

MATH1318 Time Series Analysis / MATH2204 Time Series and Forecasting Final Project Report

Declaration of contributions:

No	Name of Team Member	Contribution to the project
1	Bhavya Dinesh (s3980321)	1/4
2	Rishit Turakhia (s3935687)	1/4
3	Saksham Sharma (s3986713)	1/4
4	Samay Jain (s3963844)	1/4
5		
6		
	Sum:	1

RMIT University,
School of Science
2022

Time Series Final Project

Bhavya Dinesh (s3980321)

Rishit Turakhia (s3935687)

Saksham Sharma (s3986713)

Samay Jain (s3963844)

2024-06-06

- Introduction
- Functions
 - Seasonal lag function
 - Diagnostic test
 - Residual analysis
- Data Exploration
 - Converting to time series object
- Descriptive Analysis
 - Descriptive statistics
 - Time series plot
 - Scatter plot
- Test of stationary
- Test of Normality
- Box-Cox Transformation
- Differencing
 - First differencing
 - First difference series stationary test
- ARIMA models
 - Significant lags in ACF and PACF plot
 - EACF plot
 - BIC plot
- Parameter Estimation
 - ARIMA(0,1,1)
 - ARIMA(1,1,1)
 - ARIMA(0,1,4)
 - AIC and BIC values
 - Error measures
- Over-parameterisation
 - ARIMA(0,1,2)
- SARIMA models
 - ACF and PACF plot
 - EACF
 - BIC
- Parameter Estimation
 - SARIMA(2,1,2)x(1,1,2)_12
 - SARIMA(0,1,1)x(1,1,2)_12
 - SARIMA(0,1,2)x(1,1,2)_12
 - SARIMA(1,1,1)x(1,1,2)_12
- Goodness of fit
 - AIC and BIC values
 - Error measures
- Over-parameterisation

- Model Specification ARMA x GARCH Part 1
 - Stationary test
 - EACF ARMA
 - BIC ARMA
- Parameter Estimation ARMA
 - ARMA (0,4)
 - ARMA (0,1)
 - ARMA (0,2)
 - ARMA (1,1)
 - ARMA (1,2)
- Model Specification ARMA x GARCH Part 2
 - ACF and PACF absolute
 - EACF abs garch
 - ACF and PACF squared
 - EACF squared
- Parameter Estimation ARMA x GARCH
- Forecasting
- ARIMA forecast:
- ARMA x GARCH forecast:
- SARIMA forecast:
- Conclusion
- Reference

Research Question:

What is the most accurate forecasting model for predicting the gold price over the next 10 months?

Introduction

The project aims to analyze gold price data from January 2000 to May 2024 and forecast gold prices for the next 10 months. To achieve this, ARIMA, SARIMA, and ARMA-GARCH models are fitted to the data. These models are evaluated and compared using statistical tools such as the coefficient test, AIC, BIC, and error metrics to identify the best model for predicting gold prices.

Functions

Seasonal lag function

```
helper <- function(class = c("acf", "pacf"), ...) {

  # Capture additional arguments
  params <- match.call(expand.dots = TRUE)
  params <- as.list(params)[-1]

  # Calculate ACF/PACF values
  if (class == "acf") {
    acf_values <- do.call(acf, c(params, list(plot = FALSE)))
  } else if (class == "pacf") {
    acf_values <- do.call(pacf, c(params, list(plot = FALSE)))
  }

  # Extract values and lags
  acf_data <- data.frame(
    Lag = as.numeric(acf_values$lag),
    ACF = as.numeric(acf_values$acf)
  )

  # Identify seasonal lags to be highlighted
  seasonal_lags <- acf_data$Lag %% 1 == 0

  # Plot ACF/PACF values
  if (class == "acf") {
    do.call(acf, c(params, list(plot = TRUE)))
  } else if (class == "pacf") {
    do.call(pacf, c(params, list(plot = TRUE)))
  }

  # Add colored segments for seasonal lags
  for (i in which(seasonal_lags)) {
    segments(x0 = acf_data$Lag[i], y0 = 0, x1 = acf_data$Lag[i], y1 = acf_data$ACF[i], col =
"red")
  }
}

# seasonal_acf -----

seasonal_acf <- function(...) {
  helper(class = "acf", ...)
}

# seasonal_pacf -----

seasonal_pacf <- function(...) {
  helper(class = "pacf", ...)
}
```

The seasonal lag function highlights the seasonal lags in ACF and PACF plots.

Diagnostic test

```
Diagnostic_test <- function(data, lag = 20,
                           mainacf = "ACF Plot", subacf = "",
                           mainpacf = "PACF Plot", subpacf = "",
                           mainhist = "Histogram", subhist = "",
                           mainqq = "Q-Q Plot", subqq = "",
                           test) {
  if (test == "ACF-PACF") {
    acf(data, lag.max = lag, main = mainacf, sub = subacf)
    pacf(data, lag.max = lag, main = mainpacf, sub = subpacf)
  } else if (test == 'Stationary') {
    acf(data, lag.max = lag, main = mainacf, sub = subacf)
    pacf(data, lag.max = lag, main = mainpacf, sub = subpacf)
    adf_result <- adf.test(data)
    pp_result <- pp.test(data)
    return(list(adf_test = adf_result, pp_test = pp_result))
  } else if (test == 'Normality') {
    hist(data, main = mainhist, sub = subhist)
    qqnorm(data, main = mainqq, sub = subqq)
    qqline(data, col = 2)
    shapiro_result <- shapiro.test(data)
    return(shapiro_result)
  } else if (test == "seasonal ACF-PACF") {
    seasonal_acf(data, lag.max = lag, main = paste(mainacf, "(Seasonal)"), sub = subacf)
    seasonal_pacf(data, lag.max = lag, main = paste(mainpacf, "(Seasonal)"), sub = subpacf)
  } else if (test == 'seasonal Stationary') {
    seasonal_acf(data, lag.max = lag, main = paste(mainacf, "(Seasonal)"), sub = subacf)
    seasonal_pacf(data, lag.max = lag, main = paste(mainpacf, "(Seasonal)"), sub = subpacf)
    adf_result <- adf.test(data)
    pp_result <- pp.test(data)
    return(list(adf_test = adf_result, pp_test = pp_result))
  } else {
    print("Please enter a valid type of test: 'ACF-PACF', 'Stationary', 'Normality', 'seasona
1 ACF-PACF', or 'seasonal Stationary'")
  }
}
```

The Diagnostic test function helps to do a stationarity test (ACF, PACF, ADF, PP-test), normality test (Histogram, QQ-plot, Shapiro-Wilk), and seasonal stationarity test (Seasonal ACF-PACF).

Residual analysis

```
residual.analysis <- function(model, std = TRUE, start = 2, shift = 0, class = c("ARIMA", "GARCH", "ARMA-GARCH", "garch", "fGARCH")[1]){  
  library(TSA)  
  library(FitAR)  
  library(quantmod)  
  if (class == "ARIMA"){  
    if (std == TRUE){  
      res.model = Lag(rstandard(model), shift)  
      res.model = na.omit(res.model)  
    } else {  
      res.model = Lag(residuals(model), shift)  
      res.model = na.omit(res.model)  
    }  
  }  
  par(mfrow=c(3,2))  
  plot(res.model, type='o', ylab='Standardised residuals', main="Time series plot of standardised residuals")  
  abline(h=0)  
  hist(res.model, main="Histogram of standardised residuals")  
  qqnorm(res.model, main="QQ plot of standardised residuals")  
  qqline(res.model, col = 2)  
  acf(res.model, main="ACF of standardised residuals")  
  print(shapiro.test(res.model))  
  k=0  
  if (length(res.model) < 30){  
    lagM <- length(res.model) - 1  
  } else {  
    lagM <- 30  
  }  
  LBQPlot(res.model, lag.max = lagM, StartLag = k + 1, k = 0, SquaredQ = FALSE)  
  par(mfrow=c(1,1))  
}
```

Data Exploration

```
data <- read_csv("C:/Users/HP/Downloads/RMIT/Sem 3/Time Series analysis/Gold Futures Historical Data.csv")
```

```
# Ordering the date in ascending order  
data$Date <- as.Date(data$Date, format = "%m/%d/%Y")  
data <- data[order(data$Date), ]
```

The dates are ordered in ascending order to preserve the structure of the data and, ensure that trends, and patterns are accurately captured, leading to a correct model estimation and forecasting.

Converting to time series object

```
data.ts <- ts(data$Price, start = c(2000, 1), end = c(2024, 5), frequency = 12)
```

Descriptive Analysis

- Descriptive statistics
- Time series plot
- Impact of 1st lag gold price on current gold price
- Impact of 2nd lag gold price on current gold price

Descriptive statistics

```
summary(data.ts)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	259.2	575.5	1214.5	1109.6	1564.2	2366.8

The descriptive statistics indicate a range of gold price values from \$259.2 to \$2366.8. The median value of \$1214.5 suggests that half the data points are below this value, while the mean of \$1109.6 is the average of gold price from (Jan, 2000) to (May, 2024). The first quartile (\$575.5) and third quartiles (\$1564.2) shows the gold price spread, with the middle 50% of values falling between these points.

Time series plot

```
plot(data.ts, type = "o", pch = 20, ylab = "Gold price", main = "Gold price time series",  
      sub= "Figure 1: Gold price time series")
```

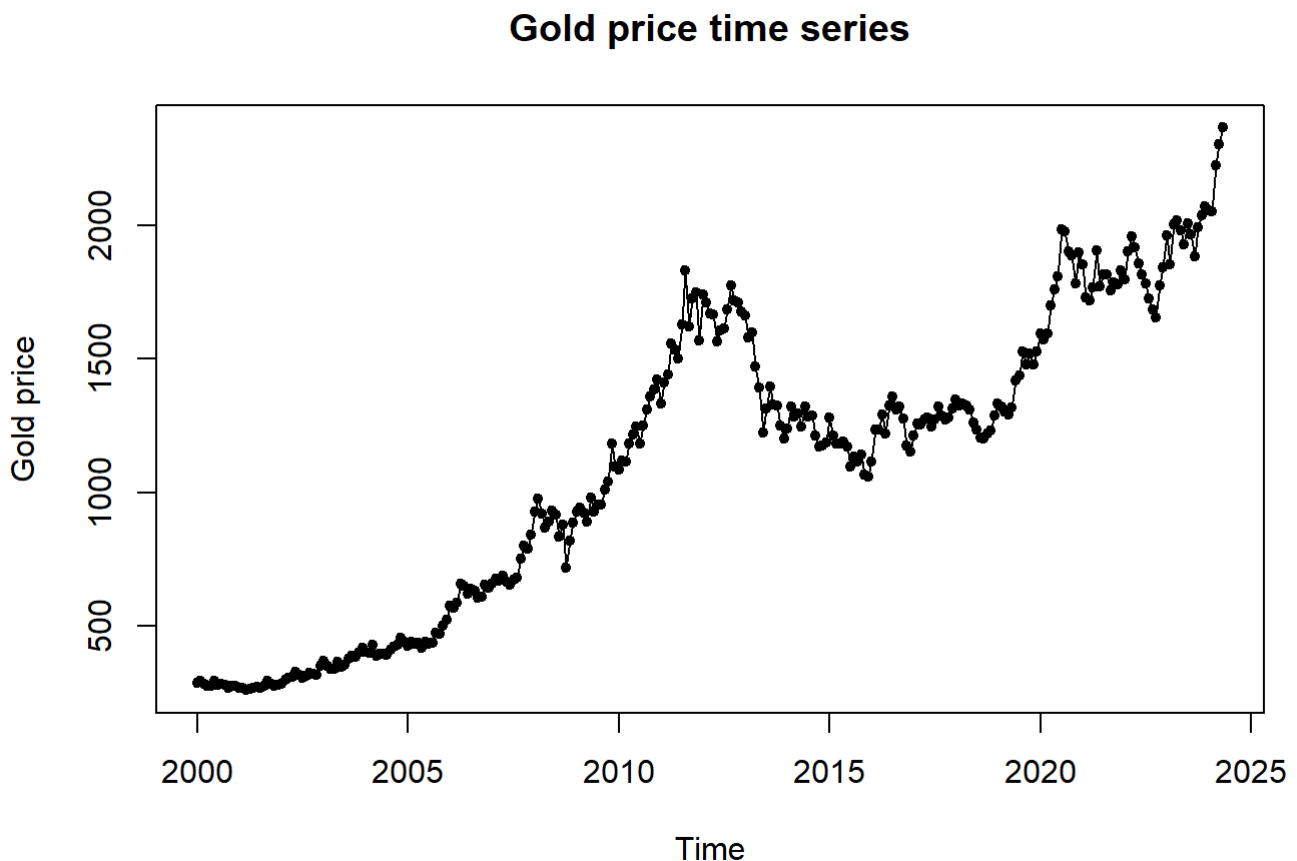


Figure 1: Gold price time series

Interpretation of Gold Price time series plot:

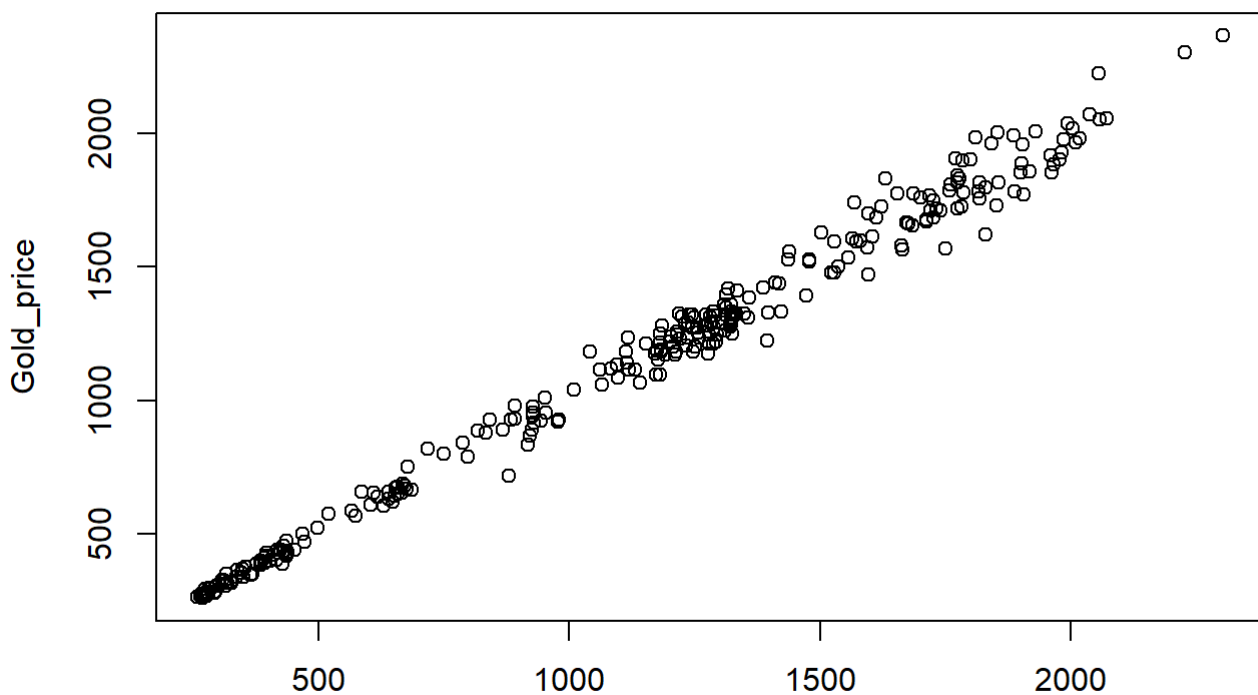
- Trend: The gold price shows a clear upward trend over the entire period from 2000 to 2024.
- Seasonality: From the plot, there is no obvious seasonal pattern observed in the data. The fluctuations and peaks do not appear to follow a regular, repeating pattern over a fixed period.
- Changing Variance: The variance in gold prices increases over time. Early in the series (2001-2005), prices fluctuate within a relatively narrow range. However, as time progresses, the range of fluctuations increases, indicating higher volatility in later years.
- Behavior: The time series plot has consecutive points following each other and indicates auto-regressive behavior, means that current gold prices are influenced by past prices.
- Intervention: No intervention point is observed. However, there are multiple shifts in trend approximately around the years 2008, 2013, and 2019 .

Scatter plot

```
# 1st lag
y = data.ts
x = zlag(data.ts)

# 1st lag plot
plot(y=data.ts, x = zlag(data.ts), ylab = 'Gold_price', xlab = 'Gold price in the previous month',
     main = "Scatter plot of neighboring gold price values",
     sub = "Figure 2: Scatter plot of gold price and 1st lag values")
```

Scatter plot of neighboring gold price values



Gold price in the previous month
Figure 2: Scatter plot of gold price and 1st lag values


```
# 1st lag correlation
index = 2:length(data.ts)
cor(y[index],x[index])
```

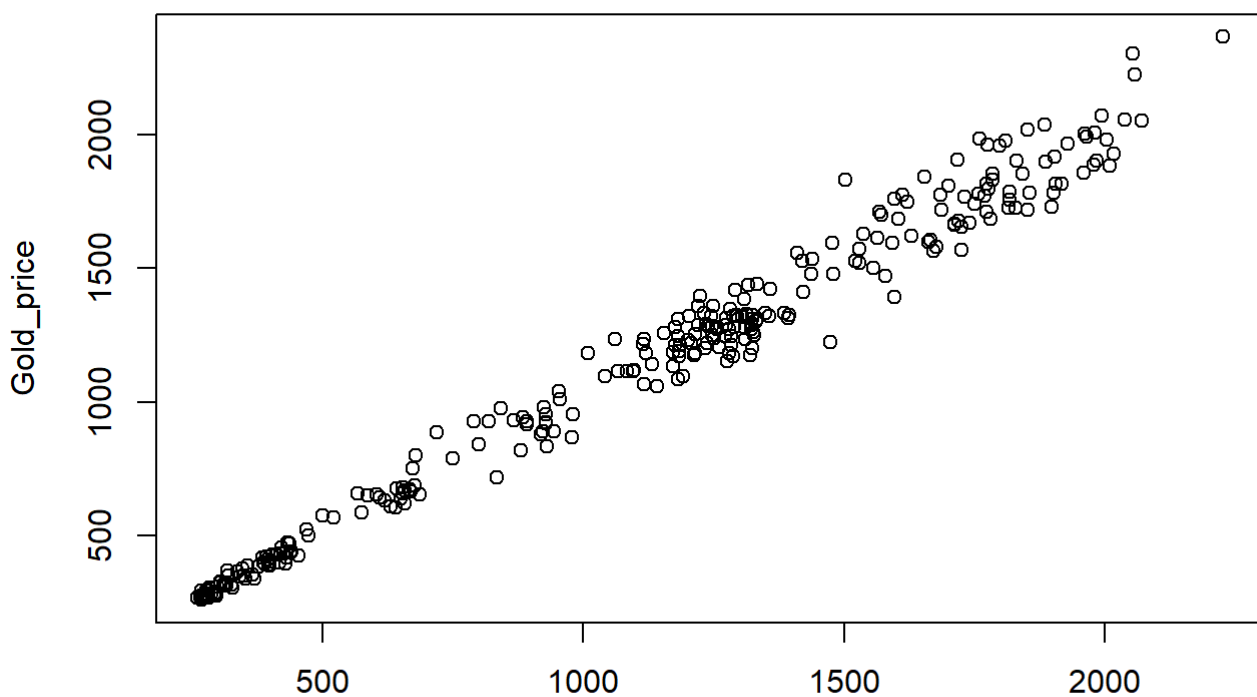
```
## [1] 0.9946506
```

The high correlations observed in the scatter plots from figures 2 indicate a strong relationship between the current month's gold price and its prices in the previous month.

```
# 2nd lag
z = zlag(x)

# 2nd alg plot
plot(y=data.ts, x = zlag(zlag(data.ts)), ylab = 'Gold_price', xlab = 'Gold price in the 2nd p
revious month', main = "Scatter plot of 2nd lag gold price adn gold price values ", sub = "Fi
gure 3: Scatter plot of gold price and 2nd lag values")
```

Scatter plot of 2nd lag gold price adn gold price values



Gold price in the 2nd previous month
Figure 3: Scatter plot of gold price and 2nd lag values

```
# 2nd lag correlation
index = 3:length(z)
cor(y[index],z[index])
```

```
## [1] 0.9904694
```

The high correlations observed in the scatter plots from figure 3 indicate a strong relationship between the current month's gold price and its prices in the previous 2 months. Specifically, gold price correlation of (0.9947) with 1st lag and (0.9905) with 2nd lag suggest that the gold price series has strong autoregressive

characteristics.

Test of stationary

The following tests are performed to check for the stationarity in the data:

- ACF plot
- PACF plot
- ADF test
- PP test

```
Diagnostic_test(data.ts, lag = 300, mainacf = 'ACF of gold price data', subacf = "Figure 4: A  
CF plot of gold price data",mainpacf = 'PACF of gold price data data', subpacf = "Figure 5: P  
ACF plot of gold price data", test = "Stationary")
```

ACF of gold price data

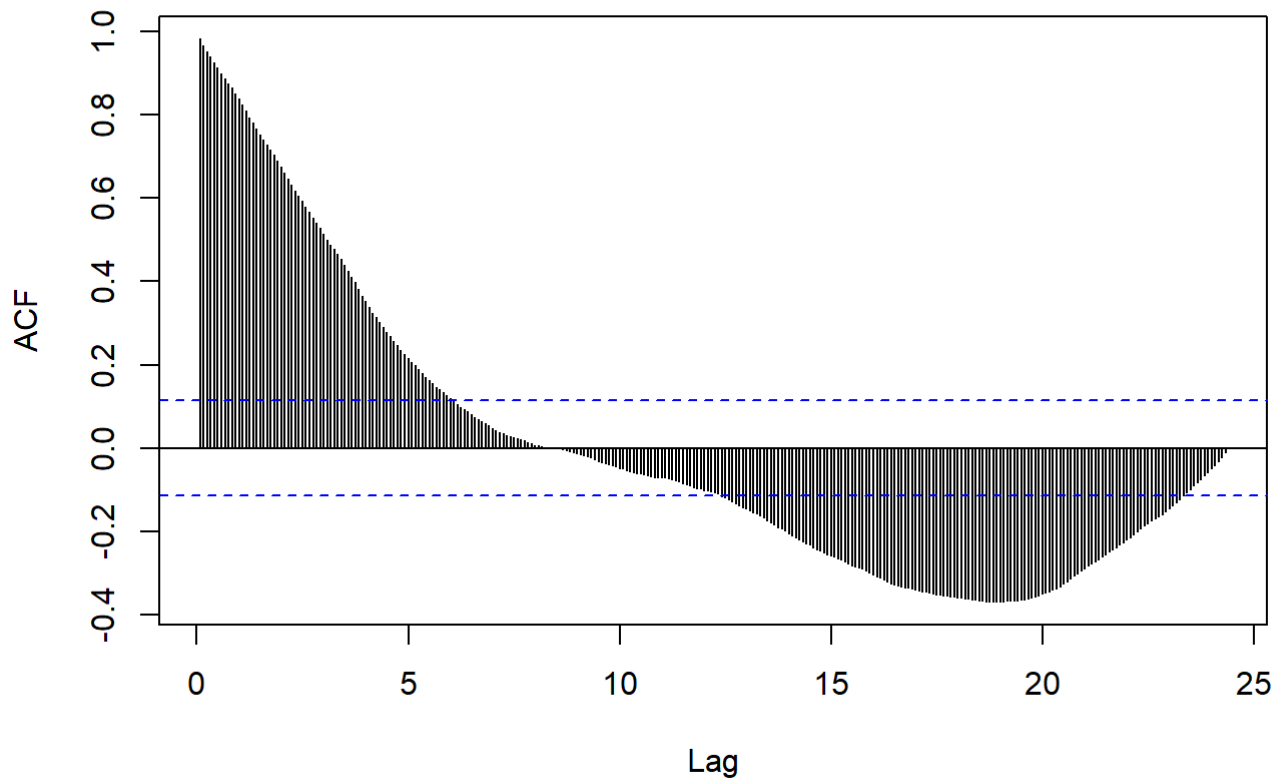


Figure 4: ACF plot of gold price data

PACF of gold price data data

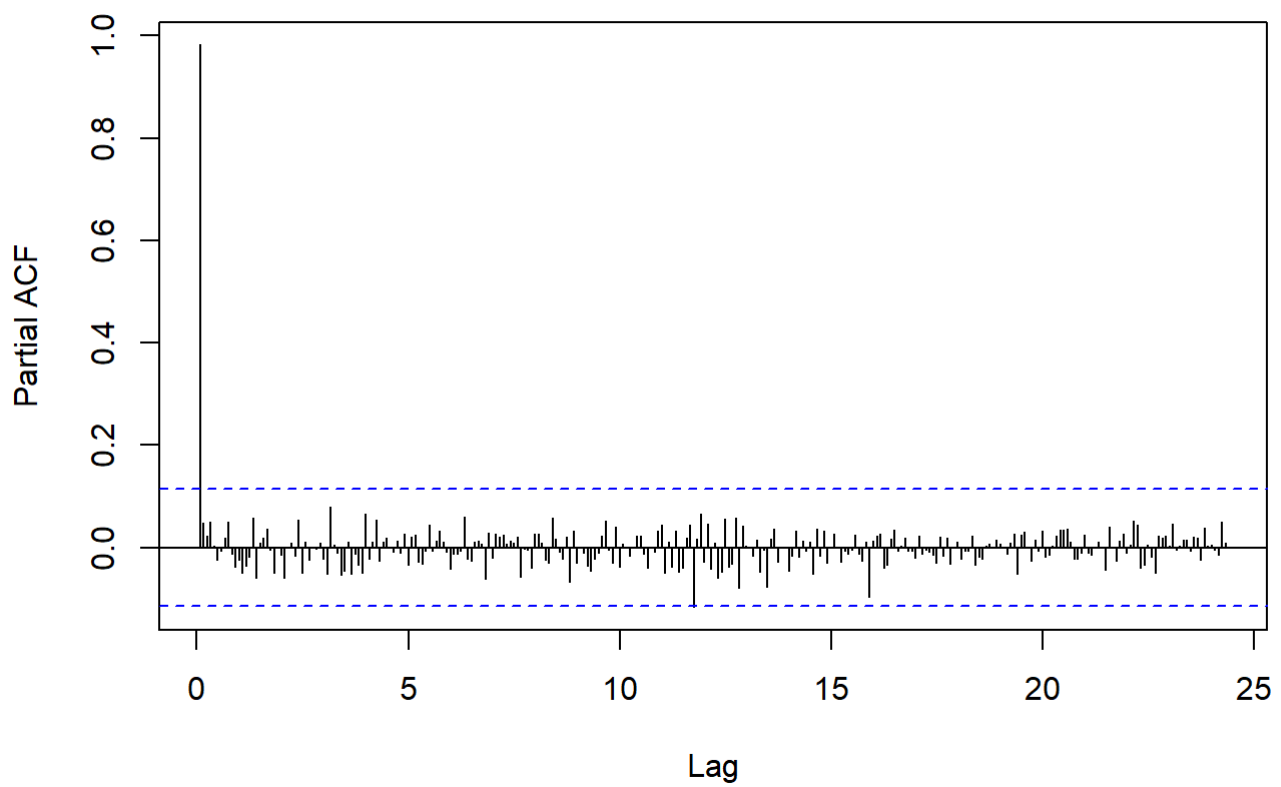


Figure 5: PACF plot of gold price data

```
## $adf_test
##
## Augmented Dickey-Fuller Test
##
## data: data
## Dickey-Fuller = -1.7657, Lag order = 6, p-value = 0.6749
## alternative hypothesis: stationary
##
##
## $pp_test
##
## Phillips-Perron Unit Root Test
##
## data: data
## Dickey-Fuller Z(alpha) = -7.4695, Truncation lag parameter = 5, p-value
## = 0.6915
## alternative hypothesis: stationary
```

The ACF plot in figure 4 shows a slowly decaying and wave pattern suggesting the presence of trend and seasonality in the series. The PACF plot in figure 5 shows 1st lag highly significant. Both ADF and PP test have a p-value > 0.05, which means that the null hypothesis can not be rejected at 5% significance level, indicating the series is non-stationary. Due to the presence of seasonality in the series, SARIMA models will be fitted.

Test of Normality

The following are checked to test for the normality in the data:

- Histogram
- Normal QQ plot
- Shapiro-Wilk test

```
Diagnostic_test(data.ts, mainhist = "Histogram of gold price series", subhist = "Figure 6: Histogram of gold price series", subqq = "Figure 7: QQ Plot of gold price series", test = "Normality")
```

Histogram of gold price series

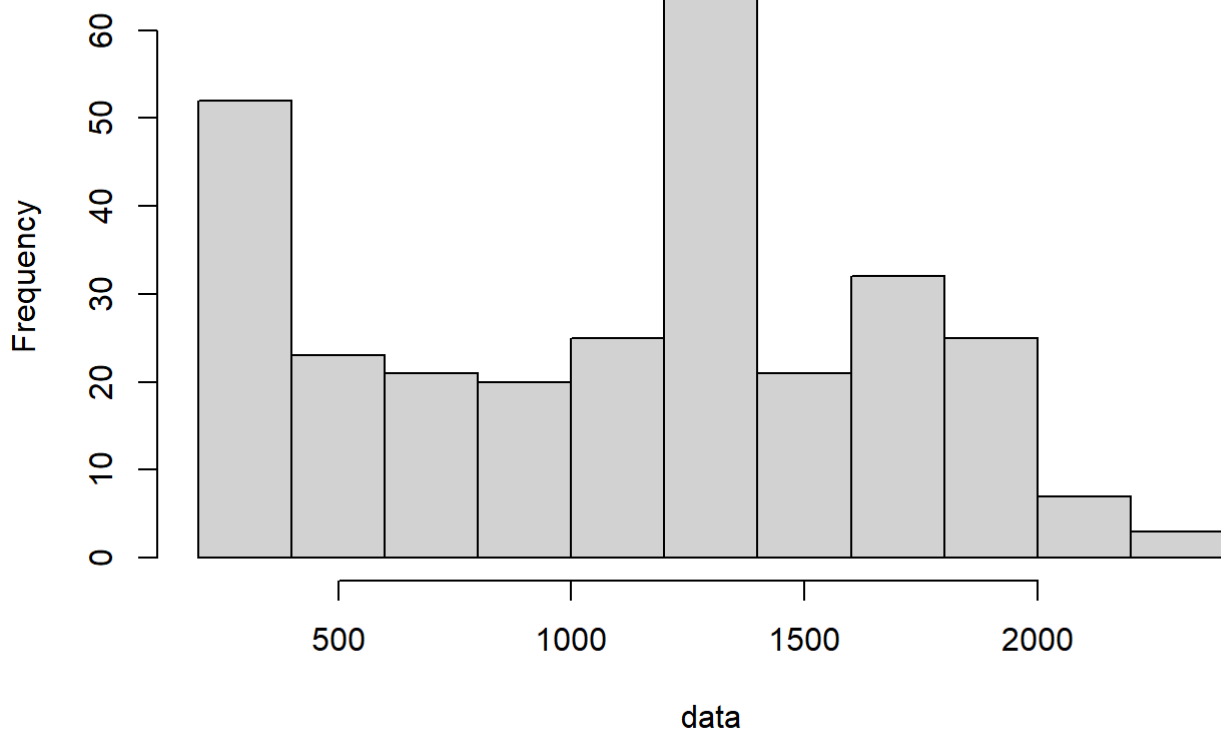


Figure 6: Histogram of gold price series

Q-Q Plot

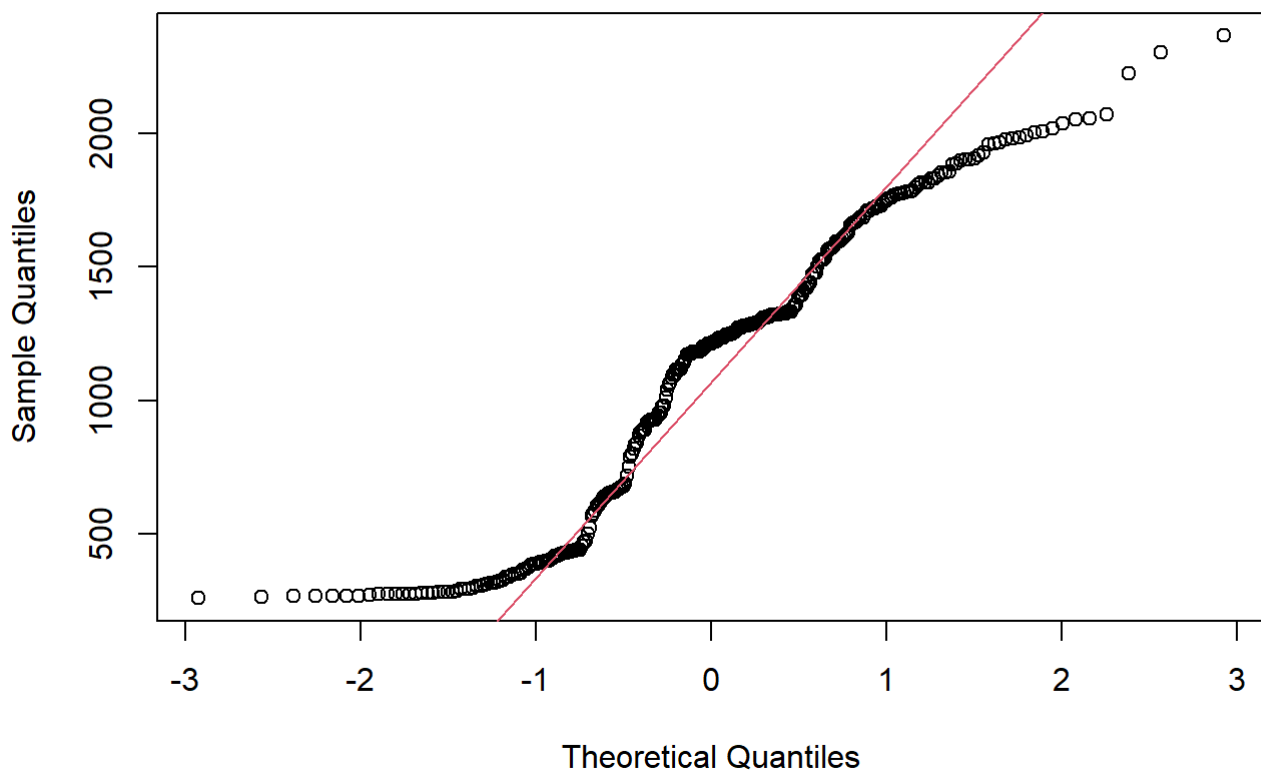


Figure 7: QQ Plot of gold price series

```
##
## Shapiro-Wilk normality test
##
## data: data
## W = 0.94027, p-value = 1.659e-09
```

The histogram in figure 6 is not symmetric, and there are outliers present in the series. The QQ-plot in figure 7 shows majority of data points deviate from the normality line and shows presence of outliers in the series. Likewise, the Shapiro test p value is < 0.05 therefore suggesting the data is not normally distributed.

To deal with normality, box transformation is applied, and to remove the trend and attain stationarity, differencing is applied.

Box-Cox Transformation

```
BC <- BoxCox.ar(y=data.ts, lambda=seq(-2, 2, 0.01))
mtext('Figure 8: Box-Cox Transformation: Log-Likelihood vs. Lambda.',line = 4, side = 1, cex
= 0.8)
```

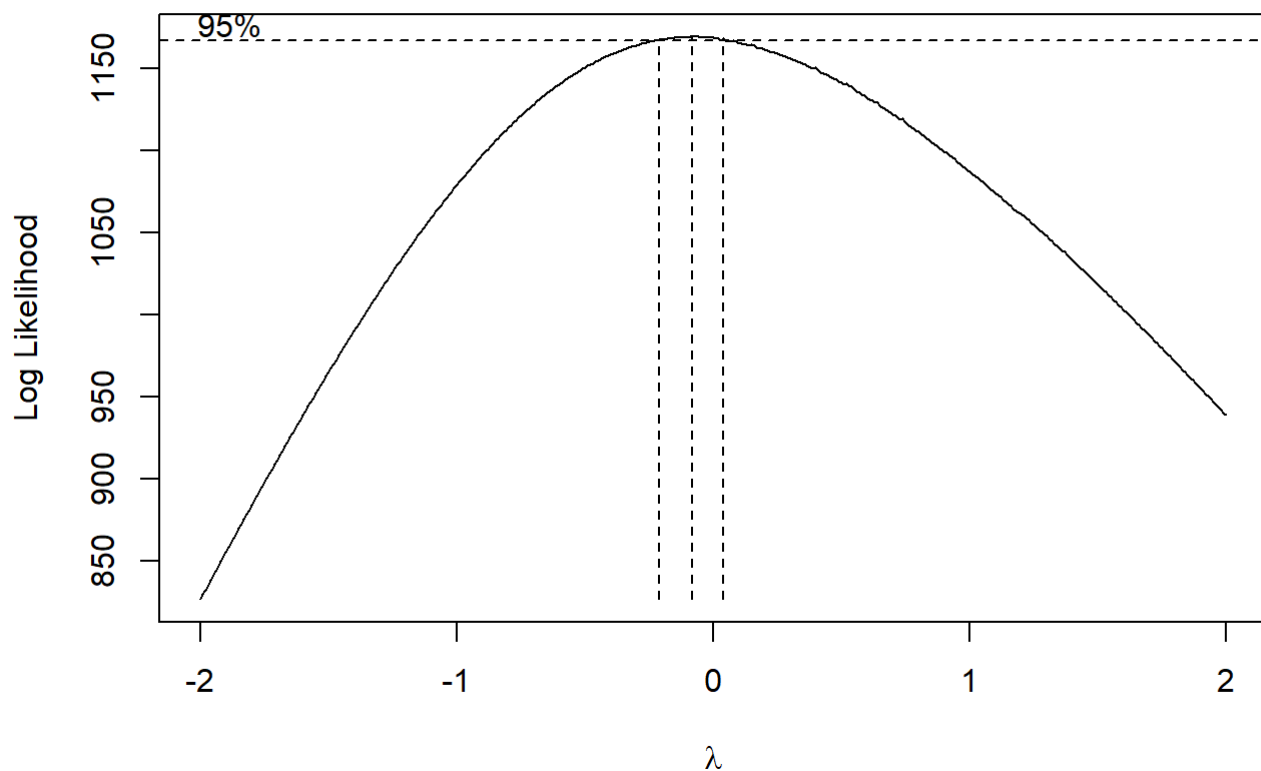


Figure 8: Box-Cox Transformation: Log-Likelihood vs. Lambda.

```
BC$ci
```

```
## [1] -0.21 0.04
```

```
# BoxCox Lambda value
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda
```

```
## [1] -0.08
```

```
BC.data = log(data.ts)
```

From figure 8, it can be observed the lambda value is -0.08 which is close to 0. Therefore, log transformation is applied on the series.

```
# Plot of normalised gold price series
plot(BC.data,type='o', pch = 20, ylab = "Normalised Gold Price", main='Log transformed gold p
rice series', sub = "Figure 9: Log transformed gold price series")
```

Log transformed gold price series

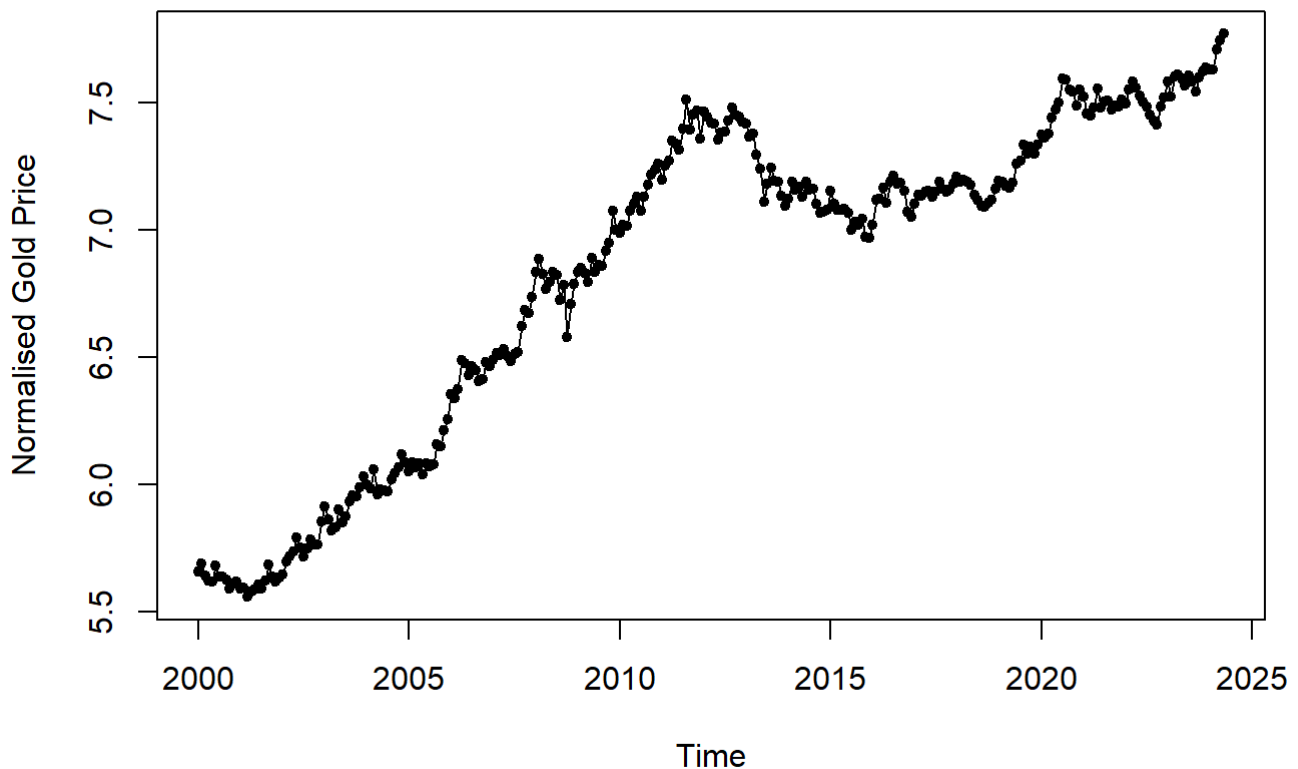


Figure 9: Log transformed gold price series

```
Diagnostic_test(BC.data, subhist = "Figure 10: Histogram of log transformed data", subqq = "F
igure 11: QQ plot of Log transformed gold price series", test = "Normality")
```

Histogram

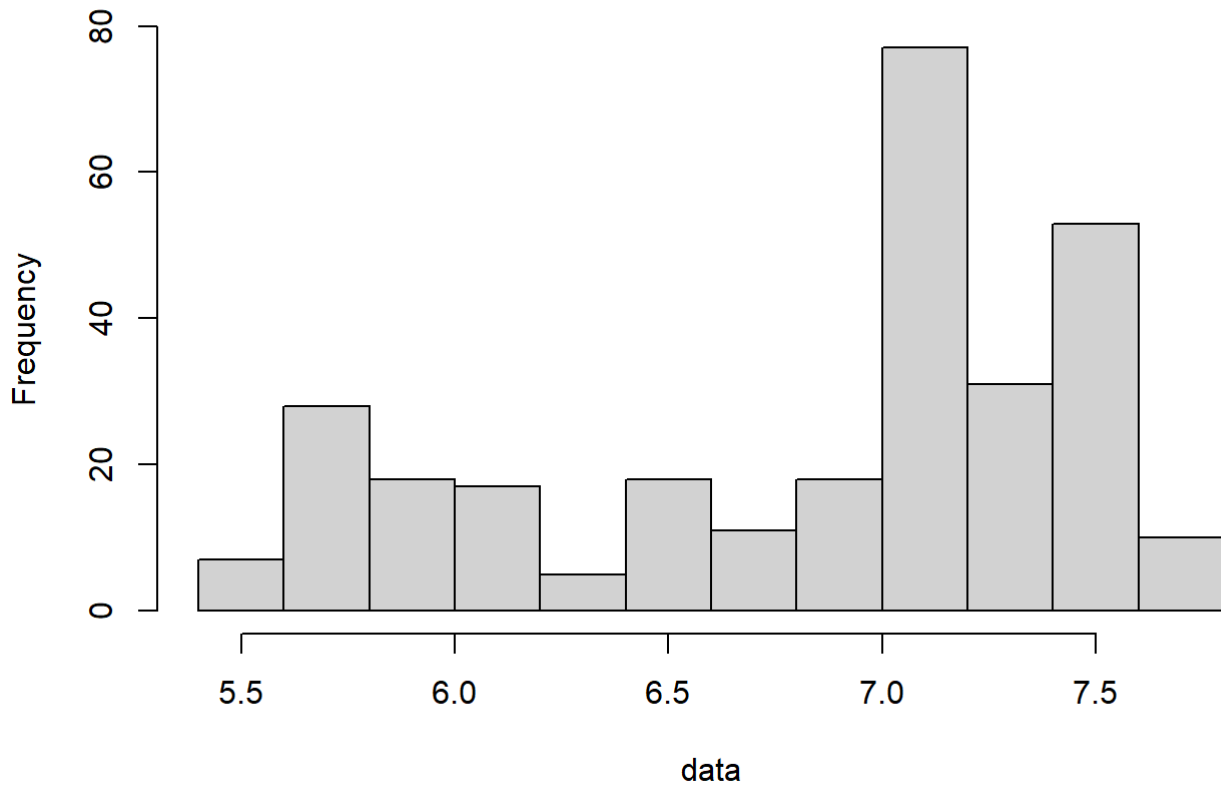


Figure 10: Histogram of log transformed data

Q-Q Plot

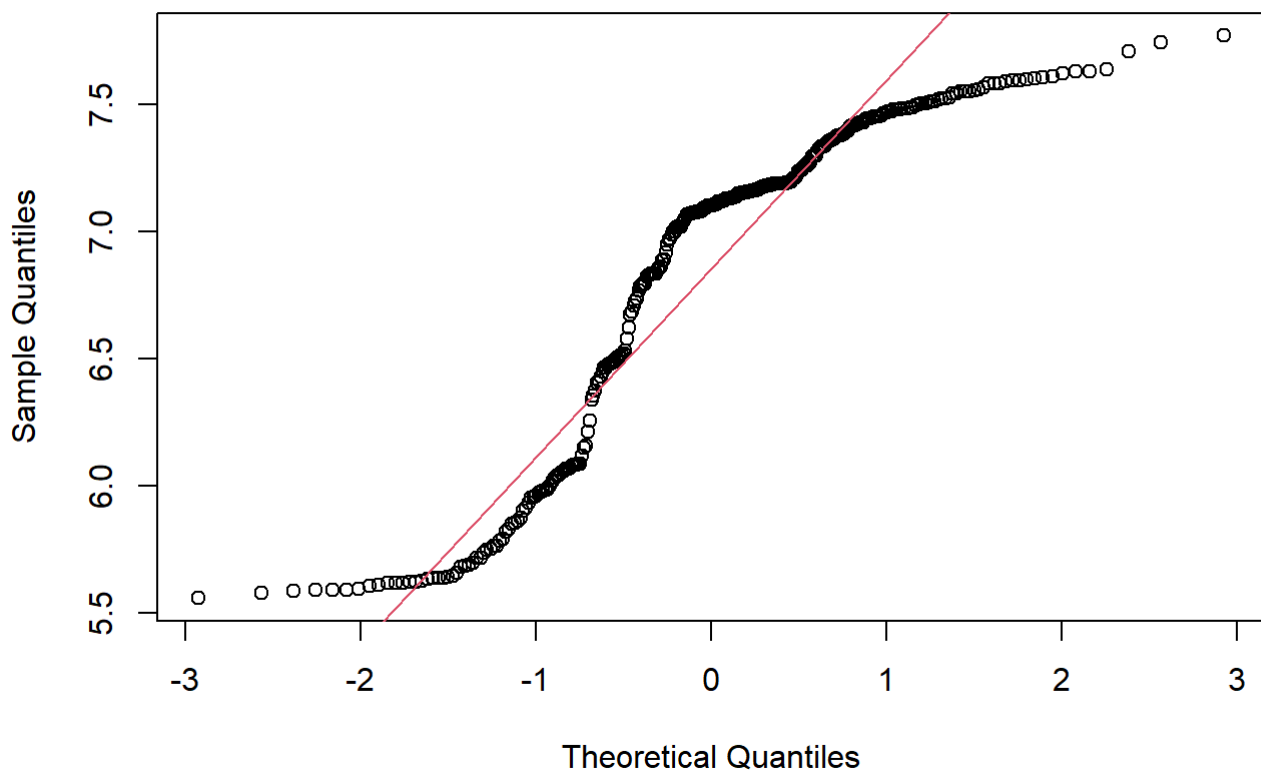


Figure 11: QQ plot of Log transformed gold price series


```
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.87904, p-value = 1.875e-14
```

After applying log transformation on the series, the series did not achieve normality. QQ-plot in figure 11 still shows high deviation of data points with respect to the line of normality. Further, Shapiro-Wilk's test of normality gives a p value < 0.05, and it is worse than that of untransformed series. Therefore, original data is used for further analysis.

Differencing

First differencing

```
diff.ts = diff(data.ts)
plot(diff.ts,type='o',ylab='First difference values', main ="Time series plot of the first di
fference of transformed gold price series.", sub = 'Figure 12: 1st difference series')
```

Time series plot of the first difference of transformed gold price series

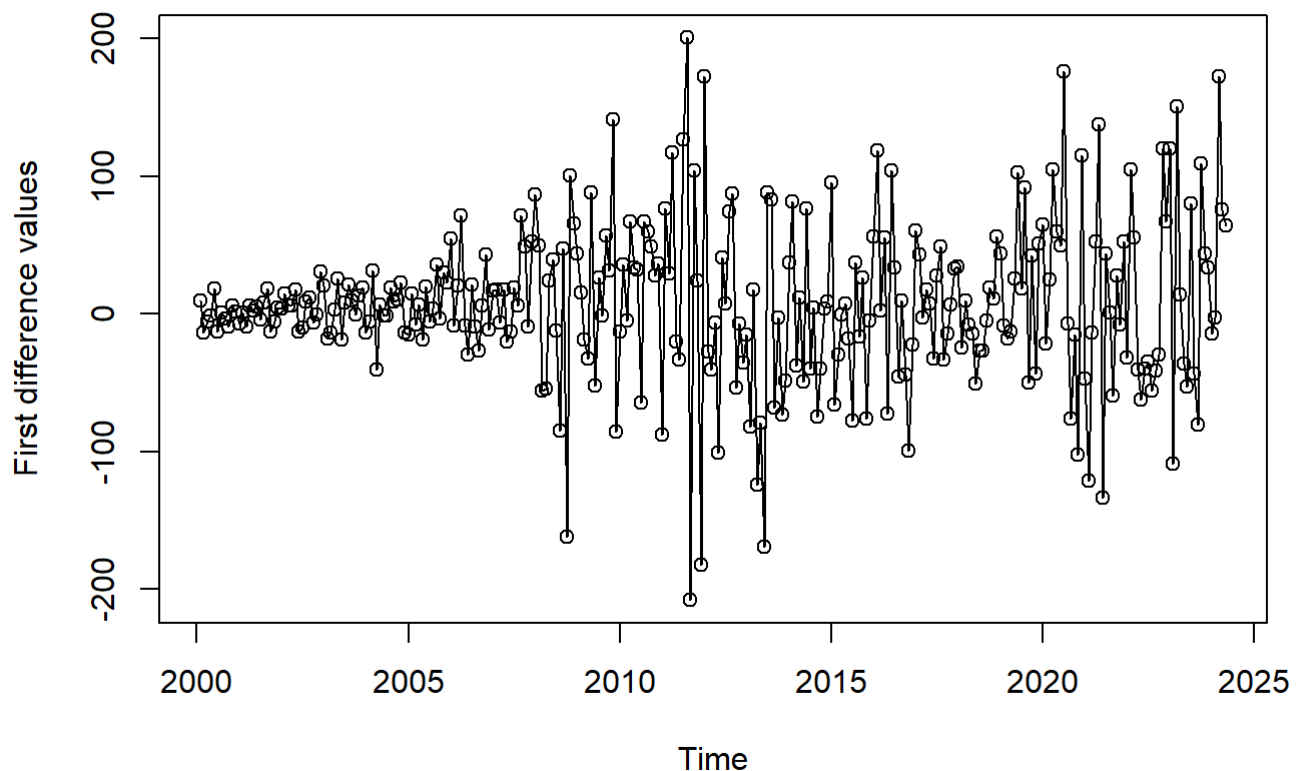


Figure 12: 1st difference series

Figure 12 shows the first difference plot of the gold price series. The plot has a flat mean level indicating no trend, however, the high fluctuations in the differenced series suggest presence of changing variance. McLeod Li test is performed to check for the changing variance in the differenced series.

```
McLeod.Li.test(y=diff.ts, main ="McLeod Li test for differenced series", sub = "Figure 13: Mc
Leod Li test for differenced series")
```

McLeod Li test for differenced series

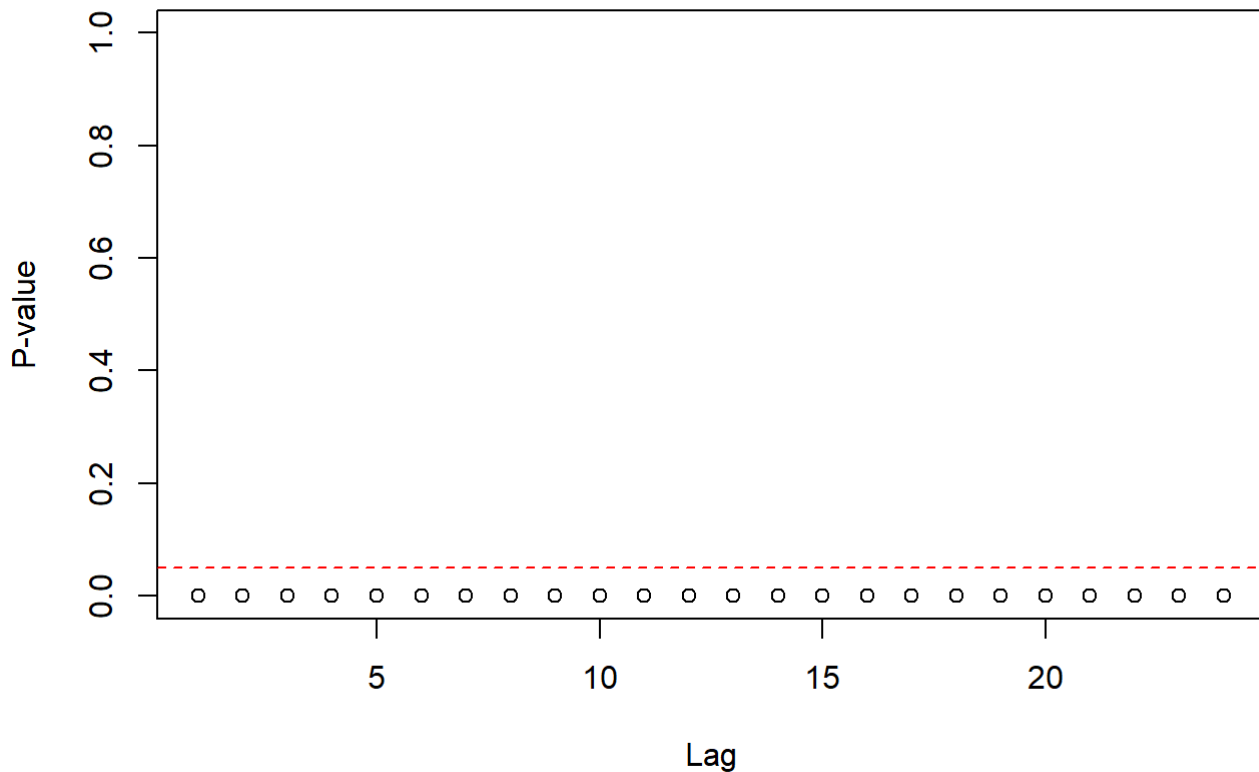


Figure 13: McLeod Li test for differenced series

Figure 13 displays the McLeod Li test for the first difference gold price series. Since the transformation does not help, so McLeod Li test is applied on first difference of gold price series. All the lags are beyond the 5% significant level which suggests that the series has changing variance. Due to changing variance, ARMA-GARCH models will be fitted.

First difference series stationary test

```
Diagnostic_test(diff.ts, 100, mainacf = "ACF for First differenced Gold price series", subacf = "Figure 14: ACF plot of first difference gold price time series", mainpacf = "PACF for First differenced Gold price series", subpacf = "Figure 15: PACF plot of first difference gold price time series", test = "Stationary")
```

ACF for First differenced Gold price series

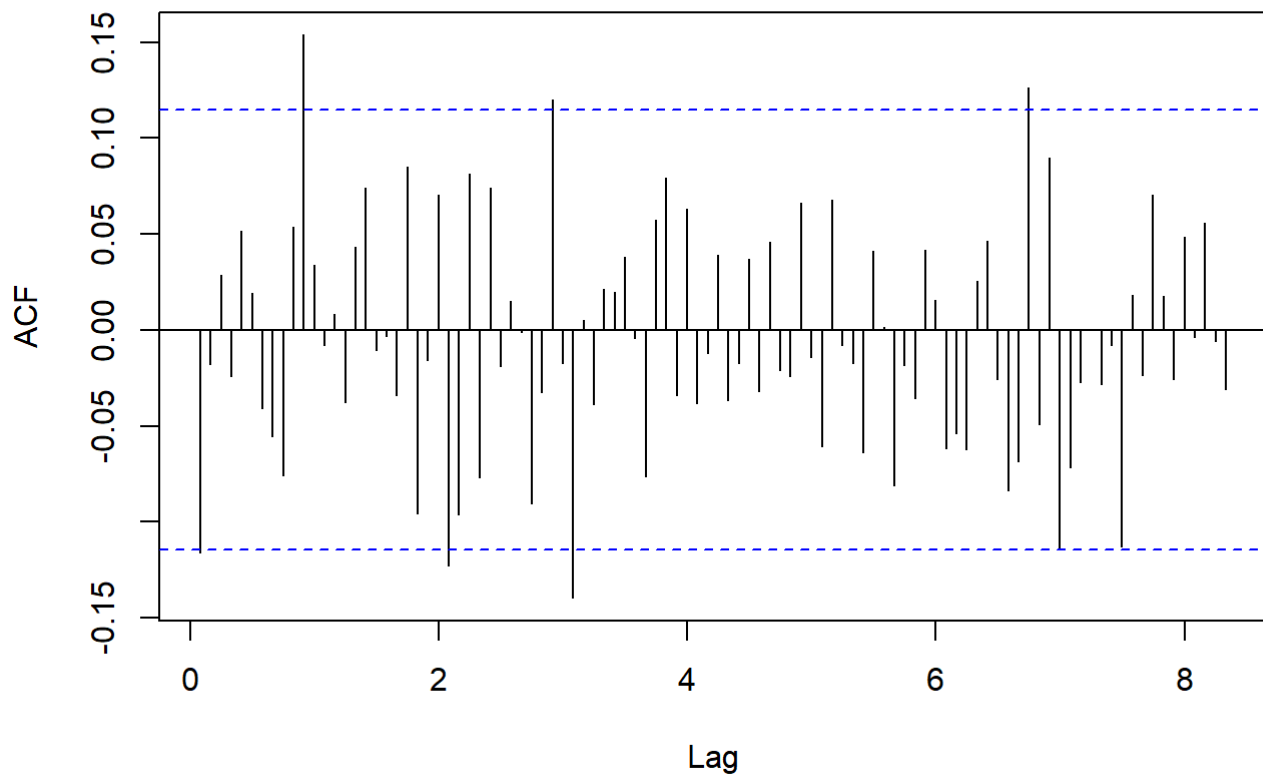


Figure 14: ACF plot of first difference gold price time series

PACF for First differenced Gold price series

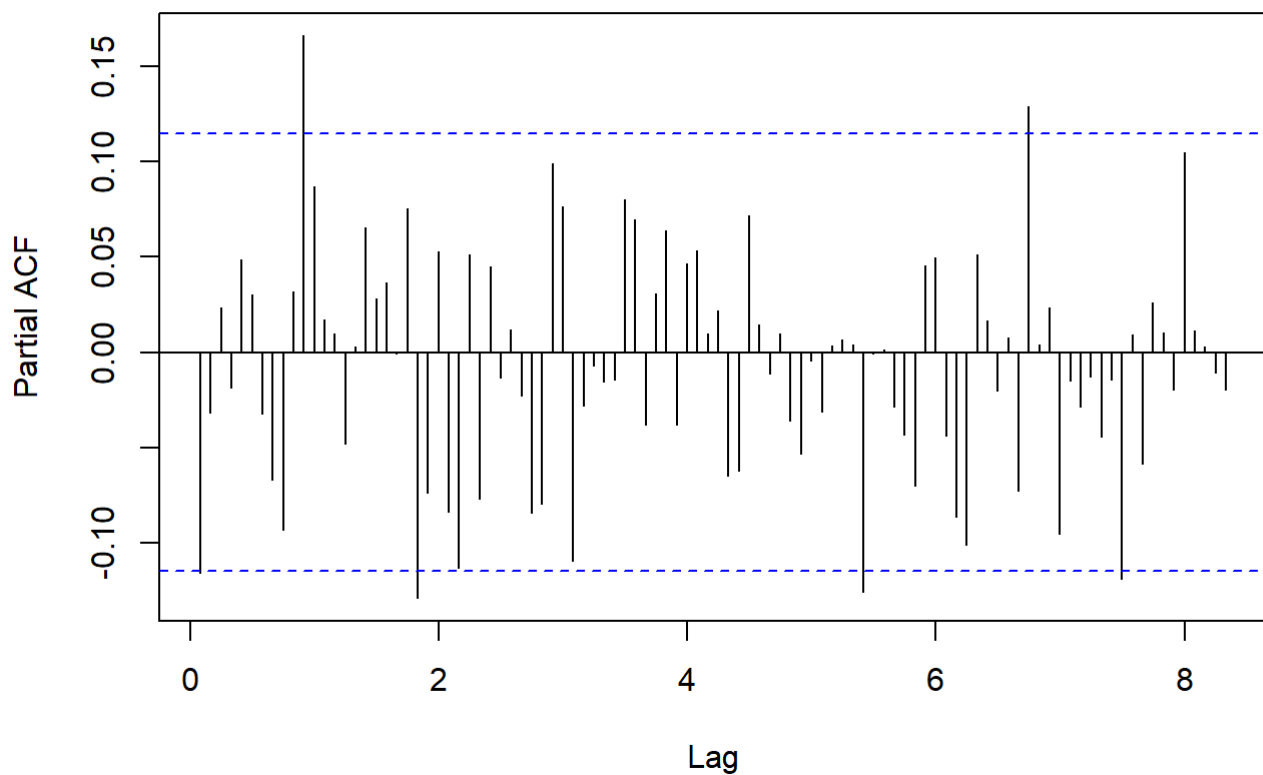


Figure 15: PACF plot of first difference gold price time series

```
## $adf_test
##
## Augmented Dickey-Fuller Test
##
## data: data
## Dickey-Fuller = -6.0229, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $pp_test
##
## Phillips-Perron Unit Root Test
##
## data: data
## Dickey-Fuller Z(alpha) = -322.81, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

The ACF plot in figure 14 doesn't show any pattern, the PACF plot in figure 15 doesn't show any highly significant lag suggesting stationary in the series.

ADF and PP test have a p-value < 0.05 , suggesting the null hypothesis can be rejected at 5% significance level, indicating stationary.

Stationarity is achieved by first difference of gold price series. Consequently, for ARIMA(p,d,q) models, 1st difference will be used.

ARIMA models

The following 3 methods are used to identify potential models.

- Significant lags from ACF and PACF plots
- EACF plot
- BIC table

Significant lags in ACF and PACF plot

From figure 14, ACF shows 1 significant lag (late lags are ignored), so $q = 1$. From figure 15, PACF shows 1 lag significant (late lags are ignored), so $p = 1$

Model identified from ACF and PACF plot of the differenced series:

- ARIMA(1,1,1)

EACF plot

```
eacf(diff.ts, ar.max = 5, ma.max = 5)
```

```
## AR/MA
##   0 1 2 3 4 5
## 0 o o o o o o
## 1 x o o o o o
## 2 x o o o o o
## 3 x o o o o o
## 4 x x x x o o
## 5 x x x x o o
```

The top-left “o” identified in the EACF plot is (0,0)

Neighbor models:

- ARIMA(0,1,0)
- ARIMA(0,1,1)
- ARIMA(1,1,1)

There are no AR/MA subsets in ARIMA(0,1,0), so is not considered for analysis.

BIC plot

```
res = armasubsets(y=diff.ts,nar=4,nma=4,y.name='p',ar.method='ols')
```

```
## Reordering variables and trying again:
```

```
plot(res)
mtext('Figure 16:BIC Table',line = 4, side = 1, cex = 0.8)
```

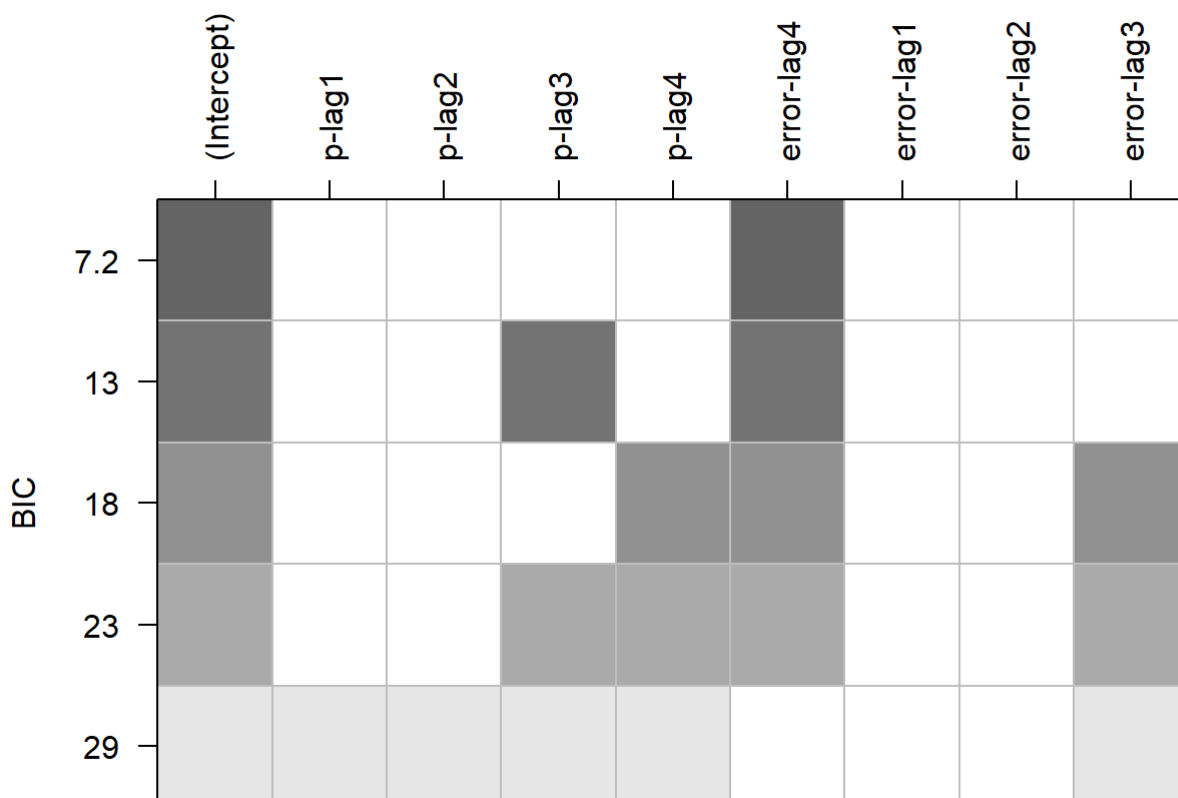


Figure 16:BIC Table

From BIC table following model the top model has the lowest BIC value(7.2):

- ARIMA(0,1,4)

Parameter Estimation

ARIMA(0,1,1)

```
model.011 = Arima(data.ts,order=c(0,1,1),method='ML')
coeftest(model.011)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.100260   0.057853  -1.733  0.08309 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.011css = Arima(data.ts,order=c(0,1,1),method='CSS')
coeftest(model.011css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.100595   0.057945 -1.7361  0.08255 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No coefficients are significant for model.011 at 5% significance level.

ARIMA(1,1,1)

```
model.111 = Arima(data.ts,order=c(1,1,1),method='ML')
coeftest(model.111)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.030240   0.433075 -0.0698   0.9443
## ma1 -0.071377   0.431113 -0.1656   0.8685
```

```
model.111css = Arima(data.ts,order=c(1,1,1),method='CSS')
coeftest(model.111css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.047353   0.434847 -0.1089   0.9133
## ma1 -0.052988   0.435171 -0.1218   0.9031
```

No coefficients are significant for model.111

ARIMA(0,1,4)

```
model.014 = Arima(data.ts,order=c(0,1,4),method='ML')
coeftest(model.014)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.1022941   0.0593224 -1.7244   0.08464 .
## ma2 -0.0054222   0.0598917 -0.0905   0.92786
## ma3  0.0415385   0.0596254  0.6967   0.48602
## ma4  0.0113208   0.0662584  0.1709   0.86434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.014css = Arima(data.ts,order=c(0,1,4),method='CSS')
coeftest(model.014css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.1026921   0.0594365 -1.7278   0.08403 .
## ma2 -0.0055445   0.0601135 -0.0922   0.92651
## ma3  0.0420180   0.0599702  0.7006   0.48352
## ma4  0.0116994   0.0668821  0.1749   0.86114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No coefficients are significant for model.014

Model.011 is the best model based in coefficient testing.

AIC and BIC values

The AIC and BIC scores are calculated using “ML” methods.

```
aic_table =AIC(model.011, model.111, model.014)
bic_table =BIC(model.011, model.111, model.014)

sorted_aic_table <- aic_table[order(aic_table$AIC), ]
sorted_bic_table <- bic_table[order(bic_table$BIC), ]

sorted_aic_table
```

```
##           df      AIC
## model.011  2 3200.211
## model.111  3 3202.208
## model.014  5 3205.699
```

```
sorted_bic_table
```

```
##           df      BIC
## model.011  2 3207.564
## model.111  3 3213.238
## model.014  5 3224.083
```

From the AIC and BIC score, model.011 is the best model.

Error measures

```
Smodel.011 = accuracy(summary(model.011))
Smodel.111 = accuracy(summary(model.111))
Smodel.014 = accuracy(summary(model.014))

headers <- c("ME", "RMSE", "MAE", "MPE", "MAPE", "MASE", "ACF1")

merged_table <- data.frame(rbind
  ( c(Smodel.011), c(Smodel.111),c(Smodel.014)))
model_names <- c("model.011", "model.111","model.014")

rownames(merged_table) <- model_names
colnames(merged_table) <- headers
merged_table
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## model.011 7.865242 57.52952 41.24505 0.6799457 3.656716 0.2946539 -0.01928541
## model.111 7.852654 57.52924 41.25099 0.6787429 3.657211 0.2946964 -0.01795060
## model.014 7.547298 57.47852 41.28719 0.6491016 3.657498 0.2949550 -0.01645941
```

The error metrics shows very close values for all the models.

Based on coef_test, AIC, BIC, and error metrics, model.011 is the best model.

Over-parameterisation

Parameter tuning is done to identify any further potential models.

The following models will be tested under parameter tuning:

- ARIMA(0,1,2)
- ARIMA(1,1,1)

We have already tested the ARIMA(1,1,1) model and concluded that it wasn't a significant model.

Parameter estimation is conducted on ARIMA(0,1,2) model.

ARIMA(0,1,2)

```
model.012 = Arima(data.ts,order=c(0,1,2),method='ML')
coeftest(model.012)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.1010029  0.0589617 -1.7130  0.08671 .
## ma2  0.0043249  0.0586188  0.0738  0.94119
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.012css = Arima(data.ts,order=c(0,1,2),method='CSS')
coeftest(model.012css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.1013471  0.0590650 -1.7159  0.08619 .
## ma2  0.0043658  0.0588095  0.0742  0.94082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No coefficients are significant for model.012 at 5% level. Hence, we can ignore this model.

ARIMA models gives us model.011 as the best option, however, we will try different approaches to get the best model.

SARIMA models

ACF and PACF plot

Plotting ACF(figure 3) and PACF(figure 4) of gold price series again

ACF for Gold price series

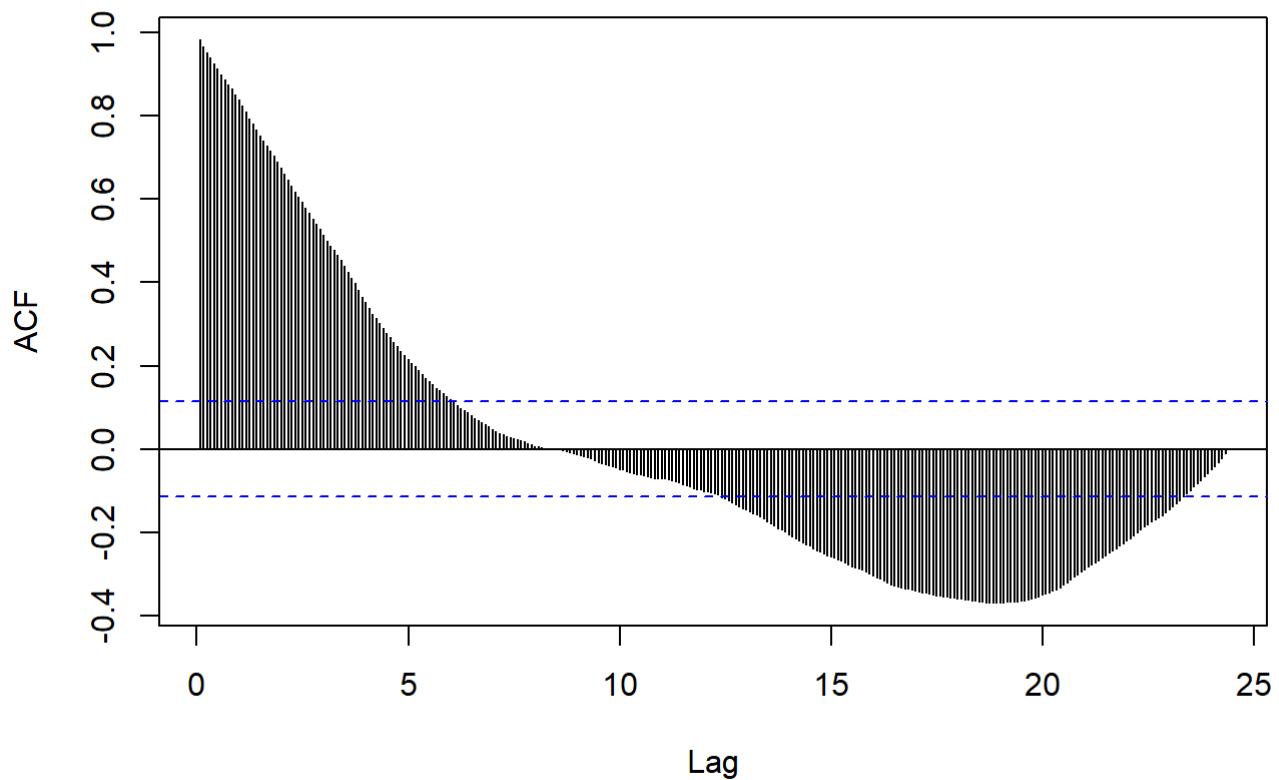


Figure 3: ACF plot of gold price time series

PACF for Gold price series

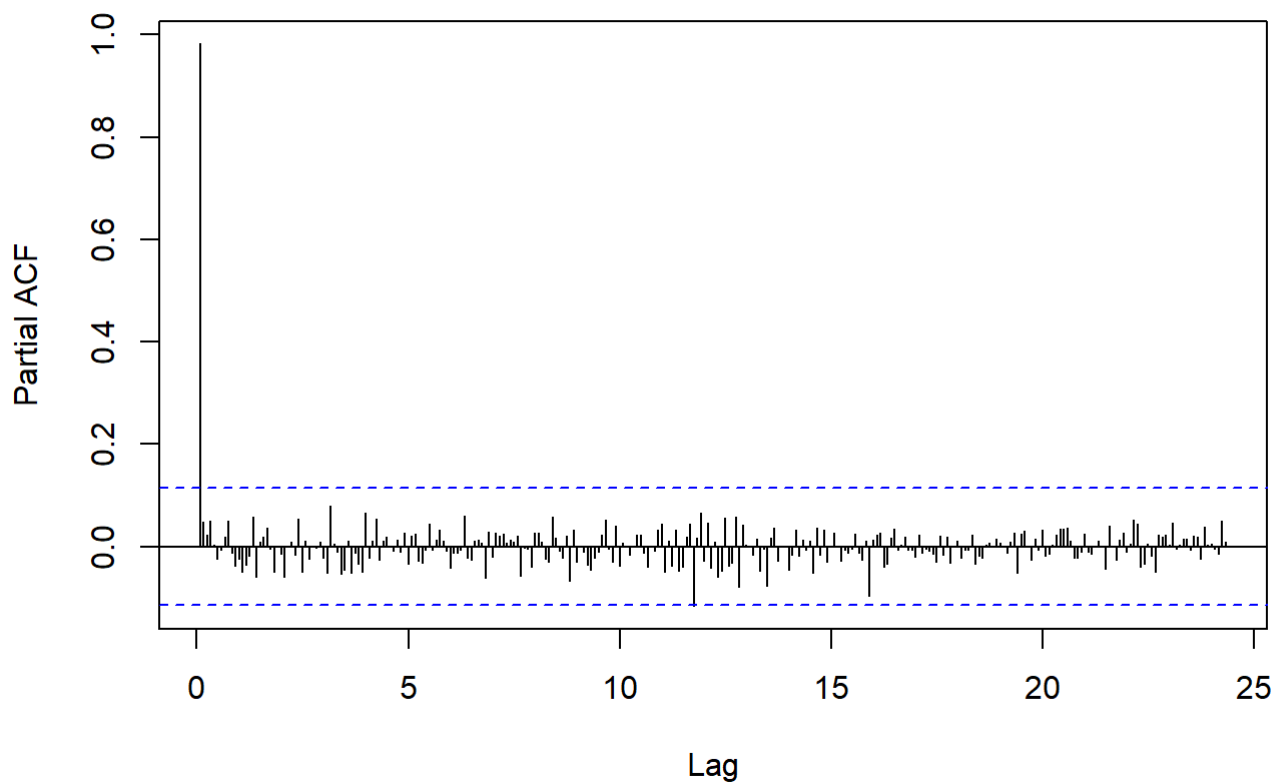


Figure 4: PACF plot of gold price time series

The ACF plots in figure 3 revealed presence of trend and seasonality, SARIMA models are fitted on the series.

To remove the trend, 1st seasonal difference is applied

Seasonal Difference (D=1)

```
# Plain model fit with 1st differencing
m1.gold = Arima(data.ts,order=c(0,0,0), seasonal=list(order=c(0,1,0), period=12))

# m1 residuals
res.m1 = rstandard(m1.gold)
plot(res.m1,xlab='Time',ylab='Residuals', main="Time series plot of the residuals.", sub = "Figure 17: Time series of residuals")
```

Time series plot of the residuals.

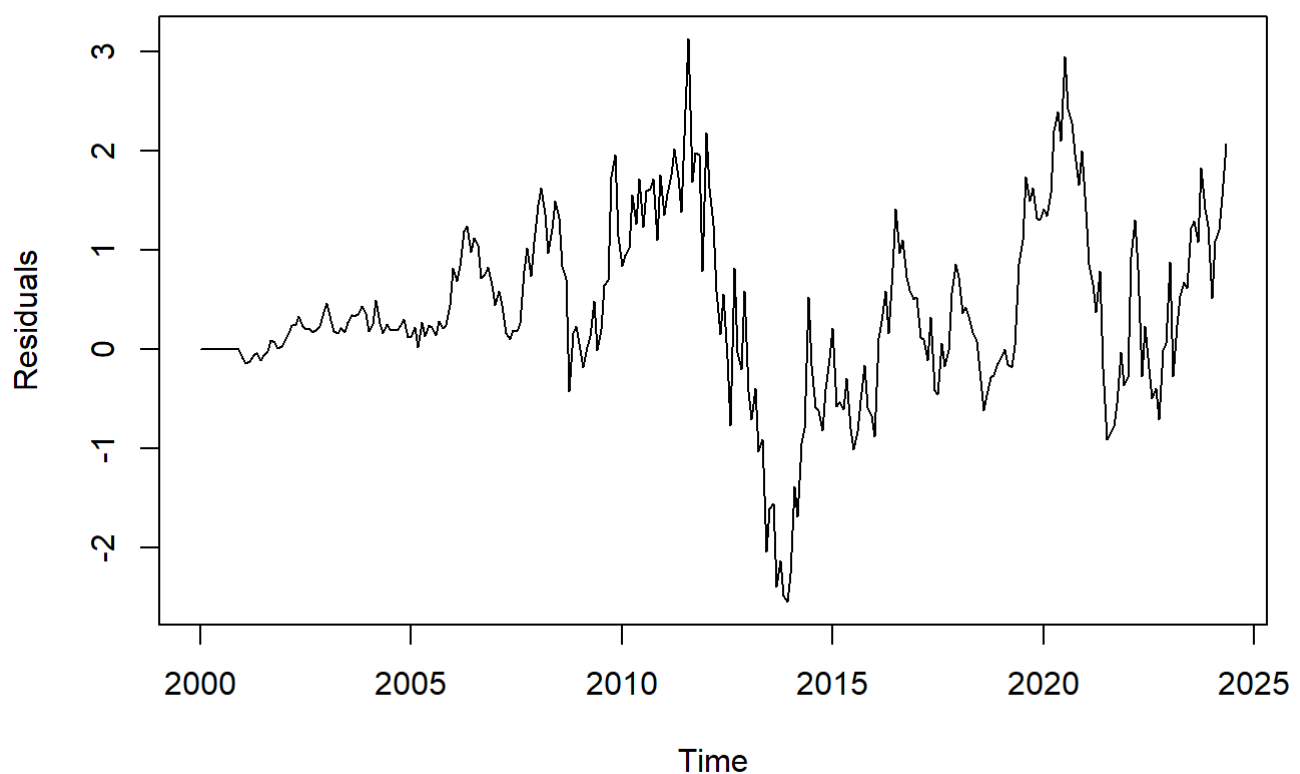


Figure 17: Time series of residuals

```
Diagnostic_test(res.m1, 48, mainacf = "The sample ACF of the residuals", subacf = "Figure 18:  
Sample ACF plot of residulas", mainpacf = "The sample PACF of the residuals", subpacf = "Figure 19:  
Sample PACF plot of residuals", test = "seasonal Stationary")
```

The sample ACF of the residuals (Seasonal)

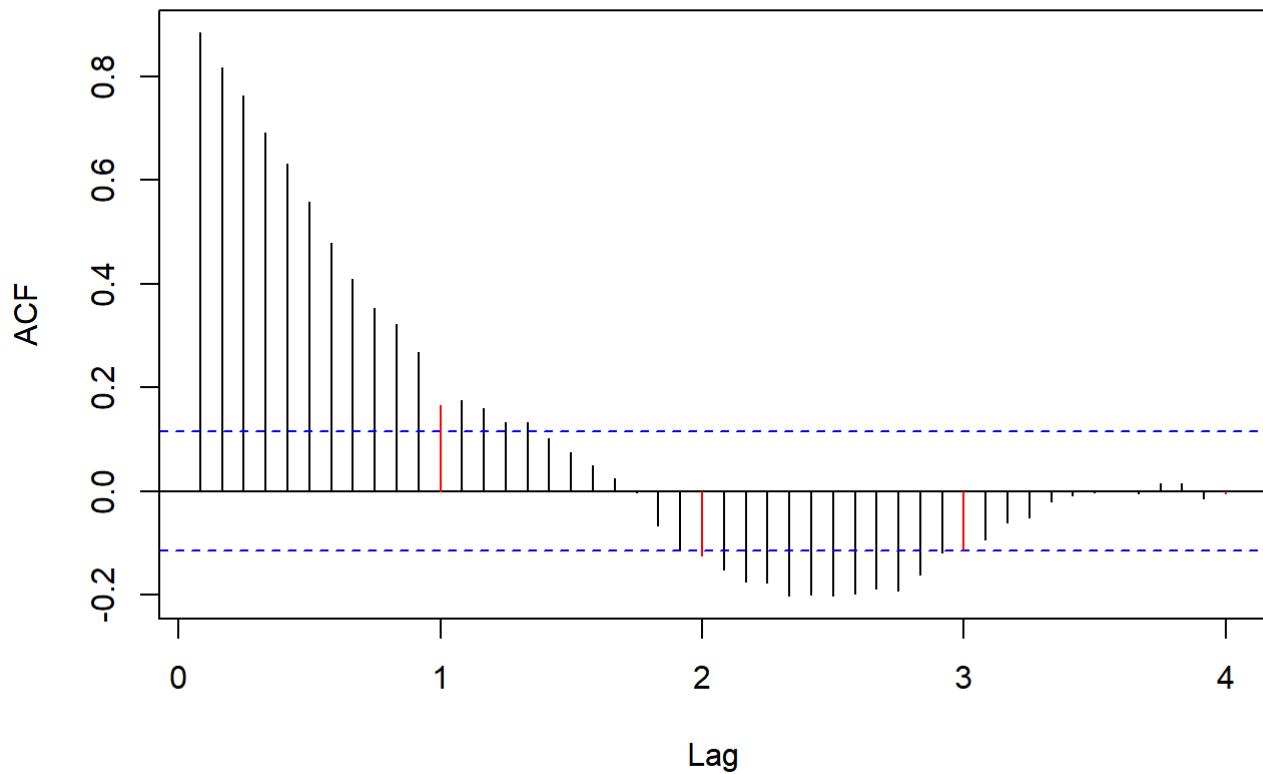


Figure 18: Sample ACF plot of residulas

The sample PACF of the residuals (Seasonal)

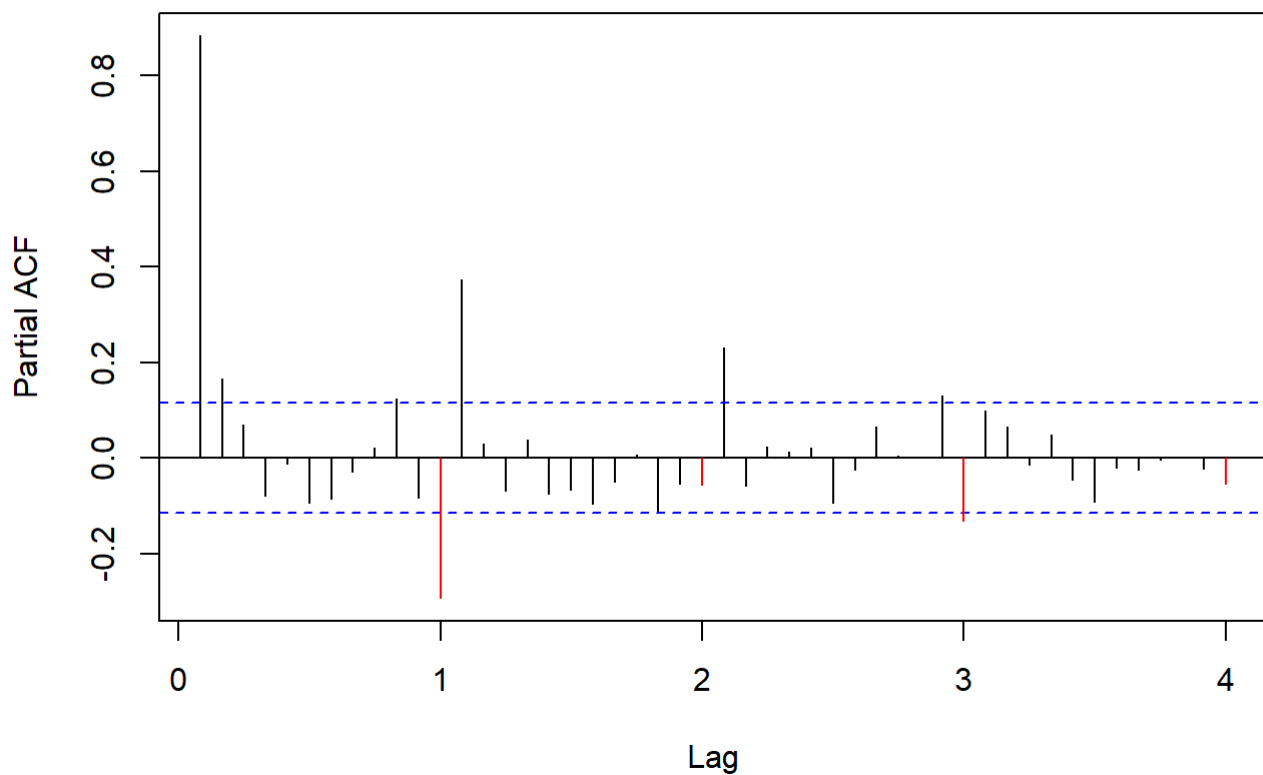


Figure 19: Sample PACF plot of residuals

```
## $adf_test
##
## Augmented Dickey-Fuller Test
##
## data: data
## Dickey-Fuller = -3.5707, Lag order = 6, p-value = 0.03631
## alternative hypothesis: stationary
##
##
## $pp_test
##
## Phillips-Perron Unit Root Test
##
## data: data
## Dickey-Fuller Z(alpha) = -28.533, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

The first seasonal difference residual plot(Figure 17) still has trend and fluctuations.

The 1st seasonal difference fitted series has a trend, but, the series is stationary as suggested by ADF and PP test. The PACF plot has 1st seasonal lag significant, so $P = 1$. The ACF plot has 1st, and 2nd seasonal lag significant, so, $Q=2$ is applied to in the model to get rid of seasonal component.

Seasonal parameter ((P,Q) = (1,2))

```
m2.gold = Arima(data.ts,order=c(0,0,0),seasonal=list(order=c(1,1,2), period=12, method="ML"))

res.m2 = rstandard(m2.gold)

plot(res.m2,xlab='Time',ylab='Residuals',main="Time series plot of the residuals.",sub = "Figure m2.1: Time series of residuals")
```

Time series plot of the residuals.

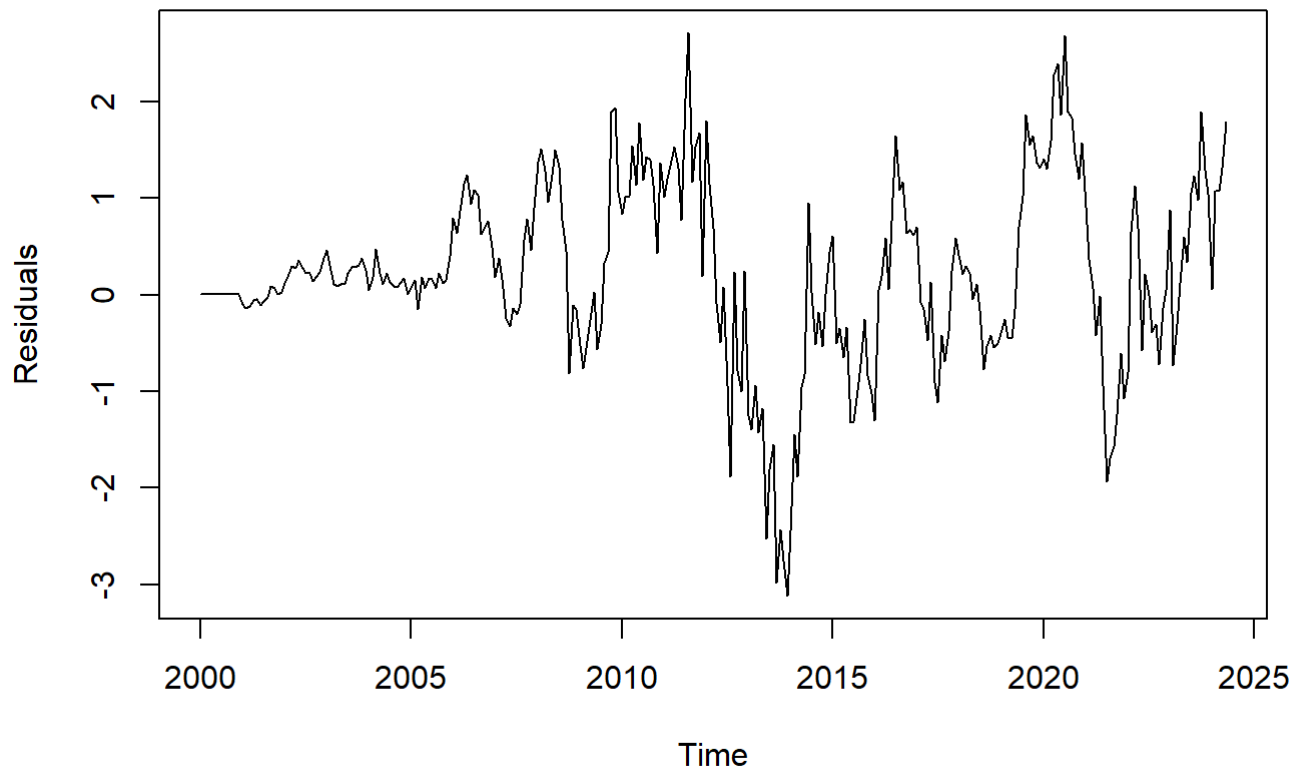


Figure m2.1: Time series of residuals

```
Diagnostic_test(res.m2, 48, mainacf = "The sample ACF of the residuals", subacf = "Figure 20:  
Sample ACF plot of residulas", mainpacf = "The sample PACF of the residuals", subpacf = "Figu  
re 21: Sample PACF plot of residuals", test = "seasonal ACF-PACF")
```

The sample ACF of the residuals (Seasonal)

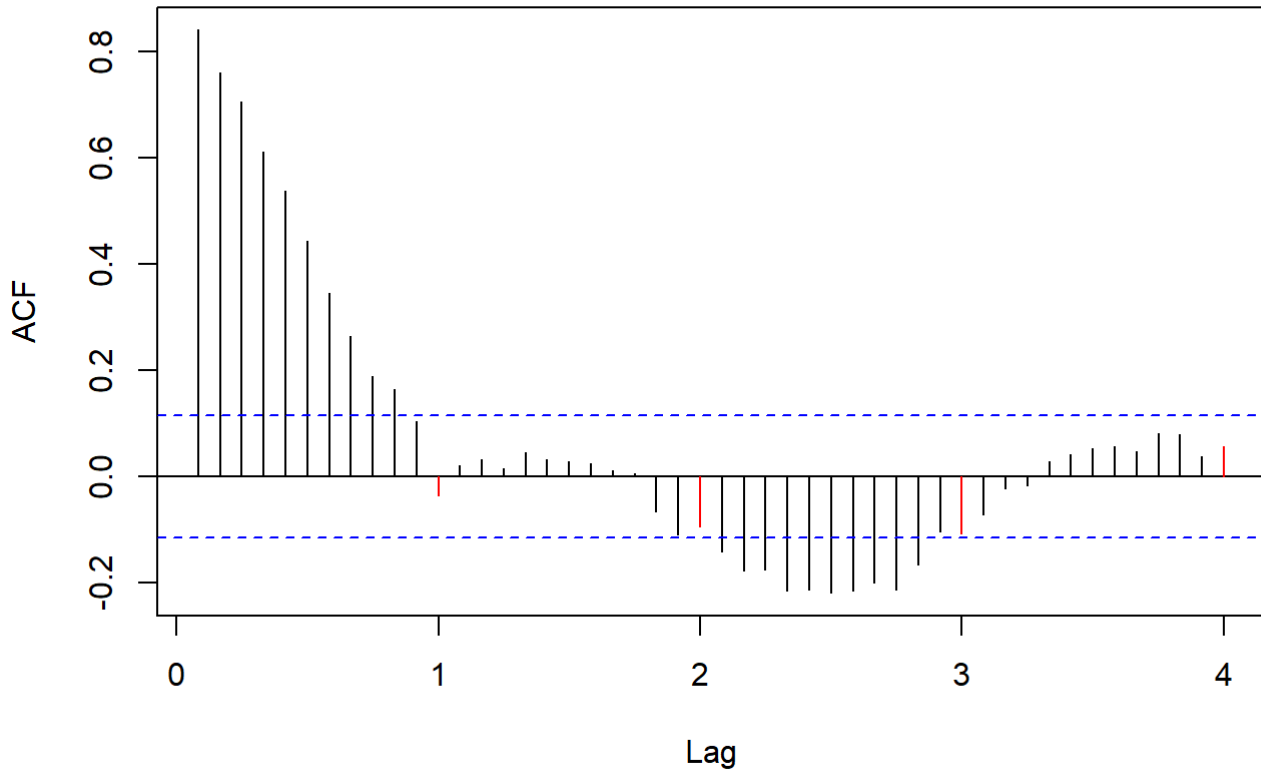


Figure 20: Sample ACF plot of residulas

The sample PACF of the residuals (Seasonal)

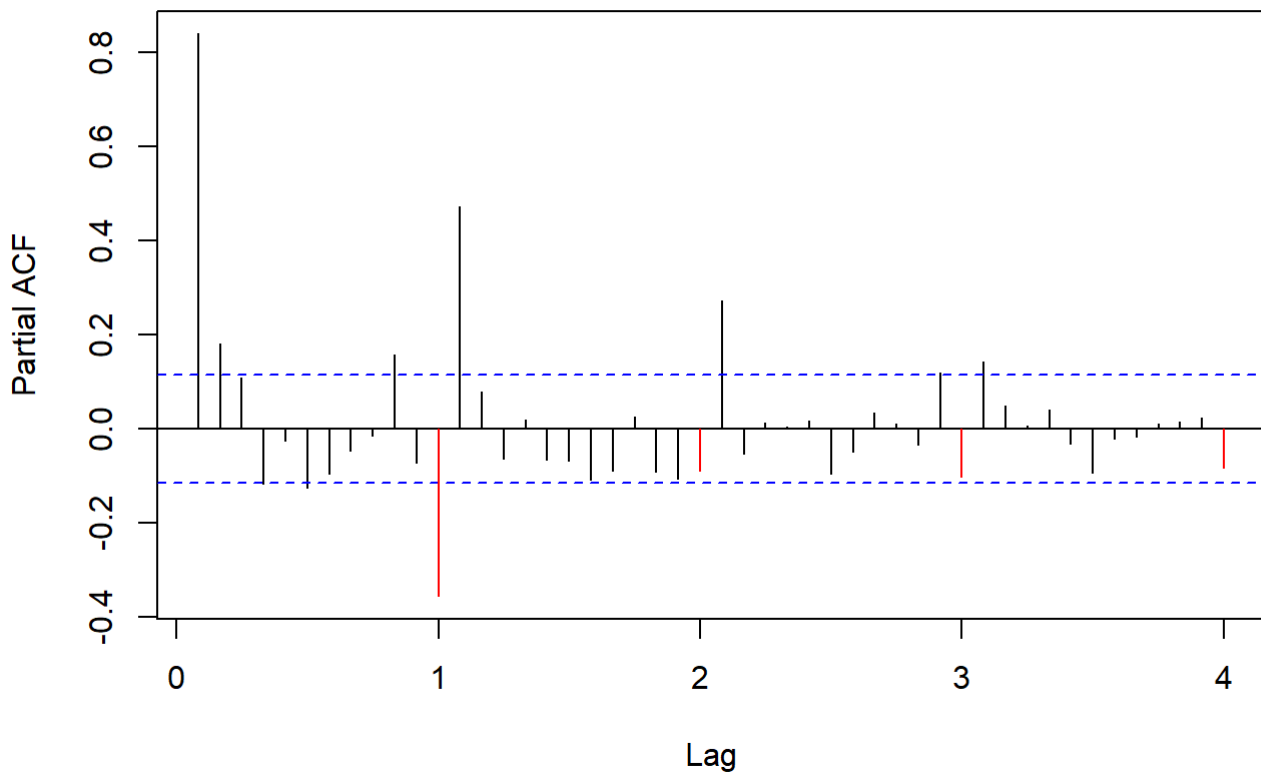


Figure 21: Sample PACF plot of residuals

The seasonal parameter fitted residual plot(Figure m2.1) still has trend and fluctuations.

There is a high auto correlation at the first lag of PACF, and nearly all the auto correlations are significant before the first seasonal lag in ACF. So, first ordinary difference is applied to remove this trend.

Ordinal difference (d=1)

```
m3.gold = Arima(data.ts, order=c(0,1,0), seasonal= list(order=c(1,1,2), period=12, method="ML"))

res.m3 = rstandard(m3.gold)
plot(res.m3,xlab='Time',ylab='Residuals',main="Time series plot of the residuals.", sub = "Figure m3.1: Time series of residuals")
```

Time series plot of the residuals.

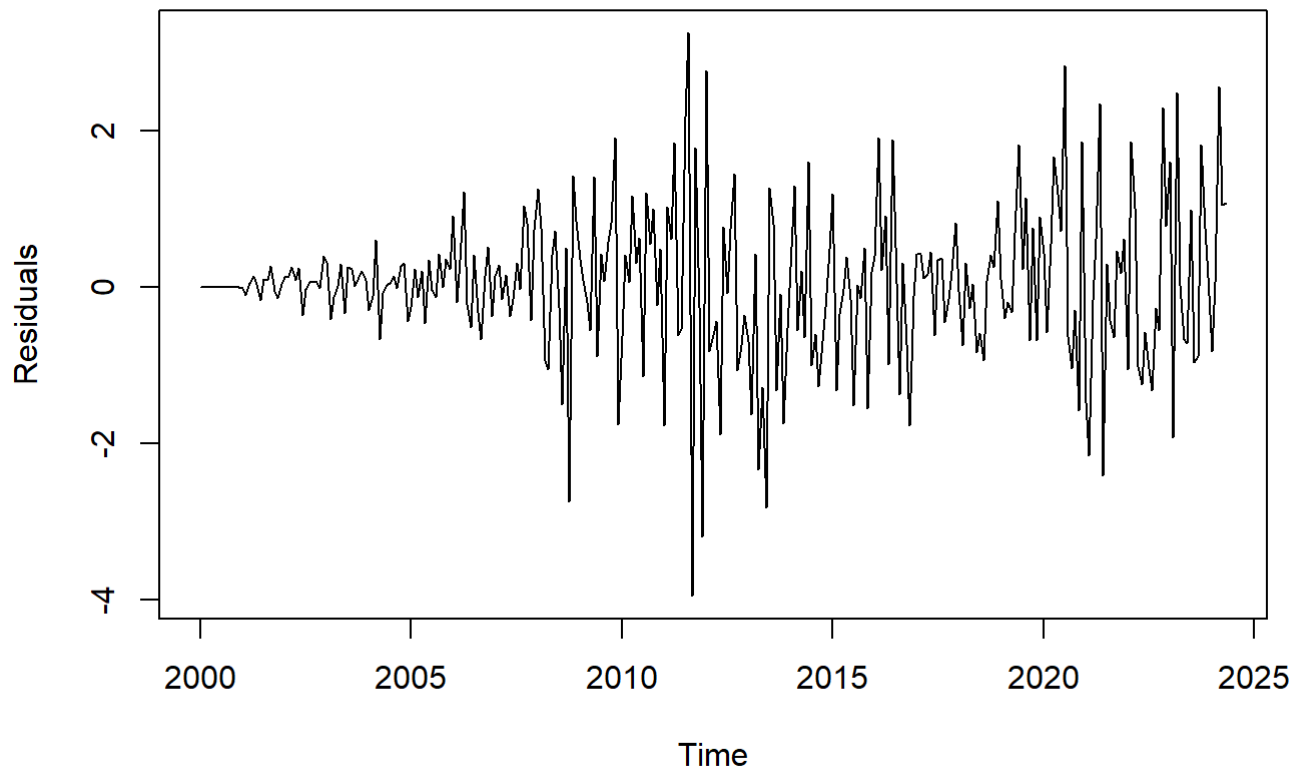


Figure m3.1: Time series of residuals

```
Diagnostic_test(res.m3, 48, mainacf = "The sample ACF of the residuals", subacf = "Figure 22: Sample ACF plot of residulas", mainpacf = "The sample PACF of the residuals", subpacf = "Figure 23: Sample PACF plot of residuals", test = "seasonal ACF-PACF")
```


The sample ACF of the residuals (Seasonal)

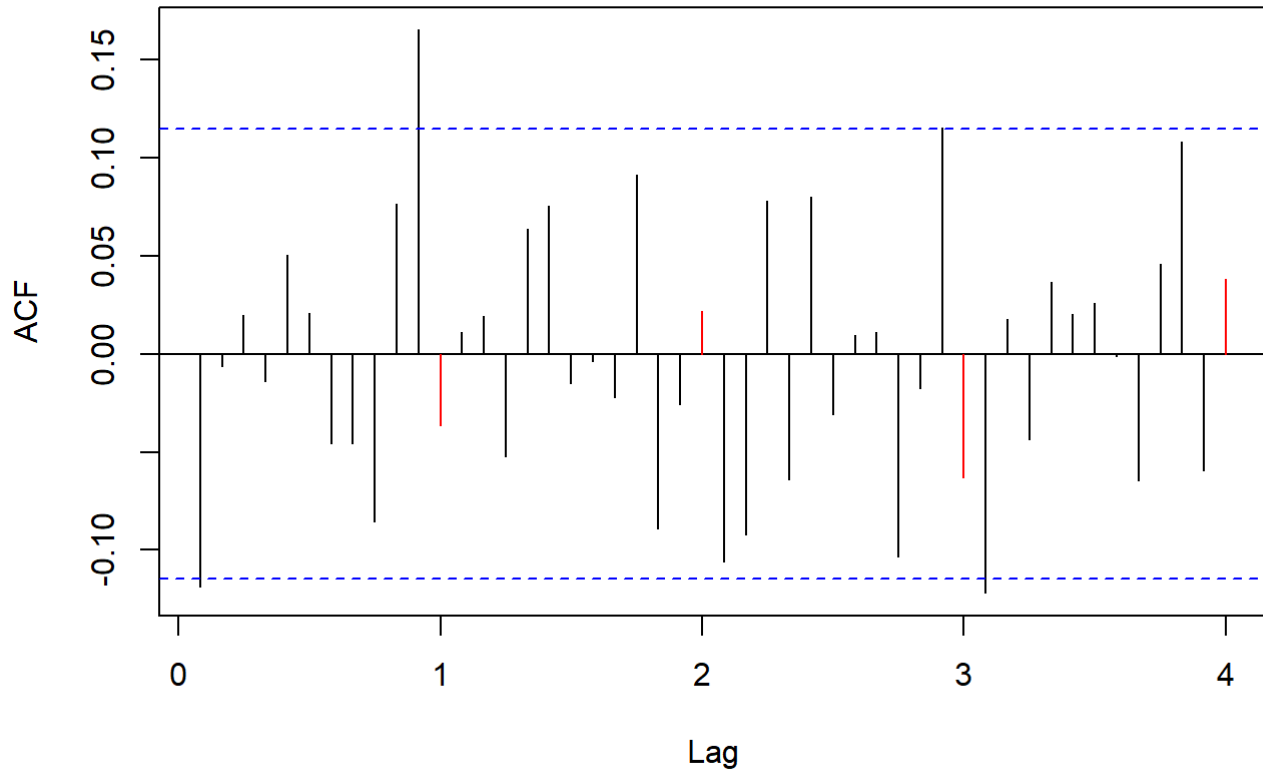


Figure 22: Sample ACF plot of residulas

The sample PACF of the residuals (Seasonal)

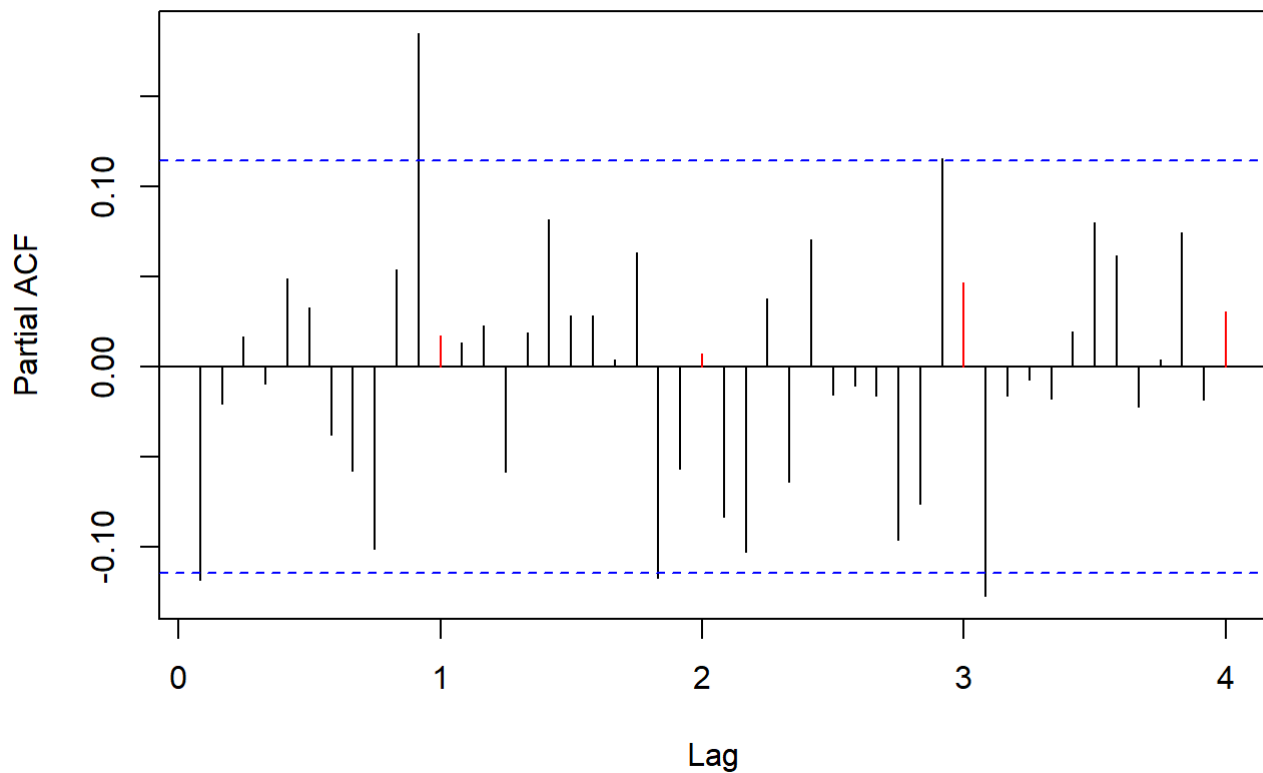


Figure 23: Sample PACF plot of residuals

The first ordinal difference residual plot(Figure m3.1) has a flat mean level but has high fluctuations.

After applying 1st ordinal differencing, there is no trend. There are 2 significant lags in ACF, $q = 2$. There are 2 significant lags in PACF, $p = 2$

Ordinal parameter (p,q) = (2,2)

```
m4.gold = Arima(data.ts, order=c(2,1,2),seasonal=list(order=c(1,1,2), period=12, method="ML"))

res.m4 = rstandard(m4.gold)
plot(res.m4,xlab='Time',ylab='Residuals', main="Time series plot of the residuals.", sub = "Figure m4.1: Time series of residuals")
```

Time series plot of the residuals.

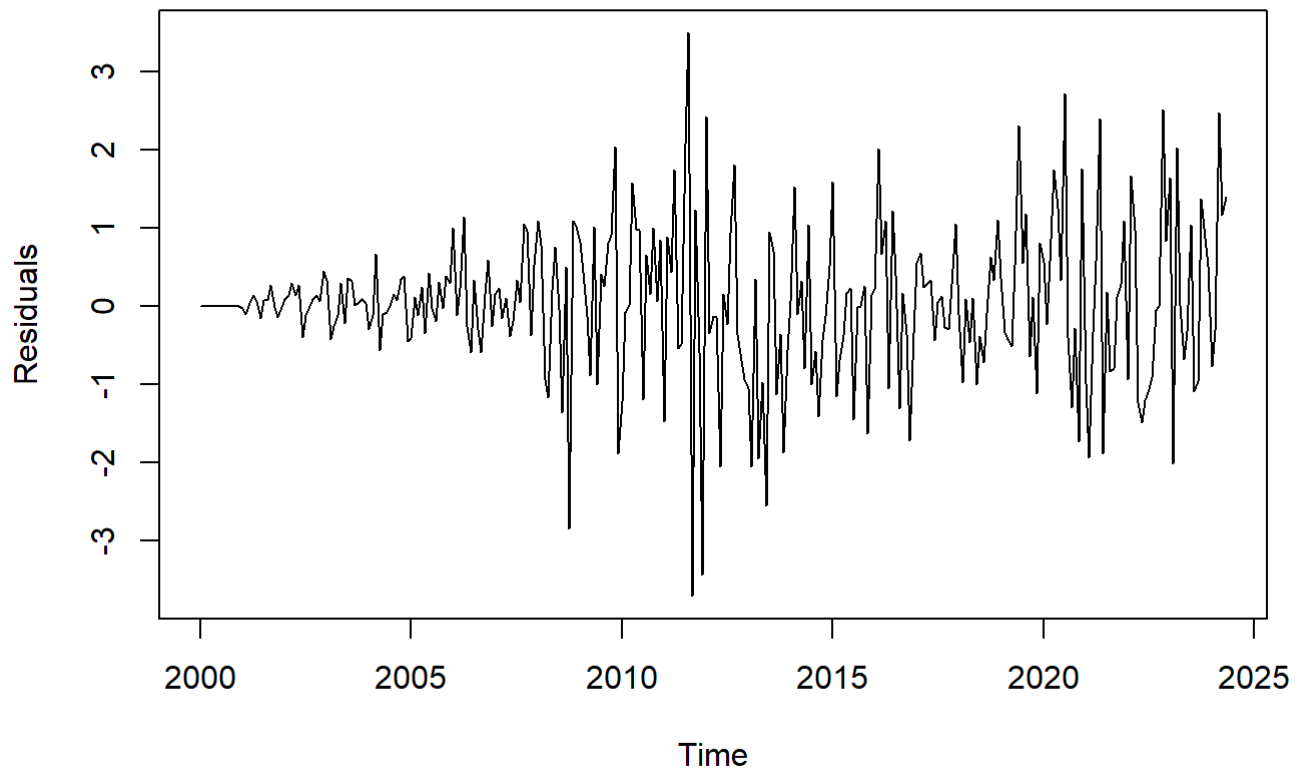


Figure m4.1: Time series of residuals

```
Diagnostic_test(res.m4, 48, "The sample ACF of the residuals", "Figure 24: Sample ACF plot of  
residuals", "The sample PACF of the residuals", "Figure 25: Sample PACF plot of residuals", t  
est = "seasonal ACF-PACF")
```

The sample ACF of the residuals (Seasonal)

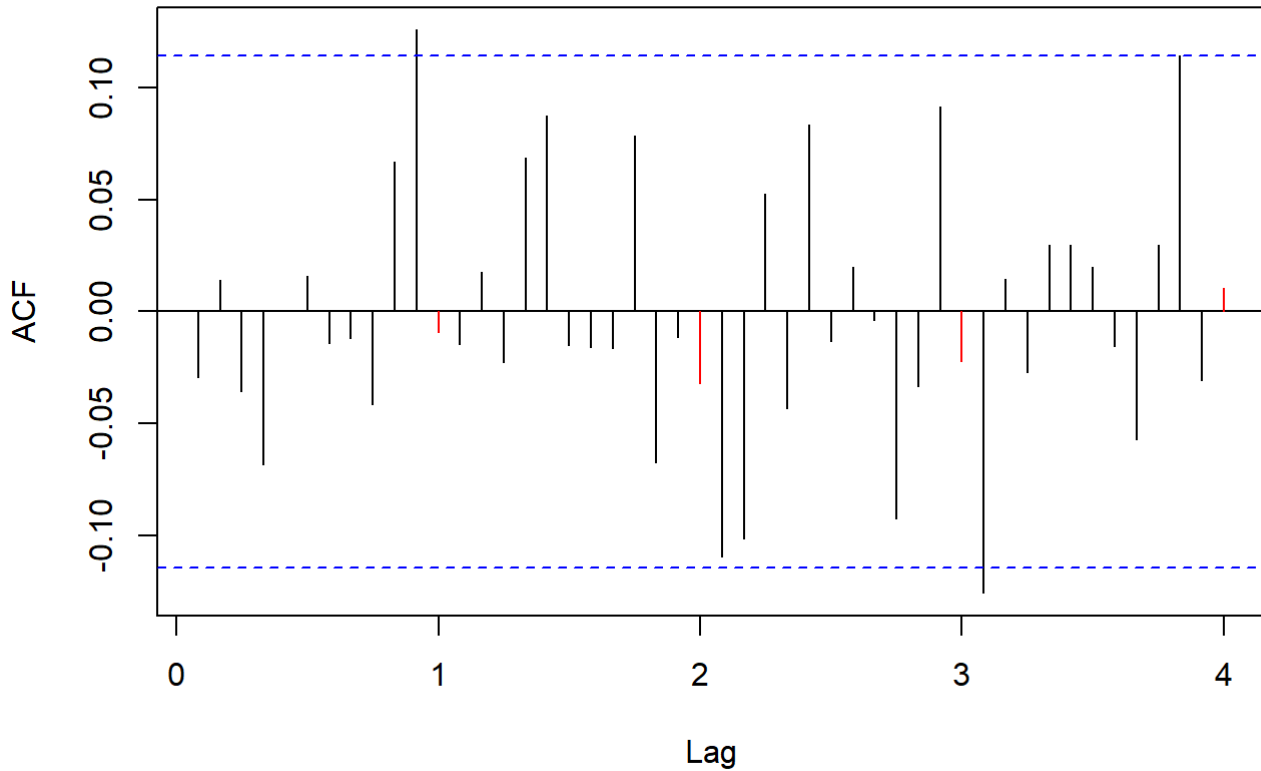


Figure 24: Sample ACF plot of residulas

The sample PACF of the residuals (Seasonal)

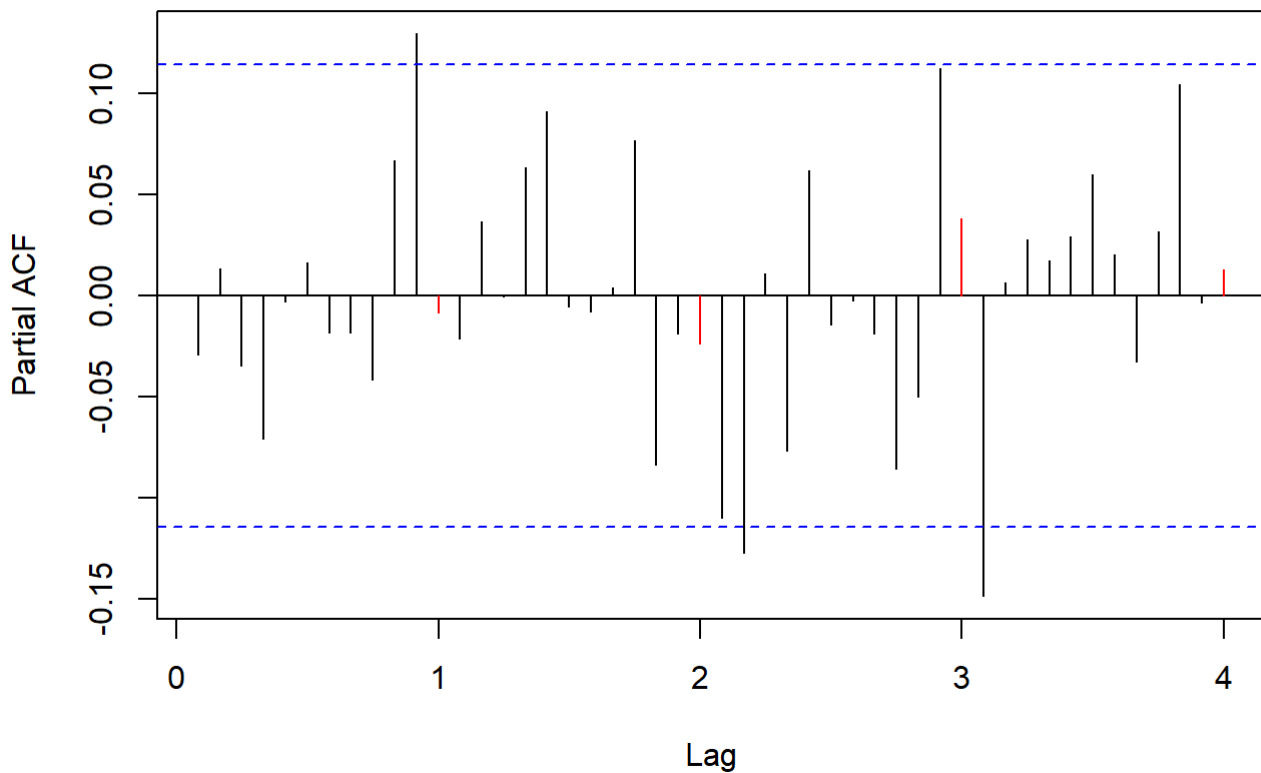


Figure 25: Sample PACF plot of residuals

The ordinal fitted parameter residual plot(Figure m4.1) has a flat mean level but has high fluctuations.

From the ACF and PACF plots of the residuals, SARIMA(2,1,2)x(1,1,2)₁₂ is the identified as a potential model.

EACF

```
eacf(res.m3)
```

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

From the EACF plot following models are identified:

- SARIMA(0,1,1)x(1,1,2)_12

Neighbor models:

- SARIMA(0,1,2)x(1,1,2)_12
- SARIMA(1,1,1)x(1,1,2)_12
- SARIMA(1,1,2)x(1,1,2)_12

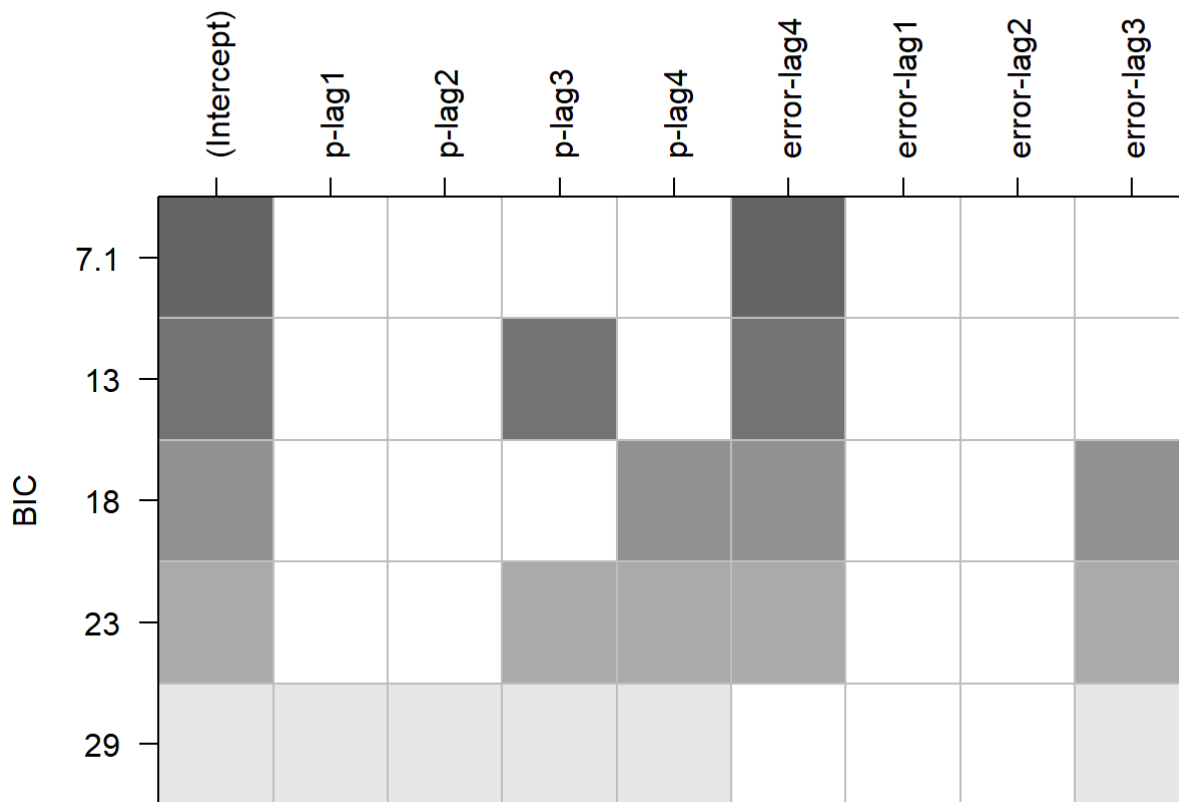
BIC

```
bic_table = armasubsets(y=res.m3,nar=4,nma=4,y.name='p',ar.method='ols')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 3 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
plot(bic_table)
```



From the BIC table following model is identified: - SARIMA(0,1,4)x(1,1,2)_12

Parameter Estimation

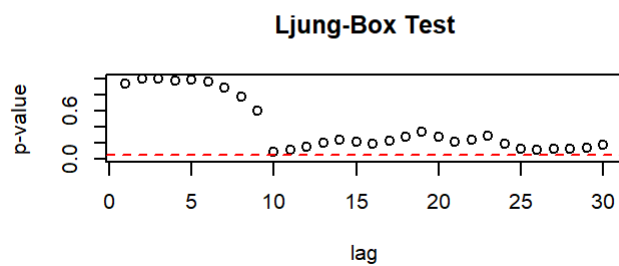
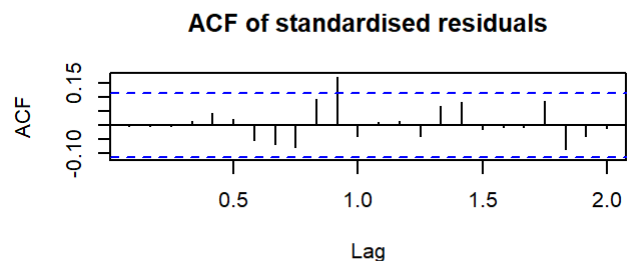
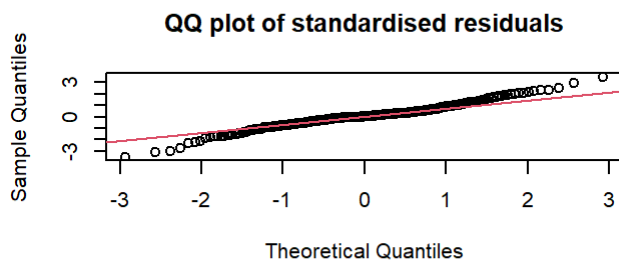
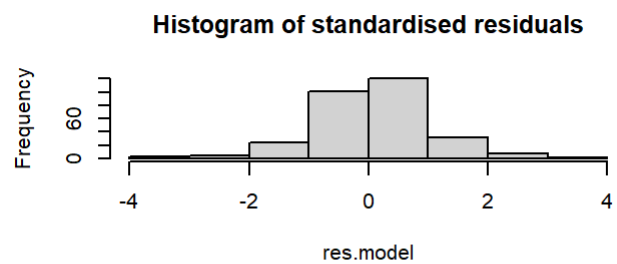
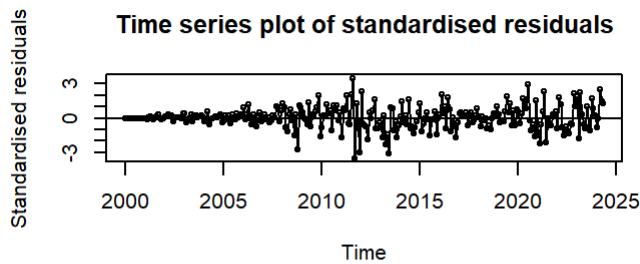
SARIMA(2,1,2)x(1,1,2)_12

```
m_212_112_12 = Arima(data.ts, order=c(2,1,2), seasonal=list(order=c(1,1,2), period=12), method = "ML")
coeftest(m_212_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  -0.74926    0.76657 -0.9774  0.328359
## ar2  -0.21582    0.61478 -0.3510  0.725555
## ma1   0.62663    0.77771  0.8057  0.420390
## ma2   0.12220    0.62078  0.1969  0.843939
## sar1  0.76815    0.28441  2.7008  0.006916 **
## sma1 -1.67331    0.35934 -4.6567 3.214e-06 ***
## sma2  0.67412    0.32241  2.0909  0.036540 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_212_112_12)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97099, p-value = 1.212e-05
```



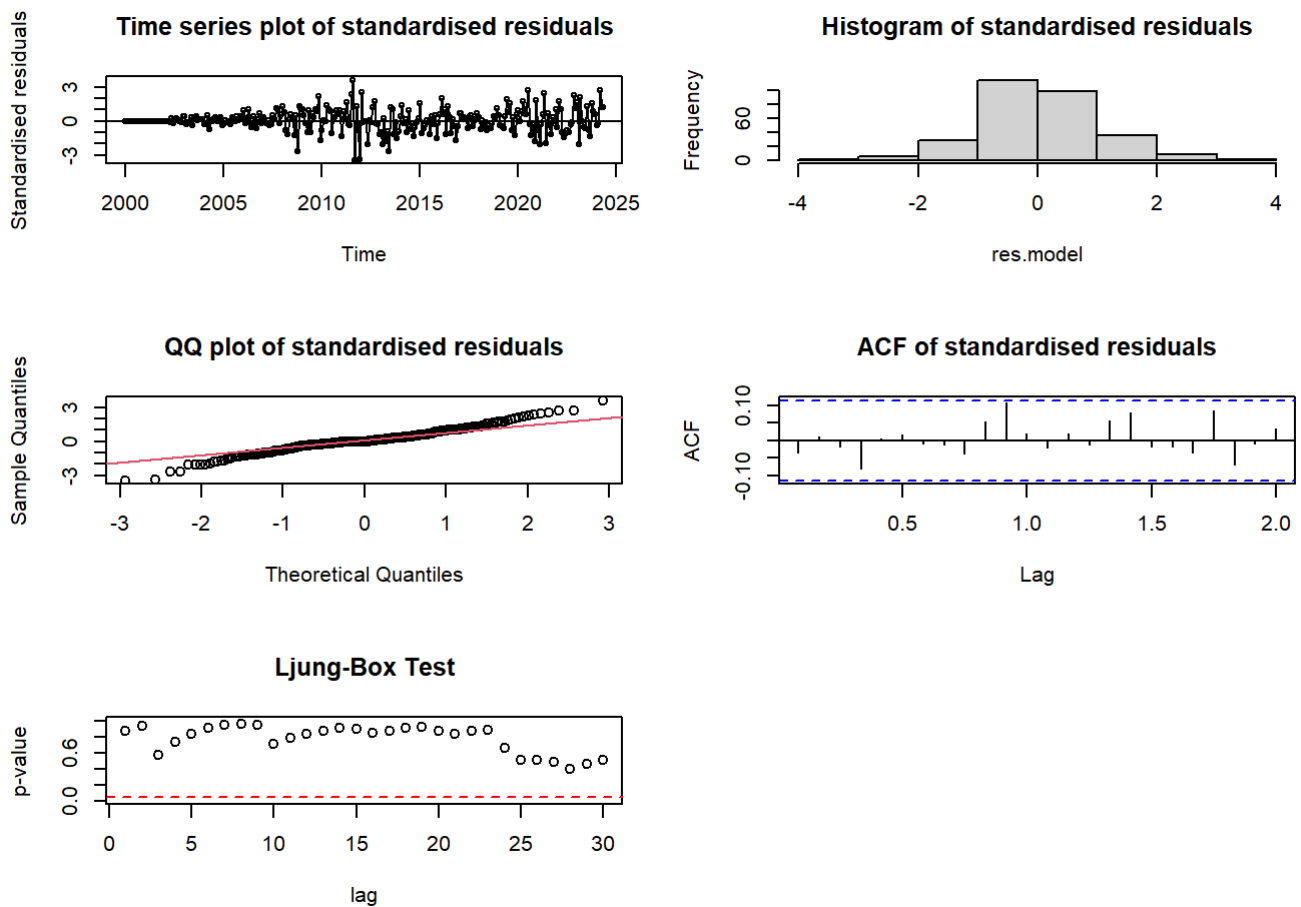
For SARIMA(2,1,2)x(1,1,2)₁₂ “ML” model, all seasonal coefficients are significant whereas, ordinal coefficients are insignificant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

```
m_212_112_12CSS = Arima(data.ts, order=c(2,1,2), seasonal=list(order=c(1,1,2), period=12), method = "CSS")
coeftest(m_212_112_12CSS)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1    1.203065   0.084336  14.2651 <2e-16 ***
## ar2   -0.830350   0.087463  -9.4938 <2e-16 ***
## ma1   -1.310747   0.051232 -25.5846 <2e-16 ***
## ma2    0.946876   0.056345  16.8049 <2e-16 ***
## sar1  -0.422156   0.605564  -0.6971  0.4857
## sma1  -0.561639   0.585456  -0.9593  0.3374
## sma2  -0.374496   0.554813  -0.6750  0.4997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_212_112_12CSS)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97, p-value = 8.531e-06
```



For SARIMA(2,1,2)x(1,1,2)₁₂ "CSS" model, all seasonal coefficients are insignificant whereas, ordinal coefficients are significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

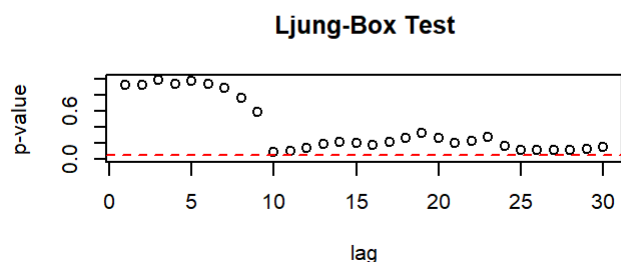
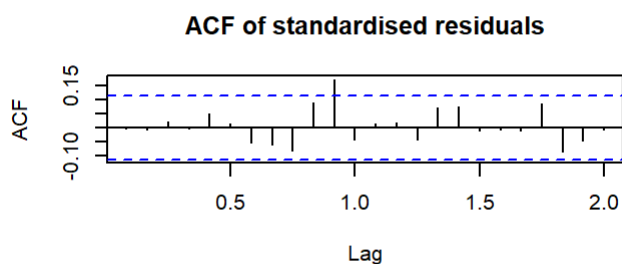
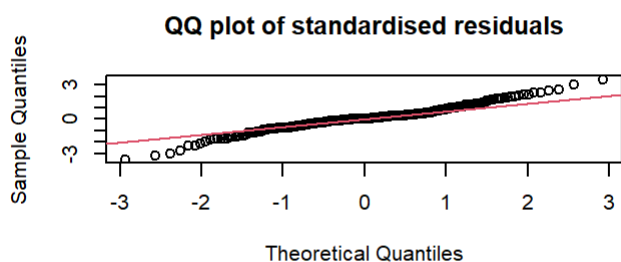
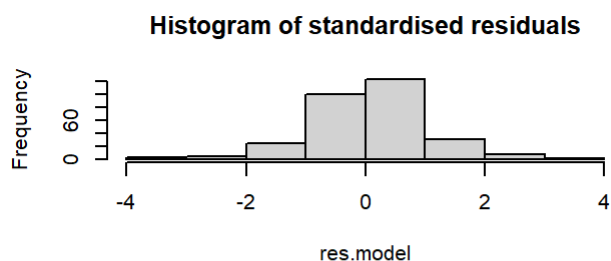
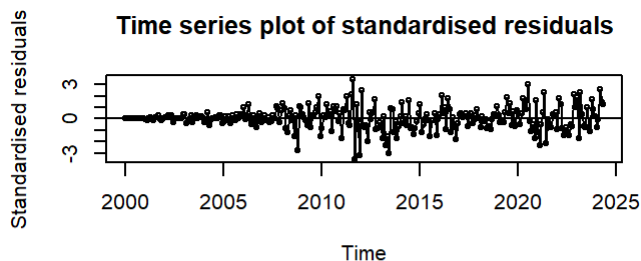
SARIMA(0,1,1)x(1,1,2)_12

```
m_011_112_12 = Arima(data.ts,order=c(0,1,1),seasonal=list(order=c(1,1,2), period=12),method =
"ML")
coeftest(m_011_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.12441    0.05988 -2.0776 0.0377481 *
## sar1   0.75479    0.31216  2.4179 0.0156085 *
## sma1  -1.65389    0.49154 -3.3647 0.0007662 ***
## sma2   0.65704    0.40822  1.6095 0.1075006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_011_112_12)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.96954, p-value = 7.248e-06
```



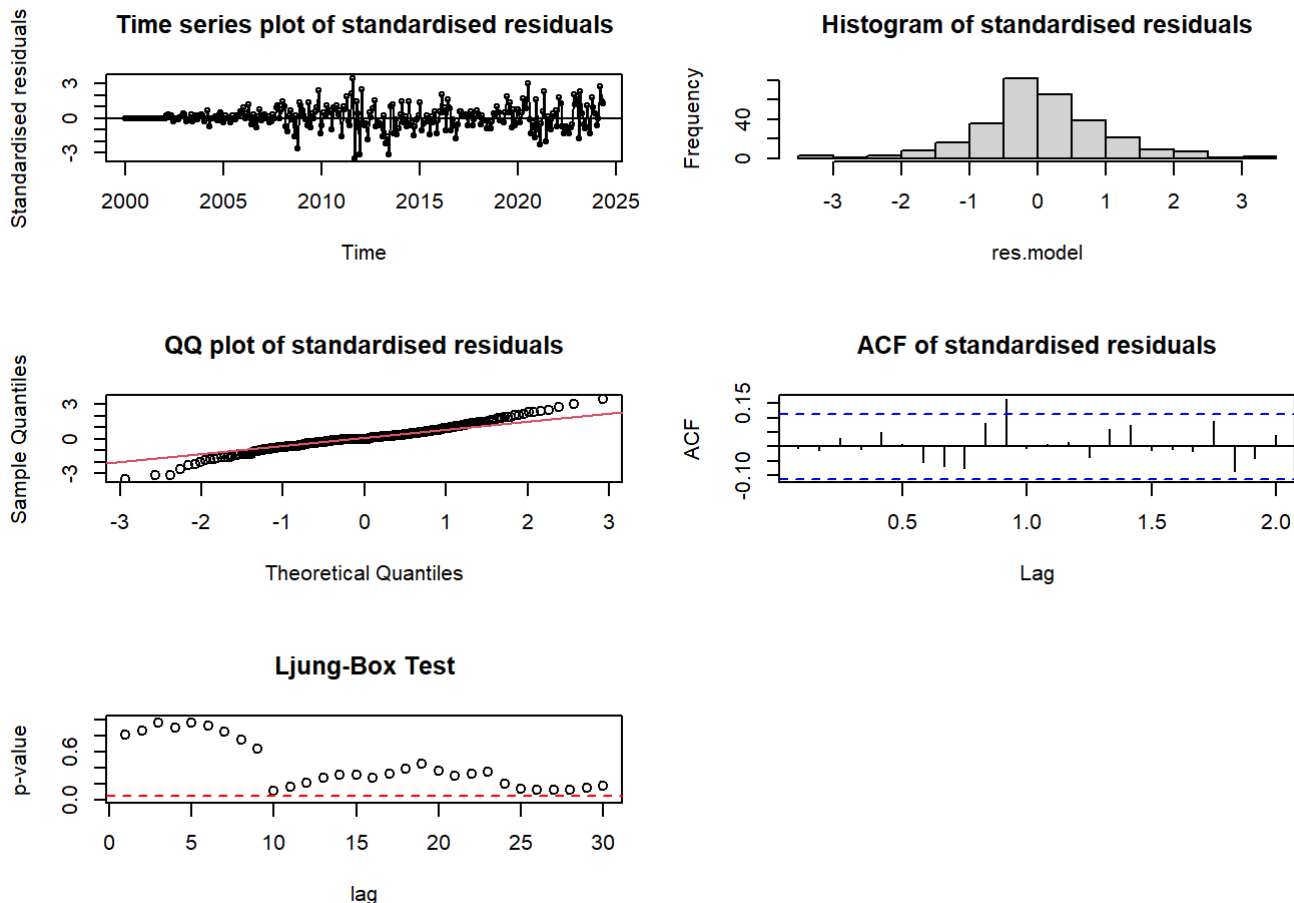
For SARIMA(0,1,1)x(1,1,2)_12 “ML” model, all coefficient except “sma2”, are significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

```
m_011_012_12CSS = Arima(data.ts,order=c(0,1,1),seasonal=list(order=c(1,1,2), period=12),method = "CSS")
coeftest(m_011_012_12CSS)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.127333   0.060858 -2.0923 0.036413 *
## sar1   0.087110   0.398916  0.2184 0.827144
## sma1 -0.997808   0.385440 -2.5887 0.009633 **
## sma2   0.038891   0.365215  0.1065 0.915195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_011_012_12CSS)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97198, p-value = 1.736e-05
```



For SARIMA(0,1,1)x(1,1,2)₁₂ “CSS” model, only “ma1” and “sma1” are significant, whereas, “sar1” and “sma2” are insignificant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviations at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

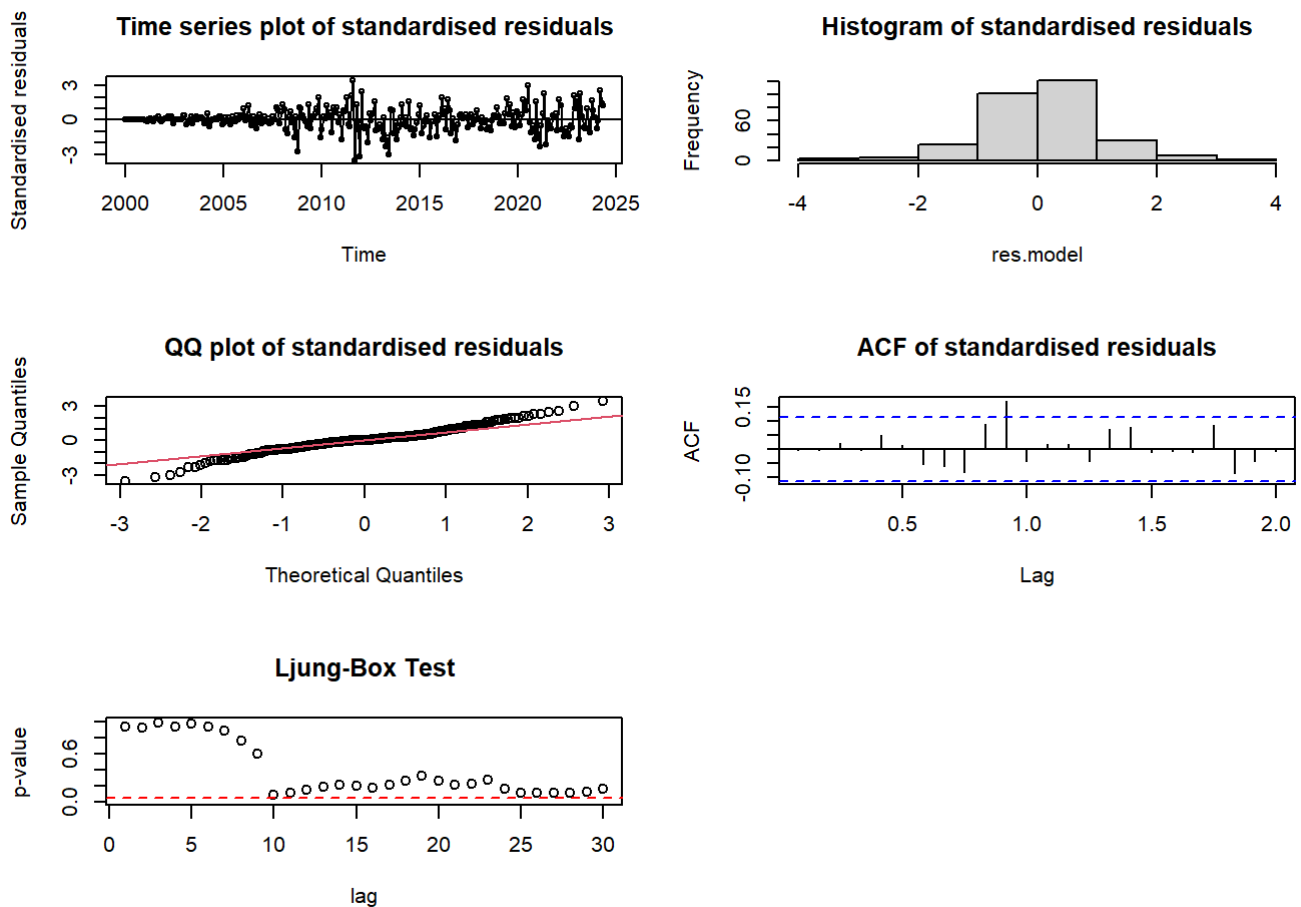
SARIMA(0,1,2)x(1,1,2)₁₂

```
m_012_112_12 = Arima(data.ts,order=c(0,1,2),seasonal=list(order=c(1,1,2), period=12),method =
"ML")
coeftest(m_012_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.1240004  0.0605497 -2.0479  0.040569 *
## ma2   -0.0014879  0.0599799 -0.0248  0.980209
## sar1   0.7760672  0.2791549  2.7801  0.005435 **
## sma1  -1.6821929  0.3672676 -4.5803 4.643e-06 ***
## sma2   0.6825690  0.3234879  2.1100  0.034856 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_012_112_12)
```

```
##
## Shapiro-Wilk normality test
##
## data: res.model
## W = 0.96946, p-value = 7.068e-06
```



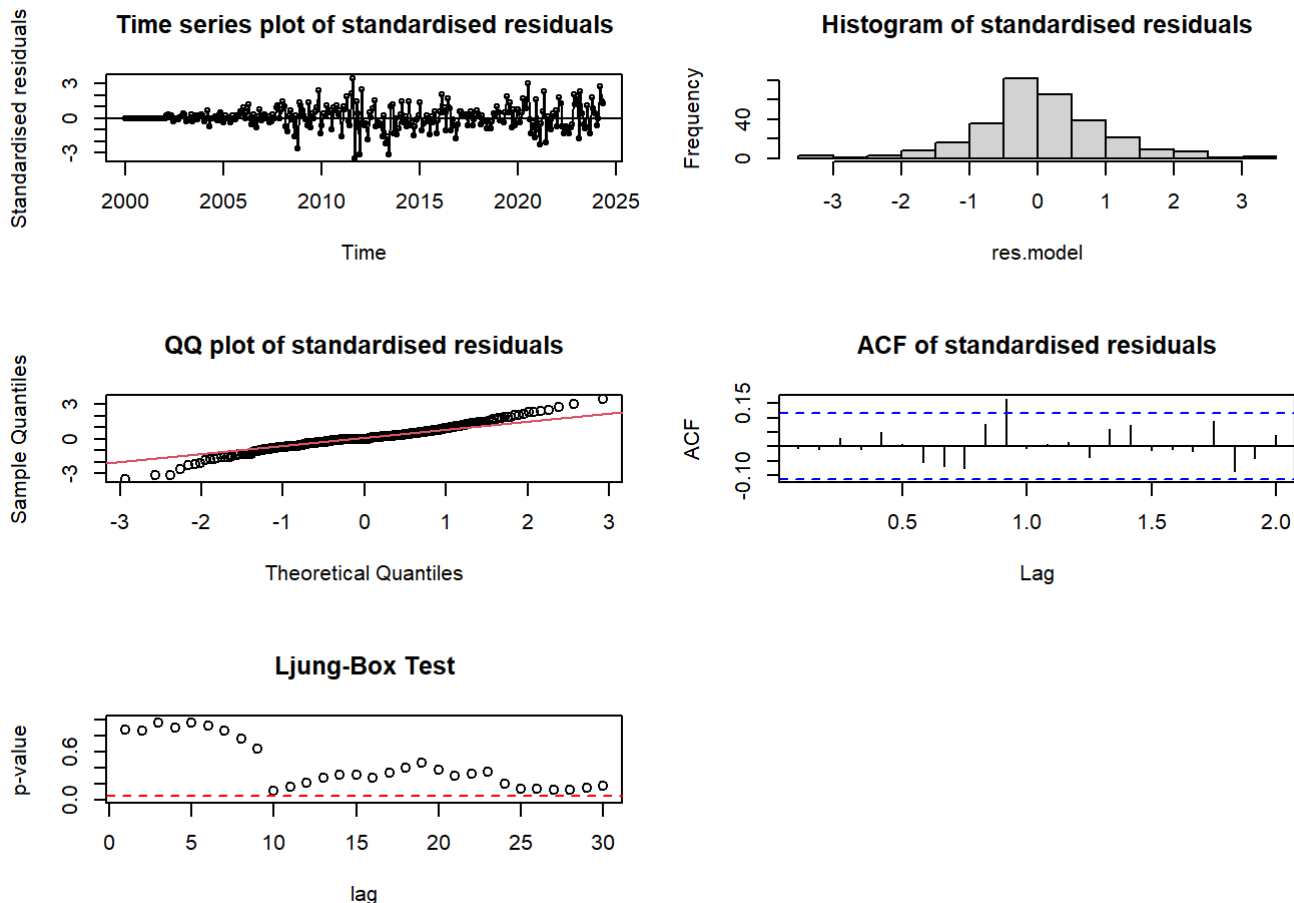
For SARIMA(0,1,2)x(1,1,2)_12 “ML” model, all coefficients except “ma2” are significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviations at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no autocorrelation is present in the residuals.

```
m_012_112_12CSS = Arima(data.ts,order=c(0,1,2),seasonal=list(order=c(1,1,2), period=12),method = "CSS")
coeftest(m_012_112_12CSS)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.1266208  0.0614870  -2.0593  0.03946 *
## ma2   -0.0042727  0.0607099  -0.0704  0.94389
## sar1   0.0864579  0.4007471   0.2157  0.82919
## sma1  -0.9970637  0.3871374  -2.5755  0.01001 *
## sma2   0.0380059  0.3668397   0.1036  0.91748
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_012_112_12CSS)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: res.model  
## W = 0.97182, p-value = 1.638e-05
```



For SARIMA(0,1,2)x(1,1,2)₁₂ “CSS” model, only “ma1” and “sma1” are significant, whereas, “sar1” and “sma2” are insignificant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviations at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

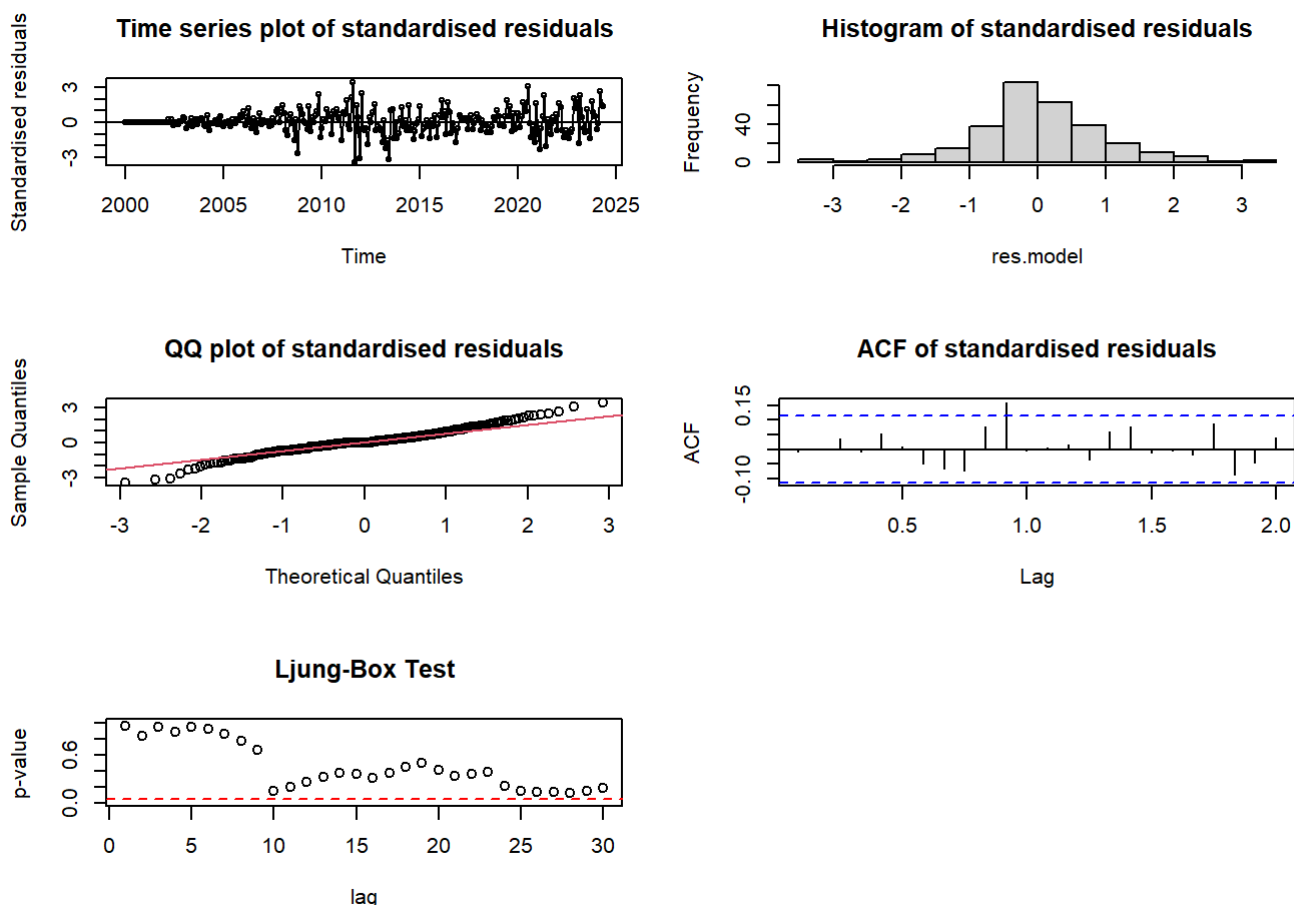
SARIMA(1,1,1)x(1,1,2)₁₂

```
m_111_112_12 = Arima(data.ts,order=c(1,1,1),seasonal=list(order=c(1,1,2), period=12),method =  
"CSS")  
coeftest(m_111_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.146881   0.371661  0.3952  0.69269
## ma1   -0.276334   0.354457 -0.7796  0.43563
## sar1    0.094398   0.365935  0.2580  0.79643
## sma1   -1.004664   0.355303 -2.8276  0.00469 **
## sma2    0.042400   0.337063  0.1258  0.89990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_111_112_12)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97155, p-value = 1.486e-05
```



For SARIMA(1,1,1)x(1,1,2)_12 “CSS” model, only “sma1” is significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

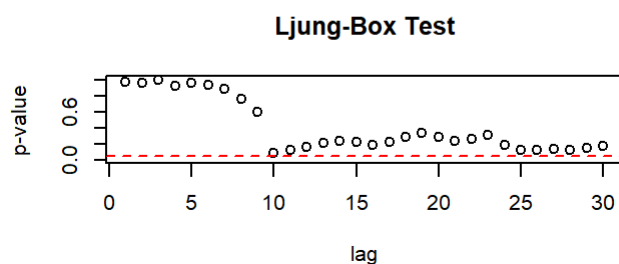
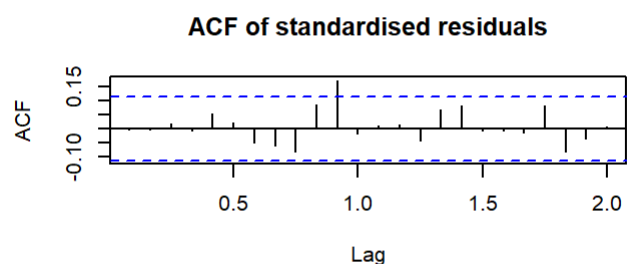
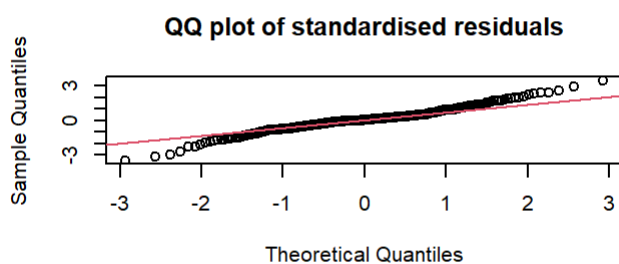
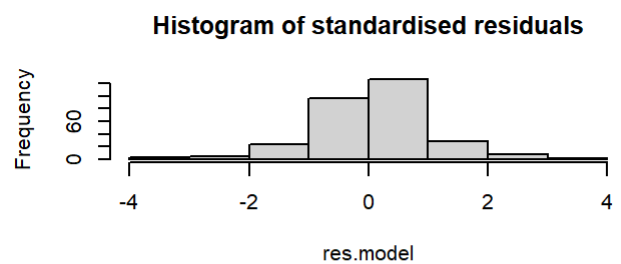
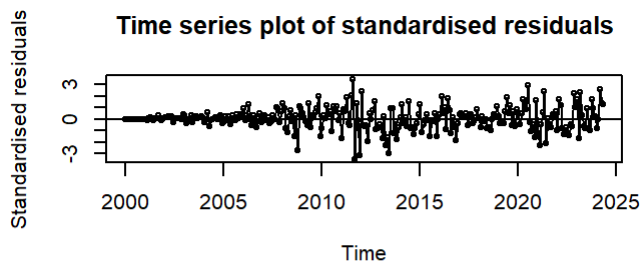
```
##SARIMA(1,1,2)x(1,1,2)_12
```

```
m_112_112_12 = Arima(data.ts,order=c(1,1,2),seasonal=list(order=c(1,1,2), period=12),method =
"ML")
coeftest(m_112_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  -0.085334      NaN      NaN      NaN
## ma1  -0.034553      NaN      NaN      NaN
## ma2  -0.014057      NaN      NaN      NaN
## sar1 -0.327372      NaN      NaN      NaN
## sma1 -0.556799      NaN      NaN      NaN
## sma2 -0.294625      NaN      NaN      NaN
```

```
residual.analysis(model = m_112_112_12)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.96974, p-value = 7.772e-06
```



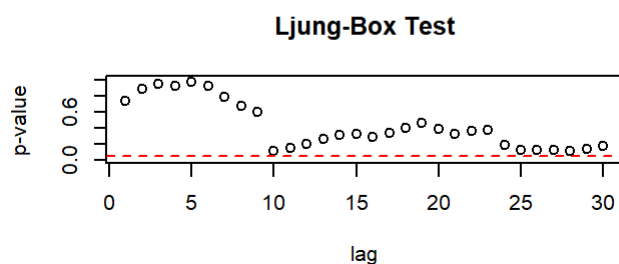
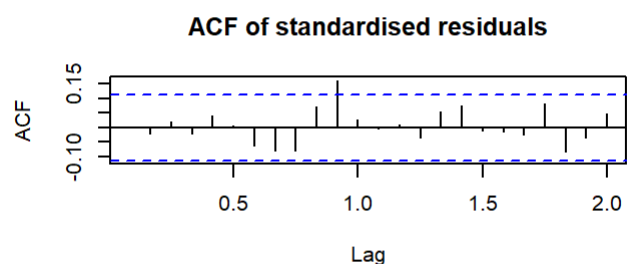
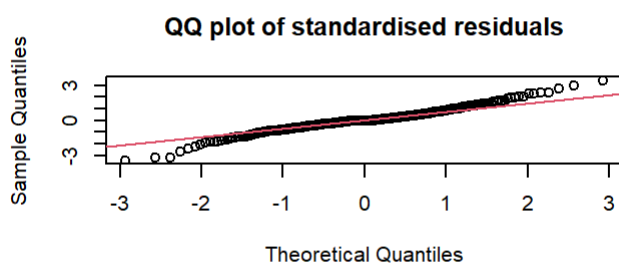
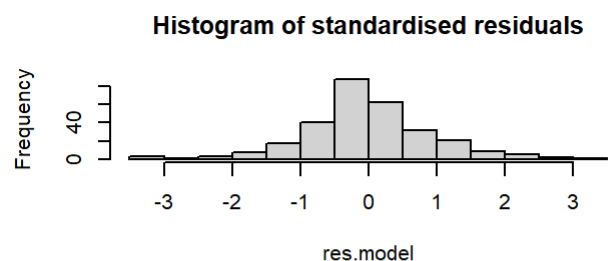
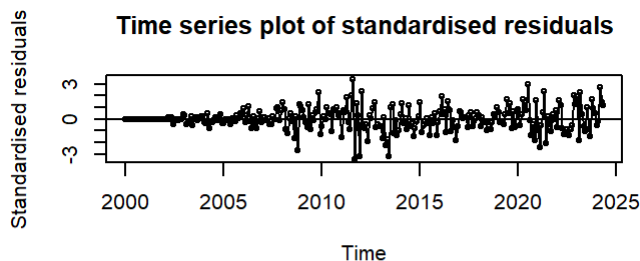
We observe NaN values in the coefficient test. This could be because of 0 values in the model.

```
m_112_112_12CSS = Arima(data.ts,order=c(1,1,2),seasonal=list(order=c(1,1,2), period=12),method = "CSS")
coeftest(m_112_112_12CSS)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.908585   0.343082  2.6483 0.008090 **
## ma1  -1.042586   0.302885 -3.4422 0.000577 ***
## ma2   0.130596   0.062684  2.0834 0.037216 *
## sar1 -0.361433   2.085287 -0.1733 0.862396
## sma1 -0.585736   1.897328 -0.3087 0.757537
## sma2 -0.364041   1.788012 -0.2036 0.838665
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_112_112_12CSS)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97274, p-value = 2.298e-05
```



For SARIMA(1,1,2)x(1,1,2)₁₂ “CSS” model, all ordinal components are significant, whereas, all seasonal components are insignificant. The residual plot has a flat mean level indicating no trend. The histogram is

symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

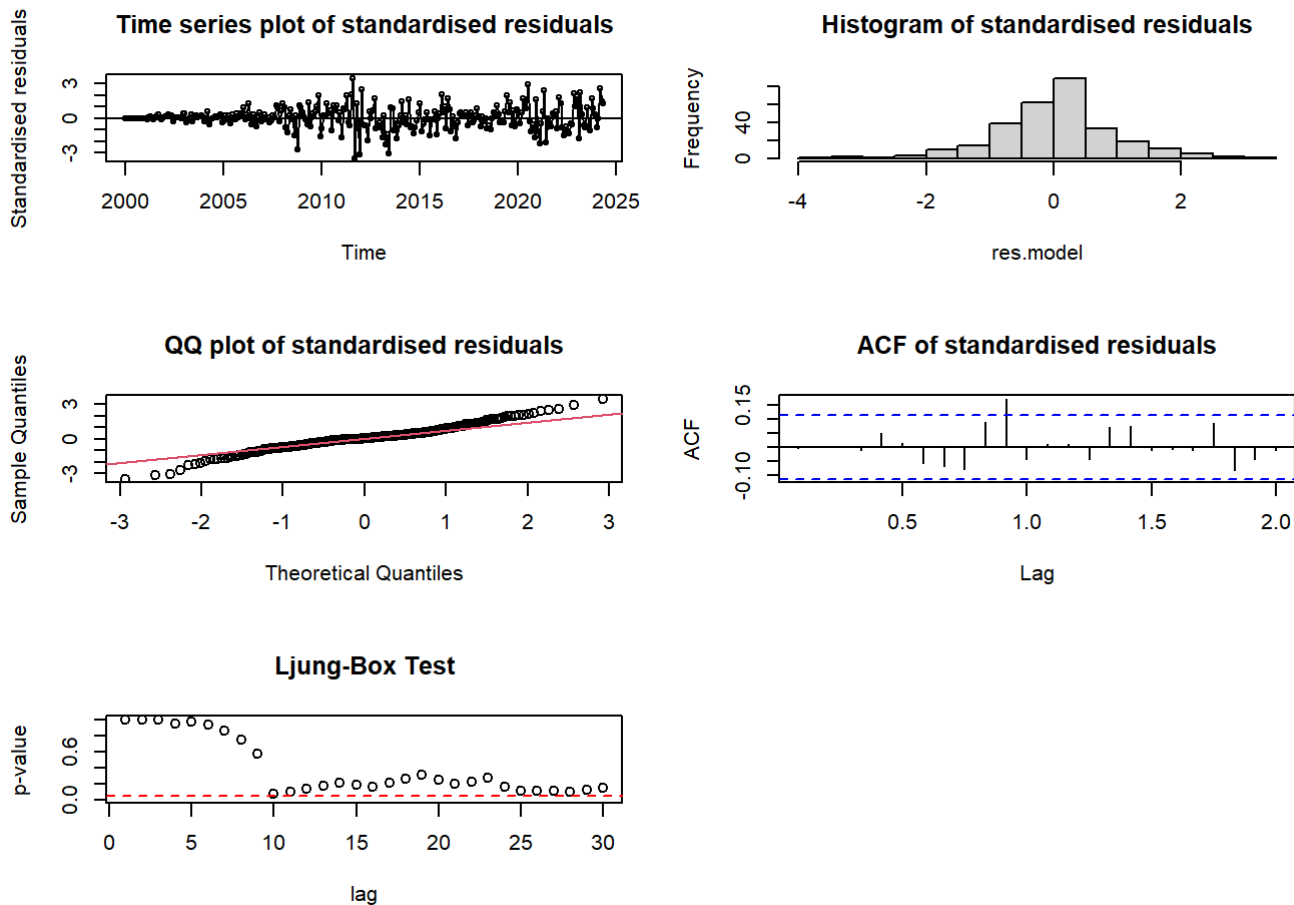
```
##SARIMA(0,1,4)x(1,1,2)_12
```

```
m_014_112_12 = Arima(data.ts,order=c(0,1,4),seasonal=list(order=c(1,1,2), period=12),method =
"ML")
coeftest(m_014_112_12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.1250362  0.0607214 -2.0592  0.03948 *
## ma2  -0.0067193  0.0609401 -0.1103  0.91220
## ma3   0.0258773  0.0621805  0.4162  0.67729
## ma4   0.0042836  0.0669249  0.0640  0.94897
## sar1  0.7505003  0.2959207  2.5362  0.01121 *
## sma1 -1.6525416  0.4069906 -4.0604 4.899e-05 ***
## sma2  0.6539994  0.3508248  1.8642  0.06230 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_014_112_12)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97101, p-value = 1.221e-05
```

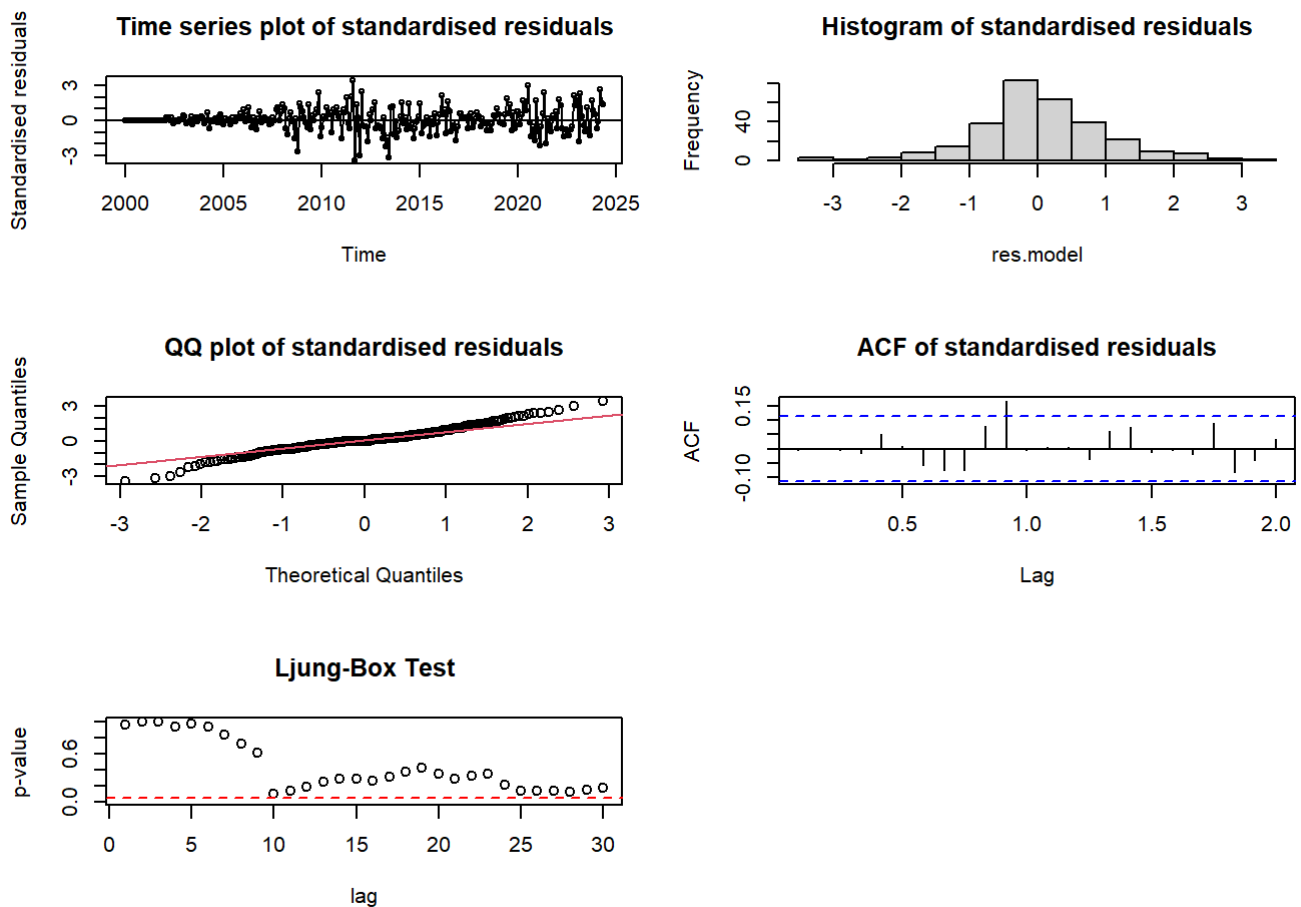
For SARIMA(0,1,4)x(1,1,2)_12 “ML” model, only “ma1”, “sar1”, and “sma1” are significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but there are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no autocorrelation is present in the residuals.

```
m_014_112_12CSS = Arima(data.ts,order=c(0,1,4),seasonal=list(order=c(1,1,2), period=12),method = "CSS")
coeftest(m_014_112_12CSS)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  -0.127928   0.061647 -2.0752 0.037970 *
## ma2  -0.013038   0.061584 -0.2117 0.832330
## ma3   0.039836   0.064433  0.6182 0.536412
## ma4   0.003782   0.067957  0.0557 0.955619
## sar1  0.076672   0.392715  0.1952 0.845209
## sma1 -0.983590   0.381432 -2.5787 0.009918 **
## sma2  0.022072   0.363463  0.0607 0.951577
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model = m_014_112_12CSS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97368, p-value = 3.252e-05
```



For SARIMA(0,1,4)x(1,1,2)₁₂ “CSS” model, only “ma1” and “sma1” are significant. The residual plot has a flat mean level indicating no trend. The histogram is symmetric and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the tails. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

Based on coefficient testing and residual analysis, SARIMA(0,1,1)x(1,1,2)₁₂ is the best model option.

Goodness of fit

AIC and BIC values

The AIC and BIC scores are calculated using “ML” methods. SARIMA_{111_112_12} is not a compatible model for “ML” method. Therefore, it is excluded from the matrix.

```

aic_table =AIC(m_212_112_12,m_011_112_12,m_012_112_12,m_112_112_12, m_014_112_12)
bic_table =BIC(m_212_112_12,m_011_112_12,m_012_112_12,m_112_112_12, m_014_112_12)

sorted_aic_table <- aic_table[order(aic_table$AIC), ]
sorted_bic_table <- bic_table[order(bic_table$BIC), ]

sorted_aic_table

```

```

##           df      AIC
## m_011_112_12  5 3107.079
## m_012_112_12  6 3109.077
## m_112_112_12  7 3112.029
## m_212_112_12  8 3112.815
## m_014_112_12  8 3112.904

```

```
sorted_bic_table
```

```

##           df      BIC
## m_011_112_12  5 3125.253
## m_012_112_12  6 3130.885
## m_112_112_12  7 3137.472
## m_212_112_12  8 3141.893
## m_014_112_12  8 3141.983

```

Based on AIC and BIC scores, SARIMA(0,1,1)x(1,1,2)_12 is the best model option.

Error measures

```

Smodel.011 = accuracy(summary(m_212_112_12))
Smodel.012 = accuracy(summary(m_011_112_12))
Smodel.111 = accuracy(summary(m_012_112_12))
Smodel.112 = accuracy(summary(m_111_112_12))
Smodel.014 = accuracy(summary(m_112_112_12))
Smodel.212 = accuracy(summary(m_014_112_12))

headers <- c("ME", "RMSE", "MAE", "MPE", "MAPE", "MASE", "ACF1")

merged_table <- data.frame(rbind
  ( c(Smodel.011), c(Smodel.012),c(Smodel.111),c(Smodel.112),c(Smodel.014),(Smodel.212)))
model_names <- c("model.011", "model.012", "model.111","model.112","model.014","model.212")

rownames(merged_table) <- model_names
colnames(merged_table) <- headers
merged_table

```

##		ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
##	model.011	2.565955	56.41334	40.08606	0.2467946	3.442390	0.2863742	-0.002880773
##	model.012	2.594659	56.59559	40.14168	0.2506678	3.447710	0.2867715	-0.002502538
##	model.111	2.594967	56.42091	40.01146	0.2509124	3.436849	0.2858412	-0.002436669
##	model.112	3.597293	57.65522	40.92263	0.2496347	3.467470	0.2923506	-0.008512861
##	model.014	2.645187	57.82150	40.99477	0.2560791	3.520331	0.2928660	-0.002645721
##	model.212	2.555405	56.44648	40.15336	0.2450779	3.445569	0.2868549	-0.002032425

The error measures have very close values for all models.

In conclusion, we use SARIMA(0,1,1)x(1,1,2)_12 model.

Over-parameterisation

Parameter tuning is done to identify any further potential models.

The following models will be tested under parameter tuning:

- SARIMA(0,1,2)x(1,1,2)_12
- SARIMA(1,1,1)x(1,1,2)_12

We have already tested the SARIMA(0,1,2)x(1,1,2)_12, and SARIMA(1,1,1)x(1,1,2)_12 model and concluded that they weren't significant models.

Model Specification ARMA x GARCH Part 1

Since our time series data had changing variance we had to consider ARMA x GARCH model for making our predictions. After applying the log transformation with the first order of differencing the following time series plot is displayed.

```
r.gold <- diff(log(data.ts))*100
plot(r.gold, ylab = "Gold price", main = "Return series for Gold price", sub = "Figure 26: Return series for Gold price")
```

Return series for Gold price

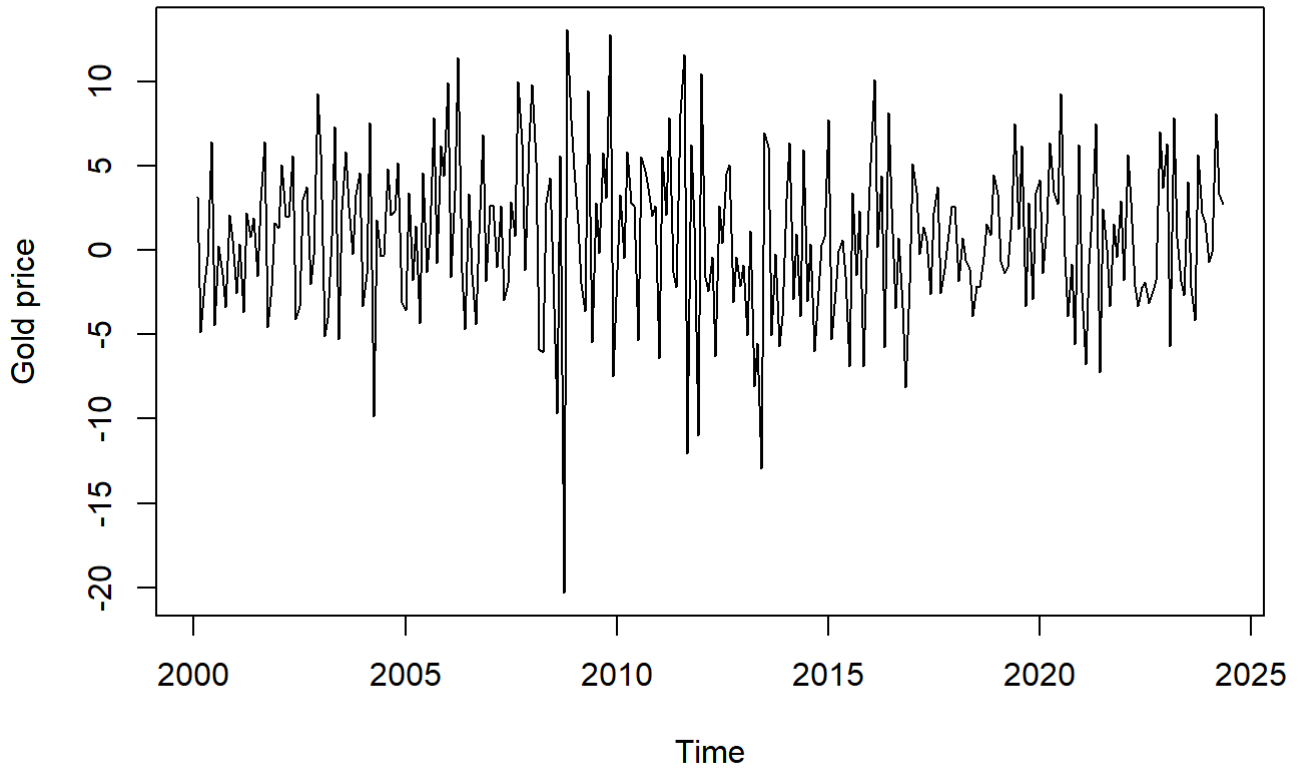


Figure 26: Return series for Gold price

It is clear from the Figure 26 that there is a changing variance in the return series. To verify this we will conduct the McLeod Li test on the return series.

```
McLeod.Li.test(y=r.gold,main ="McLeodLi Test for Changing Variance", sub = "Figure 27: McLeod  
Li Test for Changing Variance")
```

McLeodLi Test for Changing Variance

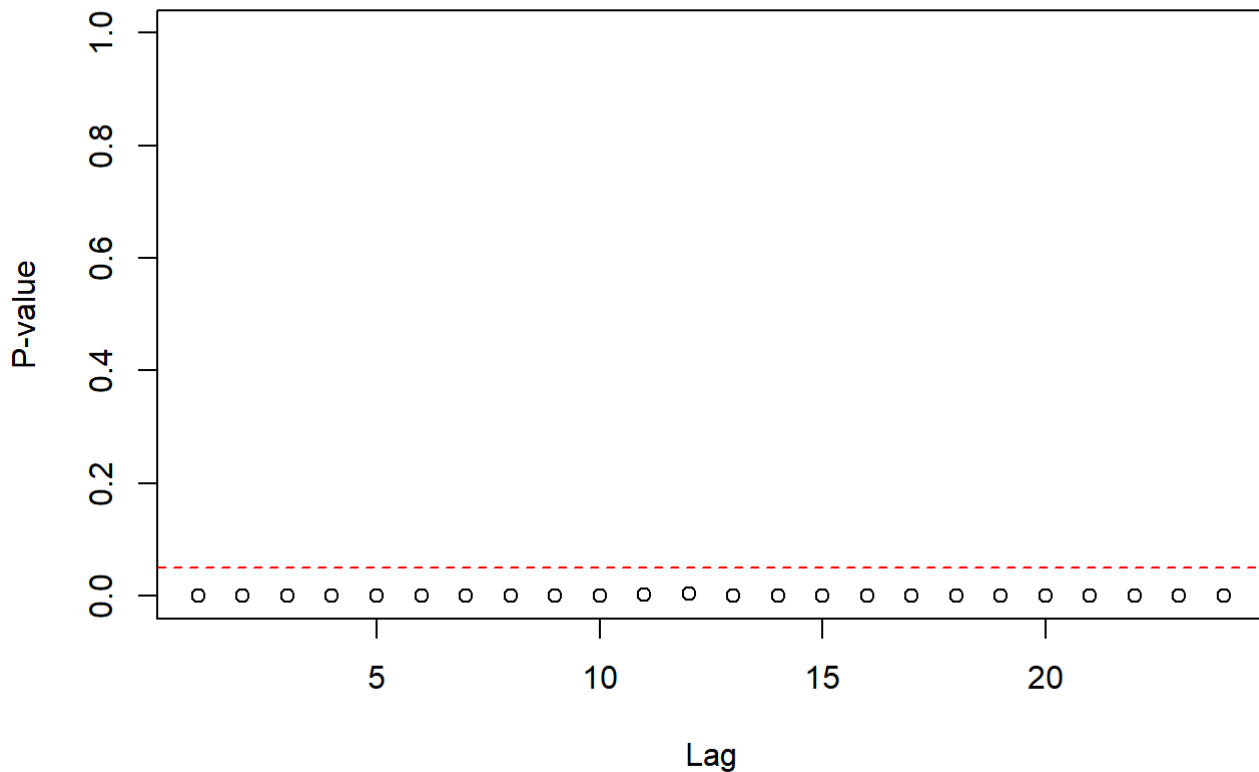


Figure 27: McLeodLi Test for Changing Variance

From the above Figure 27, the p-value at each lag is below the significance level of 0.05 that means there is changing variance in the return series.

Stationary test

In the return series, the volatility is obvious and there is no sense of trend or seasonality. The stationarity of the return series is confirmed by the ADF test. To support the ADF test, we will go on with displaying the ACF and PACF plots.

```
Diagnostic_test(r.gold, 48, mainacf = "ACF plot for return series", subacf = "Figure 28: ACF  
plot for return series", mainpacf = "PACF plot for return series", subpacf = "Figure 29: PACF p  
lot for return series", test = "Stationary")
```

ACF plot for return series

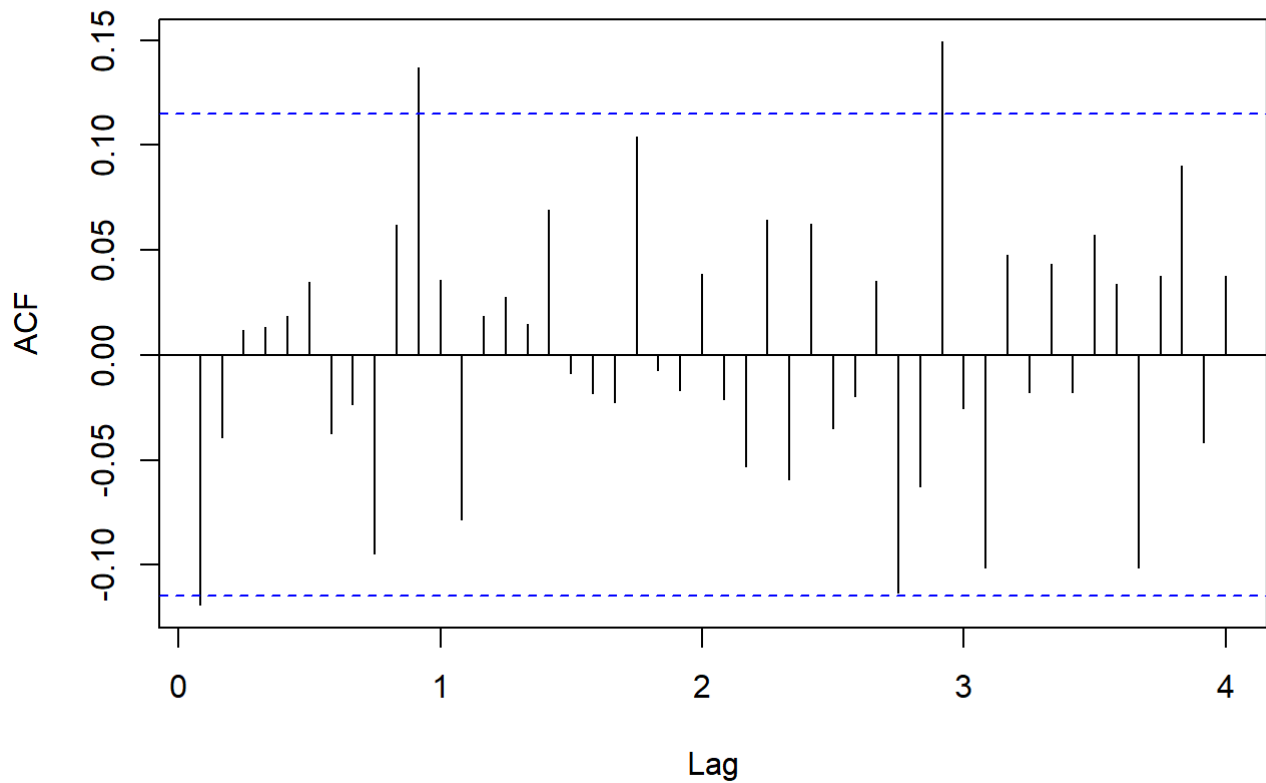


Figure 28: ACF plot for return series

PACF plot for return series

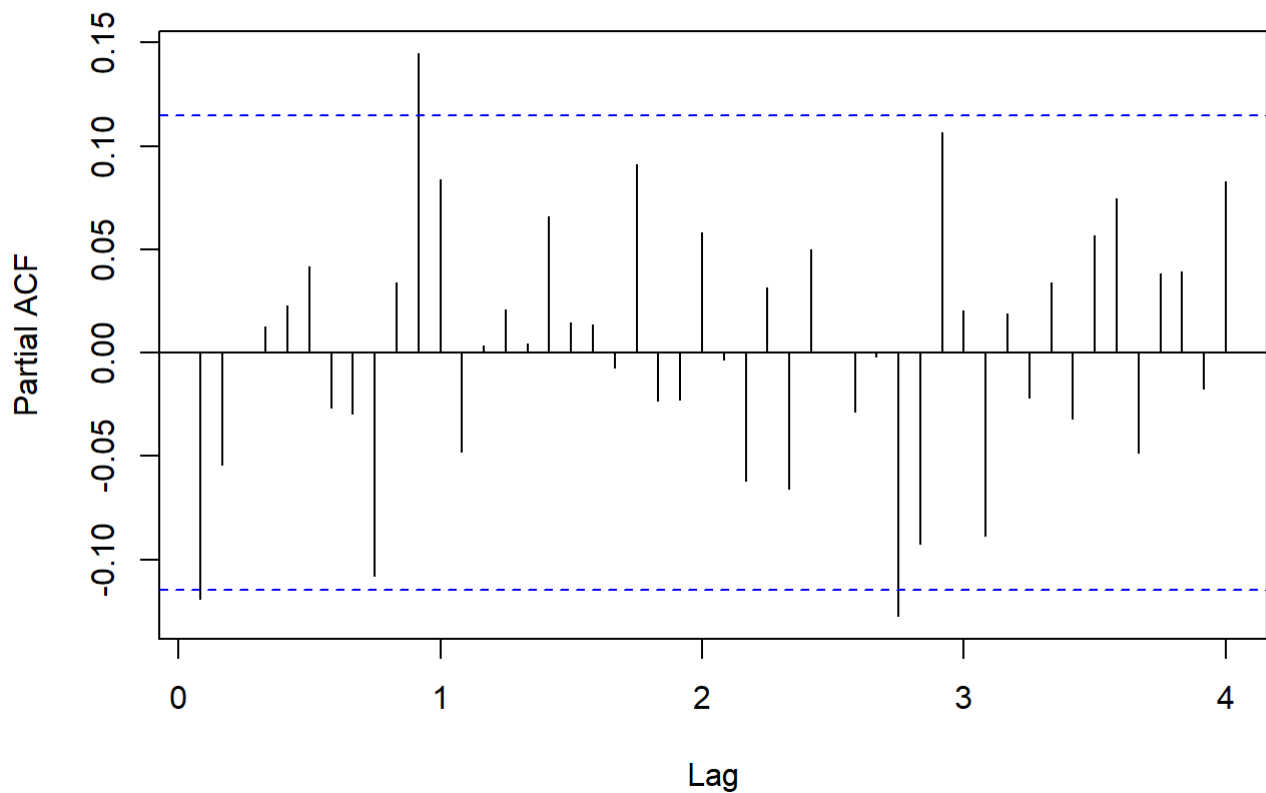


Figure 29: PACF plot for return series

```
## $adf_test
##
## Augmented Dickey-Fuller Test
##
## data: data
## Dickey-Fuller = -6.2687, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $pp_test
##
## Phillips-Perron Unit Root Test
##
## data: data
## Dickey-Fuller Z(alpha) = -317.97, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

The PACF plot in figure 29 shows the 1st lag significant indicating the value of $p=1$. Whereas, the ACF plot in figure 28 shows 1st lag is significant indicating the value of $q=1$.

In total ACF and PACF Plots propose 1 set of possible models: $\{ARMA(1,1)\}$

EACF ARMA

```
eacf(r.gold)
```

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

The top-left “o” is identified in the EACF plot is at (0,1). From the EACF plot following models are identified:

- $ARMA(0,1)$

Neighbor models:

- $ARMA(0,2)$
- $ARMA(1,1)$
- $ARMA(1,2)$

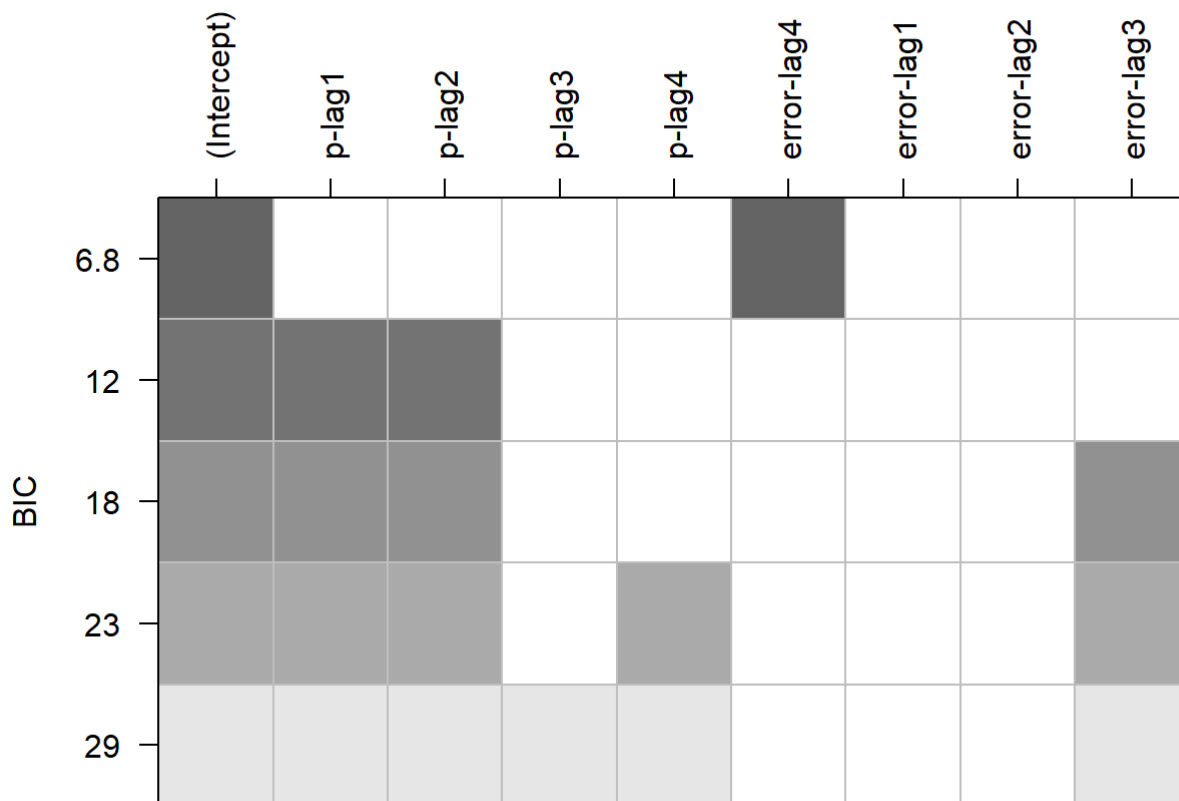
BIC ARMA

```
par(mfrow=c(1,1))
bic_table = armasubsets(y=r.gold,nar=4,nma=4,y.name='p',ar.method='ols')
```



```
## Reordering variables and trying again:
```

```
plot(bic_table)
```



From the BIC table following model is identified: - ARMA(0,4)

Parameter Estimation ARMA

Combining all the possible models from ACF, PACF, EACF and BIC table, we get

- ARMA(0,4)
- ARMA(0,1)
- ARMA(0,2)
- ARMA(1,1)
- ARMA(1,2)

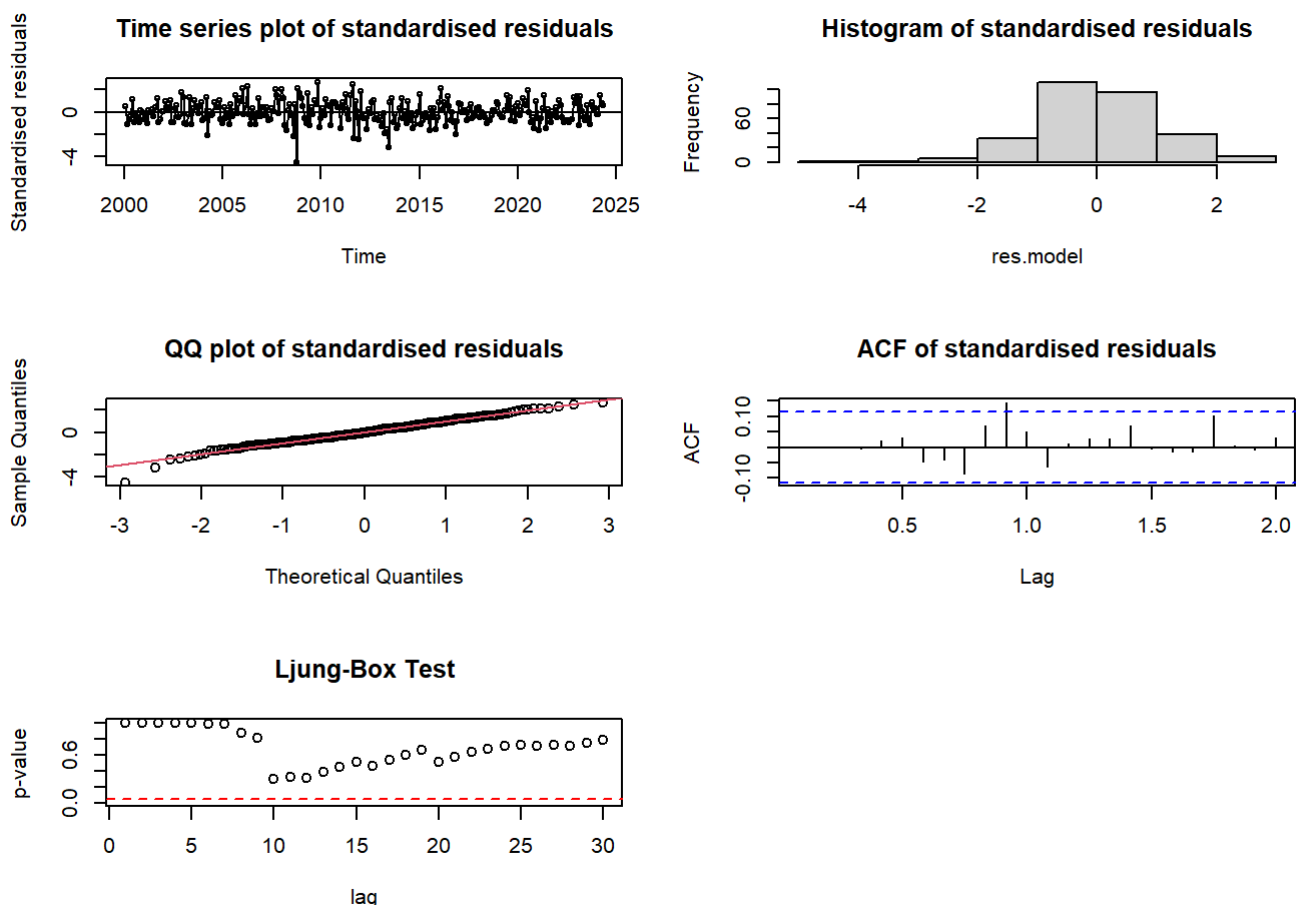
ARMA (0,4)

```
m.04 = Arima(r.gold,order=c(0,0,4),method='ML')  
coefest(m.04)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1      -0.126663    0.058535 -2.1639 0.030473 *
## ma2      -0.039129    0.059176 -0.6612 0.508462
## ma3       0.018187    0.058836  0.3091 0.757238
## ma4       0.025408    0.061579  0.4126 0.679899
## intercept 0.722343    0.237584  3.0404 0.002363 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model=m.04, std = TRUE, class = "ARIMA")
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98749, p-value = 0.01248
```



For ARMA(0,4) “ML” model, only “ma1” is significant, whereas, all components are insignificant. The residual plot has a flat mean level indicating no trend. The histogram is left skewed and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the left tail. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

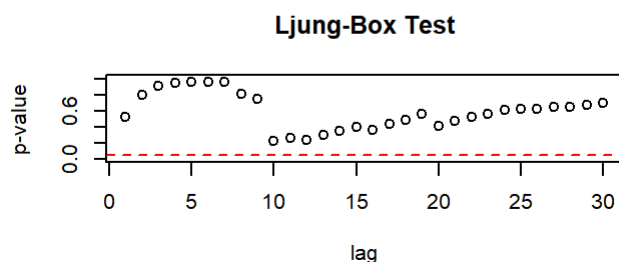
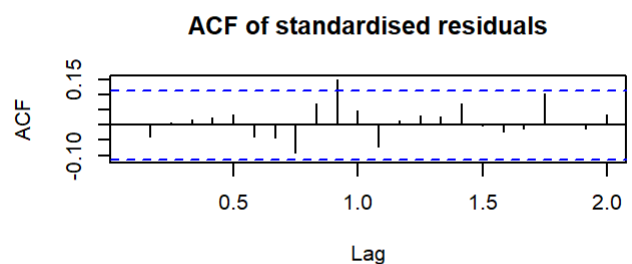
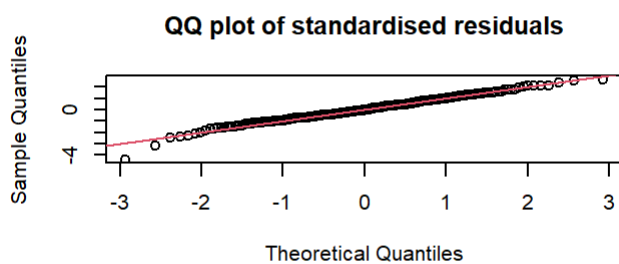
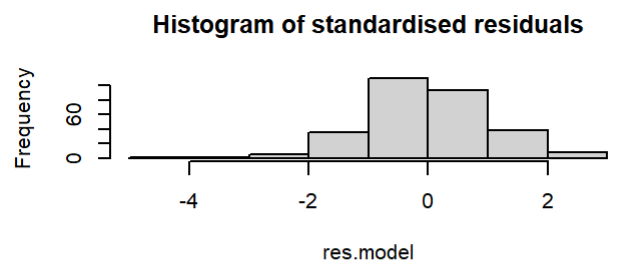
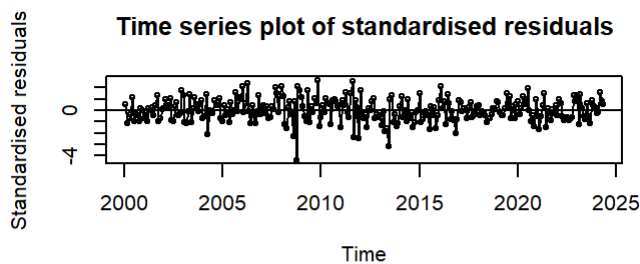
ARMA (0,1)

```
m.01 = Arima(r.gold,order=c(0,0,1),method='ML')
coeftest(m.01)
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ma1      -0.130280   0.060008 -2.1711 0.029927 *
## intercept 0.721647   0.235722  3.0614 0.002203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model=m.01, std = TRUE, class = "ARIMA")
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98886, p-value = 0.02437
```



For ARMA(0,1) "ML" model had all coefficient significant. The residual plot has a flat mean level indicating no trend. The histogram is left skewed and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the left tail. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

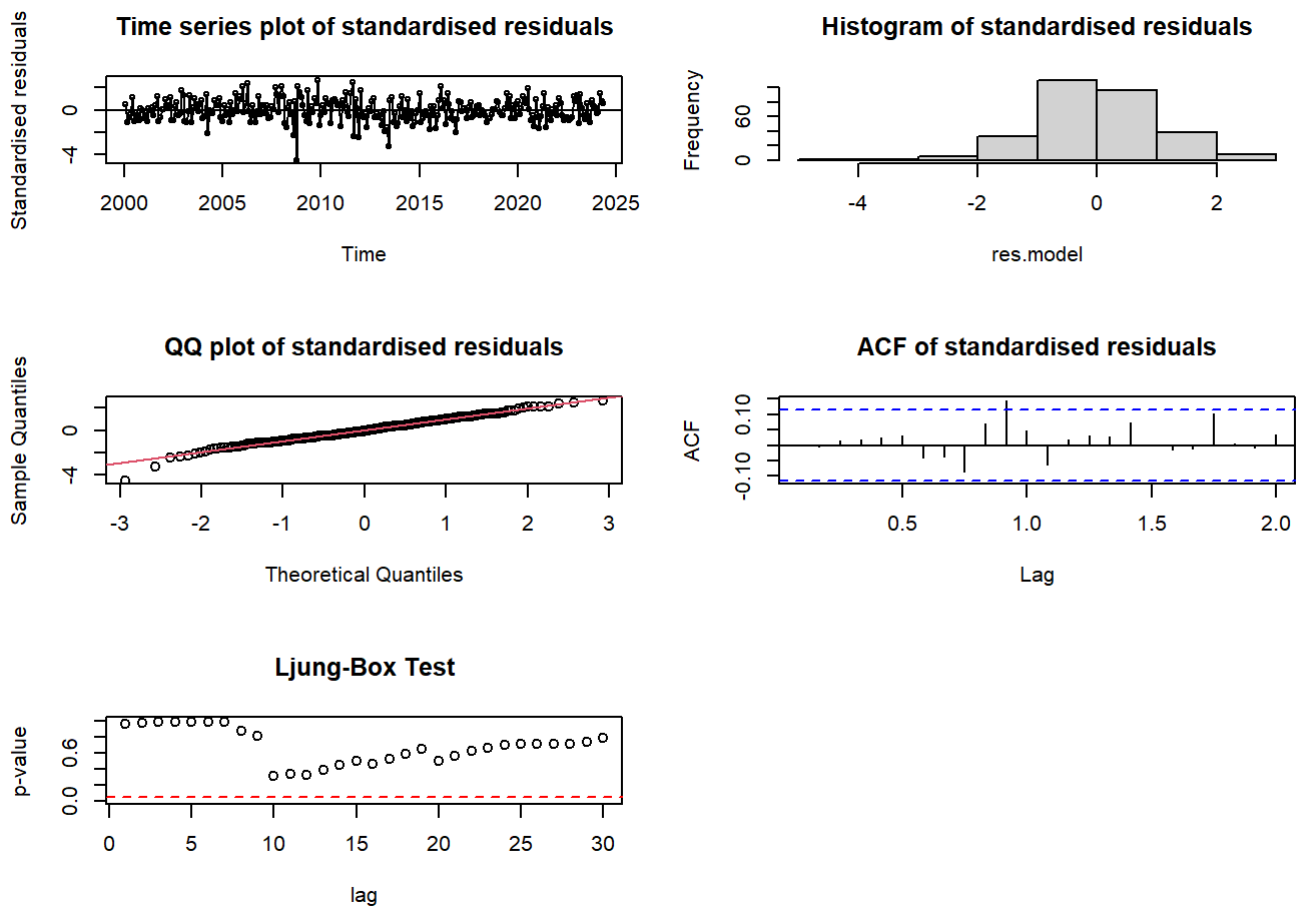
ARMA (0,2)

```
m.02 = Arima(r.gold,order=c(0,0,2),method='ML')
coeftest(m.02)
```

```
##
## z test of coefficients:
##
##          Estimate Std. Error z value Pr(>|z|)
## ma1      -0.124433   0.058572 -2.1244 0.033634 *
## ma2      -0.034710   0.057458 -0.6041 0.545784
## intercept  0.721616   0.227820  3.1675 0.001538 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model=m.02, std = TRUE, class = "ARIMA")
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98752, p-value = 0.01268
```



For ARMA(0,2) “ML” model had “ma1” coefficient significant. The residual plot has a flat mean level indicating no trend. The histogram is left skewed and indicates outliers. Most of the residuals follows the line of normality but there are some deviation at the left tail. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality.

Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

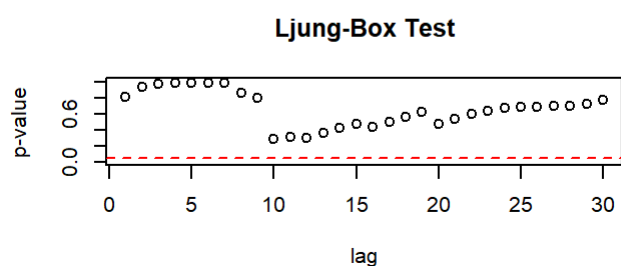
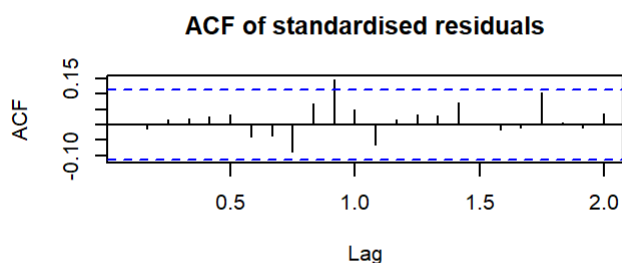
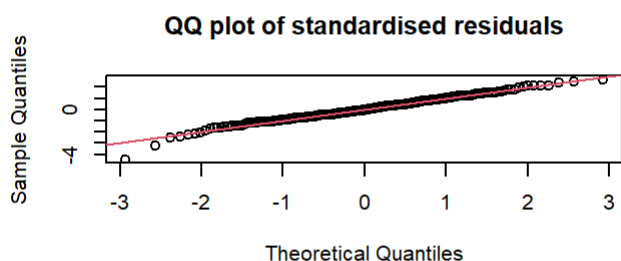
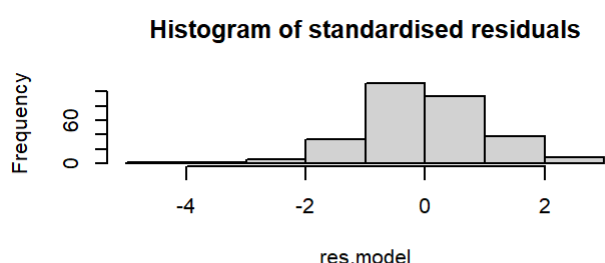
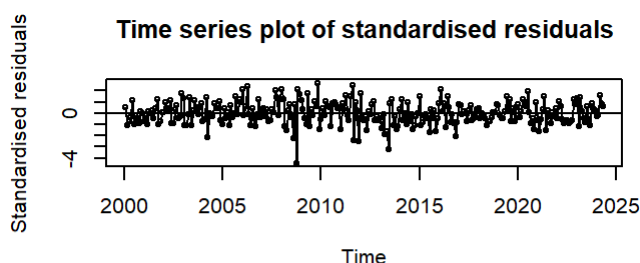
ARMA(1,1)

```
m.11 = Arima(r.gold,order=c(1,0,1),method='ML')
coeftest(m.11)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.18421   0.34410  0.5354 0.592407
## ma1     -0.31117   0.33132 -0.9392 0.347640
## intercept 0.72144   0.22881  3.1530 0.001616 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model=m.11, std = TRUE, class = "ARIMA")
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98793, p-value = 0.01546
```



For ARMA(1,1) “ML” model had no coefficient significant. The residual plot has a flat mean level indicating no trend. The histogram is left skewed and indicates outliers. Most of the residuals follows the line of normality but there are some deviation at the left tail. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

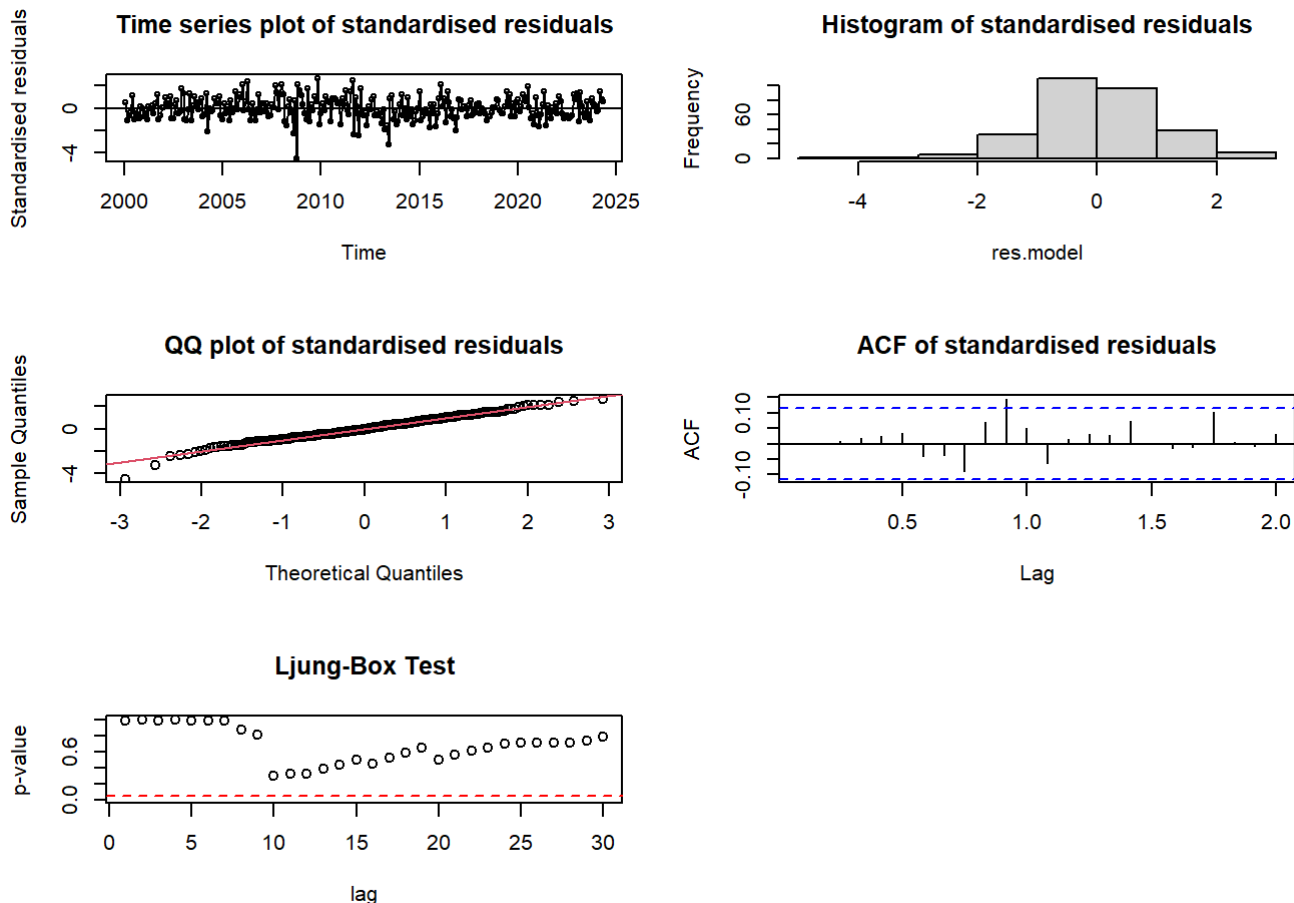
ARMA (1,2)

```
m.12 = Arima(r.gold,order=c(1,0,2),method='ML')
coeftest(m.12)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1      -0.178454   1.086952 -0.1642 0.869591
## ma1       0.053723   1.085102  0.0495 0.960513
## ma2      -0.059109   0.149284 -0.3959 0.692144
## intercept  0.721397   0.228644  3.1551 0.001604 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residual.analysis(model=m.12, std = TRUE, class = "ARIMA")
```

```
##
## Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98749, p-value = 0.01246
```



For ARMA(1,2) “ML” model had no coefficient significant. The residual plot has a flat mean level indicating no trend. The histogram is left skewed and indicates outliers. Most of the residuals follows the line of normality but their are some deviation at the left tail. Shapiro-Wilk test has a p-value < 0.05, indicating non-normality. Ljung-Box test shows all lags above the significance level, therefore, no auto-correlation is present in the residuals.

To conclude, ARMA(0,1) had all the coefficients significant. Therefore we can say ARMA(0,1) is the most suited model.

Model Specification ARMA x GARCH Part 2

```
abs.r.res.gold = abs(rstandard(m.01))
sq.r.res.gold = rstandard(m.01)^2
```

ACF and PACF absolute

```
Diagnostic_test(abs.r.res.gold, mainacf="ACF plot for abs residual return series", subacf =
"Figure 30: ACF plot for abs residual return series", mainpacf = "PACF plot for abs residual
return series", subpacf = "Figure 31: PACF plot for abs residual return series", test = "ACF-
PACF")
```

ACF plot for abs residual return series

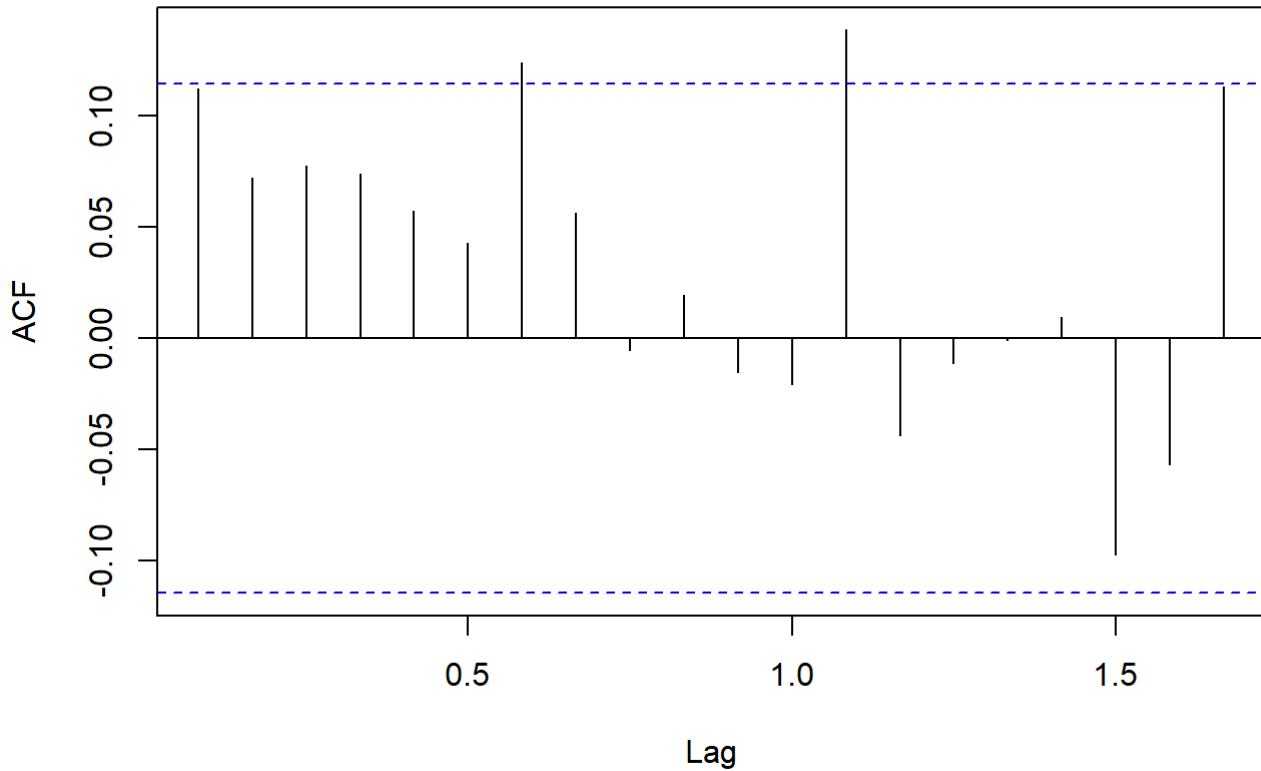


Figure 30: ACF plot for abs residual return series

PACF plot for abs residual return series

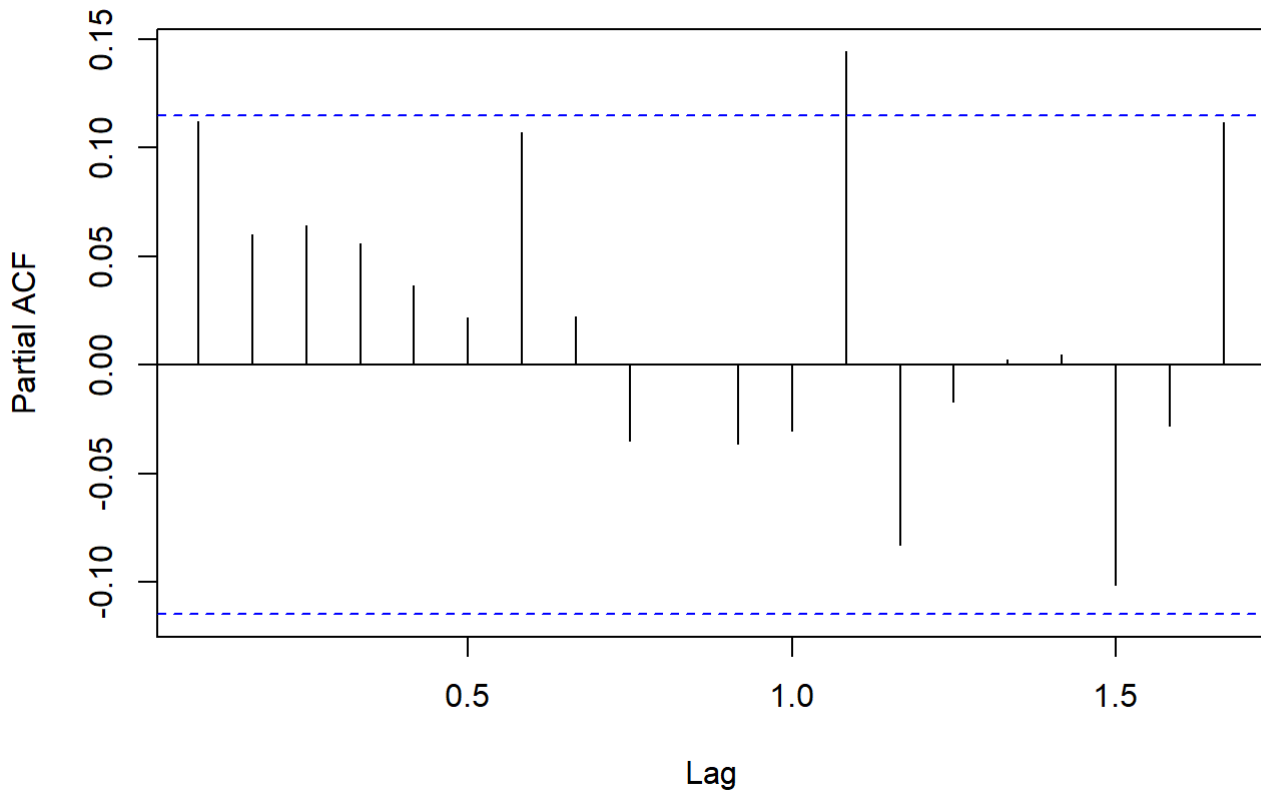


Figure 31: PACF plot for abs residual return series

To get the value of (p,q) , first 5 lags are considered from ACF and PACF plot

From Figure 30 ACF plot, there are no significant lags, $q = 0$ and from Figure 31 PACF plot, there are no lags significant, so $p = 0$.

$\max(p,q) = 0$ $q = 0$ $\max(p,q = 0)$ does not lead to any models.

EACF abs garch

```
eacf(abs.r.res.gold)
```

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

Top left “o” is in (0,0) - $\max(p,q) = 0$ and $q=0 \Rightarrow \max(p,q = 0) = 0$; does not lead any models.

Neighbor models:

- $\max(p,q) = 0$ and $q = 1 \Rightarrow \max(p,q = 1) = 0$; does not lead any models.
- $\max(p,q) = 1$ and $q = 1 \Rightarrow \max(p,q = 1) = 1$, hence $p = 0$ or 1

{GARCH(0,1) and GARCH(1,1)}

ACF and PACF squared

```
Diagnostic_test(sq.r.res.gold, mainacf="ACF plot for squared residual return series", subacf = "Figure 32: ACF plot for squared residual return series", mainpacf = "PACF plot for squared residual return series", subpacf = "Figure 33: PACF plot for squared residual return series", test = "ACF-PACF")
```

ACF plot for squared residual return series

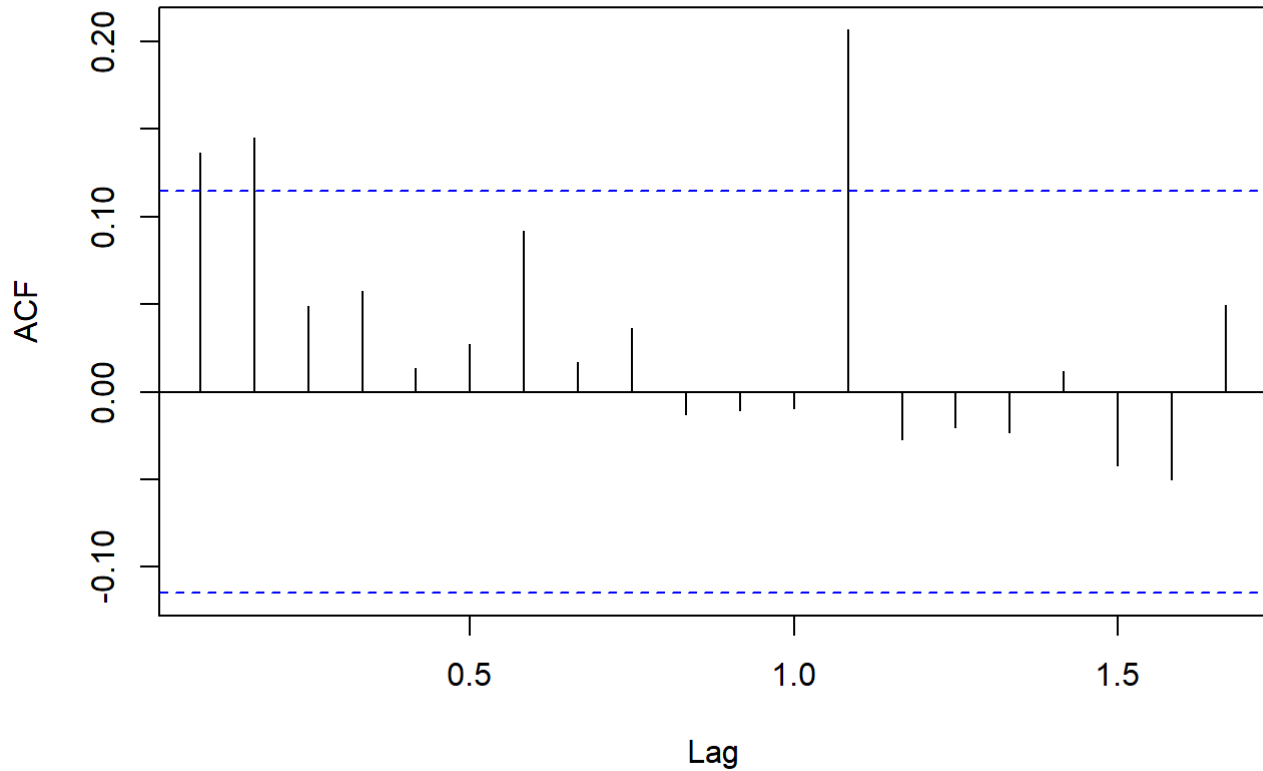


Figure 32: ACF plot for squared residual return series

PACF plot for squared residual return series

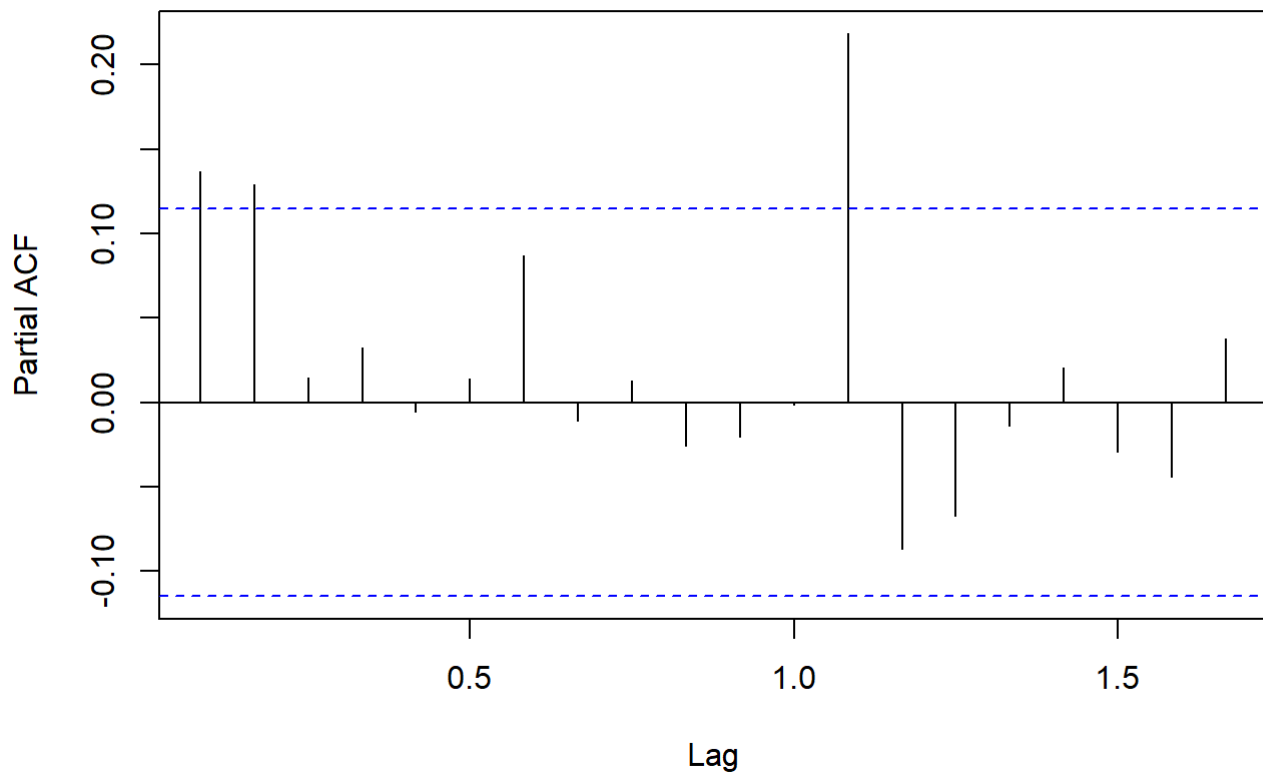


Figure 33: PACF plot for squared residual return series

From Figure 32 ACF plot we get $q = 2$ and from Figure 33 PACF plot we get $p = 2$.

$\max(p, q) = 2$ $q = 2$ $\max(p, q = 2)$, we get $p = 0, 1, 2$

GARCH {0,2}, GARCH {1,2} and GARCH {2,2}

EACF squared

```
eacf(sq.r.res.gold)
```

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

- $\max(p,q) = 0$ and $q = 2 \Rightarrow \max(p,q = 2) = 0$; does not lead any models.

Neighbor models:

- $\max(p,q) = 0$ and $q = 3 \Rightarrow \max(p,q = 3) = 0$; does not lead any models.
- $\max(p,q) = 1$ and $q = 2 \Rightarrow \max(p,q = 2) = 1$; does not lead any models.
- $\max(p,q) = 1$ and $q = 3 \Rightarrow \max(p,q = 3) = 1$; does not lead any models

GARCH {0,2} GARCH {1,2} GARCH {2,2}

Parameter Estimation ARMA x GARCH

```
m.01.01<- fGarch::garchFit(~ arma(0,1)+garch(1,0),
data = r.gold, trace=F)
m.01.11<- fGarch::garchFit(~ arma(0,1)+garch(1,1),
data = r.gold, trace=F)
m.01.02<- fGarch::garchFit(~ arma(0,1)+garch(2,0),
data = r.gold, trace=F)
m.01.12<- fGarch::garchFit(~ arma(0,1)+garch(2,1),
data = r.gold, trace=F)
m.01.22<- fGarch::garchFit(~ arma(0,1)+garch(2,2),
data = r.gold, trace=F)
```

```
summary(m.01.01)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## fGarch::garchFit(formula = ~arma(0, 1) + garch(1, 0), data = r.gold,
##   trace = F)
##
## Mean and Variance Equation:
## data ~ arma(0, 1) + garch(1, 0)
## <environment: 0x000002b5d6d5ec48>
## [data = r.gold]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ma1      omega      alpha1
## 0.617973 -0.061223 18.513131 0.134143
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.61797    0.25263   2.446  0.0144 *
## ma1     -0.06122    0.06945  -0.881  0.3781
## omega   18.51313    2.03017   9.119 <2e-16 ***
## alpha1   0.13414    0.07917   1.694  0.0902 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## -859.1102    normalized: -2.942158
##
## Description:
## Thu Jun  6 19:08:23 2024 by user: HP
##
##
## Standardised Residuals Tests:
##
##      Statistic      p-Value
## Jarque-Bera Test  R    Chi^2 14.0913851 0.0008711533
## Shapiro-Wilk Test R    W      0.9893178 0.0306262931
## Ljung-Box Test    R    Q(10)  5.0977723 0.8845516951
## Ljung-Box Test    R    Q(15) 13.9437475 0.5297996750
## Ljung-Box Test    R    Q(20) 16.1940736 0.7045135769
## Ljung-Box Test    R^2 Q(10)  9.0240701 0.5298207696
## Ljung-Box Test    R^2 Q(15) 27.3561308 0.0259651604
## Ljung-Box Test    R^2 Q(20) 30.0591929 0.0689001850
## LM Arch Test      R    TR^2   8.4759943 0.7469152657
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## 5.911714 5.962080 5.911345 5.931889
```

No coefficients are significant for m.01.01

```
summary(m.01.11)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  fGarch::garchFit(formula = ~arma(0, 1) + garch(1, 1), data = r.gold,
##    trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 1) + garch(1, 1)
## <environment: 0x000002b5d5554cb0>
##  [data = r.gold]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      mu      ma1      omega      alpha1      beta1
##  0.68780 -0.08278  1.98535  0.11590  0.79422
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.68780    0.23419   2.937  0.00332 **
## ma1     -0.08278    0.06371  -1.299  0.19382
## omega    1.98535    1.69370   1.172  0.24112
## alpha1   0.11590    0.05476   2.117  0.03430 *
## beta1    0.79422    0.11338   7.005 2.48e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -854.7267    normalized:  -2.927146
##
## Description:
##  Thu Jun  6 19:08:23 2024 by user: HP
##
##
## Standardised Residuals Tests:
##
##      Statistic  p-Value
## Jarque-Bera Test  R    Chi^2  0.8945865 0.6393564
## Shapiro-Wilk Test  R    W      0.9941019 0.3158700
## Ljung-Box Test     R    Q(10)  4.2274411 0.9365039
## Ljung-Box Test     R    Q(15) 11.7499547 0.6978485
## Ljung-Box Test     R    Q(20) 12.9923204 0.8777133
## Ljung-Box Test     R^2  Q(10)  4.0367199 0.9456753
## Ljung-Box Test     R^2  Q(15) 20.9957574 0.1369649
## Ljung-Box Test     R^2  Q(20) 27.2772419 0.1276427
## LM Arch Test       R    TR^2   5.3563064 0.9450076
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## 5.888539 5.951497 5.887966 5.913758
```

Only 2 coefficients are significant for m.01.11

```
summary(m.01.02)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## fGarch::garchFit(formula = ~arma(0, 1) + garch(2, 0), data = r.gold,
##   trace = F)
##
## Mean and Variance Equation:
## data ~ arma(0, 1) + garch(2, 0)
## <environment: 0x000002b5d3cf0770>
## [data = r.gold]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ma1      omega      alpha1      alpha2
## 0.703328 -0.082857 16.474510 0.101305 0.124140
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.70333    0.24036   2.926 0.00343 **
## ma1     -0.08286    0.06959  -1.191 0.23381
## omega   16.47451    2.12282   7.761 8.44e-15 ***
## alpha1   0.10131    0.06976   1.452 0.14644
## alpha2   0.12414    0.07417   1.674 0.09420 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## -856.9624    normalized: -2.934803
##
## Description:
## Thu Jun  6 19:08:23 2024 by user: HP
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test R Chi^2 0.9695397 0.6158389
## Shapiro-Wilk Test R W 0.9962201 0.7135513
## Ljung-Box Test R Q(10) 5.1626066 0.8800543
## Ljung-Box Test R Q(15) 13.3966096 0.5716930
## Ljung-Box Test R Q(20) 15.2553057 0.7616173
## Ljung-Box Test R^2 Q(10) 6.9489060 0.7302596
## Ljung-Box Test R^2 Q(15) 21.5439469 0.1203316
## Ljung-Box Test R^2 Q(20) 27.6226388 0.1186376
## LM Arch Test R TR^2 7.1048214 0.8506094
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## 5.903852 5.966810 5.903278 5.929070
```


No coefficients are significant for m.01.02

```
summary(m.01.12)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## fGarch::garchFit(formula = ~arma(0, 1) + garch(2, 1), data = r.gold,
## trace = F)
##
## Mean and Variance Equation:
## data ~ arma(0, 1) + garch(2, 1)
## <environment: 0x000002b5ce3e81d8>
## [data = r.gold]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ma1      omega      alpha1      alpha2      beta1
## 0.720678 -0.091151 3.042020 0.084049 0.077411 0.701344
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.72068    0.23287   3.095 0.00197 **
## ma1     -0.09115    0.06322  -1.442 0.14934
## omega    3.04202    2.55682   1.190 0.23414
## alpha1   0.08405    0.06664   1.261 0.20721
## alpha2   0.07741    0.10537   0.735 0.46256
## beta1    0.70134    0.17864   3.926 8.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## -854.5314    normalized: -2.926477
##
## Description:
## Thu Jun  6 19:08:23 2024 by user: HP
##
##
## Standardised Residuals Tests:
##
##      Statistic    p-Value
## Jarque-Bera Test  R    Chi^2  0.3496448 0.83960612
## Shapiro-Wilk Test  R    W      0.9958064 0.62562284
## Ljung-Box Test     R    Q(10)  4.3902078 0.92803215
## Ljung-Box Test     R    Q(15) 11.9511476 0.68272299
## Ljung-Box Test     R    Q(20) 13.2328564 0.86717108
## Ljung-Box Test     R^2  Q(10)  3.8771917 0.95271704
## Ljung-Box Test     R^2  Q(15) 21.6361673 0.11770494
## Ljung-Box Test     R^2  Q(20) 28.9631969 0.08848655
## LM Arch Test       R    TR^2   5.0209708 0.95727485
##
## Information Criterion Statistics:
```

##	AIC	BIC	SIC	HQIC
##	5.894050	5.969600	5.893228	5.924313

Only 1 coefficient is significant for m.01.12

```
summary(m.01.22)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## fGarch::garchFit(formula = ~arma(0, 1) + garch(2, 2), data = r.gold,
##   trace = F)
##
## Mean and Variance Equation:
## data ~ arma(0, 1) + garch(2, 2)
## <environment: 0x000002b5d9dbf830>
## [data = r.gold]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ma1      omega      alpha1      alpha2      beta1      beta2
## 0.716377 -0.089948 3.655391 0.087388 0.106455 0.362793 0.277344
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.71638    0.23356   3.067 0.00216 **
## ma1     -0.08995    0.06381  -1.410 0.15868
## omega    3.65539    3.57465   1.023 0.30650
## alpha1   0.08739    0.06837   1.278 0.20121
## alpha2   0.10645    0.12822   0.830 0.40640
## beta1    0.36279    1.25619   0.289 0.77273
## beta2    0.27734    1.04574   0.265 0.79085
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## -854.5153    normalized: -2.926422
##
## Description:
## Thu Jun  6 19:08:23 2024 by user: HP
##
##
## Standardised Residuals Tests:
##
##      Statistic    p-Value
## Jarque-Bera Test  R    Chi^2    0.2522403 0.8815089
## Shapiro-Wilk Test  R    W        0.9957120 0.6056971
## Ljung-Box Test     R    Q(10)     4.4582808 0.9243146
## Ljung-Box Test     R    Q(15)    11.9297594 0.6843378
## Ljung-Box Test     R    Q(20)    13.2251157 0.8675177
## Ljung-Box Test     R^2  Q(10)     3.8941063 0.9519976
## Ljung-Box Test     R^2  Q(15)    20.4326163 0.1559575
## Ljung-Box Test     R^2  Q(20)    27.8302268 0.1134796
## LM Arch Test       R    TR^2      5.0417624 0.9565695
##
## Information Criterion Statistics:
```

```
##          AIC          BIC          SIC          HQIC
## 5.900789 5.988931 5.899676 5.936095
```

No coefficients are significant for m.01.22

Summary analysis gave ARMA(0,1) X GARCH(1,1) as best model for forecast.

Forecasting

ARIMA forecast:

```
m5_011 = Arima(data.ts,order=c(0,1,1),
               lambda = 0, method = "ML")
preds1 = forecast(m5_011, lambda = 0, h = 10)
preds1
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jun 2024	2359.698	2221.531	2506.458	2151.695	2587.808
## Jul 2024	2359.698	2175.361	2559.655	2083.682	2672.276
## Aug 2024	2359.698	2139.554	2602.492	2031.457	2740.975
## Sep 2024	2359.698	2109.459	2639.622	1987.919	2801.007
## Oct 2024	2359.698	2083.116	2673.002	1950.077	2855.360
## Nov 2024	2359.698	2059.480	2703.679	1916.340	2905.629
## Dec 2024	2359.698	2037.916	2732.287	1885.739	2952.781
## Jan 2025	2359.698	2018.004	2759.248	1857.632	2997.457
## Feb 2025	2359.698	1999.448	2784.856	1831.572	3040.107
## Mar 2025	2359.698	1982.030	2809.328	1807.227	3081.060

```
plot(preds1, sub = "Figure 34: Forecast from ARIMA model")
```

Forecasts from ARIMA(0,1,1)

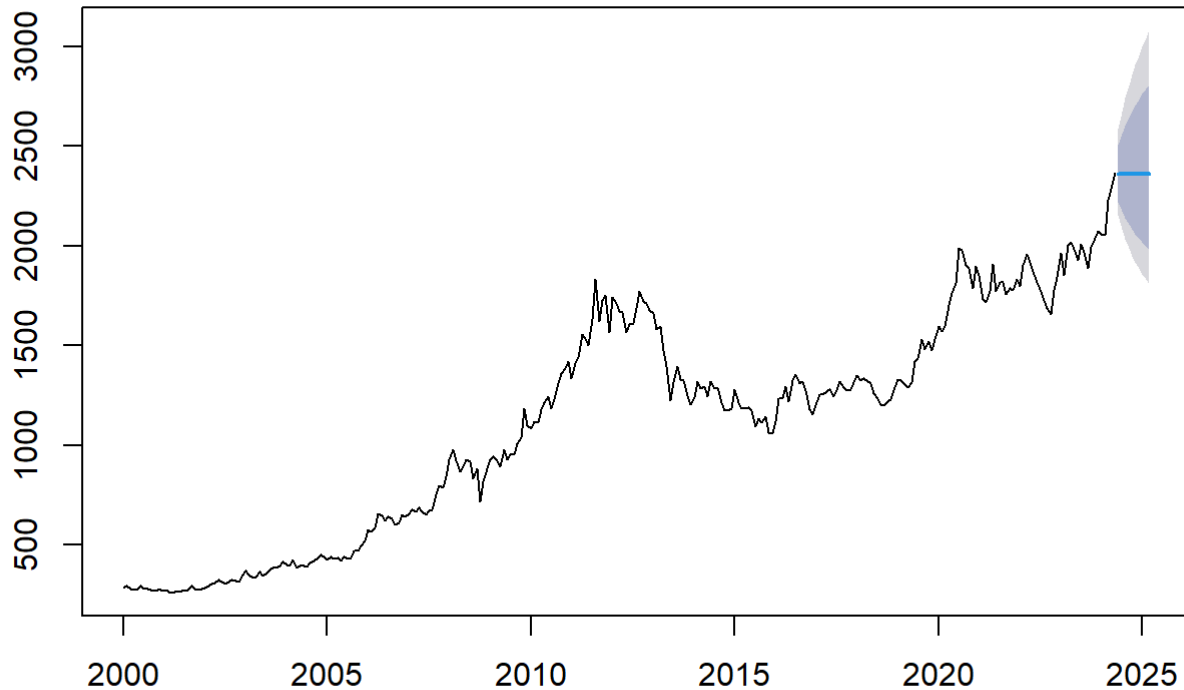


Figure 34: Forecast from ARIMA model

The best ARIMA model is ARIMA(0,1,1). The forecast from this model shows consistent values over the next 10 months.

ARMA x GARCH forecast:

```
spec <- ugarchspec(variance.model = list(model = "sGARCH",  
garchOrder = c(1, 1)  
),  
mean.model = list(armaOrder = c(0, 1)))  
model_401_11_2 <- ugarchfit(spec = spec, data = r.gold,  
solver = "hybrid",  
solver.control = list(trace=0))  
frc <- ugarchforecast(model_401_11_2,n.ahead=10,data=r.gold)  
frc
```

```
##
## *-----*
## *          GARCH Model Forecast          *
## *-----*
## Model: sGARCH
## Horizon: 10
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=May 2024]:
##      Series Sigma
## T+1  0.4943 4.167
## T+2  0.6953 4.218
## T+3  0.6953 4.265
## T+4  0.6953 4.307
## T+5  0.6953 4.344
## T+6  0.6953 4.378
## T+7  0.6953 4.409
## T+8  0.6953 4.437
## T+9  0.6953 4.462
## T+10 0.6953 4.485
```

```
plot(frc, which = 1, sub = "Figure 35: Forecast of ARMA-GARCH series")
mtext('Figure 35: Forecast of ARMA-GARCH series',line = 4, side = 1, cex = 0.8)
```

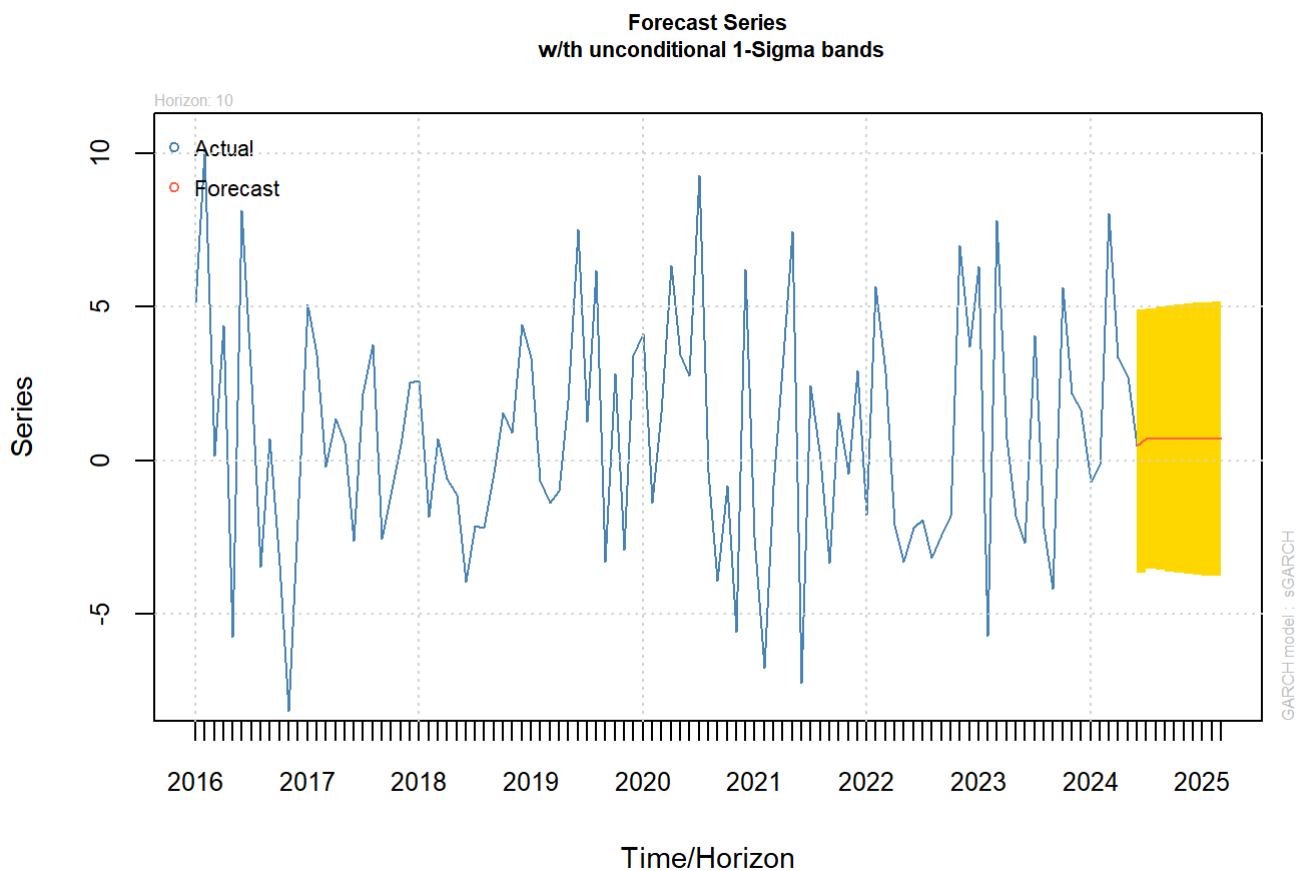


Figure 35: Forecast of ARMA-GARCH series

```
plot(frc, which = 3, sub = "Figure 36: Forecast of ARMA-GARCH sigma")
mtext('Figure 36: Forecast of ARMA-GARCH sigma',line = 4, side = 1, cex = 0.8)
```

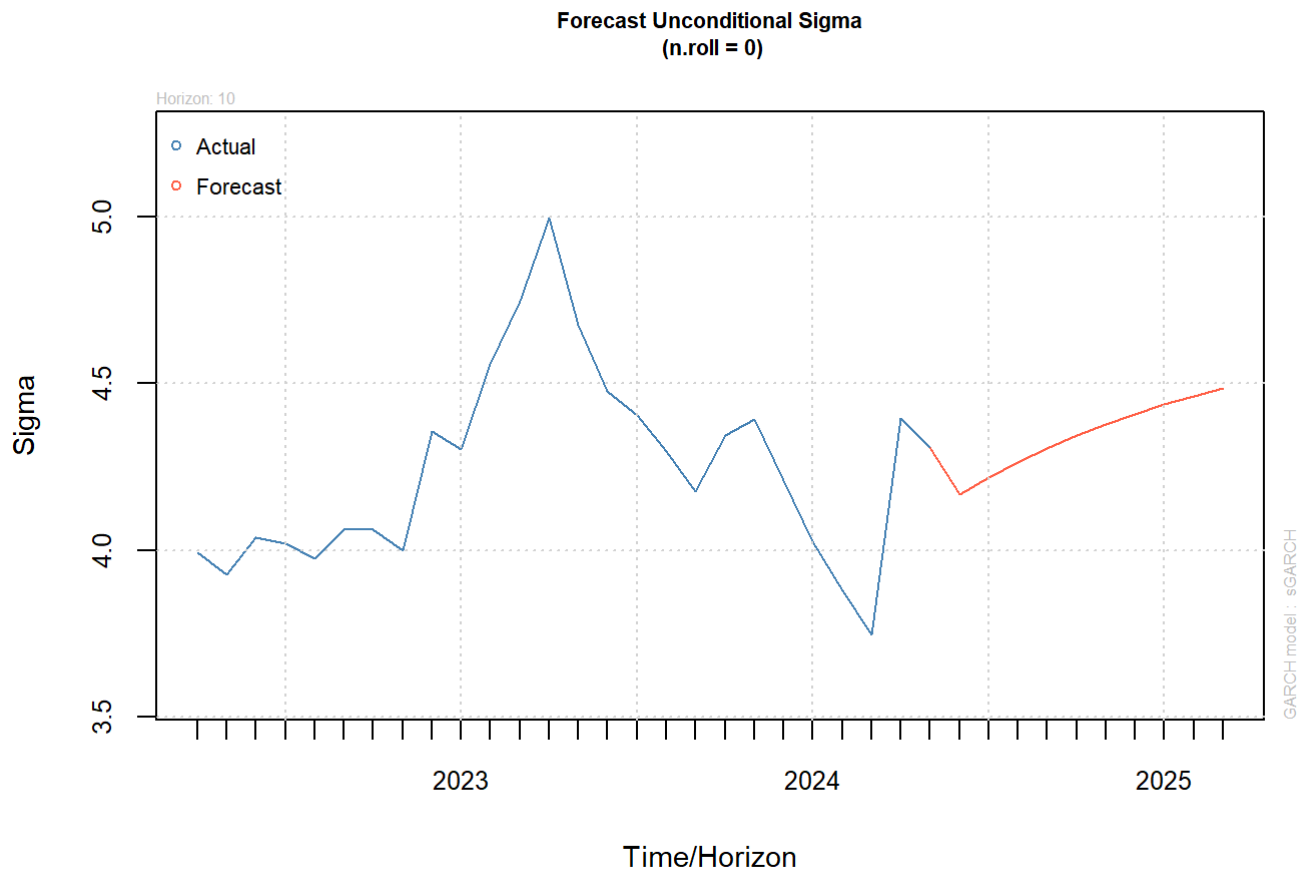


Figure 36: Forecast of ARMA-GARCH sigma

Based on the ARMA-GARCH model, the forecast for the next 10 months is consistent, indicating stable predicted values. However, the high variance within the yellow highlighted area suggests significant uncertainty in the forecast, underscoring the potential for considerable deviation from the central trend.

The ARMA-GARCH model's forecast indicates a rising trend in volatility over the next 10 months. While the historical sigma values indicates fluctuations, the forecast suggests that volatility will increase consistently. This rising trend in forecasted volatility highlights the growing uncertainty and potential risk in the series during the forecast period.

SARIMA forecast:

```
m_011_112_12 = Arima(data.ts,order=c(0,1,1),seasonal=list(order=c(1,1,2), period=12), lambda
= 0, method = "ML")
preds1 = forecast(m_011_112_12, lambda = 0, h = 10)
preds1
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jun 2024	2340.072	2200.317	2488.704	2129.746	2571.169
## Jul 2024	2363.698	2179.262	2563.742	2087.527	2676.404
## Aug 2024	2386.897	2166.255	2630.013	2057.834	2768.581
## Sep 2024	2373.474	2125.122	2650.849	2004.352	2810.574
## Oct 2024	2375.637	2101.590	2685.420	1969.557	2865.442
## Nov 2024	2405.892	2105.147	2749.602	1961.474	2951.004
## Dec 2024	2446.021	2118.690	2823.925	1963.535	3047.066
## Jan 2025	2499.756	2144.846	2913.393	1977.845	3159.388
## Feb 2025	2508.555	2133.329	2949.778	1957.981	3213.946
## Mar 2025	2538.021	2140.261	3009.703	1955.593	3293.912


```
plot(preds1, main = "Forecast from SARIMA_011_112_12", sub = "Figure 37: Forecast of SARIMA s  
eries")
```

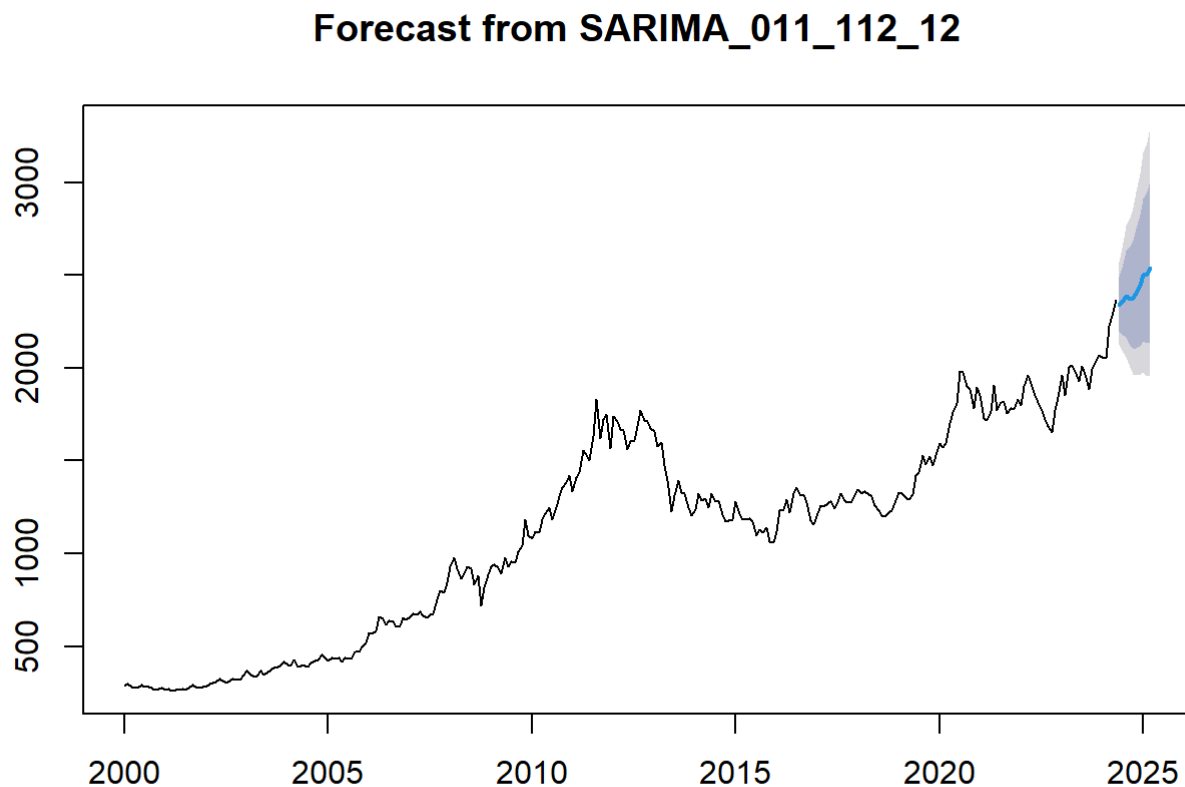


Figure 37: Forecast of SARIMA series

The SARIMA(0,1,1)(1,1,2)[12] model forecasts a continued upward trend over the next 10 months, with increasing uncertainty as time progresses.

Conclusion

Based on the analysis, both ARIMA and ARMA-GARCH models did not get significant results, making their forecasts unreliable. In contrast, the SARIMA model proved to be the most effective, as evidenced by residual analysis, goodness of fit metrics (AIC, BIC), error metrics, and significant coefficients. Specifically, the SARIMA((0,1,1)(1,1,2)_12 model emerged as the best fit.

The forecast from the SARIMA model indicates an upward trend for the next 10 months.

Reference

- [1] Demirhan, H 2024, lecture and lab notes, Time Series, RMIT University, Melbourne
- [2] Tran L, Pham, T 2024, seasonality function, Time Series, RMIT University, Melbourne