GARCH & ARIMA Modelling on Apple Stock Price

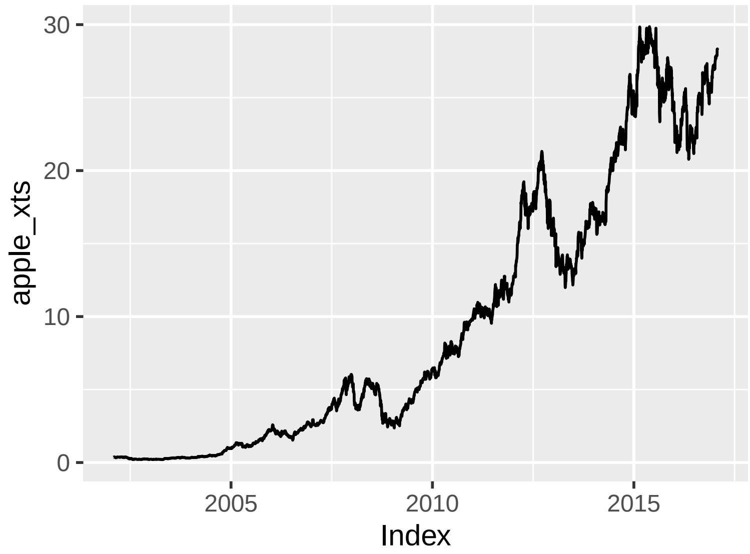
2024-04-28

Part 1

1. Dataset

The following data is of Apple stock prices from 1 February, 2002 to 31 January, 2017.

```
data = read.csv("AAPL.csv")
apple_xts = xts(data$Adj.Close, order.by = as.Date(data$Date))
autoplot(apple_xts)
```



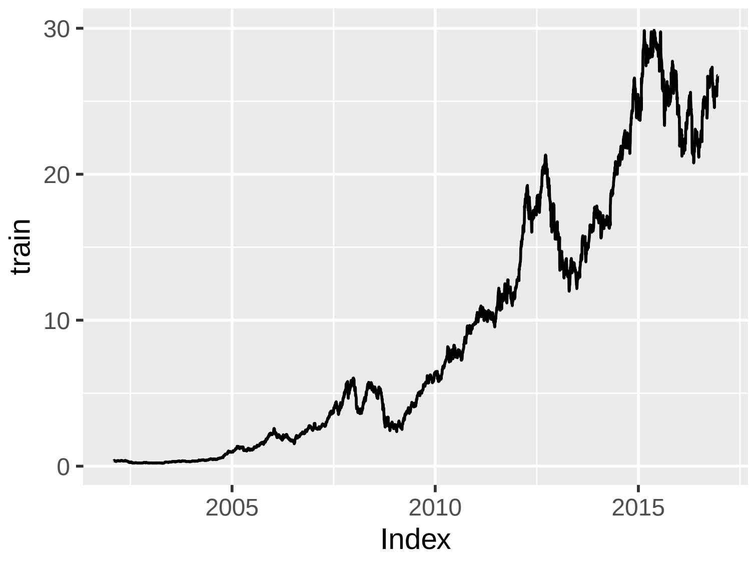
I will first split the data into train and test sets for forecasting later. I will keep the last 30 days for the test set.

```
split_point = length(apple_xts) - 30

# Split the data into training and test sets
train = apple_xts[1:split_point]
test = apple_xts[(split_point + 1):length(apple_xts)]
```

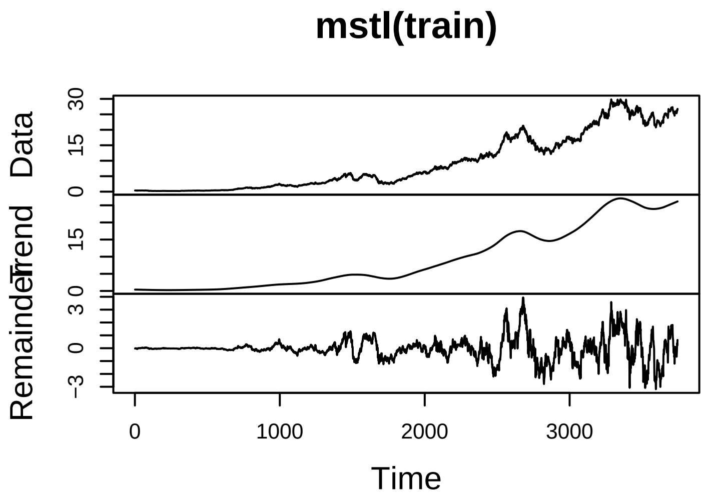
Next, I will analyse the train set.

```
autoplot(train)
```



Observing the above time plot, there seems to be volatility clustering. The time series is also clearly not stationary.

```
plot(mstl(train))
```



By decomposing the time series, it can be observed that there is an increasing, non-linear trend in the data. The non-linearity, along with the volatility clustering seen in the previous plot, suggests that a GARCH model might potentially be appropriate to model this data.

I will first make the time series stationary.

2. Stationarity

Based on the time plot, there looks to be exponential growth. The plot also exhibits heteroskedasticity. Hence, I will log-transform the data to stabilise the variance.

```r
log_apple_xts = log(train)
```

Next, I will check if any differencing is needed.

```r
# nsdiffs(log_apple_xts) # no seasonality
```

```r
ndiffs(log_apple_xts)
```

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

One non-seasonal differencing is necessary to make the data stationary.

```r
stat_ts = diff(log_apple_xts)
```

In order to verify that the time series is indeed stationary, I will conduct the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

Augmented Dickey-Fuller (ADF) Test

```r
adf.test(stat_ts[2:length(stat_ts)], alternative = "stationary")
```

```
## Warning in adf.test(stat_ts[2:length(stat_ts)], alternative = "stationary"):
## p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  stat_ts[2:length(stat_ts)]
## Dickey-Fuller = -14.861, Lag order = 15, p-value = 0.01
## alternative hypothesis: stationary
```

The results show that the p-value < 0.01. Hence, based on the ADF test, at 1% significance level, we reject the null hypothesis that the time series has a unit root, meaning the time series is stationary.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

```r
kpss.test(stat_ts)
```

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

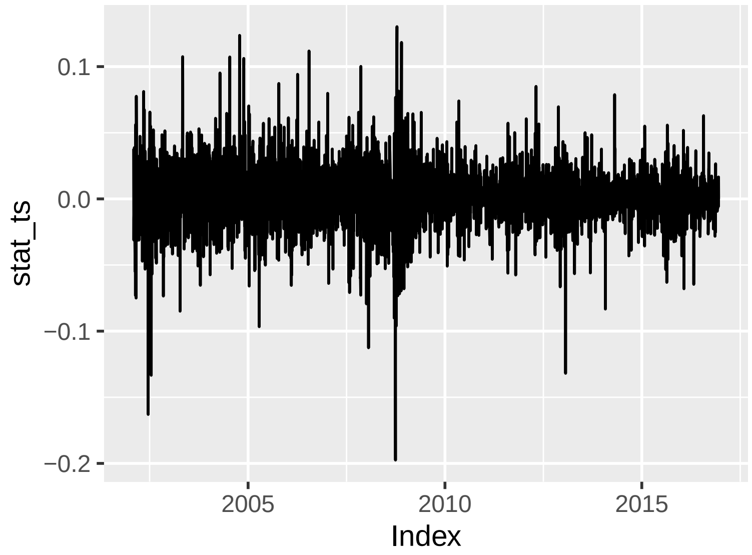
```
##
##  KPSS Test for Level Stationarity
##
## data:  stat_ts
## KPSS Level = 0.16501, Truncation lag parameter = 9, p-value = 0.1
```

The results show that the p-value > 0.1. Hence, based on the KPSS test, at 10% significance level, we do not reject the null hypothesis that the time series is stationary.

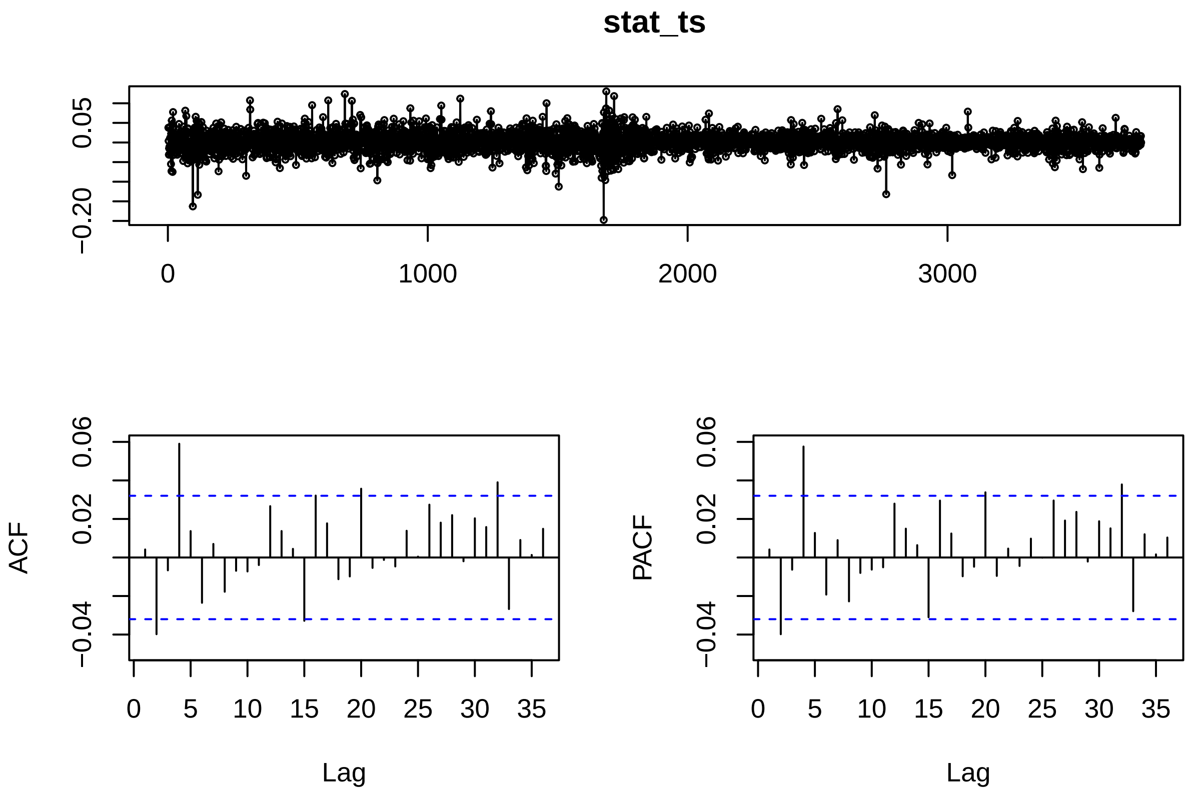
Both the ADF and KPSS tests show that the time series is stationary.

3. Serial Correlation & i.i.d

```
autoplot(stat_ts)
```



```
tsdisplay(stat_ts)
```

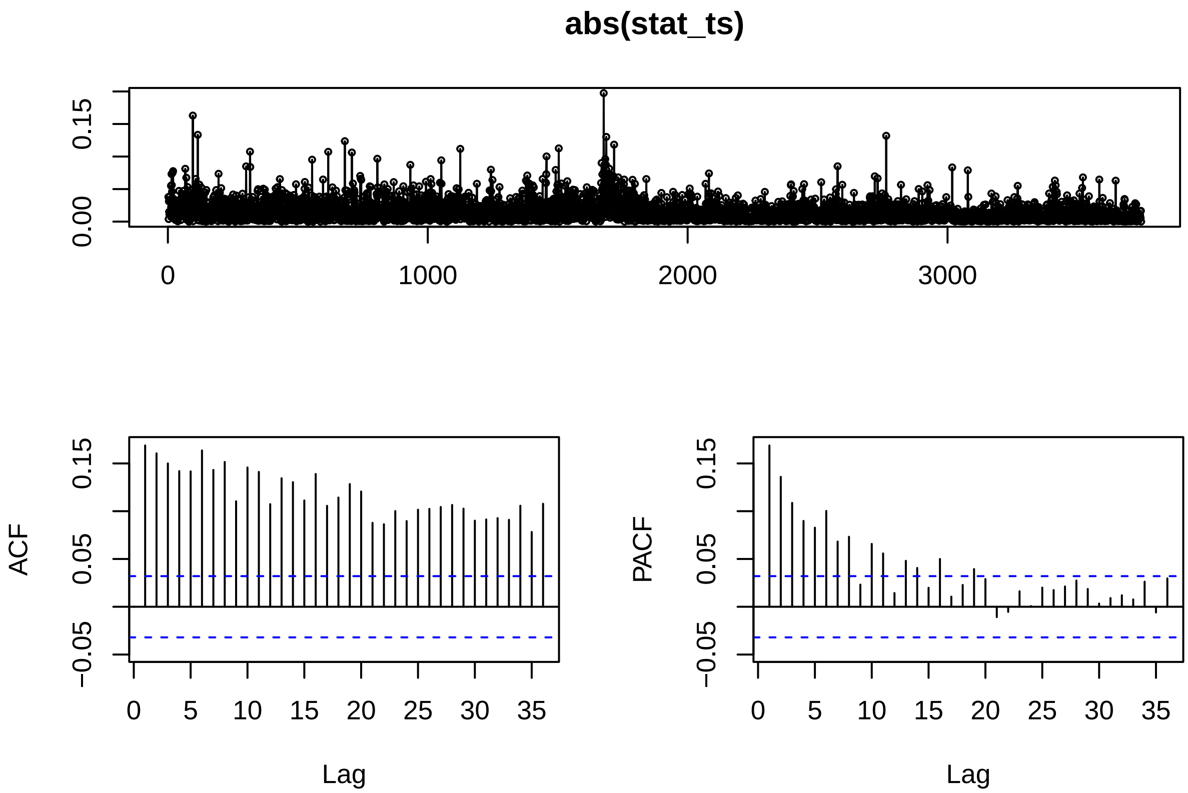


After applying log transformation and once differencing, the mean of the time series appears to be more constant over time and hovers around zero, which suggests that the transformations have been effective in stabilising the mean. Looking at the variance, there seems to be some periods with larger variations and some extreme spikes, particularly around 2008-2009, which coincides with the global financial crisis. This indicates periods of high volatility, which is common in financial time series during market turmoils.

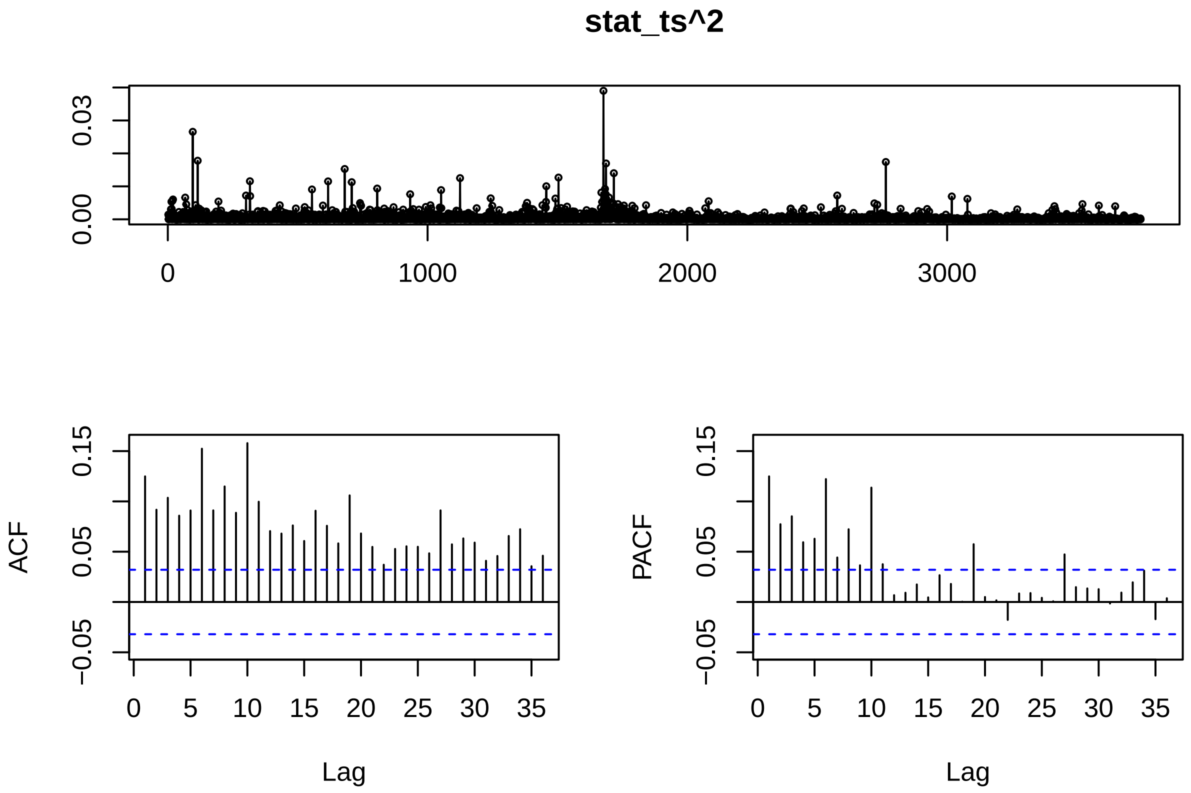
Based on the ACF and PACF plots, there seems to be little serial correlation in the data.

Looking at the time plot, since the variance is not constant over time and there is volatility clustering observed, it suggests that the data might not be independently and identically distributed (i.i.d.). To look deeper into this, I will plot the absolute and squared values of the data.

```
tsdisplay(abs(stat_ts))
```



```
tsdisplay(stat_ts^2)
```

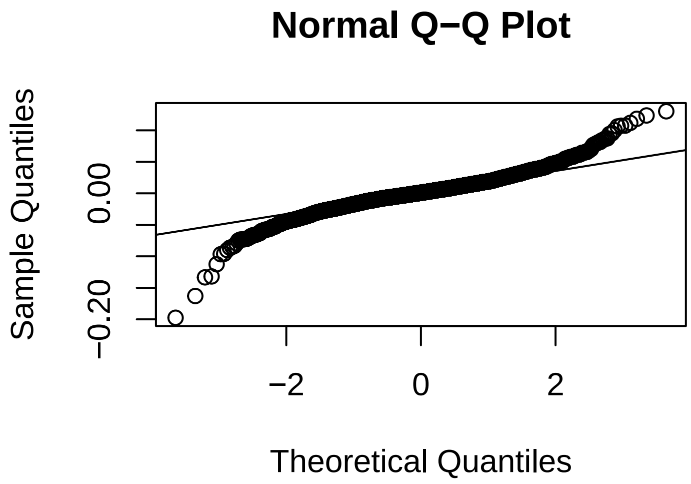


The ACF and PACF plots of the absolute and squared values of the data are very clearly autocorrelated. This suggests that the data is not i.i.d.

4. Q-Q Plot

Next, I will explore the distributional shape of the data using a Normal Q-Q plot.

```
qqnorm(stat_ts)
qqline(stat_ts)
```



```
kurtosis(stat_ts[2:length(stat_ts)])
```

```
## [1] 5.383254
```

The Q-Q plot exhibits skewness and heavy tails. This is further supported by the positive kurtosis value of 8.383254, suggesting that it is a heavy-tailed distribution.

In summary, the data is serially uncorrelated. However, it admits a higher-order dependence structure, namely volatility clustering, and a heavy-tailed distribution. Therefore, it is appropriate to use GARCH models for this data.

5. GARCH Models

To determine the p and q values of candidate GARCH models, I will examine the EACF of the absolute values of the time series.

```
eacf(abs(stat_ts[2:length(stat_ts)]), ar.max = 10, ma.max = 10)
```

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

The EACF table shows that an ARMA(1,1) model is a likely candidate. Alternatively, an ARMA(2,2) model can also be considered. Hence, I will consider both the GARCH(1,1) and GARCH(2,2) models.

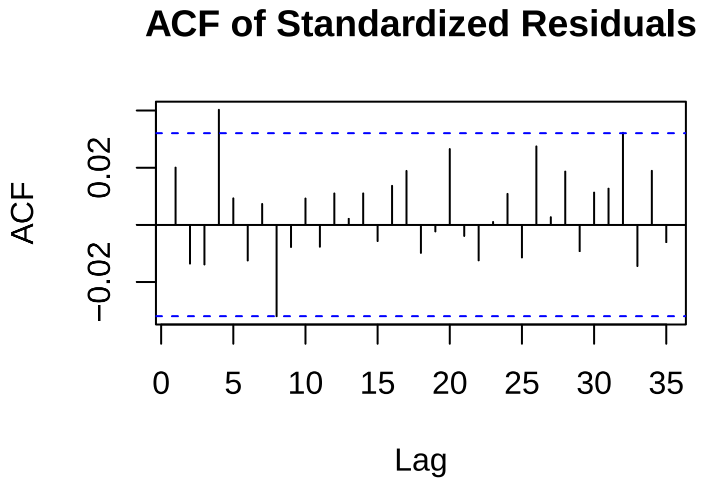
I will fit the GARCH models using the rugarch package in R.

GARCH(1,1)

```
spec_g11 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

g11 = ugarchfit(spec = spec_g11, data = stat_ts[2:length(stat_ts)])
```

```
g11_res = residuals(g11, standardize = TRUE)
acf(g11_res, main = "ACF of Standardized Residuals")
```



```
Box.test(g11_res, type = "Ljung-Box", lag = 20)
```

```
##
##  Box-Ljung test
##
## data:  g11_res
## X-squared = 20.823, df = 20, p-value = 0.4076
```

```
jarque.bera.test(g11_res)
```

```
##
##  Jarque Bera Test
##
## data:  g11_res
## X-squared = 1874.6, df = 2, p-value < 2.2e-16
```

```
infocriteria(g11)
```

```
##
## Akaike       -4.998507
## Bayes        -4.988527
## Shibata      -4.998513
## Hannan-Quinn -4.994958
```

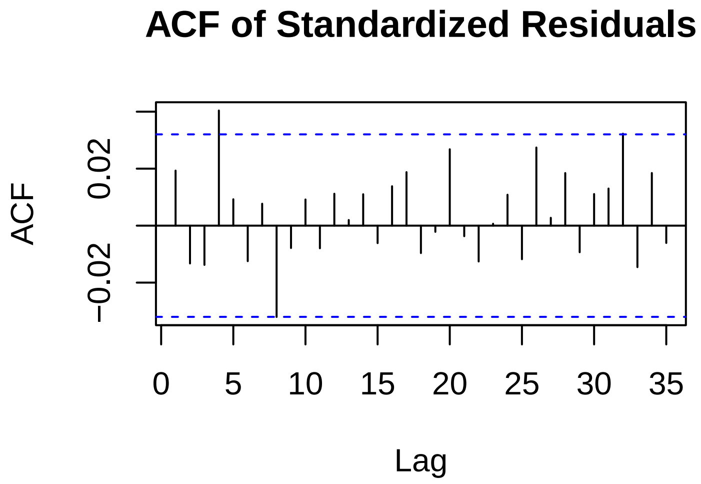
In the Jarque Bera test, the p-value < 2.2e-16. Hence, we reject the null hypothesis that the residuals are normally distributed. This implies that the skewness and/ or kurtosis of the residuals significantly deviate from that of a normal distribution.
In the Ljung-Box test, the p-value is 0.4076. Hence, at 5% significance level (and also all significance levels), we do not reject the null hypothesis that there is no time series information left in the residuals. This is supported by the ACF plot of the residuals, which generally shows that there is no autocorrelation in the residuals. Hence, the GARCH(1,1) model is adequate.
The AIC of the model is -4.998507.

GARCH(2,2)

```
spec_g22 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 2)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

g22 = ugarchfit(spec = spec_g22, data = stat_ts[2:length(stat_ts)])
```

```
g22_res = residuals(g22, standardize = TRUE)
acf(g22_res, main = "ACF of Standardized Residuals")
```



```
Box.test(g22_res, type = "Ljung-Box", lag = 20)
```

```
##
##  Box-Ljung test
##
## data:  g22_res
## X-squared = 20.871, df = 20, p-value = 0.4047
```

```r
jarque.bera.test(g22_res)
```

```
##
##  Jarque Bera Test
##
## data:  g22_res
## X-squared = 1862.6, df = 2, p-value < 2.2e-16
```

```r
infocriteria(g22)
```

```
##
## Akaike       -4.997682
## Bayes        -4.984375
## Shibata      -4.997691
## Hannan-Quinn -4.992949
```

In the Jarque Bera test, the p-value < 2.2e-16. Hence, we reject the null hypothesis that the residuals are normally distributed. This implies that the skewness and/ or kurtosis of the residuals significantly deviate from that of a normal distribution.
In the Ljung-Box test, the p-value is 0.4047. Hence, at 5% significance level (and also all significance levels), we do not reject the null hypothesis that there is no time series information left in the residuals. This is supported by the ACF plot of the residuals, which generally shows that there is no autocorrelation in the residuals. Hence, the GARCH(2,2) model is adequate.
The AIC of the model is -4.997682.

Comparing the AICs of GARCH(1,1) and GARCH(2,2), the GARCH(1,1) is the better model as it has a lower AIC of -4.998507.

6. Mixed Model: ARIMA-GARCH

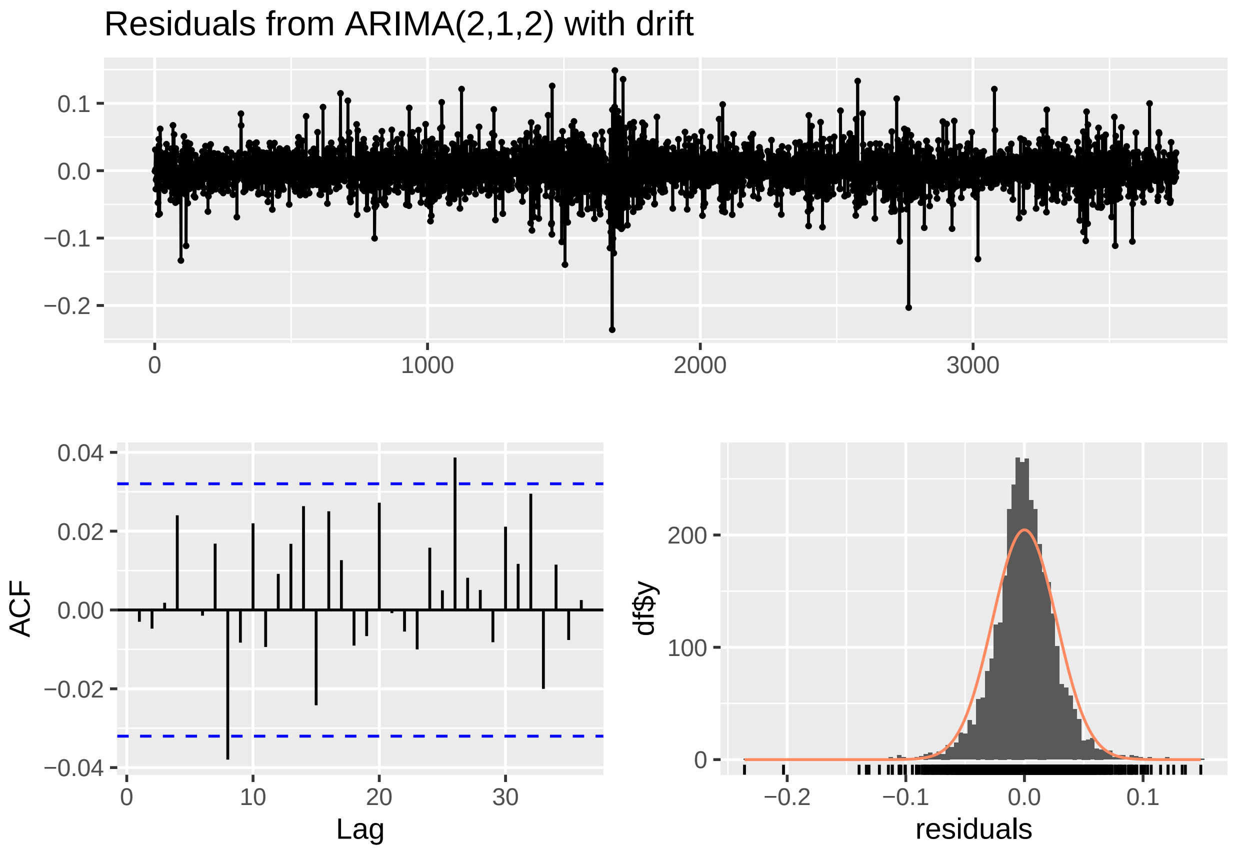
Based on the original time plot of the train data, there seems to be patterns in the data not related to volatility, namely trend. Hence, I will address this in a mean model by first fitting the data to an ARIMA model, then fitting the residuals to a GARCH model.

First, I will use the auto.arima() function to find the p, d, q values. Note that I will pass the raw train data through auto.arima(), because this function will automatically make the necessary transformations and differencing in order to make the data stationary.

```r
# pass raw train data as this function handles transformations & differencing
arima_fit = auto.arima(train, lambda = "auto")
arima_fit
```

```
## Series: train
## ARIMA(2,1,2) with drift
## Box Cox transformation: lambda= 0.1528483
##
## Coefficients:
##           ar1      ar2     ma1     ma2   drift
##       -0.0370  -0.8971  0.0471  0.8742  0.0013
## s.e.   0.1337   0.1203  0.1442  0.1383  0.0004
##
## sigma^2 = 0.0007307:  log likelihood = 8208.57
## AIC=-16405.13   AICc=-16405.11   BIC=-16367.76
```

```
checkresiduals(arima_fit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2) with drift
## Q* = 10.857, df = 6, p-value = 0.09288
##
## Model df: 4.   Total lags used: 10
```

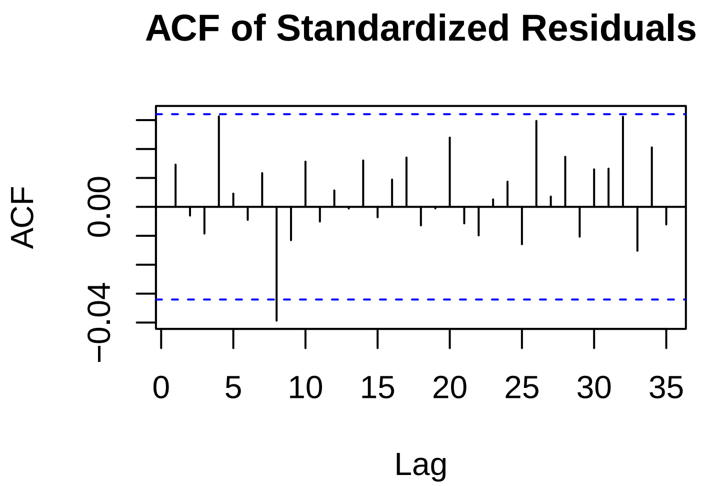
Looking at the residual ACF plot, there are some minor spikes in the lags. This suggests that there might still be time series information left in the residuals that the ARIMA model was unable to capture. This is further supported by the p-value of the Ljung-Box test of 0.09288. This suggests that at a 10% significance level, we reject the null hypothesis that there is no autocorrelation in the residuals (although at 5% significance level, we do not reject the null hypothesis). The residual ACF plot and the Ljung-Box test suggest the model might not be adequate, and that there could be some non-linear information from the volatility of the time series left in the residuals that the linear ARIMA model was unable to capture. Hence, I will fit the residuals to a GARCH model. Since I have previously determined the GARCH(1,1) model to be an adequate model and a good fit, I will use GARCH(1,1) on the residuals.

Now that I have determined the orders of both the ARMA and GARCH models for the mixed model, where ARMA(2,2) and GARCH(1,1), I will use the rugarch package in R. This will help me to fit an ARIMA model to the data, and fit a GARCH model to the residuals.

```r
# Specify the ARIMA(2,1,2) + GARCH(1,1) model
spec = ugarchspec(
  variance.model = list(
    model = "sGARCH",
    garchOrder = c(1, 1)  # GARCH(1,1)
  ),
  mean.model = list(
    armaOrder = c(2, 2),  # ARIMA(2,1,2) where I = 1 is implicit because I have
    # differenced the data before fitting
    include.mean = TRUE,
    archm = FALSE,
    arfima = FALSE
  ),
  distribution.model = "sstd"
)
```

```r
arima_garch_model = ugarchfit(spec = spec, data = stat_ts[2:length(stat_ts)])
```

```r
ag_res = residuals(arima_garch_model, standardize = TRUE)
acf(ag_res, main = "ACF of Standardized Residuals")
```



```r
Box.test(ag_res, type = "Ljung-Box", lag = 20)
```

```
##
##  Box-Ljung test
##
## data:  ag_res
## X-squared = 17.71, df = 20, p-value = 0.6065
```

```r
jarque.bera.test(ag_res)
```

```
##
```

```
##  Jarque Bera Test
##
## data:  ag_res
## X-squared = 1887, df = 2, p-value < 2.2e-16
```

```r
infocriteria(arima_garch_model)
```

```
##
## Akaike       -4.997024
## Bayes        -4.980389
## Shibata      -4.997038
## Hannan-Quinn -4.991107
```

In the Jarque Bera test, the p-value < 2.2e-16. Hence, we reject the null hypothesis that the residuals are normally distributed. This implies that the skewness and/ or kurtosis of the residuals significantly deviate from that of a normal distribution.
In the Ljung-Box test, the p-value is 0.6065. Hence, at 5% significance level (and also all significance levels), we do not reject the null hypothesis that there is no time series information left in the residuals. This is supported by the ACF plot of the residuals, which generally shows that there is no autocorrelation in the residuals. Hence, the ARIMA-GARCH model is adequate.
The AIC of the model is -4.997024.

7. Comparing the Models

Between the GARCH(1,1), GARCH(2,2), and ARIMA-GARCH models, the GARCH(1,1) model has the best AIC score of -4.998507, even though all 3 models are adequate. Since GARCH(1,1) is the best performing model during training, and I will use it for forecasting/ testing.

8. Forecasting

I will next use the GARCH(1,1) model to forecast the next 30 periods.

```r
forecasts = ugarchforecast(g11, n.ahead = 30)

mean_forecast = forecasts@forecast$seriesFor
variance_forecast = forecasts@forecast$sigmaFor
```

Undifference and un-log the forecast values to obtain the original scaled forecasts.

```r
stock_prices = as.numeric(train)

last_value = tail(log(stock_prices), n = 1)

cumulative_forecast = cumsum(c(last_value, mean_forecast))[-1]

undiff_unlog_forecast = exp(cumulative_forecast)
```

Finally, I will evalute the forecasts using RMSE as an evaluation metric.

```
rmse = sqrt(mean((undiff_unlog_forecast - test)^2))
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 0.226433594192852"
```

The forecasts have an RMSE of 0.2264.