rtest_ ,h=1)
```

```
## [1] "RMSE of out-of-sample forecasts"
## [1] 107.7378
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 74.47705
## [1] "Mean Absolute Percentage error"
## [1] NaN
## [1] "Symmetric Mean Absolute Percentage error"
## [1] NaN
```

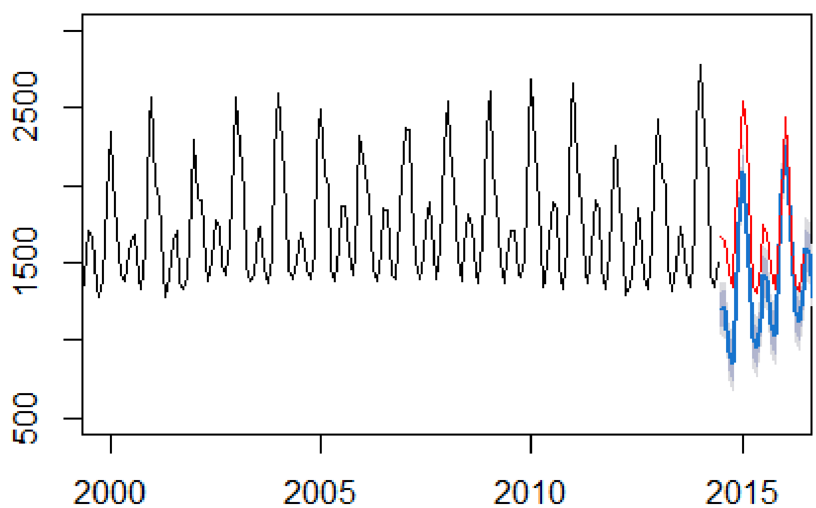
2h, fir1= ARIMA(1,0,2)(0,1,1)[12], shows more accuracy in forecasting thank fit2= Regression with ARIMA(1,0,2)(0,1,1)[12] errors. From bellow we see that forecastig on 24 period matchs well in fit 1 than fit2. In fir 2 forecast (blue) follow the right pattern as original(red) but shift below in range.

```
#length(ENERGY_ts)

rtrain= subset(ENERGY_ts , end = 498)
rtest= subset(ENERGY_ts , start = 499)
```

Forecasts from ARIMA(1,1,2)(0,1,1)[12]



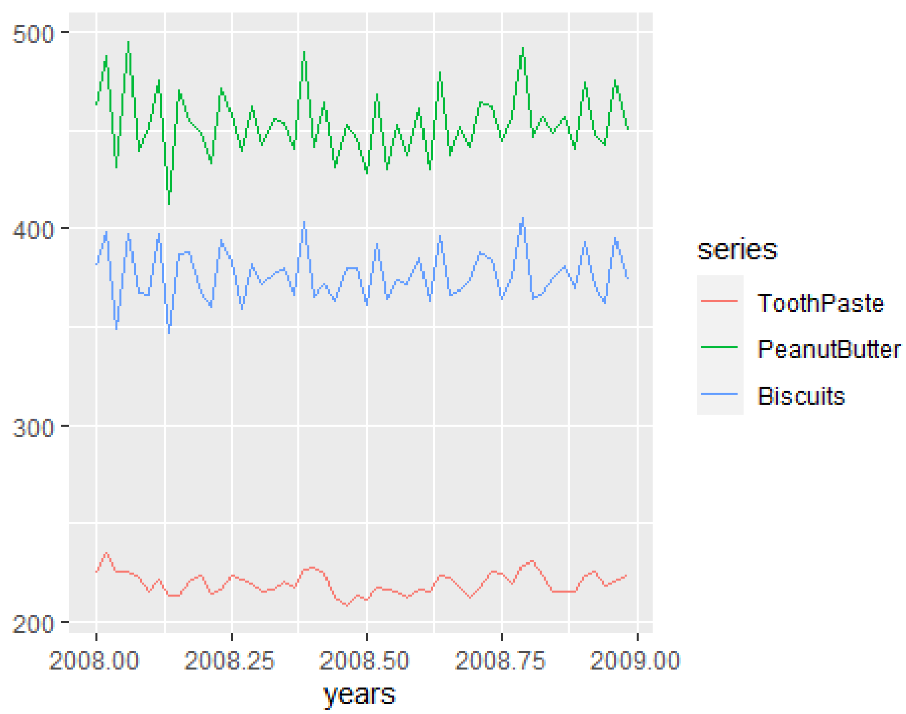
recasts from Regression with ARIMA(1,0,2)(0,1,1)[12]



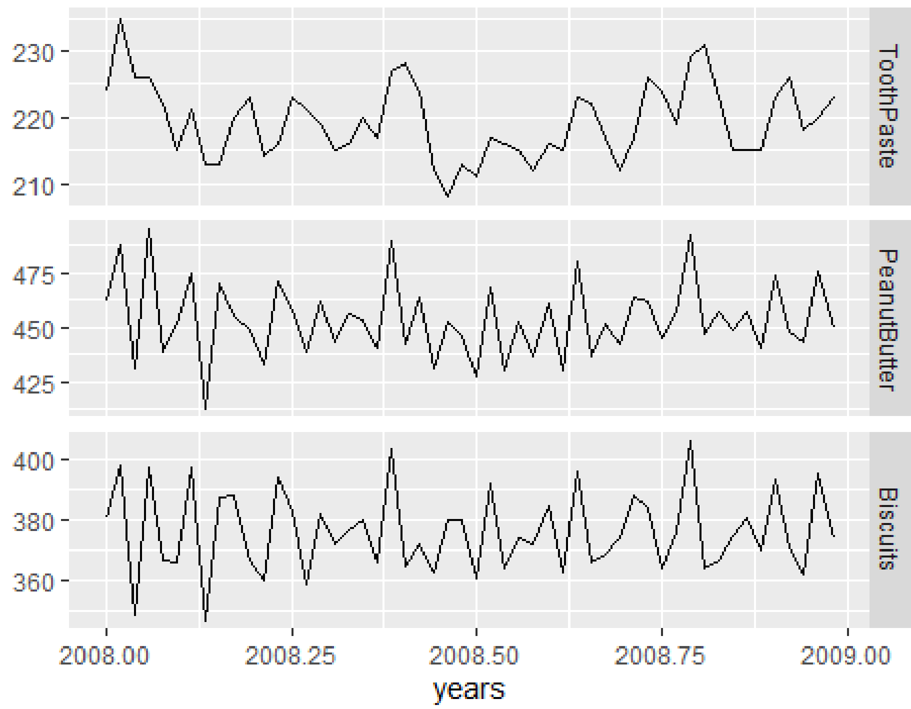
4

4a, There is a clear relationship between biscuits and peanutbutter. They all seem to follow the same trend.

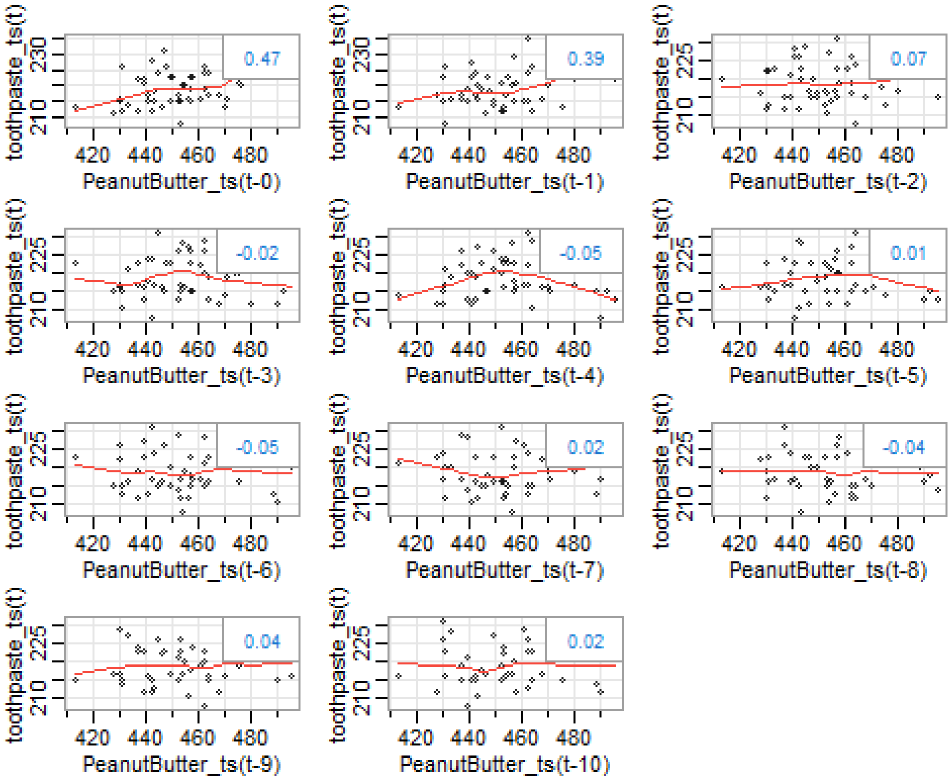
```
autoplot(groceries[, 2:4]) + xlab("years")+ ylab("")
```

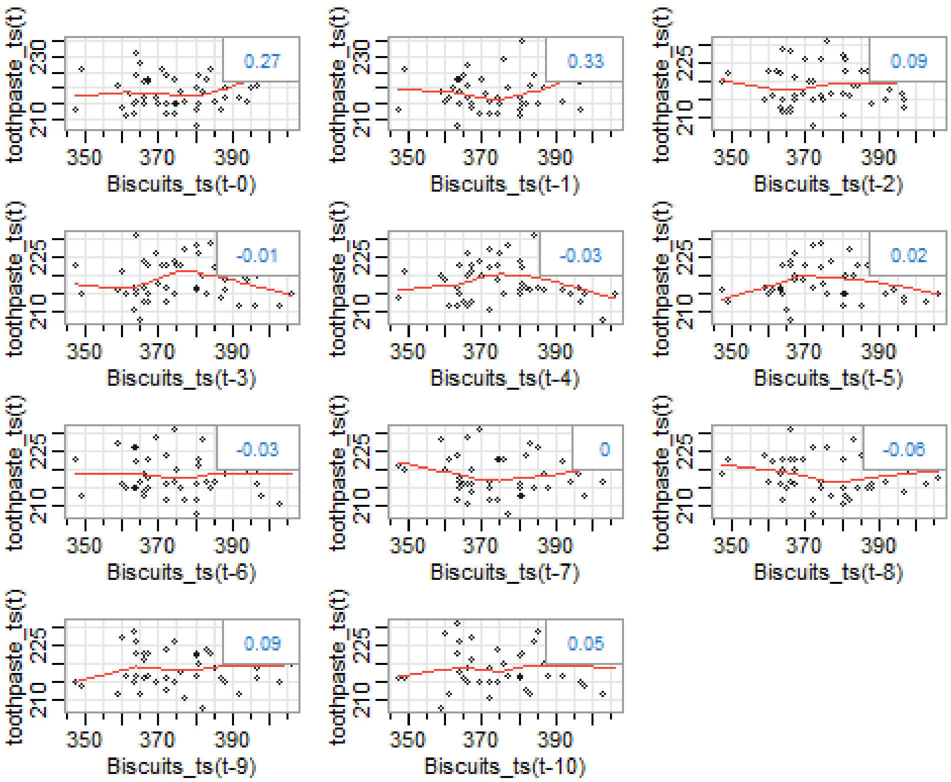



```
autoplot(groceries[, 2:4], facets = T) + xlab("years")+ ylab("")
```


230
220
210
ToothPaste
475
450
425
PeanutButter
400
380
360
Biscuits
2008.00
2008.25
2008.50
2008.75
2009.00
years

4b, for toothpaste and peanut butter has a high positive corelation at lag-2 but also has a higher correlation at lag-0 which means no lag is needed, for toothpaste and biscuit there is a high negative correlation at lag-1



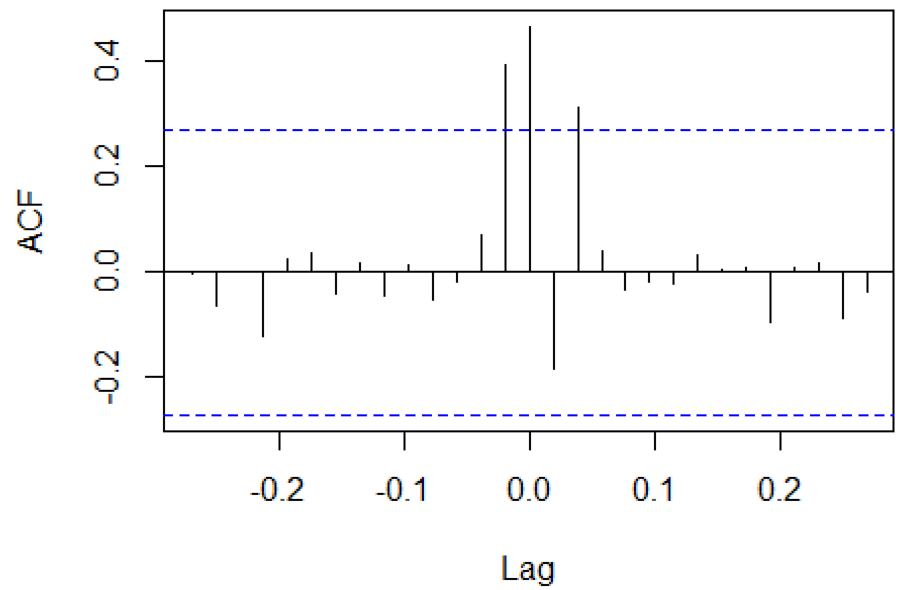


toothpaste_ts(t)
Biscuits_ts(t-0)
0.27
Biscuits_ts(t-1)
0.33
Biscuits_ts(t-2)
0.09
Biscuits_ts(t-3)
-0.01
Biscuits_ts(t-4)
-0.03
Biscuits_ts(t-5)
0.02
Biscuits_ts(t-6)
-0.03
Biscuits_ts(t-7)
0
Biscuits_ts(t-8)
-0.06
Biscuits_ts(t-9)
0.09
Biscuits_ts(t-10)
0.05
350
370
390
210
225
230

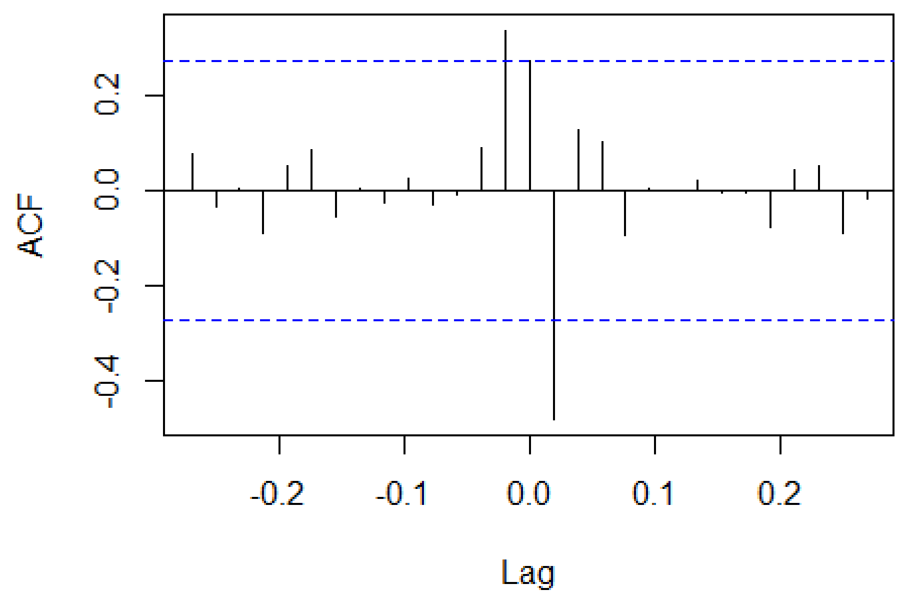
for toothpaste_ts and PeanutButter_ts a lag-0 would compute the higest information also at lag-(-1) and lag-2, while for toothpaste and biscuit a lag-1 would help compute the the hightest information of relationships between toothpaste and biscuit.

The lags affect the regression beacuse we lag them based on the correlation.

PeanutButter_ts & toothpaste_ts



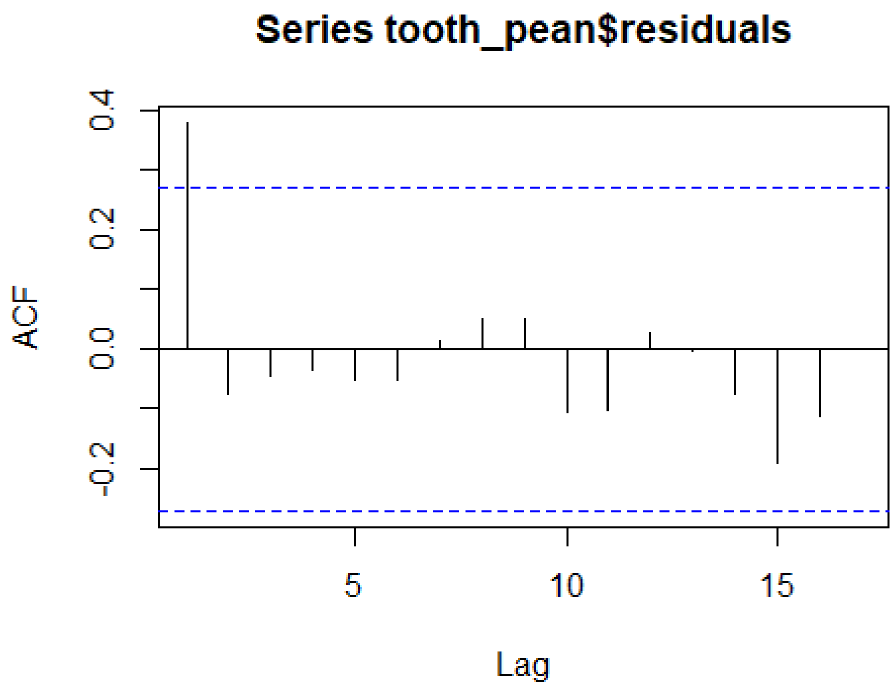
Biscuits_ts & toothpaste_ts



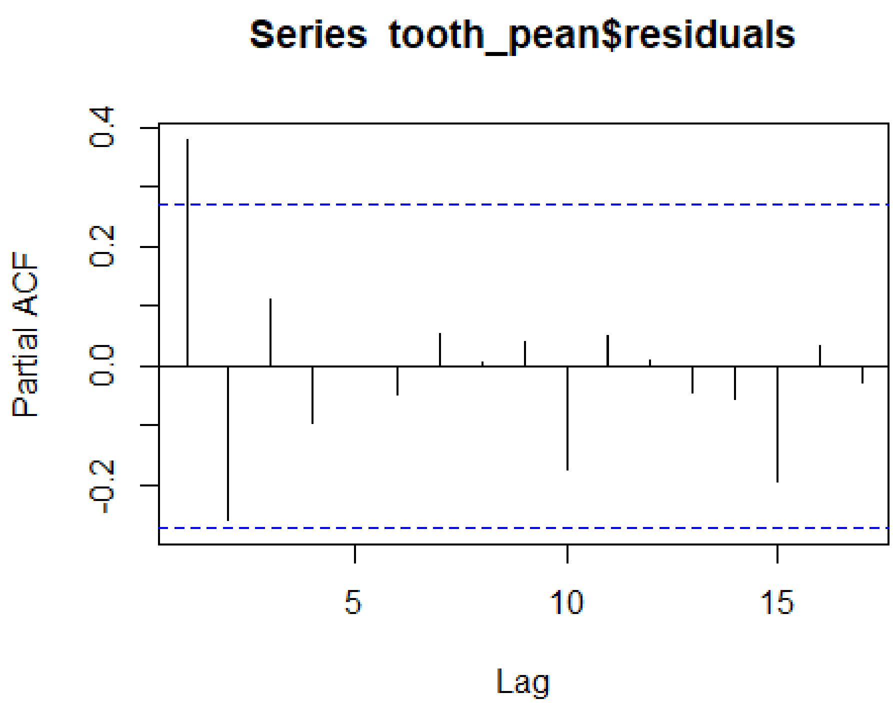
4c,

$$V_t = 0.1 + 0.42_{vt-2} + a_t$$

```
library(dynlm)

## Warning: package 'dynlm' was built under R version 4.0.5

tooth_pean = dynlm(as.numeric(toothpaste_ts) ~
lag(as.numeric(PeanutButter_ts), 0))
summary(tooth_pean)                    # Oooh, that's better!

##
## Time series regression with "numeric" data:
## Start = 1, End = 52
##
## Call:
## dynlm(formula = as.numeric(toothpaste_ts) ~
lag(as.numeric(PeanutButter_ts),
##     0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3380  -4.0023  -0.4678   3.4224  12.5658
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         151.10187   18.27746   8.267 6.44e-11
***
## lag(as.numeric(PeanutButter_ts), 0)   0.15063    0.04026   3.742 0.000472
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.094 on 50 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2031
## F-statistic:    14 on 1 and 50 DF,  p-value: 0.0004718

#plot(tooth_pean)              # Some bias, and deviation from normality
acf(tooth_pean$residuals)
```




```r
pacf(tooth_pean$residuals)
```




```r
Box.test(tooth_pean$residuals, lag = 20, type = 'Ljung')
```

```
##
##  Box-Ljung test
##
## data:  tooth_pean$residuals
## X-squared = 21.399, df = 20, p-value = 0.374
```

For toothpaste and zero lag (PeanutButter)= Regression with ARIMA(0,0,1) errors

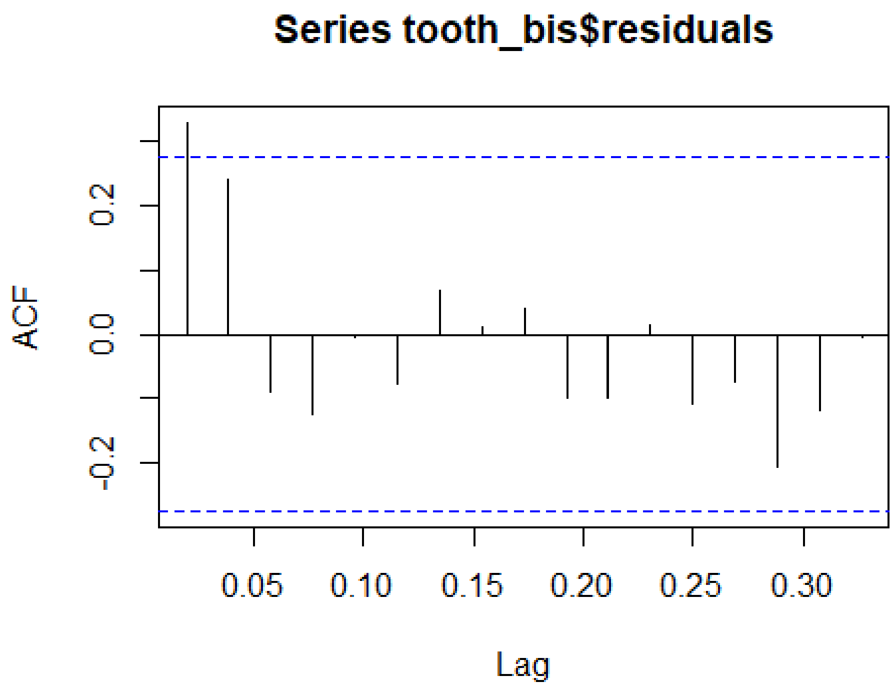
The chosen model is MA(1). The sale of toothpaste at any time t, is derived with a lag zero of peanut butter and MA(1)

$$Toothpaste = 182.4627 + 0.081 PeanutButter_t + n_t$$

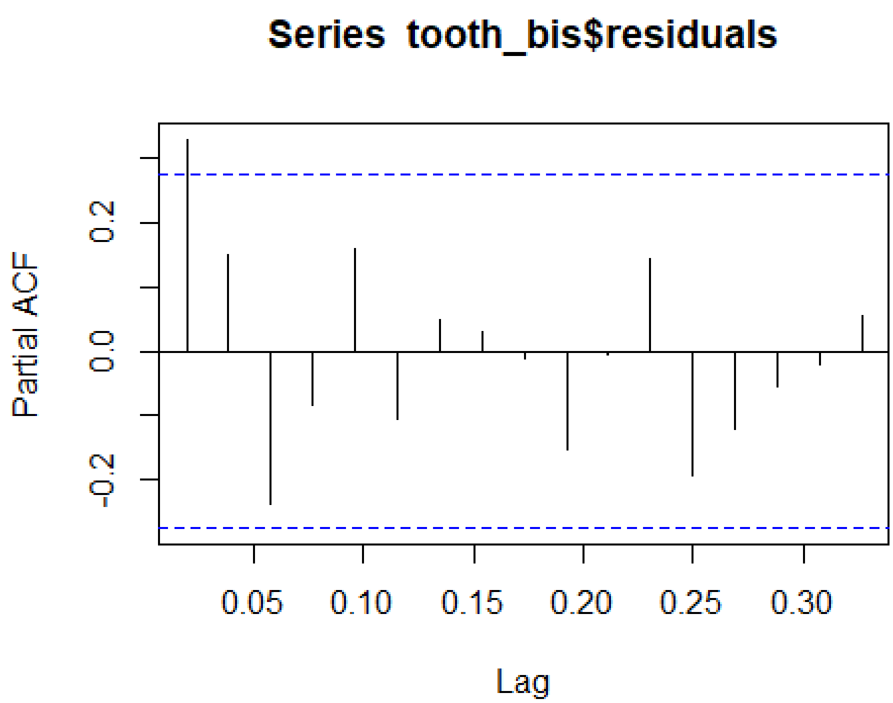
$$n_t = 1. n_{t-1}$$

```
# Let's do an Arima so that we can run a 6-sample prediction at the end.
# We will use a zoo because it allows us to more easily make and coordinate
# the lags

s = as.zoo(ts.intersect((toothpaste_ts),
pean=lag(as.numeric(PeanutButter_ts), 0)))

pean=lag(as.numeric(PeanutButter_ts),0)

length(PeanutButter_ts)

## [1] 52

                     # set, so we can use them to forecast "rec" out beyond
penut_test = subset(PeanutButter_ts, start = length(PeanutButter_ts)  + 1)

# Notice that a zoo is a little more convenient because we can use $!!


tooth_nut_fit = Arima(toothpaste_ts, xreg= pean, order=c(0, 0, 1))
#tooth_nut_fit = Arima(s$toothpaste_ts, xreg=s$pean, order=c(0, 0, 1))
tooth_nut_fit

## Series: toothpaste_ts
## Regression with ARIMA(0,0,1) errors
##
## Coefficients:
##          ma1  intercept    xreg
##       1.0000   182.4627  0.0816
## s.e.  0.0669     3.5255  0.0074
##
## sigma^2 = 13.7:  log likelihood = -142.28
## AIC=292.55   AICc=293.4   BIC=300.36
```

```r
coeftest(tooth_nut_fit)

##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ma1       1.0000e+00 6.6919e-02  14.943 < 2.2e-16 ***
## intercept 1.8246e+02 3.5255e+00  51.755 < 2.2e-16 ***
## xreg      8.1605e-02 7.4478e-03  10.957 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Box.test(tooth_nut_fit$residuals, lag = 20, type = 'Ljung')

##
##  Box-Ljung test
##
## data:  tooth_nut_fit$residuals
## X-squared = 7.6683, df = 20, p-value = 0.9938

#tooth_bis = dynlm(as.numeric(toothpaste_ts) ~ lag(Biscuits_ts, 1))
tooth_bis=dynlm(toothpaste_ts ~ lag(as.numeric(Biscuits_ts), 1))
summary(tooth_bis)

##
## Time series regression with "ts" data:
## Start = 2008(2), End = 2008(52)
##
## Call:
## dynlm(formula = toothpaste_ts ~ lag(as.numeric(Biscuits_ts),
##     1))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4917 -3.5509 -0.2448  3.5940 14.9919
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    166.7461    21.0516   7.921 2.53e-10 ***
## lag(as.numeric(Biscuits_ts), 1)   0.1398     0.0559   2.501   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.448 on 49 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1132, Adjusted R-squared:  0.09507
## F-statistic: 6.253 on 1 and 49 DF,  p-value: 0.01579

acf(tooth_bis$residuals)
```




```
pacf(tooth_bis$residuals)
```




```
eacf(tooth_bis$residuals)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x o o o o o o o o o o  o  o  o
## 7 o o o o o o o o o o o  o  o  o

Box.test(tooth_bis$residuals, lag = 20, type = 'Ljung')

## 
##  Box-Ljung test
## 
## data:  tooth_bis$residuals
## X-squared = 24.813, df = 20, p-value = 0.2087
```

For toothpaste and lag-1 biscuit= Regression with ARIMA(0,0,1) errors

$$Toothpaste = 233.2836 - 0.03661 buscuit_{t-1} + n_t$$

$$n_t = 0.7010 n_{t-1}$$

```
# Let's do an Arima so that we can run a 6-sample prediction at the end.
# We will use a zoo because it allows us to more easily make and coordinate
# the lags

s = as.zoo(ts.intersect((toothpaste_ts),
Biscuits_ts_lag=lag(as.numeric(Biscuits_ts), 1)))

Biscuits_ts_lag=lag(as.numeric(Biscuits_ts), 1)
length(Biscuits_ts)

## [1] 52

                     # set, so we can use them to forecast "rec" out beyond
penut_test = subset(PeanutButter_ts, start = length(Biscuits_ts)  + 1)

# Notice that a zoo is a little more convenient because we can use $!!


tooth_bis_fit = Arima(toothpaste_ts, xreg= Biscuits_ts_lag, order=c(0, 0, 1))
#tooth_nut_fit = Arima(s$toothpaste_ts, xreg=s$pean, order=c(0, 0, 1))
tooth_bis_fit

## Series: toothpaste_ts 
## Regression with ARIMA(0,0,1) errors 
## 
## Coefficients:
```

```
##          ma1  intercept     xreg
##       0.7010   233.2836  -0.0366
## s.e.  0.1394    17.0520   0.0452
##
## sigma^2 = 23.57:  log likelihood = -151.74
## AIC=311.47   AICc=312.34   BIC=319.2

coeftest(tooth_bis_fit)

##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z|)
## ma1          0.700953   0.139443  5.0268 4.987e-07 ***
## intercept 233.283573  17.052029 13.6807 < 2.2e-16 ***
## xreg       -0.036551   0.045231 -0.8081     0.419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Box.test(tooth_bis_fit$residuals, lag = 20, type = 'Ljung')

##
##  Box-Ljung test
##
## data:  tooth_bis_fit$residuals
## X-squared = 8.1609, df = 20, p-value = 0.9907
```

4d,

Auto Arima for toothpaste with zero lag peanut = Regression with ARIMA(1,0,0) errors

Auto Arima for toothpaste with lag one biscuit = Regression with ARIMA(1,0,0) errors

```
auto_arima_bis=auto.arima(toothpaste_ts, xreg= pean)
auto_arima_bis

## Series: toothpaste_ts
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  intercept    xreg
##       0.5026   179.7242  0.0879
## s.e.  0.1226    12.3724  0.0271
##
## sigma^2 = 20.64:  log likelihood = -151.1
## AIC=310.19   AICc=311.04   BIC=318

coeftest(auto_arima_bis)
```

```
##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z|)
## ar1          0.502595   0.122591  4.0998 4.136e-05 ***
## intercept 179.724160  12.372358 14.5263 < 2.2e-16 ***
## xreg         0.087862   0.027099  3.2423  0.001186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Box.test(auto_arima_bis$residuals, lag = 20, type = 'Ljung')
```

```
##
##  Box-Ljung test
##
## data:  auto_arima_bis$residuals
## X-squared = 19.453, df = 20, p-value = 0.4926
```

```r
auto_arima_bis=auto.arima(toothpaste_ts, xreg= Biscuits_ts_lag)
auto_arima_bis
```

```
## Series: toothpaste_ts
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  intercept    xreg
##       0.5176   199.1440  0.0547
## s.e.  0.1356    14.4642  0.0381
##
## sigma^2 = 23.78:  log likelihood = -152.3
## AIC=312.6   AICc=313.47   BIC=320.33
```

```r
coeftest(auto_arima_bis)
```

```
##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z|)
## ar1          0.517593   0.135587  3.8174 0.0001348 ***
## intercept 199.144021  14.464220 13.7680 < 2.2e-16 ***
## xreg         0.054668   0.038087  1.4353 0.1511876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Box.test(auto_arima_bis$residuals, lag = 20, type = 'Ljung')
```

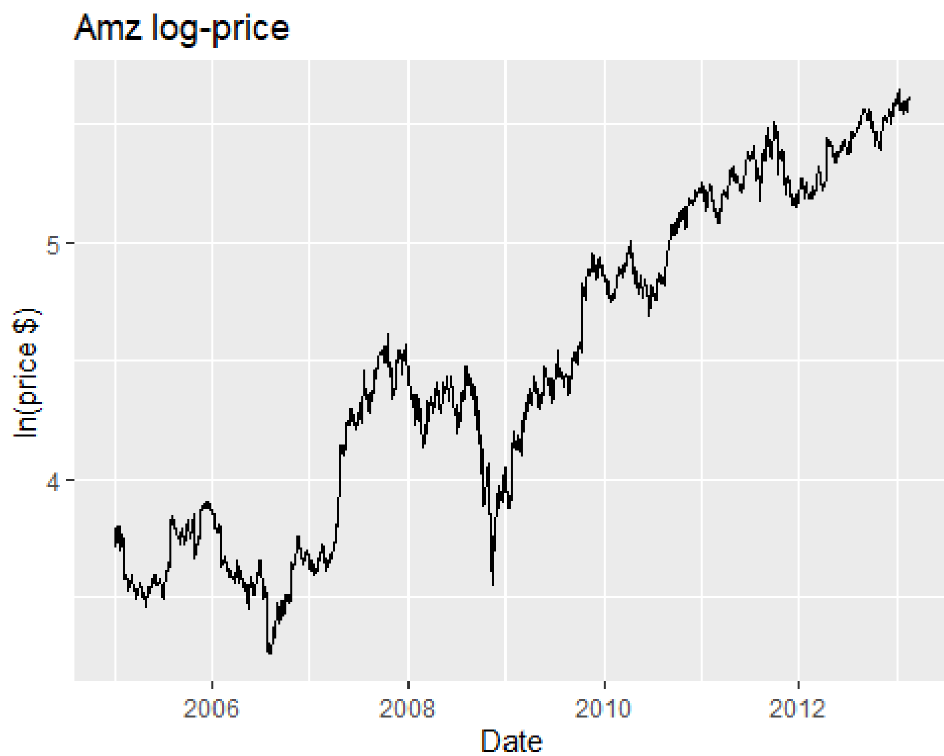
```
##
##  Box-Ljung test
##
## data:  auto_arima_bis$residuals
## X-squared = 11.88, df = 20, p-value = 0.9202
```

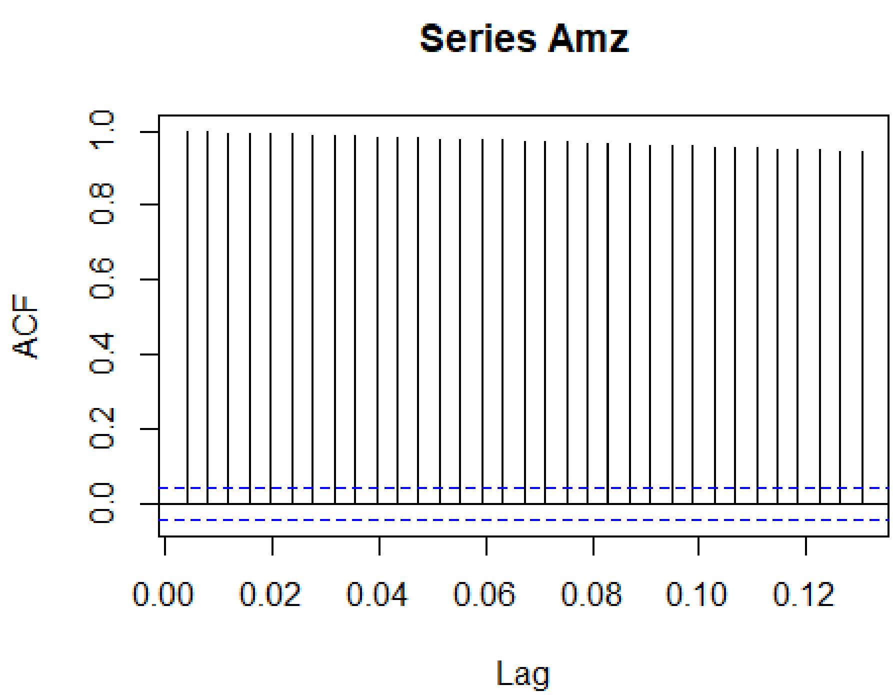
#5

5a

```
ds = read.csv("amzn_2005_13_d.csv")
ds$Date = as.Date(ds$Date,format = "%m/%d/%Y")
ds$Date = as.character(ds$Date)

amz_ts = ts(ds$Price, start = c(2005,01), end = c(2013,12), frequency = 52)
```

5b, ACF shows serial correlation, the box test rejects white noise with 5% significant level.

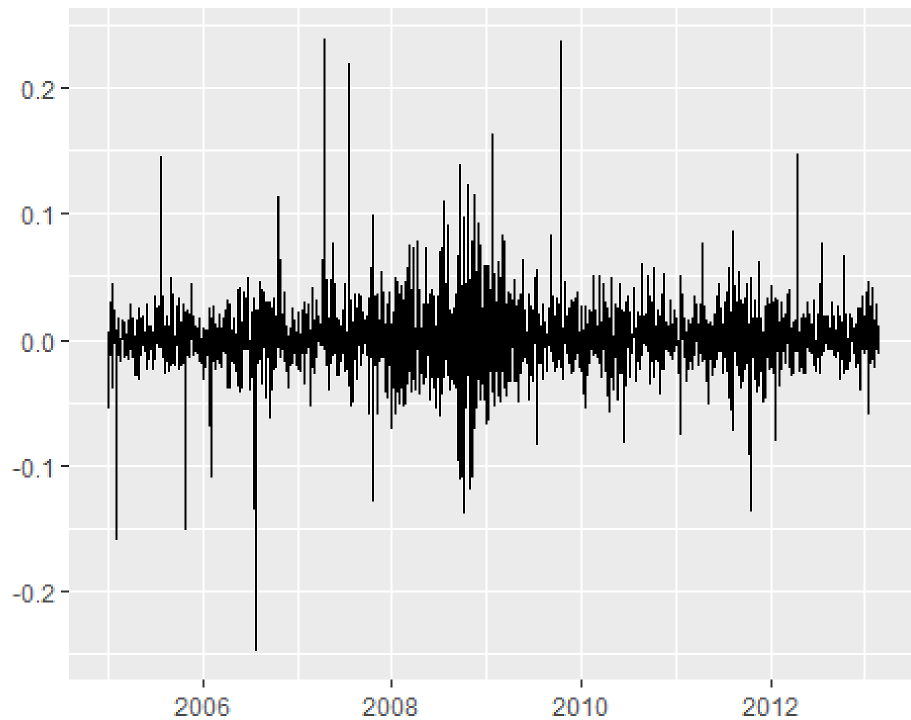




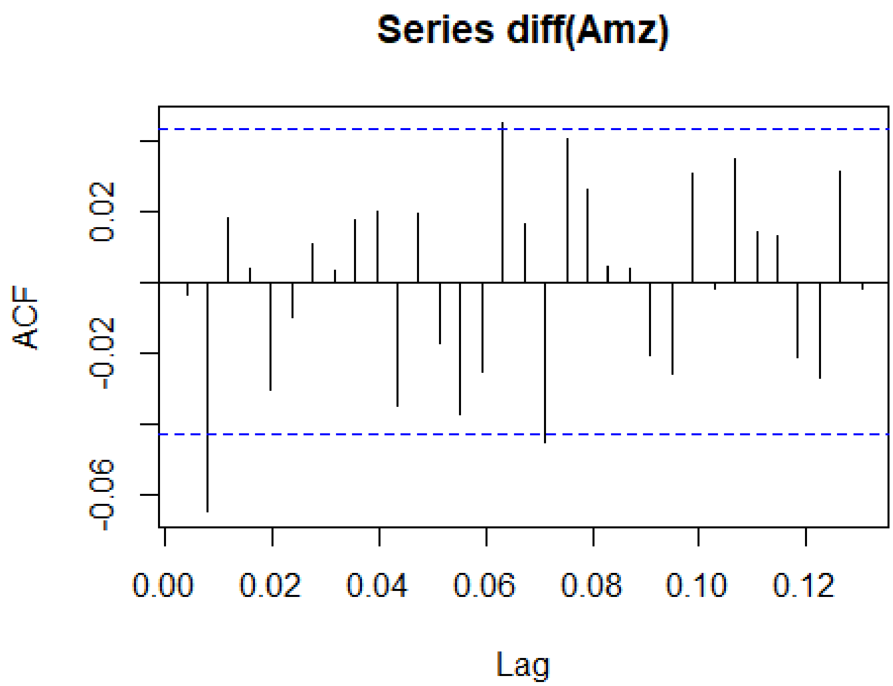
```
##
##  Box-Ljung test
##
## data:  Amz
## X-squared = 24292, df = 12, p-value < 2.2e-16
```

5C, ARCH effect is present at 5% significant level we reject white noise of the log return of amazon stock which means information are available in the log difference.

```
autoplot(diff(Amz))
```




```
acf(diff(Amz))
```



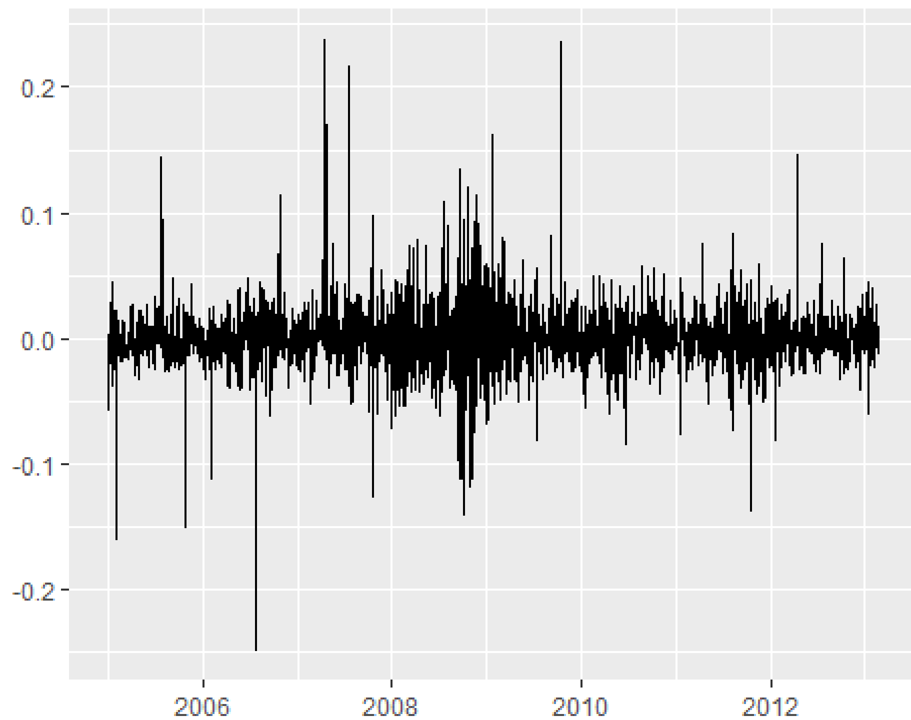

```
Box.test(diff(Amz), lag = 12, type = "Ljung")

##
##  Box-Ljung test
##
## data:  diff(Amz)
## X-squared = 16.497, df = 12, p-value = 0.1695
```

5d, The residual for the fitting faild to reject white noise. The squared residual failed to reject white noise the t-distribution is a good fit for the data.

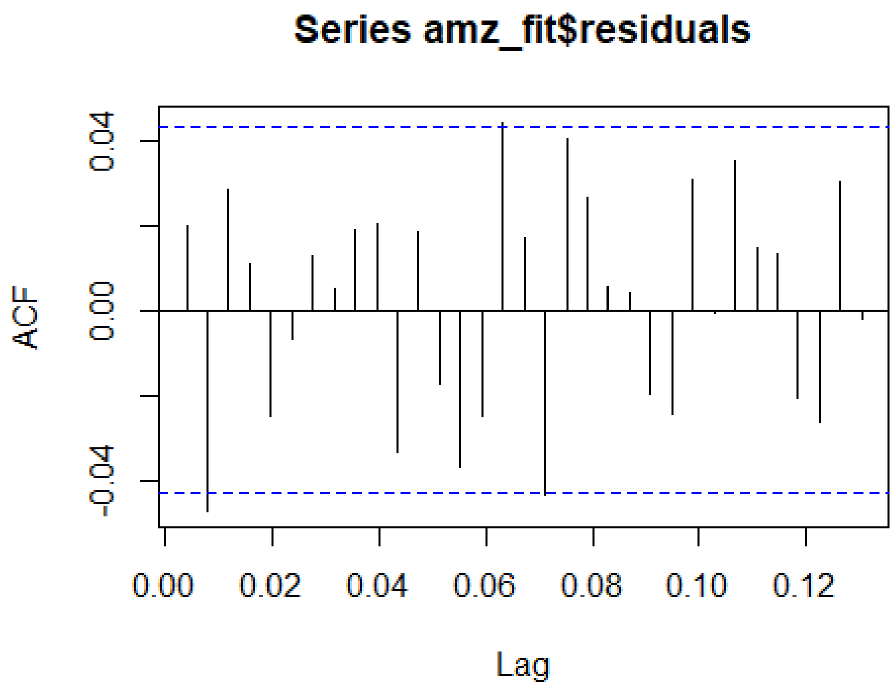
```
amz_fit = Arima(Amz, order = c(1,1,1), include.drift=T)
amz_fit

## Series: Amz
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1  drift
##       0.6722  -0.6973  9e-04
## s.e.  0.2614   0.2547  6e-04
##
## sigma^2 = 0.0007767:  log likelihood = 4453.76
## AIC=-8899.52   AICc=-8899.5   BIC=-8877

coeftest(amz_fit)
```

```
##
## z test of coefficients:
##
##          Estimate  Std. Error z value Pr(>|z|)
## ar1    0.67222992  0.26138976  2.5718 0.010119 *
## ma1   -0.69729491  0.25469389 -2.7378 0.006186 **
## drift  0.00087082  0.00056717  1.5354 0.124692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

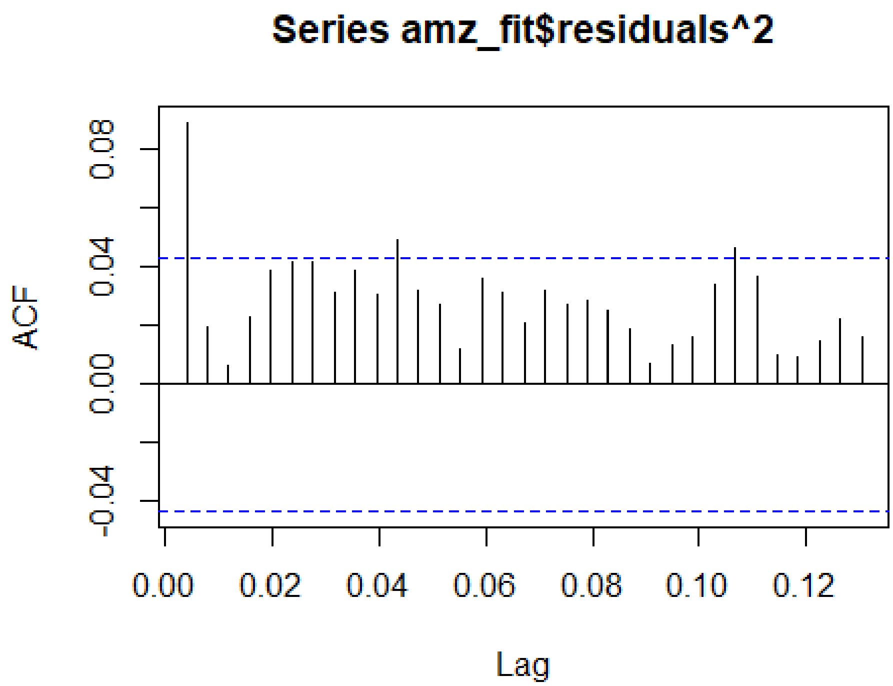
```
autoplot(amz_fit$residuals)   # Random but a lot of heteroschedasticity!
```



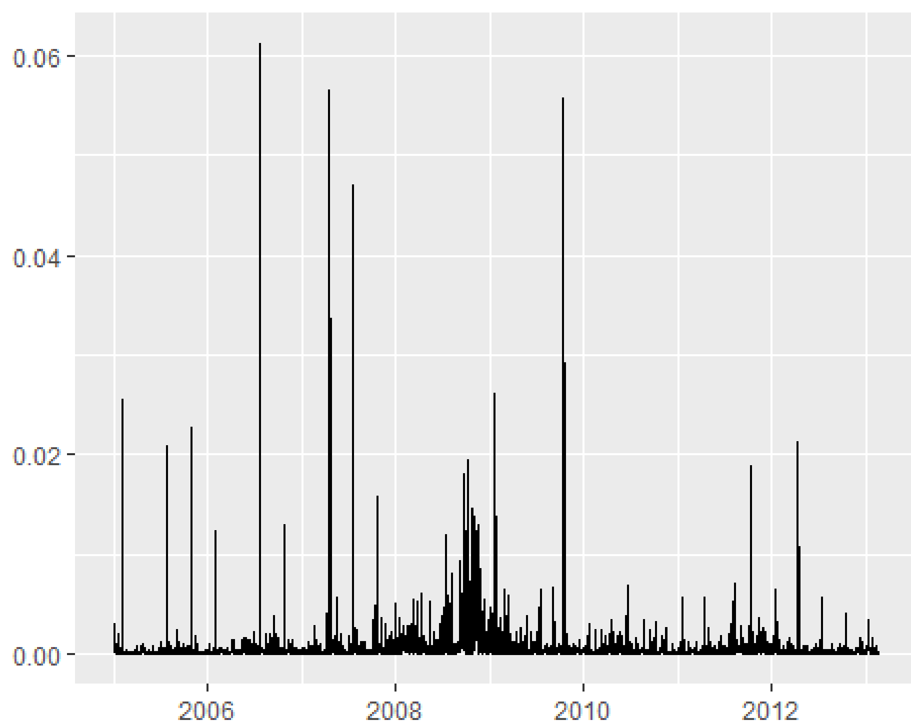

```
acf(amz_fit$residuals)
```




```r
Box.test(amz_fit$residuals, lag=10, type="Ljung")

##
##  Box-Ljung test
##
## data:  amz_fit$residuals
## X-squared = 10.672, df = 10, p-value = 0.3836

Box.test(amz_fit$residuals, lag=15, type="Ljung")

##
##  Box-Ljung test
##
## data:  amz_fit$residuals
## X-squared = 18.468, df = 15, p-value = 0.2389

acf(amz_fit$residuals^2)
```




```r
autoplot(amz_fit$residuals^2)
```



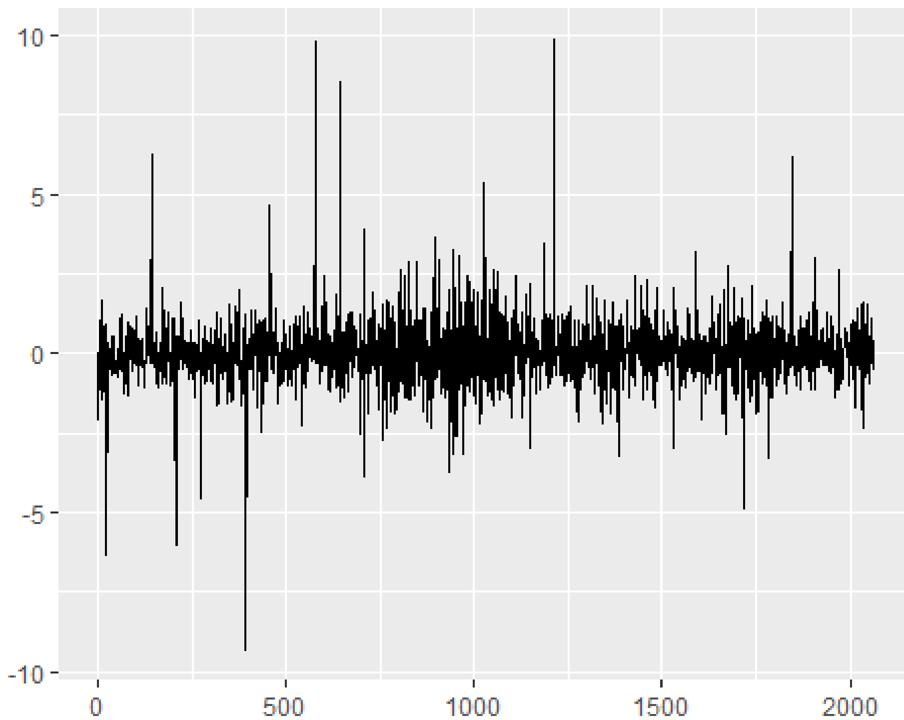

```r
Box.test((amz_fit$residuals^2), lag=10, type="Ljung")
```

```
##
##  Box-Ljung test
##
## data:  (amz_fit$residuals^2)
## X-squared = 35.671, df = 10, p-value = 9.588e-05
```

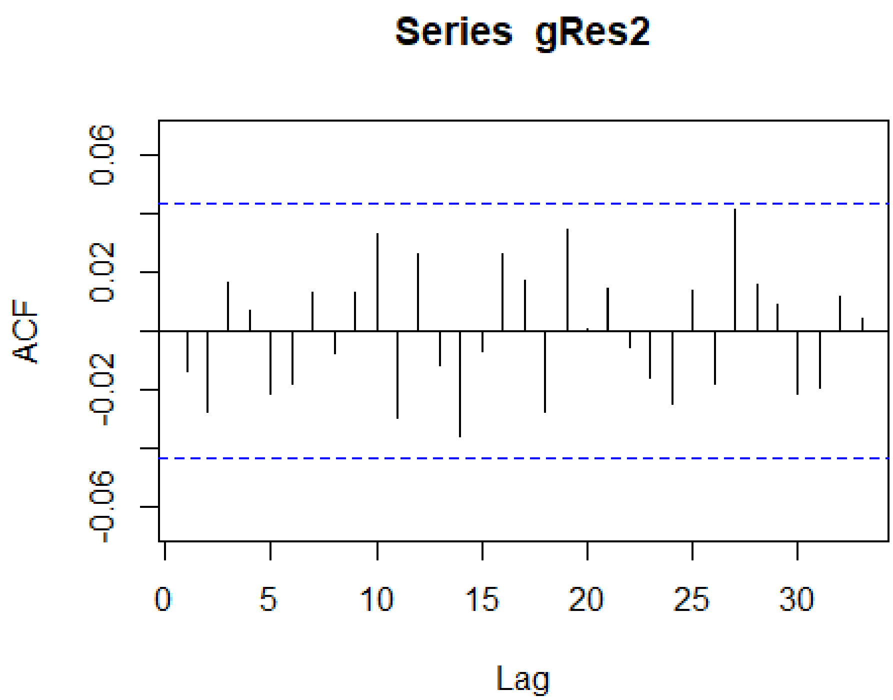
Fitting the ARMA(0,1)-GARCH(0,1). The residual fail to reject white noise, the squared residual also fail to reject white noise.

```
library(fGarch)

## Warning: package 'fGarch' was built under R version 4.0.5

# Grab the residuals and fit a garch(1, 1)
res = amz_fit$residuals

gFit2 = garchFit( ~ arma(0, 1) + garch(1, 1), data=res, trace=F)

## Warning: Using formula(x) is deprecated when x is a character vector of
length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.

gFit2

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 1) + garch(1, 1), data = res, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 1) + garch(1, 1)
## <environment: 0x000000002d9c5ae0>
##  [data = res]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu         ma1       omega      alpha1       beta1
## 0.00014241  0.03724391  0.00006769  0.05397210  0.85929885
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     1.424e-04   6.063e-04    0.235 0.814291
## ma1    3.724e-02   2.692e-02    1.384 0.166442
## omega  6.769e-05   1.803e-05    3.754 0.000174 ***
```

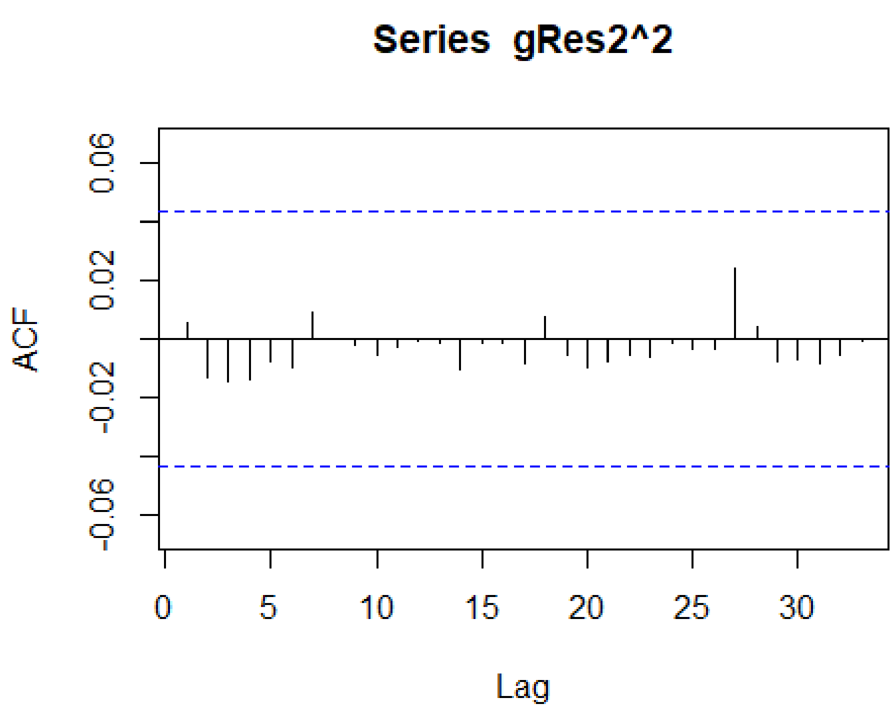
```
## alpha1 5.397e-02   1.561e-02    3.456 0.000547 ***
## beta1  8.593e-01   3.512e-02   24.465  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  4523.622    normalized:  2.194867
##
## Description:
##  Mon May 23 12:50:19 2022 by user: soboa

# Extract the residuals and make them a time series
gRes2 = ts(residuals(gFit2, standardize=T))
autoplot(gRes2)
```



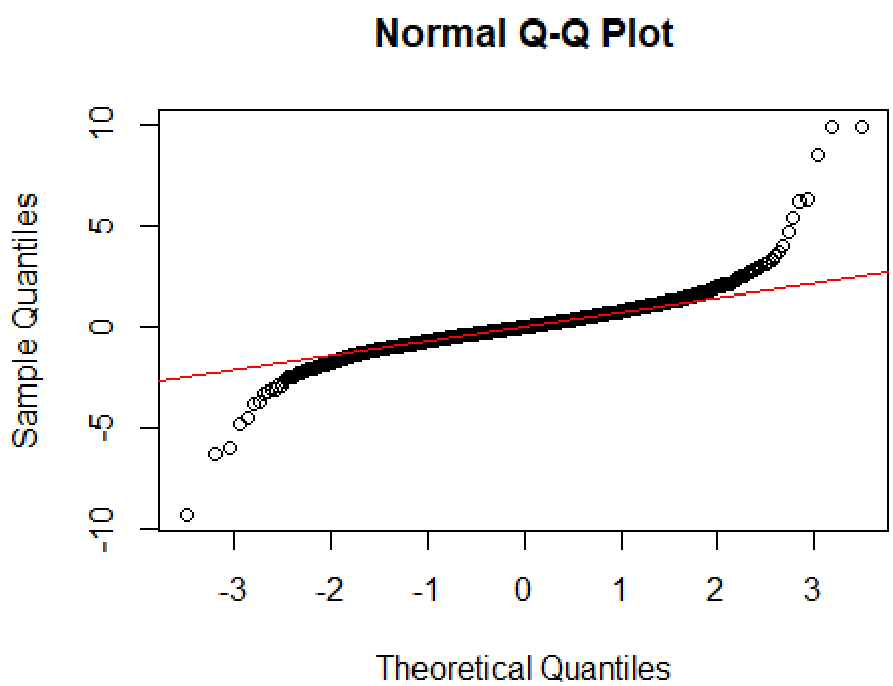

```
Acf(gRes2)            # Higher order autocorrelation, but fairly minor
```




```r
Acf(gRes2^2)          # Almost no squared autocorrelation
```



```r
# What about normality
skewness(gRes2)

## [1] 0.9213608

kurtosis(gRes2)

## [1] 18.81316

qqnorm(gRes2)
qqline(gRes2, col="red")
```




```r
Box.test(gRes2, lag=15, type="Ljung")

##
##  Box-Ljung test
##
## data:  gRes2
## X-squared = 13.535, df = 15, p-value = 0.561
```

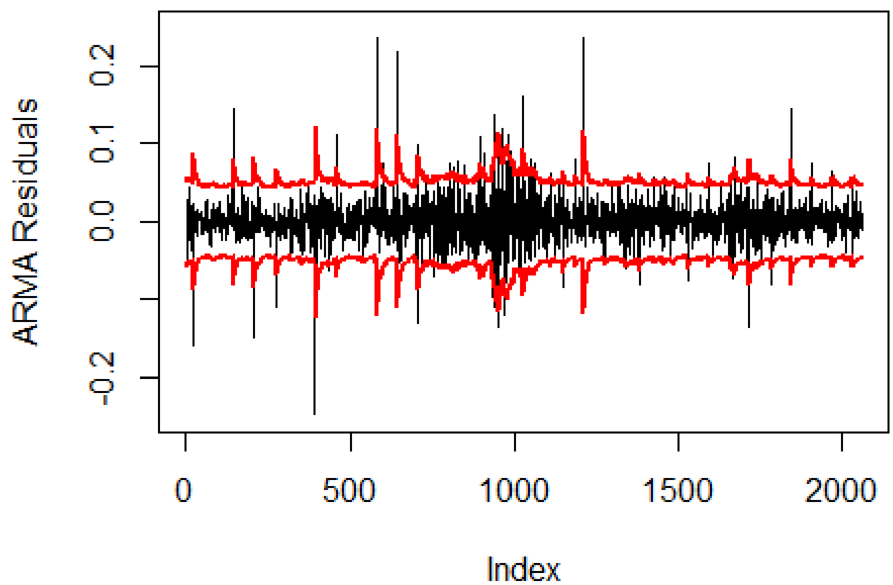
5e, data ~ arma(0, 1) + garch(1, 1), The alpha value gives the absolute standardized shock of the garch model. While the beta give the contribution of the stadardized shock with sign. The amazon stock is positively affected by the shock since beta is positive.

$$r_t = 0.00014241 + 0.03724391E_{t-1} + E_t$$

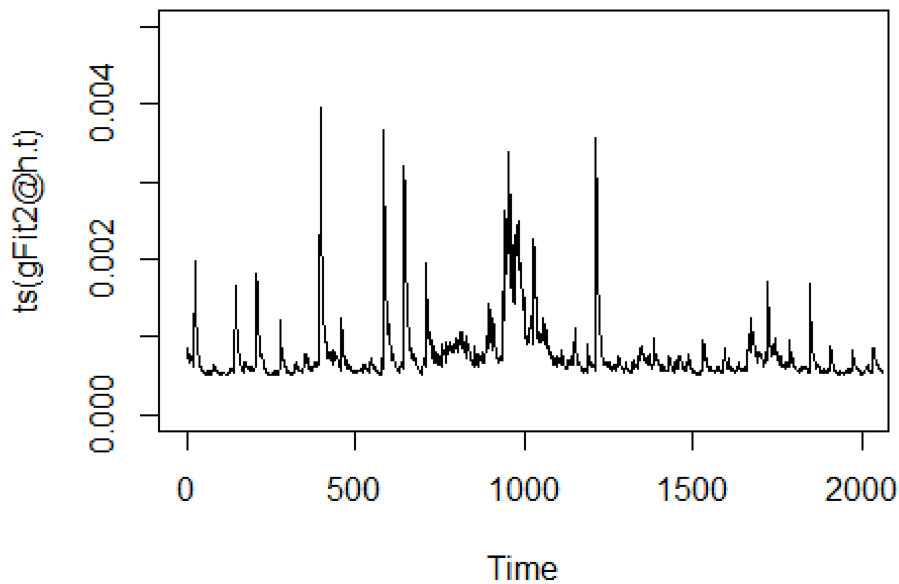
$$\sigma^2 = 0.00014241 + 0.00006769E_{t-1} + 0.85929885\sigma_{t-1}^2 + e_t$$

5f,

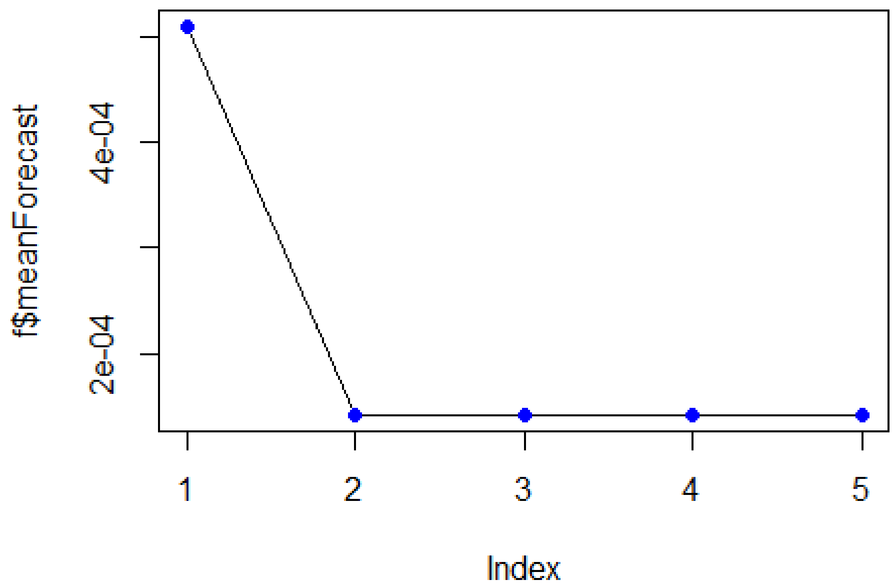
```
plot(residuals(gFit2), type="l", ylim=c(-.25, .25), ylab="ARMA Residuals")

# This time, let's plot the 95% confidence band for the returns, i.e.
# 1.96 * std.dev = 1.96 * sqrt(var)
lines(1.96 * sqrt(gFit2@h.t), col="red", lw=2)
lines(-1.96 * sqrt(gFit2@h.t), col="red", lw=2)
```



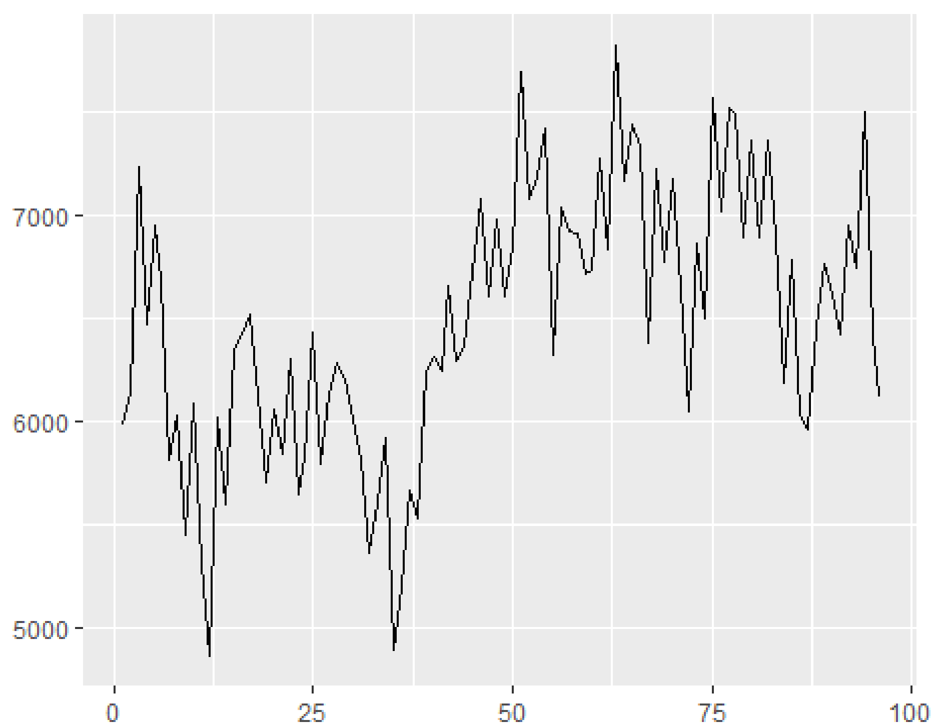

```
# Or look at the prediction itself
plot(ts(gFit2@h.t), xlim=c(0, 2000), ylim=c(0, .005))
```




```
# Let's investigate prediction a bit more.  This time we'll predict out ahead
# 20 steps.
f = predict(gFit2, n.ahead=5)

plot(f$meanForecast, type="l")
points(f$meanForecast, col="blue", pch=16)
```



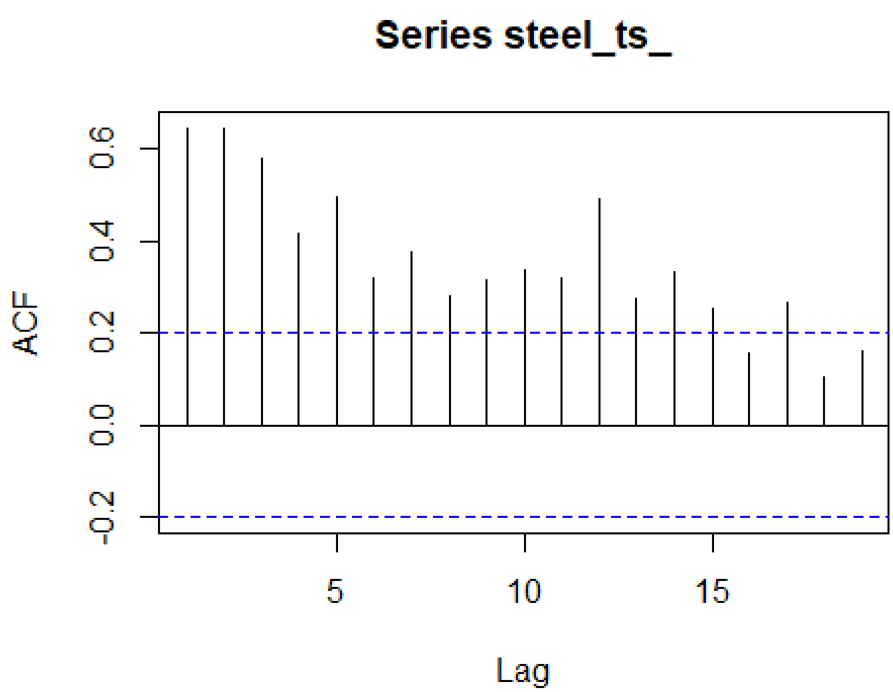
EXTRA CRESIT



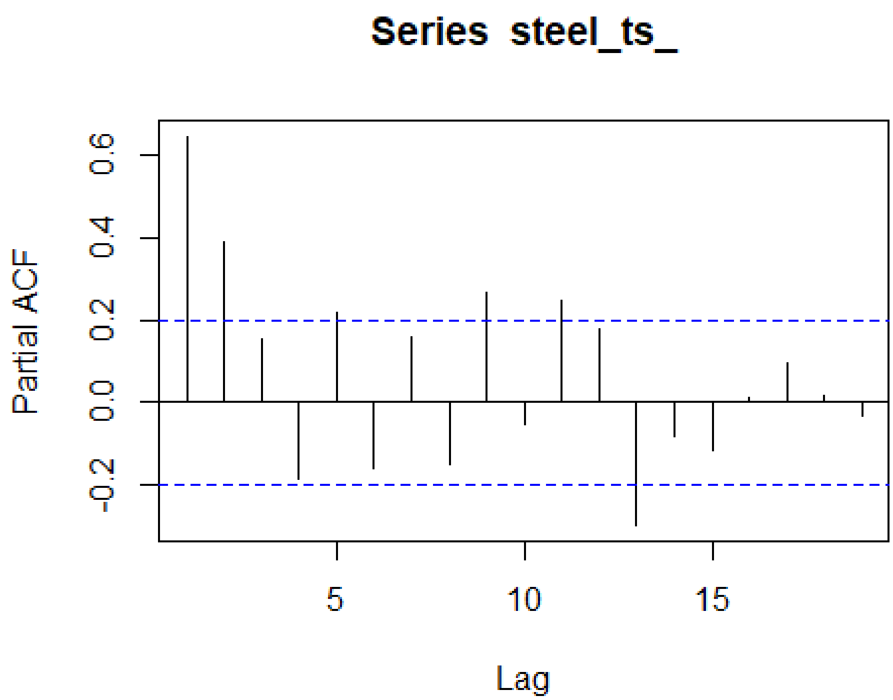
EACF shows non-stationary also shows some seasonality, pacf show AR(2). ACF shows peak at other lags.

ADF fail to fails to reject non-stationarity at with "nc", "c" and "ct". But the kpss reject statiority with trend and level.

```
acf(steel_ts_)
```




```
pacf(steel_ts_)
```




```
eacf(steel_ts_)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o o x x x x o o o o  x  x  x
## 2 x x o o o o o o o o o  x  o  x
## 3 x o o o o o o x o o o  x  o  o
## 4 x x o o o o o o o o o  x  o  o
## 5 x x o o o o o o o o o  x  o  o
## 6 x x o o o o o o o o o  x  o  o
## 7 x o o o o o o o o x o  x  o  o

library(fUnitRoots)
library(tseries)
#
# Dickey-Fuller: The series has a unit root ---> non-stationary


adfTest(steel_ts_,type = 'nc', lags = 10 )

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
```

```
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: 0.4236
##   P VALUE:
##     0.7479
##
## Description:
##  Mon May 23 12:50:20 2022 by user: soboa
```

```
adfTest(steel_ts_,type = 'c', lags = 10 )
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -1.2669
##   P VALUE:
##     0.5862
##
## Description:
##  Mon May 23 12:50:20 2022 by user: soboa
```

```
adfTest(steel_ts_,type = 'ct', lags = 10 )
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -0.8934
##   P VALUE:
##     0.9503
##
## Description:
##  Mon May 23 12:50:20 2022 by user: soboa
```

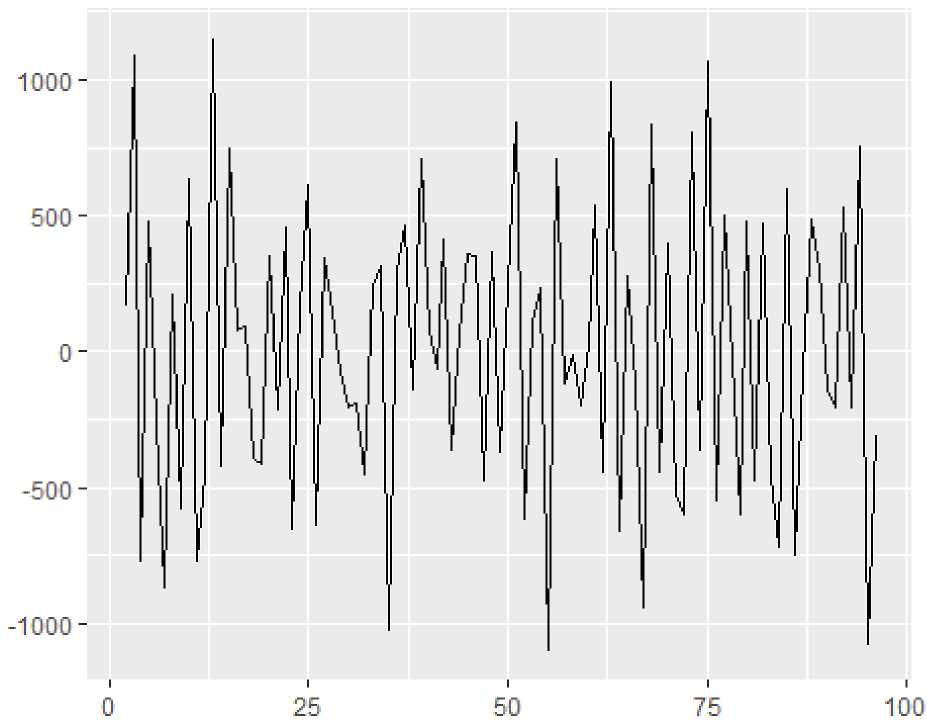
```
##
##  KPSS Test for Level Stationarity
##
## data:  steel_ts_
## KPSS Level = 1.1532, Truncation lag parameter = 3, p-value = 0.01
```

```
##
##  KPSS Test for Trend Stationarity
```

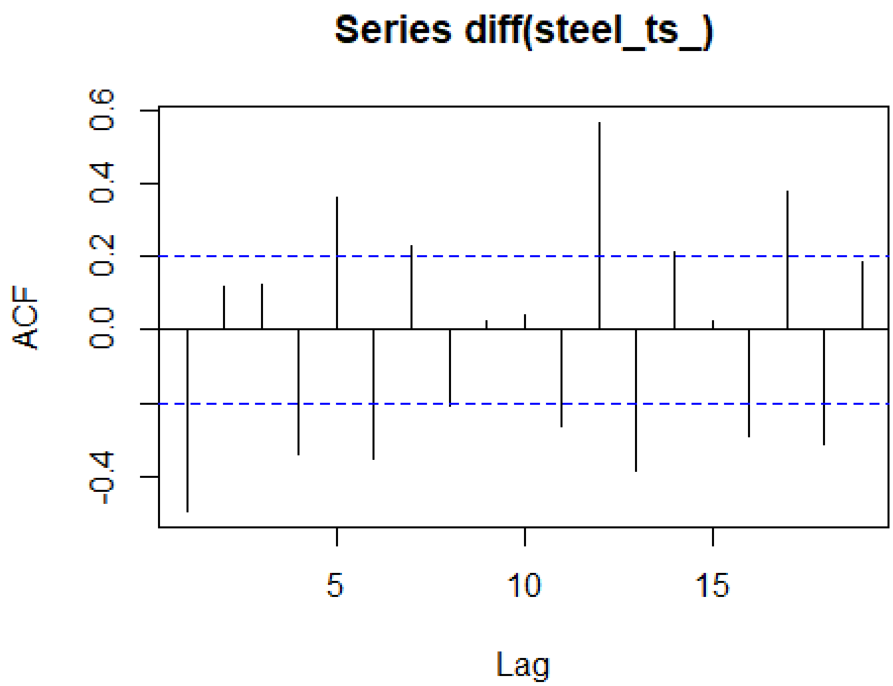
```
##
## data:  steel_ts_
## KPSS Trend = 0.18595, Truncation lag parameter = 3, p-value = 0.02127
```

The diffeence show series is stationary, but there is evidence of seasonality in the difference. ADP and KPSS agree that diff of series is stationary. There are seasonaly pattern in both the ACF and EACF.

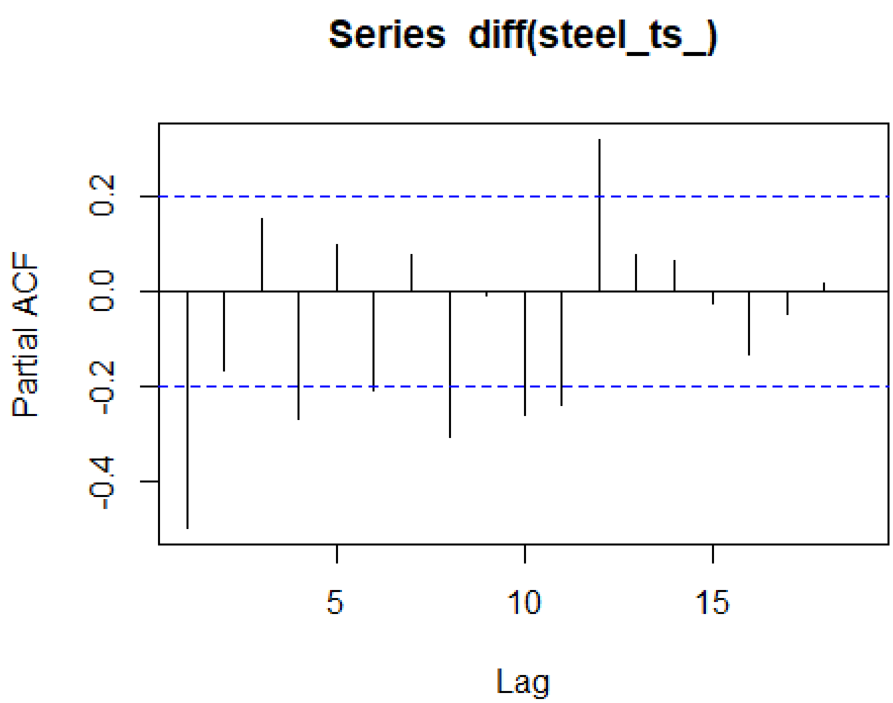
```
autoplot((diff(steel_ts_)))
```




```
acf(diff(steel_ts_))
```




```r
pacf(diff(steel_ts_))
```




```r
eacf(diff(steel_ts_))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o x x x x o o o x  x  x  o
## 1 x x o o o o o o o o o  x  o  x
## 2 x o o o o o o x o o o  x  o  o
## 3 x x o o o o o o o o o  x  o  o
## 4 x x o o o o o o o o o  x  o  o
## 5 x x o x o o o o o o o  x  o  o
## 6 x x o x o o o o o x o  x  o  o
## 7 x x x x o o o o o x o  x  o  o
```

```r
adf.test(diff(steel_ts_))
```

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

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(steel_ts_)
## Dickey-Fuller = -4.5866, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff(steel_ts_))
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  diff(steel_ts_)
## KPSS Level = 0.048618, Truncation lag parameter = 3, p-value = 0.1
```

```r
Box.test(diff(diff(steel_ts_)), lag=10, type="Ljung")
```

```
##
##  Box-Ljung test
##
## data:  diff(diff(steel_ts_))
## X-squared = 133, df = 10, p-value < 2.2e-16
```

```r
mean(diff(steel_ts_))
```

```
## [1] 1.452632
```

```r
t.test(diff(steel_ts_))
```

```
##
##  One Sample t-test
##
## data:  diff(steel_ts_)
## t = 0.026313, df = 94, p-value = 0.9791
```

```
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -108.1613  111.0665
## sample estimates:
## mean of x
##  1.452632
```

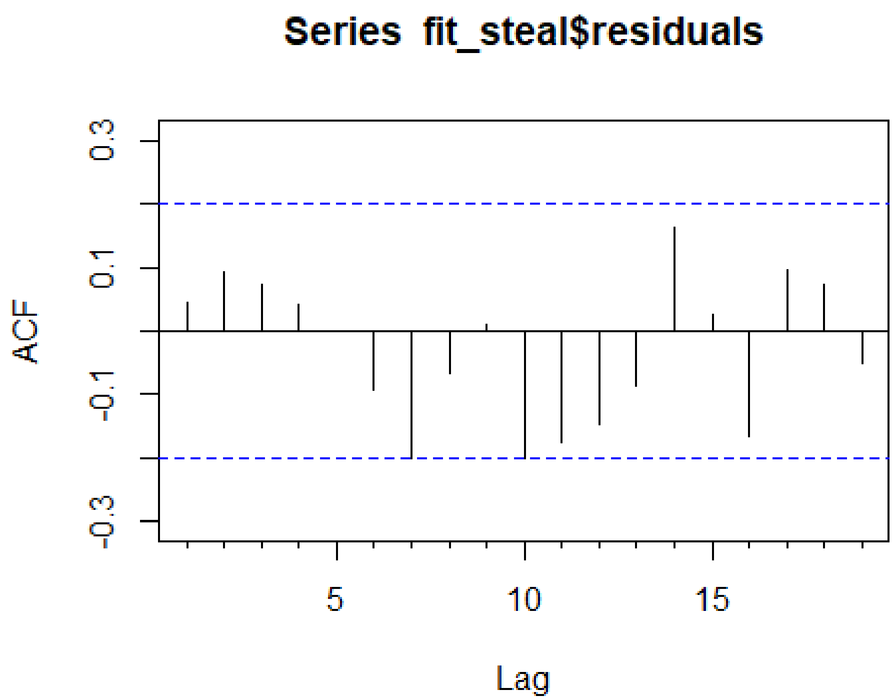
```
fit_steal =Arima(steel_ts_, order=c(2, 1, 2),  seasonal = list (order=
c(0,1,0), period=12), fixed=c(NA, NA, NA, 0))
fit_steal
```

```
## Series: steel_ts_
## ARIMA(2,1,2)(0,1,0)[12]
##
## Coefficients:
##           ar1      ar2     ma1  ma2
##       -0.9952  -0.4389  0.6043    0
## s.e.   0.2116   0.1056  0.2211    0
##
## sigma^2 = 143014:  log likelihood = -609.06
## AIC=1226.12   AICc=1226.63   BIC=1235.8
```

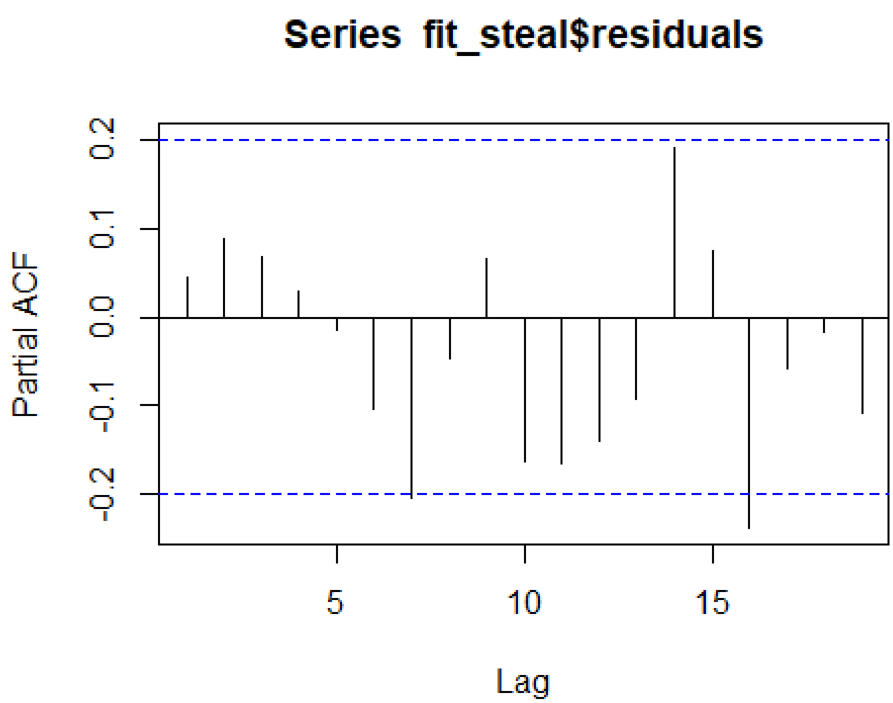
```
coeftest(fit_steal)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.99519    0.21156 -4.7040 2.551e-06 ***
## ar2 -0.43892    0.10564 -4.1547 3.257e-05 ***
## ma1  0.60433    0.22112  2.7330  0.006276 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Acf(fit_steal$residuals)
```




```r
pacf(fit_steal$residuals)
```




```r
Box.test(fit_steal$residuals, lag = 12, type = 'Ljung')
```

```
##
##  Box-Ljung test
##
## data:  fit_steal$residuals
## X-squared = 17.721, df = 12, p-value = 0.1244
```

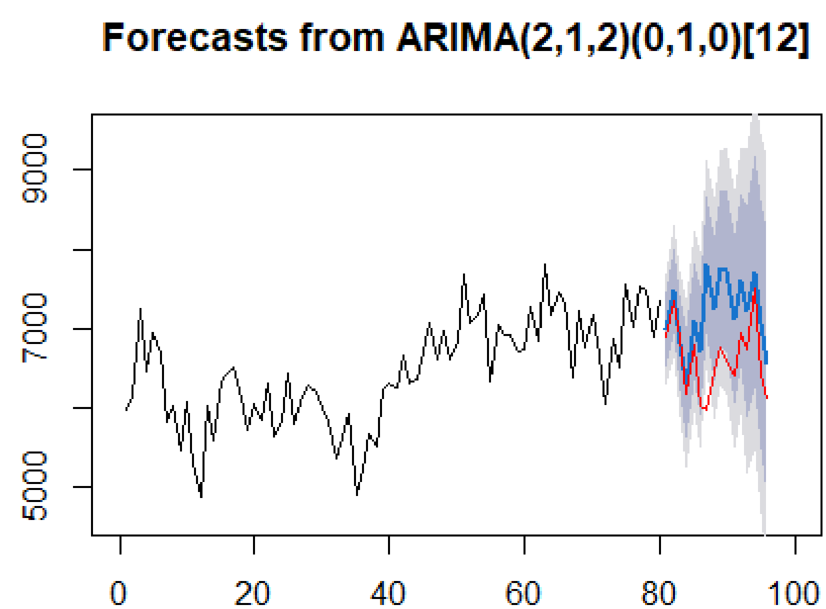
```r
length(steel_ts_)
```

```
## [1] 96
```

```r
rtrain= subset(steel_ts_ , end = 80)
rtest= subset(steel_ts_ , start = 81)
```

After running thur 3 model ARIMA(2,1,1)(1,0,1)[12] had the highest p-values for not rejecting white noise but AIC and BIC of 1391 and 1406 respecfully the forcast was not has good as ARIMA(2,1,1)(0,1,0)[12].

My final model ARIMA(2,1,1)(0,1,0)[12] showed a better forcast , also rejected white noise and a much lower AIC and BIC difference of 9.




```r
pm1_ = backtest(fit_steal,steel_ts_,80 , h=1)
```

```
## [1] "RMSE of out-of-sample forecasts"
## [1] 422.9453
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 275.2685
## [1] "Mean Absolute Percentage error"
```

```
## [1] 0.04315826
## [1] "Symmetric Mean Absolute Percentage error"
## [1] 0.04245558
```